

UNIVERSITÄT WÜRZBURG
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR KÜNSTLICHE INTELLIGENZ
UND ANGEWANDTE INFORMATIK



**Evaluation von Methoden zur
Bildverarbeitung für Objekterkennung
am Beispiel der Klassifikation von Bäumen**

Masterarbeit
zur Erlangung des akademischen Grades
Master of Science (Informatik)

Vorgelegt von: Christian Reul
Matrikel-Nr. 1716820

Datum der Abgabe: 08. Oktober 2015

Betreuer: Prof. Dr. Frank Puppe
M. Sc. Martin Toepfer

Erklärung

Ich versichere, dass ich die vorliegende Arbeit einschließlich aller beigefügten Materialien selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Werken entnommen sind, sind in jedem Einzelfall unter Angabe der Quelle deutlich als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.

Mir ist bekannt, dass Zuwiderhandlungen gegen diese Erklärung und bewusste Täuschungen die Benotung der Arbeit mit der Note 5.0 zur Folge haben kann.

Würzburg, 08.10.2015

Christian Reul

Abstract

Der Forschungszweig der Computer Vision ist ein schnell wachsendes Gebiet, das in vielen Bereichen Verwendung findet. In den letzten Jahren wurden beispielsweise bei der auf optischen Merkmalen basierenden Klassifikation große Fortschritte gemacht, so auch bei der automatischen Erkennung von Baumspezies anhand ihrer Blätter. Eine Vielzahl von Veröffentlichungen beschäftigte sich mit diesem Thema und zahlreiche Herangehensweisen wurden vorgestellt und untersucht (u. a. [Wu 2007], [Kadir 2014] und [Suk 2013]).

Die Klassifikation von Baumspezies anhand der charakteristischen Merkmale ihrer Blätter stellt den Kernpunkt dieser Arbeit dar. Es wird ein System entwickelt, das ein Bild eines Blattes einliest, verarbeitet, entsprechende Features extrahiert und mit deren Hilfe die jeweilige Spezies bestimmt.

Vor der Konzipierung des eigentlichen Systems werden zunächst die Grundlagen der Computer Vision zusammengefasst und erläutert. Anschließend werden verschiedene Feature-Klassen implementiert und getestet. Als besonders aussagekräftig erweisen sich die HOCS-Features (vgl. [Kumar 2012]), mit deren Hilfe die äußere Kontur eines Blattes sehr exakt beschrieben werden kann.

Die Evaluation erfolgt auf insgesamt sechs Datensets: fünf im Internet frei verfügbaren (u. a. [Flavia] und [MEW]) und einem selbst erstellten. Zunächst wird das System nur innerhalb einzelner Datensets getestet. Das bedeutet, dass ein Teil des Datensatzes zum Training, der Rest zum Testen verwendet wird. Die erzielten Ergebnisse liegen, je nach Datenset, im mittleren bis hohen 90%-Bereich und erreichen oder übertreffen den State of the Art.

Wird auf einem Datensatz trainiert und auf einem anderen getestet, fallen die Klassifikationsergebnisse stark ab, teilweise auf weniger als 60%. Es zeigt sich, dass einige Feature-Klassen, wie beispielsweise die Hu- oder Binary Pattern-Features, zwar bei der Klassifikation innerhalb eines Datensets gute Ergebnisse liefern, jedoch bei der Datenset-übergreifenden Klassifikation einen schädlichen Einfluss auf das Resultat haben können. Durch Anpassung des Systems und unter Berücksichtigung der besten fünf Ergebnisse, kann die Klassifikationsgenauigkeit, abhängig von den verwendeten Datensets, auf bis zu 98% gesteigert werden.

Abschließend wird ein System entwickelt, das einem Nutzer mittels einer simpel zu bedienenden grafischen Oberfläche erlaubt, eigene Fotografien von Blättern einzulesen und die abgebildete Spezies zu klassifizieren.

Inhaltsverzeichnis

1	Einleitung.....	1
2	Grundlagen.....	4
2.1	Computer Vision.....	4
2.1.1	Darstellungsformen von Bildern.....	4
2.1.2	Filter	10
2.1.3	Kantendetektion.....	12
2.1.4	Konturen	14
2.1.5	Morphologische Transformationen	16
2.1.6	Distanztransformation	19
2.1.7	Die OpenCV-Bibliothek.....	20
2.2	Machine Learning.....	20
2.2.1	Der K-Nearest-Neighbours-Klassifikator	21
2.2.2	Die Support Vector Machine	22
2.2.3	Evaluationsmethoden	28
2.2.4	Weka	29
2.2.5	Die LibSVM-Bibliothek.....	31
2.3	Framework zur Bildverarbeitung, Feature-Extraktion und Klassifikation.....	31
2.4	Blattmerkmale.....	32
2.4.1	Aufbau eines Blattes	32
2.4.2	Zusammenfassung der klassischen Blattmerkmale.....	32
3	Datensets.....	35
3.1	Flavia	35
3.2	Foliage	36
3.3	Swedish Leaf Dataset (SLD)	37
3.4	Middle European Woody Plants (MEW)	37
3.5	Intelligent Computing Laboratory (ICL).....	39
3.6	Bavaria Leaf Dataset (BLD)	39
3.7	Überblick	41
3.8	Weitere, nicht verwendete Datensets	41
3.8.1	Leafsnap.....	42
3.8.2	LifeCLEF.....	42
3.9	Begriffsklärung	44

4	Vorverarbeitung und Segmentierung	45
4.1	Standardverfahren zur automatischen Segmentierung.....	46
4.1.1	Otsus Methode.....	46
4.1.2	K-Means-Clustering.....	47
4.1.3	Watershed	49
4.2	Der GrabCut-Algorithmus.....	52
4.2.1	GraphCut.....	52
4.2.2	Erweiterung zu GrabCut.....	55
4.2.3	Praktische Umsetzung der manuellen Segmentierung mittels GrabCut.....	56
4.2.4	Diskussion der praktischen Umsetzung	57
4.3	Entwickeltes System.....	60
4.3.1	Grundidee	60
4.3.2	Sichere Vordergrund-/Hintergrundpixel	60
4.3.3	Zuordnung der noch unbekanntes Pixel.....	61
4.3.4	Erstellung der finalen GrabCut-Eingabemaske.....	63
4.4	Vorläufiger, qualitativer Vergleich der Segmentierungsergebnisse.....	65
4.5	Nachbearbeitungsschritt: Füllen der Kontur.....	65
4.6	Detektion und Entfernen des Stiels.....	66
4.6.1	Algorithmus	66
4.6.2	Evaluation	67
4.7	Vergleich der vorgestellten Methoden	69
4.7.1	Evaluationsmodus	69
4.7.2	Ergebnisse.....	70
5	Verwendete Features	73
5.1	Kontur-Features	73
5.2	Farb-Features	76
5.3	Kurvatur-Features	77
5.4	Bildmomente.....	78
5.4.1	Räumliche Momente.....	78
5.4.2	Hu-Momente	79
5.4.3	Zernike-Momente	81
5.5	HOCS-Features	82
5.5.1	Grundidee	82
5.5.2	Beispiel.....	83
5.5.3	Speed-Ups.....	84
5.5.4	Bilaterale Filterung	87
5.6	Binary Pattern-Features	88
5.6.1	Grundidee	88
5.6.2	Anpassungen	89
5.7	Nicht verwendete Features.....	91
5.7.1	Venerungs-Features	91
5.7.2	Gray-Level-Co-Occurrence-Matrix (GLCM)-Features	92
5.7.3	Histogram Of Orientated Gradients (HOG)-Features.....	93

6	Konzept	95
6.1	Überblick	95
6.2	Allgemeine Herangehensweise	97
6.2.1	Blattklassifikation durch einen Menschen	97
6.2.2	Auftretende Probleme bei der computergestützten Klassifikation	98
6.2.3	Verwendeter Ansatz	99
6.3	Handhabung zusammengesetzter Blätter	99
6.3.1	Problematik zusammengesetzter Blätter	99
6.3.2	Mögliche Ansätze	100
6.4	Fehlende Klassifikation von Nadelbäumen	102
7	Evaluation	103
7.1	Allgemeines Vorgehen	103
7.2	Verwendete Klassifikatoren	103
7.2.1	Der K-Nearest-Neighbours-Klassifikator	103
7.2.2	Die Support Vector Machine	105
7.2.3	Vergleich der beiden Klassifikatoren	107
7.3	Optimierung der Feature-Parameter	108
7.3.1	Kurvatur-Features	108
7.3.2	HOCS-Features	109
7.3.3	Binary Pattern-Features	111
7.3.4	Übersicht	114
7.4	Evaluation der Feature-Klassen	115
7.4.1	Klassifikationsergebnisse der einzelnen Feature-Klassen	115
7.4.2	Hinzufügen einer weiteren Feature-Klasse	116
7.4.3	Schrittweiser Aufbau des Systems	117
7.4.4	Einfluss der Wegnahme einzelner Feature-Klassen	118
7.5	Evaluation des Gesamtsystems innerhalb eines Datensets	119
7.5.1	Ablauf der Evaluation	119
7.5.2	Übersicht	119
7.5.3	Evaluation auf dem Flavia-Datensatz	120
7.5.4	Evaluation auf dem Swedish Leaf Datenset	123
7.5.5	Evaluation auf dem MEW-Datensatz	124
7.5.6	Ergebnisse unter Verwendung einer SVM	126
7.5.7	Vergleich mit anderen Systemen	126
7.5.8	Evaluation auf dem BLD	129
7.6	Farbproblematik	131
7.6.1	Einfluss der Farb-Features auf das Klassifikationsergebnis	131
7.6.2	Auswirkungen von Umwelteinflüssen auf die Farb-Features	131
7.6.3	Resultierende Problematik bei der Klassifikation	132
7.7	Evaluation der Datenset-übergreifenden Klassifikation	133
7.7.1	Gewählte Datensets	133
7.7.2	Vorgehensweise	134
7.7.3	Ergebnisse mit dem Standardsystem	134

7.7.4	Entfernung potentiell schädlicher Feature-Klassen	136
7.7.5	Wertung mehrerer Ergebnisse	138
7.7.6	Zusammenfassung mehrerer Spezies	140
7.7.7	Evaluation auf künstlich kreierte Datensets	145
7.7.8	Einfluss der Feature-Klassen des finalen Systems	146
7.7.9	Test auf zwei unabhängig erstellten Evaluationsdatensets	148
7.8	Evaluation der Klassifikation von automatisch segmentierten Blättern.....	151
7.8.1	Vergleich der Ergebnisse.....	151
7.8.2	Veranschaulichung anhand von Beispielen	151
7.9	Evaluation der Laufzeiten.....	154
7.9.1	Zeitaufwand der Feature-Extraktion	154
7.9.2	Zeitaufwand der Klassifikation.....	155
8	Finales Programm.....	156
8.1	Das Verarbeitungsfenster	156
8.1.1	Laden eines Bildes	157
8.1.2	Segmentierung des Blattes	157
8.2	Das Ergebnisfenster	159
9	Vergleich mit anderen Domänen.....	161
9.1	Gesichtserkennung.....	161
9.1.1	Methoden	161
9.1.2	Übertragbarkeit	163
9.2	Klassifikation handschriftlicher Ziffern	164
9.2.1	Methoden	164
9.2.2	Übertragbarkeit	168
9.3	Medizinische Anwendung: Mammographie	168
9.3.1	Automatische Detektion	169
9.3.2	Entscheidungsunterstützung.....	170
9.3.3	Übertragbarkeit	171
10	Diskussion und Ausblick.....	172
10.1	Zusammenfassung der Ergebnisse	172
10.2	Ausblick	173
10.2.1	Gewichtung der Features.....	173
10.2.2	Zurückweisen von Instanzen	173
10.2.3	Einführen von Merkmalsklassen und Lösungsklassen.....	175
10.2.4	Zusammenfassung ähnlicher Blätter durch Hierarchisches Clustering.....	175
10.2.5	Automatische Segmentierung am Baum	177
10.2.6	Behandlung zusammengesetzter Blätter	177
10.2.7	Mögliche Erweiterungen des finalen Programms	177
10.2.8	Fazit.....	178
11	Quellenverzeichnis	180
12	Anhang	185

1 Einleitung

Sowohl das Gebiet der Computer Vision als auch das der Objekterkennung finden heutzutage immer mehr Verwendung in verschiedensten Bereichen. Dazu zählen zum einen alltägliche Anwendungen wie die Gesichtserkennung, die mittlerweile in jedem Mobiltelefon integriert ist, oder die Gepäckkontrollen am Flughafen, bei denen Koffer und Taschen nach auffälligen und evtl. verbotenen Gegenständen, wie beispielsweise Waffen oder Drogen, gescannt werden. Zum anderen wird den Methoden der Computer Vision auch Vertrauen in kritischen Anwendungsbereichen geschenkt, bei denen teilweise menschliches Eingreifen nicht mehr vorgesehen oder sogar unmöglich ist. Bei einem der größten Anwendungsgebiete der Computer Vision, dem militärischen Einsatz, ist dabei eine hohe Zuverlässigkeit von großer Bedeutung. So existieren beispielsweise sehr fortgeschrittene Raketenleitsysteme, die nicht streng an vorgegebenen Zielkoordinaten festhalten, sondern ihr Ziel anhand neuer, vor Ort erworbener, visueller Informationen selbstständig anpassen.

Die genannten Anwendungsbereiche stellen nur einen kleinen Ausschnitt in der sich stetig und schnell entwickelnden Welt der Computer Vision dar. Um neue Projekte sinnvoll anzugehen, ist fundiertes Wissen über bisherige Erkenntnisse unerlässlich.

Diese Arbeit beschäftigt sich mit der Klassifikation von Baumspezies anhand von Bildern, d. h. Fotografien oder Scans ihrer Blätter. Zur Bestimmung der jeweiligen Spezies dienen charakteristische Merkmale, wie die Beschaffenheit der äußeren Kontur oder die Färbung des Blattes.

Zu Beginn ist eine ausführliche Zusammenfassung von Methoden der Bildverarbeitung zur automatischen und Computer-basierten Erkennung und Klassifikation von Objekten nötig. Dabei wird zunächst ausführlich der Stand der Technik aufgezeigt, sowie die zugehörigen Techniken und Algorithmen anhand von Beispielen zusammengefasst und erläutert. Im zweiten Teil der Arbeit wird das erworbene Wissen praktisch angewendet, indem eine Baumerkennung anhand von Blattmerkmalen implementiert und evaluiert wird.

Das Gebiet der Computer Vision hat sich über Jahrzehnte hinweg stetig weiterentwickelt. Daher existiert bereits eine große Menge sowohl an etablierten Lösungen als auch an möglichen Ansätzen für eine Vielzahl von Problemstellungen. Dieses erarbeitete Wissen liegt in Büchern, wissenschaftlichen Publikationen oder frei zugänglichen Computer Vision-Bibliotheken für verschiedenste Programmiersprachen vor und muss zunächst gesammelt, bewertet und ausführlich zusammengefasst werden. Die so erworbenen Kenntnisse sind unerlässlich für den weiteren Verlauf der Arbeit. Auch wird eine Auseinandersetzung mit einer Computer Vision-Bibliothek für die Programmiersprache Java erfolgen. Da der praktische Teil der Arbeit sich mit Bäumen und deren Blattmerkmalen beschäftigt, ist auch zu diesem Thema eine ausführliche Einarbeitung vonnöten.

Die Blätter der zu bestimmenden Bäume liegen als Fotos vor. Genutzt werden zum einen die im Internet frei verfügbaren Datensätze Flavia (vgl. [Flavia]), Foliage (vgl. [Foliage]), ICL (vgl. [ICL]), Middle European Woody (vgl. [MEW]) und das Swedish Leaf Dataset (vgl. [SLD]), in denen eine Vielzahl von Blättern verschiedener Bäume zum Teil vorsegmentiert, also vom Hintergrund freigestellt, zusammengetragen wurden. Zum anderen wird eine Bilddatenbank erstellt, in der Fotos von zehn einheimischen,

d. h. „ursprünglich“ in der Oberpfalz wachsenden Bäumen gesammelt wurden. Die Blätter werden dabei sowohl direkt am Baum, also vor natürlichem Hintergrund, fotografiert, als auch auf weißem Hintergrund, um die automatische Segmentierung des Blattes zu erleichtern.

Bei der Beschreibung des Standes der Technik wird zunächst ein detaillierter Überblick über grundlegende und fortgeschrittene Herangehensweisen im Bereich der Computer Vision gegeben. Dabei steht im Vordergrund, dass nachfolgende Arbeiten auf den erlangten Erkenntnissen aufbauen können. Der Stand der Technik umfasst auch eine Vorstellung der Open Source Computer Vision-Bibliothek OpenCV (vgl. [OpenCV]), der Data-Mining-Tools Weka (vgl. [Weka]) und der Support Vector Machine-Bibliothek LibSVM (vgl. [LibSVM]).

Anschließend wird das erworbene Wissen genutzt, um Bäume anhand ihrer Blätter zu klassifizieren. Dabei werden verschiedene Merkmale unterschiedlicher Komplexität in Betracht gezogen: zum einen relativ simple Feature-Klassen, wie bestimmte Größenverhältnisse oder Abstandseigenschaften der Blattkontur (vgl. [Wu 2007]) oder statistische Momente der Farbe des Blattes (vgl. [Kadir 2014]); zum anderen komplexe Merkmale, wie die HOCS-Features (vgl. [Kumar 2012]), die eine deutlich exaktere Charakterisierung der äußeren Kontur des Blattes zulassen. Der Prozess der Feature-Extraktion erfolgt mit Hilfe der bereits erwähnten OpenCV-Bibliothek.

Zur letztendlichen Bestimmung des Baumes dienen Machine Learning-Konzepte, wie die Support Vector Machine oder der K-Nearest-Neighbours-Klassifikator. Dazu wird das Data-Mining-Tool Weka verwendet.

Abschließend wird ein Programm entwickelt, das anhand einer Fotoaufnahme eines Blattes den zugehörigen Baum identifizieren kann (vgl. [Leafsnap]). Die Klassifikation erfolgt dabei vollkommen autonom. Lediglich bei der Segmentierung des Blattes von einem komplexen Hintergrund kann mit möglichst geringer menschlicher Unterstützung gearbeitet werden.

Bei der praktischen Umsetzung müssen vor dem Feature-Extraktionsprozess zunächst die Blätter segmentiert, also vom Hintergrund freigestellt werden. Bei den am Baum fotografierten Blättern erfolgt dieser Schritt unter Verwendung des GrabCut-Algorithmus (vgl. [Rother 2004]) und mit menschlicher Unterstützung. Gescannte Blätter können leicht mit Hilfe von Otsus Methode (vgl. [Otsu 1979]) segmentiert werden. Bei den auf Papier fotografierten Blättern gestaltet sich die automatische Segmentierung als schwierig, hauptsächlich aufgrund des teils sehr starken Blattschattens. Es werden verschiedene Standardmethoden zur Segmentierung getestet und jeweils die Fehlerrate mit Hilfe eines selbst erstellten Goldstandards ermittelt. Überdies wird ein eigenes System entwickelt, das unter Verwendung des GrabCut-Algorithmus in den meisten Fällen eine zuverlässige und hochwertige Segmentierung ermöglicht. Die Fehlerrate der selbst erarbeiteten Methode ist mit 4,10% signifikant niedriger als das Ergebnis der besten Standardmethode, dem Watershed-Algorithmus (vgl. [Beucher 1991]), mit 5,06%.

Es werden insgesamt sechs Feature-Klassen extrahiert. Diese beschreiben die äußere Kontur und die Färbung der Blätter. Zur Darstellung der Kontur werden sowohl relativ primitive Verfahren, wie die bereits erwähnten einfachen, geometrischen Verhältnisse, als auch komplexere Methoden, wie der HOCS-Ansatz oder Binary Patterns (vgl. [Pietikäinen 2011]), verwendet.

Um die Aussagekraft der extrahierten Features zu überprüfen, kommen sechs Datensets zum Einsatz, die Scans und Fotos von Blättern einer Vielzahl unterschiedlicher Spezies enthalten. Neben fünf online frei verfügbaren Datensets findet auch ein eigens zusammengestellter Datensatz Verwendung. Dieser besteht zum einen aus Bildern, die vor einheitlichem, hellem Hintergrund aufgenommen wurden. Zum anderen wurden die Blätter derselben Spezies auch direkt am Baum, also vor komplexem und stark variierendem Hintergrund aufgenommen. Überdies werden zu Evaluationszwecken weitere, künstliche Datensets aus den oben genannten Bildquellen zusammengestellt. Besonders hervorzuheben ist dabei eine Kombination aller online verfügbaren Datensätze. Dies führt zu einer Ansammlung von 430 verschiedenen Spezies mit knapp 36.000 einzelnen Instanzen.

Es werden Klassifikationsversuche auf allen der sechs Datensets durchgeführt und die Ergebnisse mit denen anderer Systeme verglichen. Die Erkennungsraten liegen dabei, je nach Schwierigkeitsgrad des verwendeten Datensets, im mittleren bis hohen 90%-Bereich und erreichen oder übertreffen damit den State of the Art. Als besonders nützlich erweisen sich dabei die HOCS-Features. Die besten Ergebnisse werden mit Hilfe des KNN-Klassifikators mit $K=1$ und einer Support Vector Machine erzielt.

Die eben erwähnten Resultate werden ausschließlich bei der Klassifikation innerhalb einzelner Datensets erreicht. Verwendet man dagegen ein Datenset als Trainings- bzw. Vergleichsdatsatz und ein anderes als Testdatsatz, nimmt die Anzahl der korrekten Klassifikationsergebnisse drastisch ab. Die Erkennungsrate fällt teilweise auf unter 60%. Besonders auffällig ist dabei, dass die Farb-Features, die bei den Klassifikationen innerhalb einzelner Datensets noch deutlich zu einer hohen Klassifikationsrate beitragen, nun das Ergebnis klar verschlechtern. Dies legt nahe, dass sich Farb-Features nicht ohne weiteres zur Klassifikation eignen, möglicherweise wegen des Einflusses von Umweltfaktoren. Überdies ist zu beobachten, dass die am Baum fotografierten Blätter im Schnitt etwas schlechter klassifiziert werden als die gescannten oder auf Papier aufgenommenen Blätter.

Eine Verbesserung kann durch das Zusammenfassen von Spezies, die derselben Gattung angehören und sich demnach oft stark ähneln, erzielt werden. Außerdem bietet es sich an, nicht nur das Top-Klassifikationsresultat anzugeben, sondern z. B. die fünf passendsten Spezies. Diese Erweiterung führt zu einem signifikanten Anstieg der Klassifikationsergebnisse. Bei dem zum Abschluss erstellten Evaluationsdatenset kann unter Beachtung der Top5-Resultate eine korrekte Klassifikationsrate von 94% erreicht werden. Dabei wird als Trainingsmenge das oben erwähnte, kombinierte Datenset mit 430 Spezies verwendet. Wird die Vergleichsmenge auf 153 in Mitteleuropa vorkommende Spezies begrenzt, werden sogar 98% der Blätter korrekt klassifiziert.

Im Verlauf dieser Arbeit werden zunächst die Grundlagen der Computer Vision, des Machine Learnings und der Blattmerkmale erläutert (Kapitel 2). Anschließend werden die sechs verwendeten und die zwei nicht verwendeten Datensets vorgestellt (Kapitel 3). Im Folgenden wird die Vorverarbeitung der Blätter, insbesondere deren Segmentierung behandelt (Kapitel 4). Die zur Klassifikation verwendeten Features werden in Kapitel 5 erläutert und anschließend zusammen mit dem Gesamtsystem, dessen Konzept in Kapitel 6 vorgestellt wird, evaluiert (Kapitel 7). Das erstellte Programm wird in Kapitel 8 erläutert. Ein kurzer Vergleich mit anderen Domänen erfolgt in Kapitel 9. Im 10. Kapitel werden die erhaltenen Ergebnisse noch einmal zusammengefasst und diskutiert. Anschließend wird ein Ausblick auf mögliche Verbesserungen und Erweiterungen des Systems gegeben.

2 Grundlagen

In einem ersten Schritt werden Grundlagen aus dem Bereich der Computer Vision und des Machine Learning erläutert. Abschließend wird Basiswissen über den Aufbau von Blättern und deren charakteristische Merkmale vermittelt.

2.1 Computer Vision

Unter Computer Vision versteht man die Transformation der von einer Kamera erfassten Daten zu einer neuen Repräsentation hin, um anschließend Entscheidungen zu bestimmten Fragestellungen zu treffen. [Bradski 2008]

Im folgenden Abschnitt werden zu Beginn verschiedene Darstellungsformen von Bildern aufgezeigt. Im Anschluss werden Standardoperationen und Begriffe wie Filter, Kantendetektion, Konturen, Morphologische Operationen und die Distanztransformation erläutert. Abschließend wird die in dieser Arbeit verwendete OpenCV-Bibliothek vorgestellt.

2.1.1 Darstellungsformen von Bildern

Es gibt viele Möglichkeiten, Farbbilder in einer für einen Computer verständlichen Form darzustellen. Im Folgenden sollen die am häufigsten verwendeten Methoden vorgestellt werden.

2.1.1.1 Das RGB-Modell

1850 entwickelte von Helmholtz die sogenannte „Drei-Farben-Theorie“. Danach lässt sich jede beliebige Farbe durch die Kombination verschieden großer Anteile von rotem, grünem und blauem (RGB) Licht darstellen. Dies lässt sich durch die Beschaffenheit des menschlichen Auges erklären, in dem unterschiedliche Rezeptoren, die sogenannten „Zäpfchen“, mehr oder weniger stark auf Licht verschiedener Wellenlängen reagieren. [Reul 2013]

Das aus der Drei-Farben-Theorie abgeleitete RGB-Modell stellt die Intensität des jeweiligen Farbreizes quantisiert dar. Ein Bildpunkt (Pixel) wird dabei durch je einen Rot-, Grün- und Blau-Wert beschrieben. In der Informatik bietet sich hierbei nicht die ursprüngliche Variante, also eine Verwendung eines Wertes zwischen 0 und 1 an, vielmehr wird als Maximalwert eine ganzzahlige Potenz zur Basis 2 verwendet. Je höher der gewählte Wert, desto mehr unterschiedliche Farben können dargestellt bzw. desto feinere Farbunterschiede können visualisiert werden. Besonders häufige Anwendung findet heutzutage die 8- und die 16-Bit-Darstellung, bei der jeder Pixel durch drei Werte im Bereich von 0 bis 255 (entspricht 2^8-1) bzw. 0 bis 65.535 ($2^{16}-1$) beschrieben wird. Während die letztere speziell bei hochauflösenden Fotos zum Einsatz kommt, reicht in vielen technischen Anwendungen die 8-Bit-Darstellung völlig aus und bietet überdies signifikant schnellere Berechnungszeiten. Deshalb findet sie auch in dieser Arbeit Verwendung. [Efford 2000]

Abbildung 1 zeigt den RGB-Farbraum, dargestellt als Würfel. Dieser besteht aus drei zueinander orthogonalen Achsen, von denen jede einer Primärfarbe RGB(Rot, Grün, Blau) entspricht. Der Nullpunkt liegt bei RGB(0, 0, 0), also bei einem Rot-, Grün- und Blau-Wert

von jeweils 0, und entspricht einem komplett schwarzen Pixel. Analog beschreiben die Werte RGB(255, 255, 255) einen völlig weißen Pixel und die Werte RGB(0, 255, 0) ein sattes Grün.

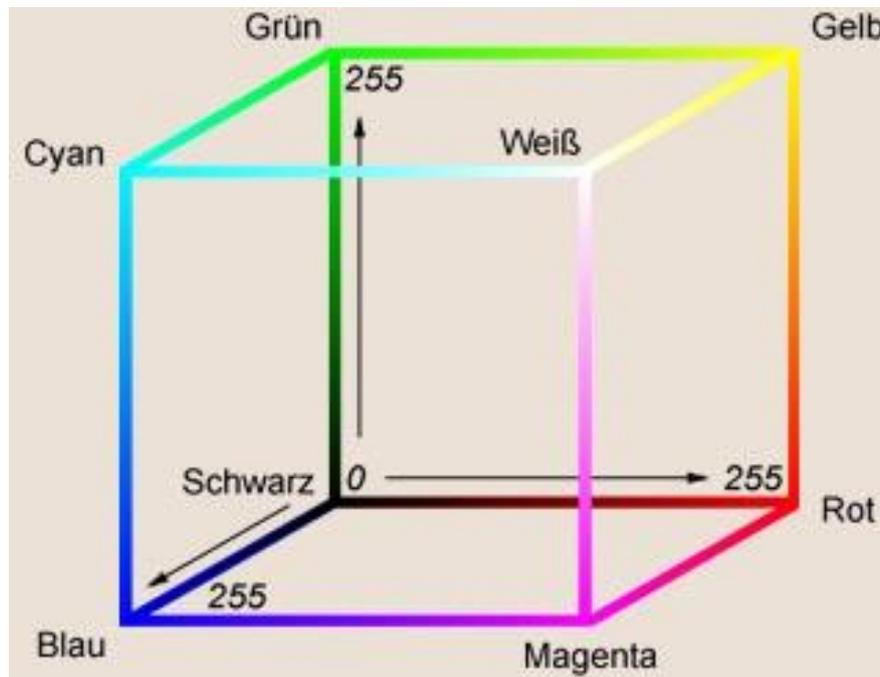


Abbildung 1: Der RGB-Farbraum. [Reul 2013]

Das RGB-Modell ist leicht zu interpretieren und stellt eine Schnittstelle zwischen Computer und Mensch dar. Trotz seiner Einfachheit ermöglicht dieses Modell selbst in der 8-Bit-Darstellung das Unterscheiden von über 16 Millionen Einzelfarben ($256^3=16.777.216$). [Davies 2012]

2.1.1.2 Graustufenbilder

Eine besondere Bedeutung im eben vorgestellten RGB-Farbraum hat die Diagonale, die im Farbwürfel zwischen Schwarz RGB(0, 0, 0) und Weiß RGB(255, 255, 255) verläuft. Alle Punkte auf dieser Geraden haben die Eigenschaft, dass ihre jeweiligen Rot-, Grün- und Blau-Werte identisch sind. Da bei diesen Punkten somit keine Primärfarbe überwiegt, unterscheiden sie sich lediglich durch ihre Helligkeit und verlaufen von Schwarz über Grau bis hin zu Weiß. Deshalb spricht man von sogenannten Graustufen- oder auch Intensitätswerten.

Mit Hilfe von Graustufenwerten ist es möglich, ein Bild anstatt mit drei nur mit einem Wert pro Pixel darzustellen. Dabei wird der Graustufenwert *gray* eines Pixels *x* mit dem Rot-Wert *R*, dem Grün-Wert *G* und dem Blau-Wert *B* wie folgt berechnet [Wu 2007]:

$$gray(x) = 0,299 \cdot x.R + 0,587 \cdot x.G + 0,114 \cdot x.B \quad (2.1)$$

Der auffällig große Vorfaktor beim Grün-Wert ist hierbei auf die erhöhte Grün-Empfindlichkeit des menschlichen Auges zurückzuführen.

Wird die eben beschriebene Transformation für jeden Pixel eines Bildes durchgeführt, erhält man das in Abbildung 2 zu sehende Ergebnis.

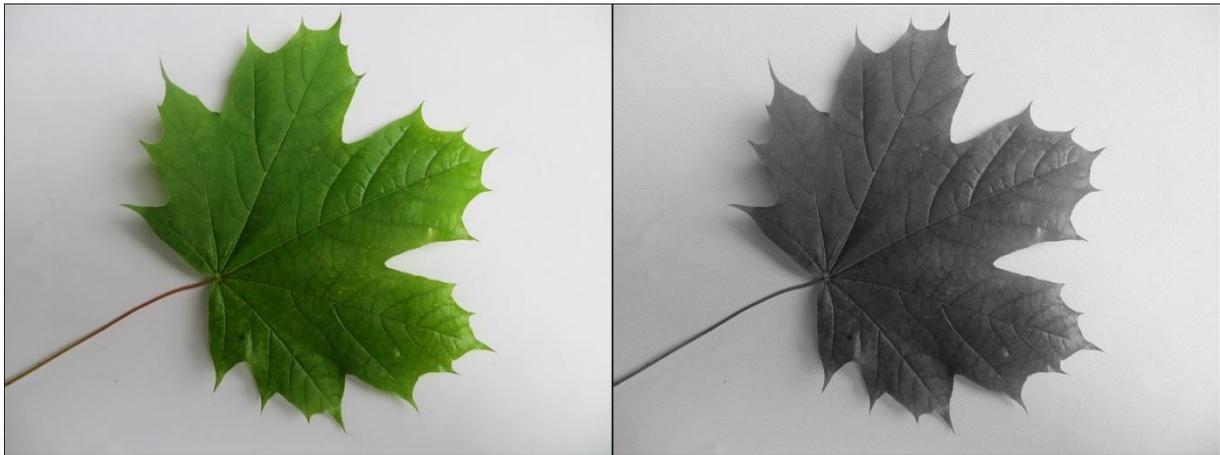


Abbildung 2: Graustufendarstellung eines Blattes. Links: Original, rechts: Graustufenbild.

Der Vorteil von Graustufenbildern liegt zum einen in der Einsparung von Speicherplatz, da pro Pixel nicht drei sondern nur ein 8-Bit-Wert gespeichert werden muss. Des Weiteren ermöglicht diese Darstellung einige Verarbeitungsschritte, wie beispielsweise die in 2.1.3 vorgestellte Kantendetektion. Der offensichtliche Nachteil von Graustufenbildern ist der erlittene Informationsverlust. Eine Rekonstruktion des ursprünglichen Farbbildes ist nicht möglich, da auch farblich völlig unterschiedliche Bilder dasselbe Graustufenbild liefern können.

2.1.1.3 Binärbilder

Wird die in einem Bild enthaltene Information auf ein Minimum reduziert, entsteht ein Binärbild. In diesem existieren lediglich noch zwei mögliche Werte, nämlich 0 und 1.

2.1.1.3.1 Grundprinzip

Um ein Farbbild in ein Binärbild umzuwandeln, erfolgt zuerst die Konvertierung in ein Graustufenbild. Danach wird ein Schwellenwert festgelegt. Pixel, die einen Intensitätswert kleiner als diesen Schwellenwert haben, erhalten den Wert 0, alle anderen den Wert 1 oder umgekehrt. Um eine gut interpretierbare Darstellung zu erreichen, wird im Normalfall den Pixeln mit Wert 1 stattdessen der Wert 255 bzw. der jeweils vorgesehene Maximalwert zugewiesen.

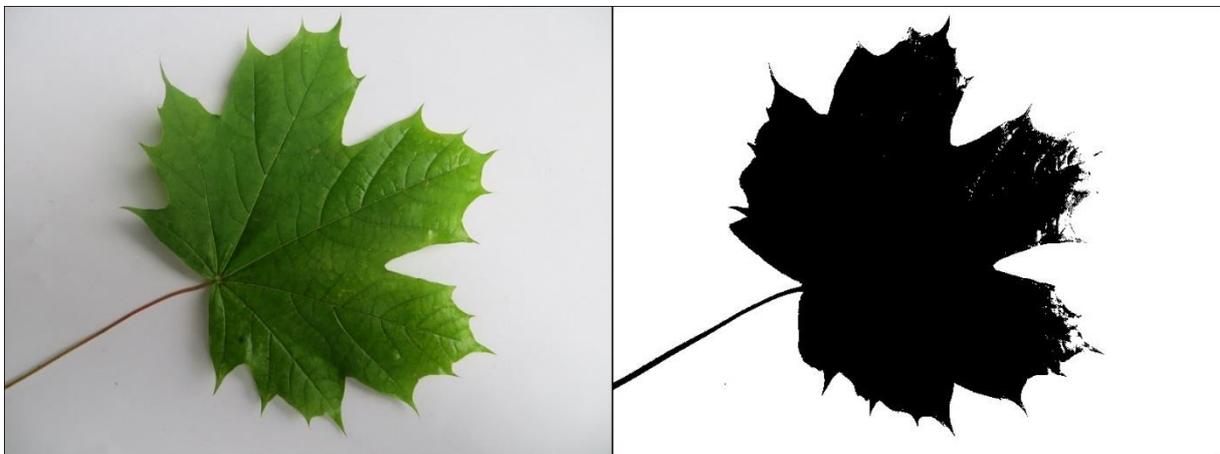


Abbildung 3: Binärdarstellung eines Blattes. Links: Original, rechts: Binärbild nach Otsu.

In realen, technischen Anwendungen ist das Festlegen eines passenden Schwellenwertes oft nicht einfach. Das in Abbildung 3 zu sehende Ergebnis wurde mit Hilfe eines automatisch erstellten Schwellenwertes erzeugt. Dieser wurde wiederum durch das sogenannte Otsu-Verfahren oder Otsus Methode bestimmt.

2.1.1.3.2 Erstellung eines Binärbildes nach Otsu

Otsus Methode hat das Ziel, ohne die Vorgabe weiterer Parameter den idealen Schwellenwert zu bestimmen. Dadurch wird erreicht, dass die Streuung der Graustufenwerte der Pixel, die sich innerhalb der jeweiligen Klasse (0/1) befinden, möglichst klein ist. Gleichzeitig sollen sich die Pixel innerhalb verschiedener Klassen so deutlich wie möglich unterscheiden, also eine große Streuung aufweisen. Als Maß für die Streuung der Graustufenwerte dient die Empirische Varianz. Im Folgenden soll der Algorithmus kurz erläutert werden:

Zu Beginn werden die Pixel in zwei initiale Klassen $K_0(t)$ und $K_1(t)$ eingeteilt. Der zu optimierende Parameter t ist dabei der Schwellenwert, durch den jeder Pixel der jeweiligen Klasse zugewiesen wird.

Jeder Graustufenwert $gray$ tritt mit einer bestimmten Wahrscheinlichkeit $P(gray)$ auf. Für alle Werte gilt dabei $0 < gray < grayMax$, wobei $grayMax$ dem maximal möglichen Graustufenwert entspricht. Bei einer 8-Bit-Auflösung wäre das der Wert 255.

Die Wahrscheinlichkeiten, dass ein Pixel einer der bereits definierten Klassen zugeschrieben wird, betragen also:

$$K_0: P_0(t) = \sum_{gray=0}^t P(gray) \quad (2.2)$$

$$K_1: P_1(t) = \sum_{gray=t+1}^{grayMax} P(gray) = 1 - P_0(t) \quad (2.3)$$

Analog lassen sich die Varianzen innerhalb der beiden Klassen berechnen:

$$\sigma_0^2(t) = \sum_{gray=0}^t (gray - \overline{gray_0})^2 \cdot P(gray) \quad (2.4)$$

$$\sigma_1^2(t) = \sum_{gray=t+1}^{grayMax} (gray - \overline{gray_1})^2 \cdot P(gray) \quad (2.5)$$

Hierbei bezeichnen $\overline{gray_0}$ und $\overline{gray_1}$ die Mittelwerte der Grauwerte innerhalb der beiden Klassen.

Da es nach wie vor das Ziel ist, die Varianz innerhalb der beiden Klassen $\sigma_{in}^2(t)$ zu minimieren und die Varianz zwischen den Klassen $\sigma_{zw}^2(t)$ zu maximieren, ergibt sich ein Optimierungsproblem, bei dem folgender Quotient maximiert werden muss:

$$Q(t) = \frac{\sigma_{zw}^2(t)}{\sigma_{in}^2(t)} \quad (2.6)$$

Die Varianz zwischen den beiden Klassen ist dabei gegeben durch:

$$\sigma_{zw}^2(t) = P_0(t) \cdot (\overline{gray_0} - \overline{gray})^2 + P_1(t) \cdot (\overline{gray_1} - \overline{gray})^2 \quad (2.7)$$

Die Varianz innerhalb der beiden Klassen lässt sich aus der Summe der Einzelvarianzen berechnen:

$$\sigma_{in}^2(t) = P_0(t) \cdot \sigma_0^2(t) + P_1(t) \cdot \sigma_1^2(t) \quad (2.8)$$

Wurde ein Schwellenwert t gefunden, der $Q(t)$ maximiert, so ist die Teilung der Pixel in zwei Klassen entsprechend der Varianz optimal. [Otsu 1979] [Wiki: Otsu]

2.1.1.4 Der HSV-Farbraum

Eine weitere Darstellungsform, die sich, anders als das RGB-Modell, eher an intuitiven Methoden, wie sie z. B. ein Maler anwenden würde, orientiert, wird durch den HSV-Farbraum beschrieben. Dieser lässt sich zwar nicht als Würfel, sondern, wie in Abbildung 4 zu sehen, als auf den Kopf gestellter Kegel darstellen.

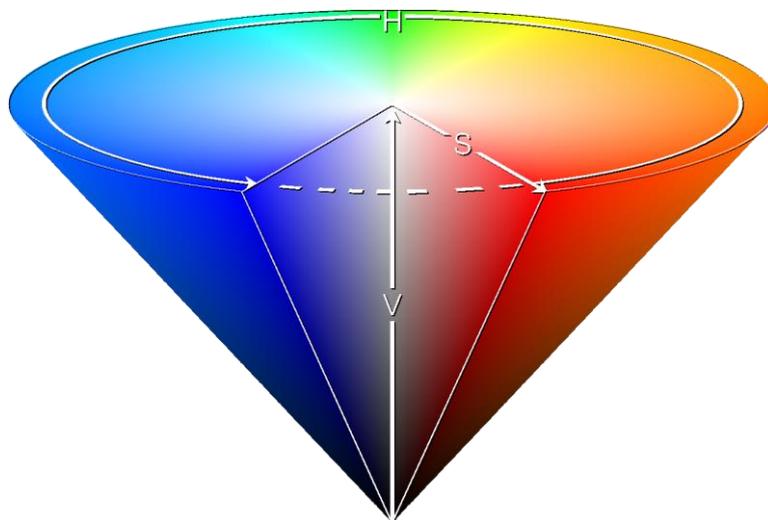


Abbildung 4: Der HSV-Farbraum. [Wiki: HSV]

Genau wie beim RGB-Modell werden auch hier drei Werte pro Pixel benötigt:

Der Farbwert H (englisch *hue*: Farbton, Färbung) liegt zwischen 0° und 360° und gibt die Position auf dem Farbkreis an. Auf diesem liegen sich Komplementärfarben gegenüber, d. h. jeweils um 180° verschoben. Ein H -Wert von 0 entspricht Rot.

Ist H festgelegt, so beschreibt der Sättigungswert S (englisch *saturation*: Sättigung) die Reinheit der Farbe im Verhältnis zu ihrer maximalen Reinheit. Dementsprechend liegt der S -Wert in einem Intervall zwischen 0 und 1 bzw. 0% und 100%, wobei 100% einer komplett gesättigten, also einer „reinen“ Farbe entspricht. Der Nullpunkt des S -Wertes ist ein Graustufenwert, der auf der Verbindung des Mittelpunktes des Farbkreises und der Spitze des Kegels liegt. Der Maximalwert liegt auf dem Farbkreis.

Abschließend gibt der Helligkeitswert V (englisch *value*: Wert) die Dunkelstufe der Farbe an. Bildlich gesprochen beschreibt V , wieviel reines Schwarz oder reines Weiß der ursprünglichen Farbe hinzugefügt werden musste, um die gewünschte Farbe zu erreichen. Dies beeinflusst zusätzlich auch die Sättigung. Genau wie der H -Wert wird der S -Wert in einem Intervall zwischen 0 und 1 bzw. 0% und 100% angegeben. [Hermes 2005]

Für eine reibungslose Verwendung der Werte bietet es sich an, diese jeweils auf einen Bereich von 0 bis 255 abzubilden. Die einzelnen Kanäle der HSV-Darstellung sind in Abbildung 5 zu sehen.

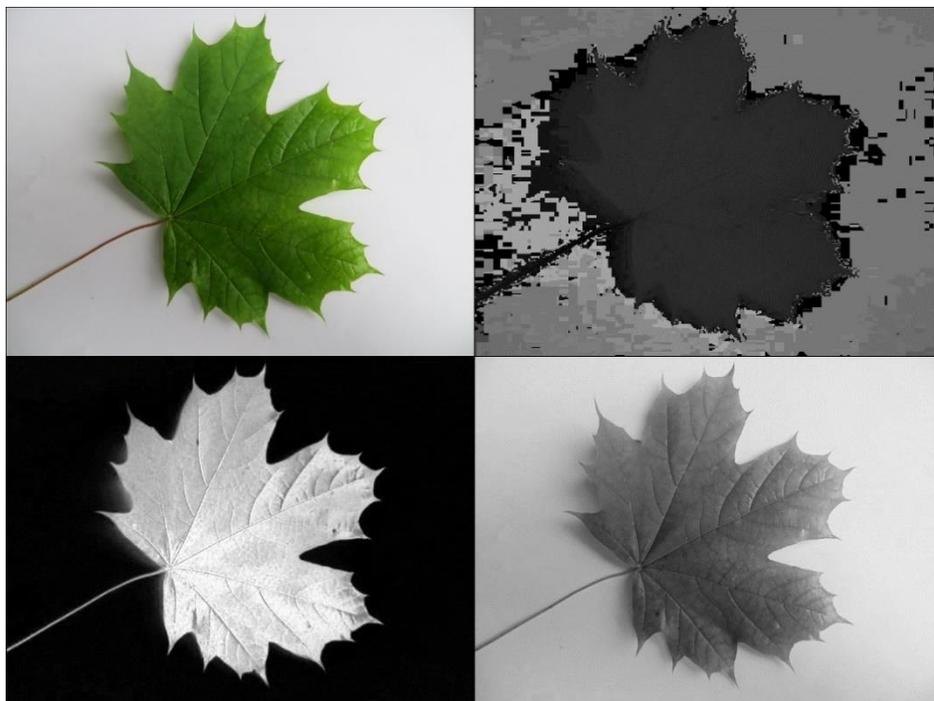


Abbildung 5: HSV-Darstellung eines Blattes. Oben: Originalbild (links), H-Kanal (rechts), unten: S-Kanal (links), V-Kanal (rechts).

Es fällt auf, dass der V -Kanal exakt dem Graustufenbild aus Abbildung 2 entspricht.

2.1.1.5 Der Lab-Farbraum

Eine seltener verwendete, aber für diese Arbeit sehr nützliche Darstellungsform ist der Lab-Farbraum, der sich am besten als Kugel visualisieren lässt (siehe Abbildung 6).

Zum Beschreiben eines Pixels werden wiederum drei Werte benötigt:

Der L -Wert (englisch *lightness*: Helligkeit) entspricht der Graustufendarstellung bzw. dem V -Wert aus dem HSV-Modell. Er erstreckt sich von 0 (tiefstes Schwarz) bis 100 (hellstes Weiß) und gibt die grundsätzliche Helligkeit der Farbe an.

Die a -Koordinate definiert das in der darzustellenden Farbe vorherrschende Verhältnis von Rot zu Grün. Im Normalfall wird dazu ein Intervall von -100 bis +100 verwendet, wobei ein Wert von -100 einem reinen Grün und ein Wert von +100 einem reinen Rot entspricht. 0 steht für ein neutrales Grau.

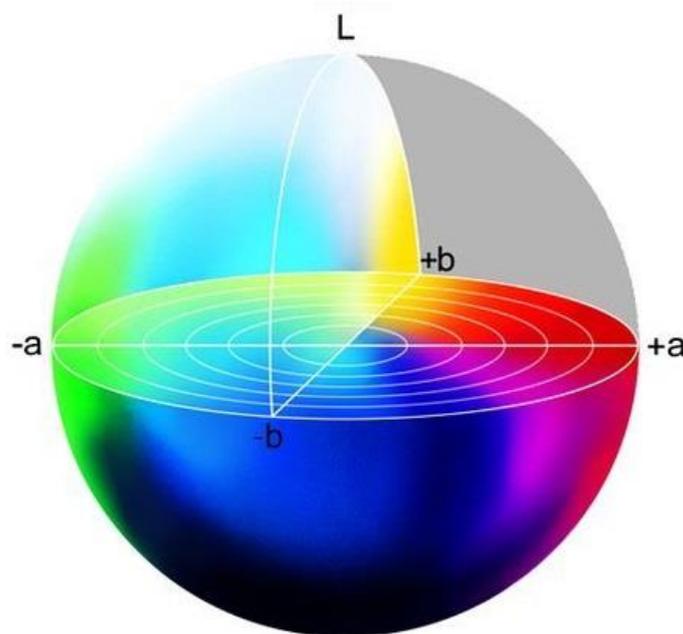


Abbildung 6: Der Lab-Farbraum. [Heise]

Analog zur a -Koordinate verhält sich die b -Koordinate. Hier beschreiben ein Wert von -100 reines Blau und ein Wert von +100 reines Gelb. b stellt also das Verhältnis von Gelb zu Blau in der Farbe dar. [Ebner 2007]

Auch beim Lab-Farbraum empfiehlt es sich, die Werte auf einer gemeinsamen Skala, z. B. 0-255, abzubilden.

2.1.2 Filter

Liegt ein Bild in einer der oben beschriebenen Darstellungen vor, so können darauf verschiedene Operationen durchgeführt werden. Dazu eignet sich häufig eine sogenannte Filtermatrix, wie in Abbildung 7 zu sehen.

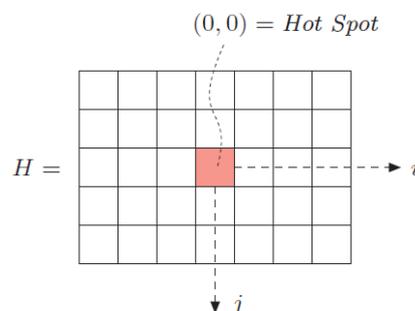


Abbildung 7: Filtermatrix mit Hot Spot. [Burger 2009]

Die Anwendung der Filtermatrix ist schematisch in Abbildung 8 dargestellt.

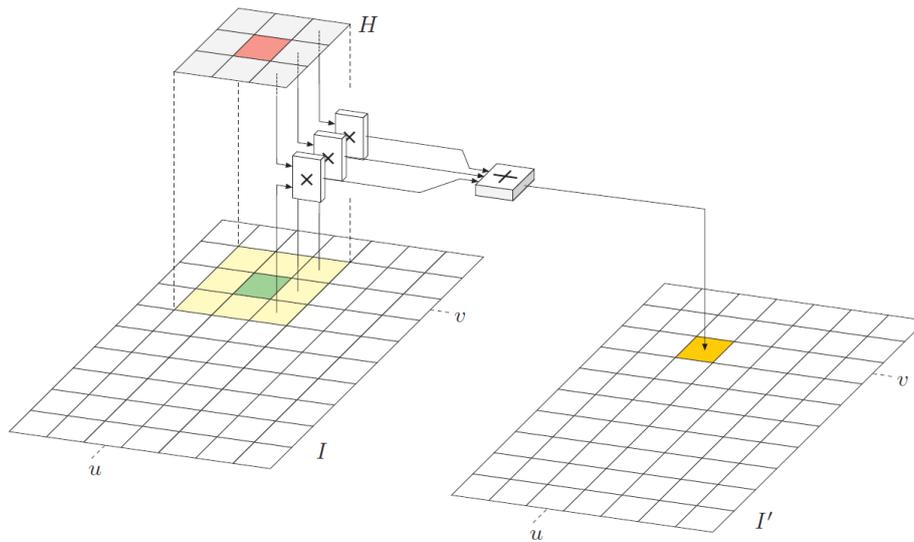


Abbildung 8: Anwendung einer Filter-Matrix. [Burger 2009]

Für jeden Pixel (u, v) im Originalbild I wird die Filtermatrix H so platziert, dass sich der sogenannte Hot Spot (hier: rot) über dem jeweiligen Pixel befindet. Anschließend werden alle Filterkoeffizienten $H(i, j)$ mit dem korrespondierenden Bildelement $I(v+i, u+i)$ multipliziert und die Ergebnisse addiert. Zuletzt wird die Summe an der entsprechenden Position im Ergebnisbild I' gespeichert. [Burger 2009]

Ein Anwendungsbeispiel für Filter-Matrizen ist das sogenannte Blurring, bei dem feine Konturen geglättet werden und das Bild insgesamt unschärfer wird. Dies klingt zunächst wenig erstrebenswert, ist aber in der Bildverarbeitung ein oft unverzichtbarer Vorverarbeitungsschritt. So ist es mittels Blurring beispielsweise möglich, ein Bild auf seine wesentlichen Informationen zu reduzieren. Dabei bleiben gut erkennbare Grenzen zwischen Objekten erhalten, während feine Konturen auf der Oberfläche geglättet und damit eliminiert werden. [Reul 2013]

Abbildung 9 zeigt Ergebnisse des *Blurrings* mit einer Filter-Matrix folgender Form:

$$H = \frac{1}{\#ones} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \quad (2.9)$$

Bei Anwendung dieser Matrix werden die Pixelwerte im Einzugsbereich der Matrix addiert und anschließend durch die Anzahl der aufsummierten Werte dividiert. Man spricht von einer Mittelwertfilterung.



Abbildung 9: Blurring eines Blattes. Links: Original, Mitte: Ergebnis mit 11x11-Filter-Matrix, rechts: Ergebnis mit 23x23-Filter-Matrix.

2.1.3 Kantendetektion

Eine der wichtigsten Filter-Anwendungen ist die Kantendetektion, mit deren Hilfe sich die Form eines Objektes genauer beschreiben lässt. Die Form ist neben der Farbe das wichtigste Merkmal, das ein Objekt auszeichnet.

2.1.3.1 Grundprinzip

Kanten sind die Stellen eines Bildes, an denen der Gradient des Bildes verhältnismäßig hoch ist. Unter dem Gradienten versteht man die Änderung des Intensitätswertes zwischen aneinander angrenzenden Pixel. Dieser lässt sich mit verschiedenen Filtermatrizen berechnen.

Die am häufigsten genutzten Filter sind die Sobel-Operatoren:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.10)$$

und

$$G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.11)$$

Diese berechnen getrennt den Wert der Gradienten für die horizontale X - und die vertikale Y -Richtung. Es fällt auf, dass die direkt vertikal und horizontal anliegenden Intensitätswerte doppelt so stark gewichtet werden wie die schräg benachbarten. Anschließend werden die beiden Werte wie folgt kombiniert:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.12)$$

Des Weiteren ist es auch möglich, die Richtung φ des Gradienten und damit die Richtung der detektierten Kante zu berechnen. Dies erfolgt mittels:

$$\varphi = \tan^{-1} \left(\frac{G_x}{G_y} \right) \quad (2.13)$$

Dabei hat eine vertikal verlaufende Kante den Wert 0. Positive, ansteigende Werte entsprechen einer Drehung gegen den Uhrzeigersinn.

[Bradski 2008] [Burger 2009a] [Reul 2013]

Die Kantendetektion erfolgt auf Graustufenbildern. Meist ist es hilfreich, die Bilder vorher zu blurren, um Rauschen zu unterdrücken und damit unwichtige Kanten zu entfernen. Dies ist in Abbildung 10 deutlich zu erkennen. Während im mittleren, vorher nicht geblurrten Bild eine Vielzahl kleiner Kanten im Inneren des Bildes detektiert wurde, sind im rechten Bild fast ausschließlich die Kanten zu sehen, die die äußere Kontur des Blattes bilden. Beide Resultate wurden mit Hilfe des Canny-Algorithmus erstellt.

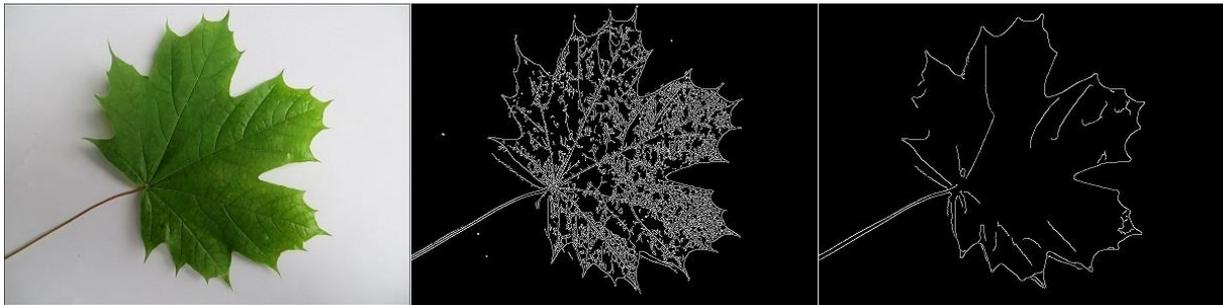


Abbildung 10: Canny-Kantendetektion am Beispiel eines Blattes. Links: Original, Mitte: Kantendetektion ohne vorheriges Blurring, rechts: Kantendetektion mit vorherigem Blurring.

2.1.3.2 Der Canny-Kantendetektor

In der Praxis wird im Normalfall die Kantendetektionsmethode von Canny (vgl. [Canny 1986]) verwendet. Diese nimmt, zusätzlich zur normalen Kantendetektion, weitere Nachverarbeitungen vor, um bessere Detektionsergebnisse zu erzielen. Der gesamte Vorgang wird im Folgenden beschrieben:

Zunächst sieht der Original-Algorithmus eine Rauschreduktion mittels eines Gauß-Filters der Dimension 5×5 vor. In modernen Anwendungen wird dieser Schritt allerdings meist weggelassen. Der Grund dafür ist, dass im Normalfall eine zusätzliche Vorverarbeitung vorgenommen wird, bei der der zu verwendende Filter und dessen Größe exakt auf die Dimension und Art des Eingabebildes und auf das gewünschte Ergebnis abgestimmt werden. Im in Abbildung 10 zu sehenden Fall wurde z. B. rechts eine Mittelwertfilterung mit einem 11×11 -Filter vorgenommen.

Anschließend werden die X - und Y -Gradienten wie gewohnt berechnet und die Gesamtstärke und Richtung des Gradienten für jeden Pixel bestimmt. Die Richtung wird dabei auf vier mögliche Werte gerundet: 0° , 45° , 90° und 135° .

Mit Hilfe der ermittelten Werte können nun sogenannte Nicht-Maxima unterdrückt werden. So wird ein Pixel nur noch dann als Kantenpixel anerkannt, wenn sein Gradient größer ist als seine in der Kantenrichtung liegenden Nachbarpixel. Dies führt im Ergebnis zu dünneren und damit klareren Kanten.

Der abschließende Schritt ist eine doppelte Grenzwertoperation. Dazu müssen vom Nutzer zwei Grenzwerte definiert werden, ein unterer und ein oberer. Dabei gilt, dass der obere größer oder gleich dem unteren sein muss und beide im Bereich zwischen 0 und dem maximal möglichen Intensitätswert des Eingabebildes liegen müssen. Bei einem 8-Bit-Bild, wie im hier vorliegenden Fall, muss er sich zwischen 0 und 255 befinden. Pixel werden als Kantenpixel akzeptiert, wenn ihr Gradient über dem oberen Grenzwert liegt. Falls der Wert niedriger als der untere Grenzwert ist, erfolgt eine direkte Abweisung. Liegt der Wert zwischen beiden Grenzen, wird der Pixel nur dann akzeptiert, wenn er sich in direkter Nachbarschaft zu einem Pixel befindet, der bereits als Kantenpixel akzeptiert wurde. Dieses Vorgehen unterstützt die oft gewünschte Konturfindung, die im nächsten Abschnitt näher erläutert werden soll. [Brahmbhatt 2013]

Abschließend ist zu erwähnen, dass die Bestimmung der optimalen Grenzwerte sich oft sehr schwierig gestaltet und stark vom Eingabebild und dem gewünschten Ergebnis abhängt. Ein guter Ansatz ist es, einen oberen Grenzwert zu wählen, der zwei- bis dreimal

höher liegt als der untere. Die in Abbildung 10 zu sehenden Ergebnisse entstanden mittels eines unteren Grenzwertes von 20 und eines oberen von 60.

2.1.4 Konturen

Es besteht ein wichtiger Unterschied zwischen Kanten und Konturen. Kanten beschreiben lediglich lokale Maxima der Intensitätsgradienten in einem Bild. Dies führt dazu, dass Ergebnisbilder, wie im vorherigen Abschnitt zu sehen, oft sehr verrauscht sind und die Kantenpunkte häufig keinerlei Verbindung zueinander haben. Eine Ausnahme bilden die mit Hilfe des Canny-Algorithmus detektierten Kanten, da diese noch eine Reihe von Nachverarbeitungsschritten durchlaufen, die fortlaufende Kanten fördern sollen. Konturen dagegen beschreiben stets eine zusammenhängende Menge von Punkten. In dieser Arbeit wird diesbezüglich immer nur die äußere Kontur eines Blattes betrachtet.

Im Folgenden soll kurz die Konturdetektion erläutert werden. Anschließend werden einige Beispiele für nützliche, in der OpenCV-Bibliothek implementierte Funktionen gegeben, die es erlauben, Informationen aus einer gegebenen Kontur zu extrahieren.

2.1.4.1 Konturdetektion

Zum Auffinden von Konturen in einem Bild wird im Normalfall als Eingabe ein Binärbild verwendet. OpenCV bedient sich zur Detektion der Konturen des Algorithmus von Suzuki und Abe (vgl. [Suzuki 1985]), der in dieser Arbeit nicht näher erläutert werden soll, da er sehr komplex ist und nur wenige der dort behandelten Spezialfälle für die hier genutzten Bilder benötigt werden. Grundsätzlich ist das Ziel, zusammenhängende Kantenpunkte zu detektieren.

„Zusammenhängend“ lässt sich dabei auf zwei unterschiedliche Weisen definieren, wie in Abbildung 11 zu sehen ist.

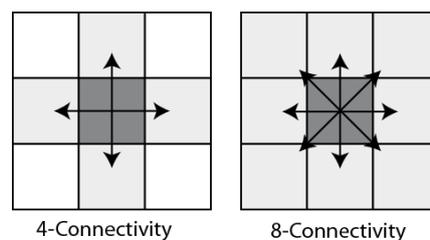


Abbildung 11: Zwei unterschiedliche Definitionen zusammenhängender Pixel. [University of Alberta]

In der linken Darstellung werden lediglich die direkt horizontal und vertikal angrenzenden Pixel als Nachbarn definiert, rechts kommen zusätzlich die schräg anschließenden Pixel hinzu. Im weiteren Verlauf wird stets die Definition mit acht Nachbarn verwendet. [Brahmbhatt 2013]

Des Weiteren wird in dieser Arbeit jeweils nur die äußerste und damit größte Kontur extrahiert. Dabei ist zu beachten, dass der oben erwähnte Algorithmus voraussetzt, dass der Hintergrund und damit auch der Rand im Binärbild schwarz ist und mit 0 kodiert wird (vgl. [Bradski 2008]). Dies muss bei der Umwandlung in ein Binärbild berücksichtigt werden, da die Bilder meist so segmentiert werden, dass der Hintergrund weiß ist und das Blatt farblich dargestellt wird. Wandelt man ein solches Bild mit einem

Standardalgorithmus in ein Binärbild um, bleibt der Hintergrund weiß und der Blattbereich wird schwarz dargestellt.

Die Folgen bei der Konturdetektion sind in Abbildung 12 zu sehen.

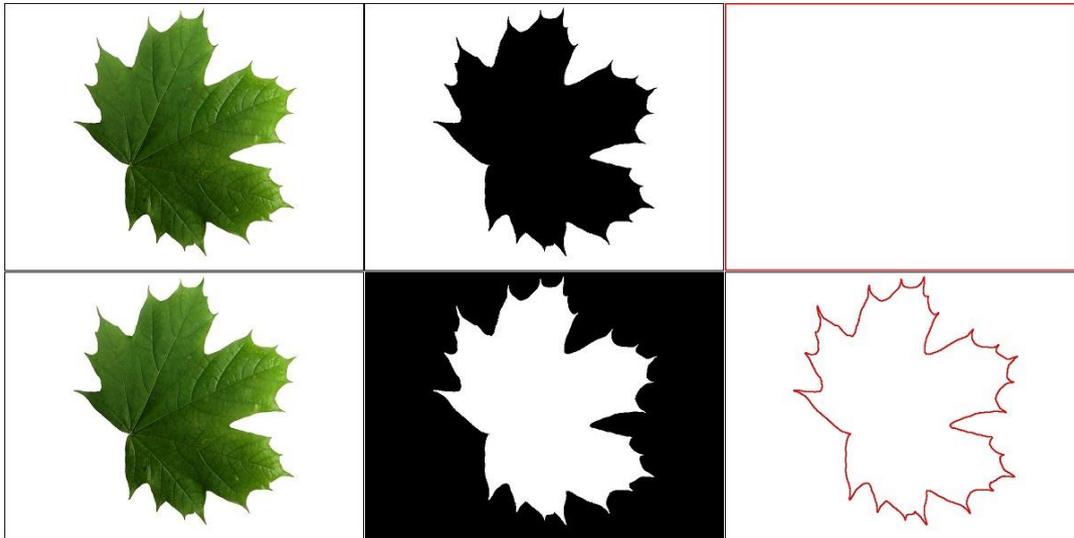


Abbildung 12: Einfluss der Wertverteilung im Binärbild. Oben: Standardkodierung (Vordergrund 1, Hintergrund 0), unten: von OpenCV erwartete Kodierung (Hintergrund 0, Hintergrund 1). Links: Original, Mitte: Binärbild, rechts: detektierte Kontur.

Ist der Randbereich des Bildes nicht mit 0 kodiert, wird der äußere Bildrand als Kontur detektiert. Die Blattkontur wird verworfen, da lediglich die größte Kontur betrachtet wird.

2.1.4.2 Nützliche Kontureigenschaften und –funktionen

OpenCV stellt eine Vielzahl von Funktionen und Methoden zur Verfügung, die das Arbeiten mit Konturen deutlich erleichtern. So lassen sich zum einen sehr einfach bestimmte Kontureigenschaften, wie deren Länge und Fläche, in Pixeln berechnen. Zum anderen ist es möglich, eine Vielzahl von Formen an die Kontur anzupassen. Diese Formen spielen bei der späteren Feature-Extraktion noch eine wichtige Rolle. Sie lassen sich unterteilen in die eckigen Formen, die in Abbildung 13 zu sehen sind, und die rundlichen Formen (siehe Abbildung 14).

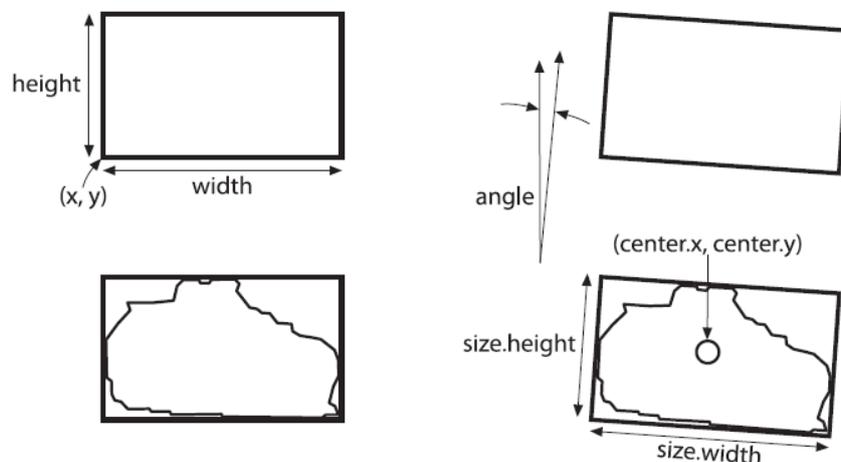


Abbildung 13: Zwei Arten von an Konturen angepasste Rechtecke. Links: `StraightBoundingRect`, rechts: `MinimumBoundingRect`. [Bradski 2008]

Die eckigen Formen beinhalten zum einen das sogenannte *StraightBoundingRect*, also das kleinste, aufrecht stehende Rechteck, das alle Konturpunkte völlig einschließt. Es wird eindeutig definiert durch seine Länge und Breite und die x - und y -Koordinaten seines linken unteren Eckpunktes. Zum anderen lässt sich auch das sogenannte *MinimumBoundingRect* an die Kontur anpassen. Dieses erweist sich als deutlich nützlicher, da es unabhängig von der Ausrichtung der Kontur immer das Rechteck mit minimaler Fläche um die Kontur bildet. Es ist also rotationsinvariant. Eindeutig definiert ist es durch seinen Mittelpunkt, die Seitenlängen des Rechtecks und den Winkel, um den es rotiert wurde.

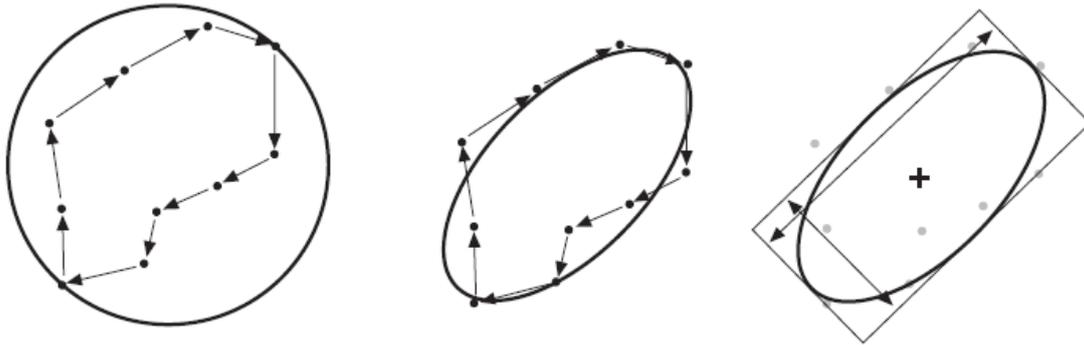


Abbildung 14: An Konturen angepasste, rundliche Formen. Links: *MinimumEnclosingCircle*, Mitte: *BestFittingEllipse*, rechts: Das die Ellipse repräsentierende Rechteck. [Bradski 2008]

Die rundlichen Formen beinhalten einerseits den sogenannten *MinimumEnclosingCircle*, der analog zum *MinimumBoundingRect* der Kreis ist, der mit minimaler Fläche alle Konturpunkte einschließt. Wie jeder andere Kreis auch, wird er in OpenCV anhand seines Mittelpunktes und seines Radius beschrieben. Andererseits gibt es die *BestFittingEllipse*, also die Ellipse, die am besten in das Punktefeld eingepasst werden kann. Dabei wird weder gefordert, dass alle Konturpunkte innerhalb der Ellipse liegen, noch muss die Ellipse komplett innerhalb der Punkte verlaufen. Beschrieben wird sie durch ihr *MinimumBoundingRect*. [Bradski 2008] [Brahmbhatt 2013]

2.1.5 Morphologische Transformationen

Die Morphologie (griechisch $\mu\omicron\rho\phi\acute{\eta}$: Gestalt, Form) ist die Lehre von der Struktur und Form. In der Bildverarbeitung beschränkt sie sich ausschließlich auf Binärbilder und erfüllt dort eine Vielzahl von Zwecken. So kommt sie beispielsweise beim Entfernen von Rauschen oder beim Isolieren und Zusammenfügen einzelner Elemente zum Einsatz.

Im Folgenden sollen zunächst die Basisoperationen Dilatation und Erosion vorgestellt werden. Aufbauend darauf folgen dann weitere Operationen wie das Opening, Closing, Top Hat und Black Hat. Für jede dieser Anwendungen wird ein Strukturelement benötigt. Dieses entspricht im Prinzip einer Filtermatrix, mit dem Unterschied, dass ein Strukturelement nicht immer rechteckig sein muss, sondern fast jede erdenkliche Form annehmen kann. In dieser Arbeit finden aber lediglich rechteckige und kreisförmige Strukturelemente Verwendung.

2.1.5.1 Dilatation und Erosion

Wie bereits erwähnt, sind Dilatation und Erosion die grundlegendsten aller morphologischen Transformationen. Bei beiden wird ein Strukturelement beliebiger Form über das Bild geschoben. Jedes Strukturelement verfügt über einen Ankerpunkt, der nacheinander über jedem Pixel des zu bearbeitenden Bildes platziert wird. Anschließend wird bei der Dilatation der minimale Pixelwert des vom Strukturelement überlagerten Bereichs bestimmt. Dieser Wert wird dem unter dem Ankerpunkt liegenden Pixel zugewiesen. Die Erosion erfolgt analog, nur wird hierbei der maximale Wert bestimmt und zugewiesen. Abbildung 15 zeigt die Auswirkung der beiden Operationen auf das Binärbild eines Blattes. Bei der Dilatation vergrößert sich die Fläche des Vordergrundes und spitz zulaufende Enden fallen deutlich massiver aus. Genau das Gegenteil geschieht bei der Erosion: Die Fläche des Vordergrundes nimmt ab und vor allem die Spitzen verlieren signifikant an Größe und werden abgestumpft.



Abbildung 15: Dilatation und Erosion am Beispiel eines Blattes. Links: Binärbild, Mitte: Ergebnis der Dilatation, rechts: Ergebnis der Erosion (Filter jeweils 21x21).

Die Dilatation kommt vor allem dann zum Einsatz, wenn es um das Auffinden verbundener Komponenten im Bild geht, z. B. bei der Konturdetektion. Sie sorgt dafür, dass eine große, zusammengehörige Region wieder verbunden wird, falls diese zuvor durch Schatten oder Rauschen aufgeteilt wurde. Analog dazu dient die Erosion in erster Linie der Beseitigung des sogenannten Sprengel-Rauschens. Kleine, zufällig entstandene Flecken Vordergrund werden zu Hintergrund, während große Flächen weitestgehend unangetastet bleiben. [Bradski 2008] [Brahmbhatt 2013]

2.1.5.2 Opening und Closing

Aufbauend auf den eben vorgestellten Verfahren der Dilatation und Erosion lassen sich zwei weitere wichtige morphologische Transformationen definieren: das Opening und das Closing.

Unter Opening versteht man die Erosion eines Binärbildes, gefolgt von einer Dilatation des Ergebnisbildes. Diese Technik eignet sich besonders, kleine Vordergrundobjekte zu beseitigen. Analog dazu ist das Closing eine Dilatation gefolgt von einer Erosion. Es ermöglicht kleine, vom Vordergrund umgebene Hintergrundbereiche zu füllen.

Opening: result = dilate(erode(source, structuring element))

Closing: result = erode(dilate(source, structuring element))

Beide Operationen sind in Abbildung 16 zu sehen. [OpenCV: Morph]



Abbildung 16: Ergebnisse einer Opening- und einer Closing-Operation. Links: Opening, rechts: Closing. Jeweils Ausgangsbild (links) und Ergebnis (rechts). [OpenCV: Morph]

2.1.5.3 Top Hat und Black Hat

Die letzten beiden hier vorzustellenden Operationen sind der Top Hat und der Black Hat. Sie sind wie folgt definiert:

Der Top Hat ist die Differenz aus dem Ausgangsbild und dem Ergebnis der Opening-Operation. Unter Black Hat versteht man die Differenz aus dem Ergebnis der Closing-Operation und dem ursprünglichen Bild:

Top Hat: result = source – open(source, structuring element)

Black Hat: result = close(source, structuring element) – source

Beide Operationen ermöglichen es, Bereiche zu extrahieren, die heller (Top Hat) oder dunkler (Black Hat) sind als ihre direkten Nachbarn. [OpenCV]



Abbildung 17: Ergebnisse einer Top Hat- und einer Black Hat-Operation. Links: Top Hat, rechts: Black Hat. Jeweils Ausgangsbild (links) und Ergebnis (rechts). [OpenCV: Morph]

2.1.6 Distanztransformation

Die Distanztransformation eines Bildes ist definiert als ein neues Bild derselben Dimension, in dem jeder Pixel den Abstand zu seinem nächstgelegenen Hintergrundpixel als Wert erhält.

Eine schematische Darstellung des Vorgangs ist in Abbildung 18 zu sehen.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	1	1	1	1	1	0
0	1	1	1	1	1	0	0	1	2	2	2	1	0
0	1	1	1	1	1	0	0	1	2	3	2	1	0
0	1	1	1	1	1	0	0	1	2	2	2	1	0
0	1	1	1	1	1	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Abbildung 18: Schematische Darstellung einer Distanztransformation. Links: Binärbild, rechts: Ergebnis der Distanztransformation. [Wiki: DT]

Offensichtlich kann das Ergebnis als Graustufenbild aufgefasst werden. Ursprüngliche Hintergrundpixel behalten den Wert 0. Alle anderen erhalten, je nach Distanz zum nächsten Hintergrundpixel, einen positiven Wert. Je höher die Distanz desto größer der zugewiesene Wert.

Abbildung 19 zeigt das Ergebnis einer Distanztransformation. Deutlich sind die Maxima im Inneren des Blattes zu erkennen. Es ist anzumerken, dass vor der Visualisierung der Transformation als Graustufenbild noch ein weiterer Verarbeitungsschritt erfolgen muss. Die Distanzen können je nach Aufbau und Größe des Bildes stark variieren und auch Werte annehmen, die oberhalb des erlaubten Bereichs, in diesem Fall zwischen 0 und 255, liegen. Deswegen müssen nach erfolgter Berechnung die Werte normalisiert werden.

Dazu wird zunächst der größte vorkommende Distanzwert max bestimmt. Anschließend wird die normalisierte Distanz $normalized$ an der Position (x, y) aus der ursprünglichen Distanz $dist$ wie folgt berechnet:

$$normalized(x, y) = \frac{dist(x, y) \cdot 255}{max} \quad (2.14)$$

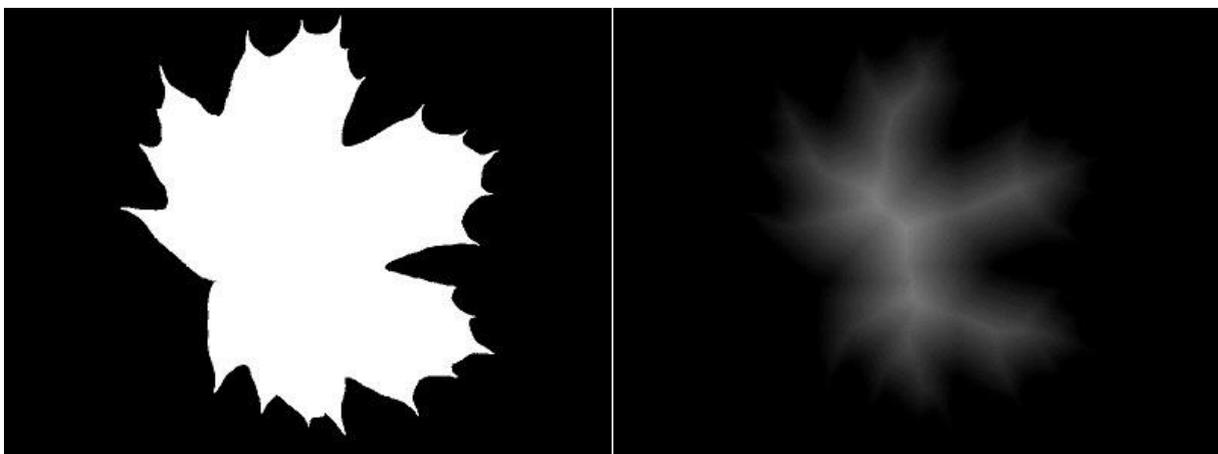


Abbildung 19: Die Distanztransformation am Beispiel eines Blattes. Links: Binärbild, rechts: Ergebnis der Distanztransformation.

Die Bedeutung der Distanztransformation wird bei der späteren Segmentierung in Kapitel 4 deutlich werden. Bei Segmentierungsaufgaben ist es oft der Fall, dass der innere Teil schnell und sicher als Vordergrund klassifiziert werden kann, während die Randbereiche schwierig zuzuordnen sind. Mit Hilfe der Distanztransformation kann leicht ein Teil des Blattes bestimmt werden, der sicher zum Vordergrund gehört. Meist erleichtert diese Erkenntnis die weitere Segmentierung erheblich.

2.1.7 Die OpenCV-Bibliothek

OpenCV (Open Source Computer Vision Library) ist die größte und verbreitetste Open Source Software-Bibliothek für Computer Vision und Machine Learning. Dies zeigen mehr als 7 Millionen Downloads und die Tatsache, dass renommierte Unternehmen wie Google, Microsoft und Yahoo auf die Bibliothek zurückgreifen.

Mittlerweile verfügt sie über mehr als 2500 optimierte Algorithmen. Diese decken sowohl grundlegende Funktionen, wie die Umwandlung von Bildern in verschiedene Farbräume oder die Kantendetektion, als auch komplexe Algorithmen zur Gesichtserkennung oder zur Erstellung von 3D-Punktwolken mittels Stereokameras ab.

OpenCV wurde ursprünglich in C++ programmiert und besitzt Schnittstellen zu C, Python, Java und MATLAB. Die Verwendung der Bibliothek wird unter Windows, Linux, Android und Mac OS unterstützt. Des Weiteren gibt es auf der Projekt-Homepage <http://www.opencv.org> eine ausführliche Dokumentation sowie zahlreiche Tutorials, die den Umgang mit der Bibliothek erleichtern.

[OpenCV]

Sowohl der Umfang als auch die leichte Handhabung führten dazu, dass OpenCV in dieser Arbeit genutzt wird. Es gibt natürlich, je nach verwendeter Programmiersprache, noch andere Computer Vision Bibliotheken, aber keine verfügt auch nur ansatzweise über den Umfang und die Ausgereiftheit der OpenCV-Bibliothek. Nach Ansicht dieser Arbeit existiert keine nennenswerte Alternative.

2.2 Machine Learning

Zur letztendlichen Bestimmung der Spezies, zu der ein Blatt gehört, sind Methoden aus dem Bereich des Machine Learnings nötig. Im Folgenden werden zunächst die beiden in dieser Arbeit verwendeten Klassifikatoren vorgestellt. Dabei handelt es sich um die Support Vector Machine (SVM) und den K-Nearest-Neighbours-Klassifikator (KNN). Im Anschluss werden Evaluationsmethoden aufgezeigt, die zum einen eine höhere Vergleichbarkeit mit anderen Systemen ermöglichen sollen. Zum anderen erlauben es diese Methoden, durch geschicktes Aufteilen eines Datensets aussagekräftigere Klassifikationsergebnisse zu erzielen. Zum Abschluss wird das frei verfügbare Data Mining und Machine Learning Tool *Weka* vorgestellt, das in dieser Arbeit Verwendung findet.

2.2.1 Der K-Nearest-Neighbours-Klassifikator

„The simplest form of learning is plain memorization.“ [Witten 2011]

Nach diesem simplen Grundsatz verfährt der K-Nearest-Neighbours-Klassifikator. Sobald erst einmal eine ausreichende Menge an Trainingsinstanzen gelernt wurde, besteht der Klassifikationsschritt nur noch darin, die unbekannte Instanz mit allen Trainingsbeispielen zu vergleichen. Dabei wird jeweils die Distanz nach einer vorgegebenen Metrik berechnet und das zu klassifizierende Objekt zugeordnet.

Im Folgenden sollen die einzelnen Schritte genauer erläutert werden.

2.2.1.1 Trainingsschritt

Jede Instanz wird durch eine Vielzahl von Features charakterisiert. Um eine Vergleichbarkeit zu gewährleisten, müssen im Trainingsschritt gegebenenfalls die Werte der Features auf numerische Werte abgebildet werden.

Liegen die Werte bereits in numerischer Form vor, ist eine Abbildung nicht mehr nötig, da bei einem Vergleich trivialerweise die Differenz gebildet werden kann. Bei nominalen Werten dagegen gestaltet sich ein Vergleich oft schwierig oder gar unmöglich, da ohne eine vorherige Abbildung keine Aussage über die Differenz zwischen „männlich“ und „weiblich“ oder zwischen „rot“ und „blau“ zu treffen ist. Deswegen müssen in diesen Fällen den nominalen Werten numerische zugewiesen werden. Die einfachste Lösung ist es anzunehmen, dass die Differenz zwischen zwei gleichen nominalen Werten 0 und die zwischen zwei unterschiedlichen Werten 1 ist. Allerdings bietet es sich oft an, eine ausgefeiltere Metrik anzuwenden. Im obigen Fall könnte beispielsweise die Differenz zwischen „rot“ und „orange“ deutlich geringer ausfallen als die zwischen „schwarz“ und „weiß“.

Ebenso kommt es häufig vor, dass einige Attribute einen höheren Stellenwert besitzen als andere. In diesem Fall müssen den Attributen entsprechende Gewichte zugewiesen werden. [Russel & Norvig 2010] [Witten 2011].

2.2.1.2 Klassifikationsschritt

Wie oben bereits beschrieben, verläuft der Klassifikationsschritt beim K-Nearest-Neighbours-Klassifikator unspektakulär. Nach dem Vergleichen der zu klassifizierenden Distanz mit den Trainingsbeispielen wird sie der Klasse zugewiesen, zu deren nächstem Vertreter sie die geringste Distanz aufweist. Dies stellt die einfachste Form des KNN-Klassifikators dar, da lediglich der eine nächste Nachbar berücksichtigt wird. Der Parameter K wäre demnach gleich 1. Wie später in dieser Arbeit gezeigt wird, kann ein solcher 1NN-Klassifikator durchaus sehr gute Ergebnisse erzielen und auch andere Klassifikationsmethoden in den Schatten stellen.

Allerdings verfügt er über einen offensichtlichen Nachteil, der in Abbildung 20 zu sehen ist: Er ist sehr anfällig gegenüber Ausreißern.

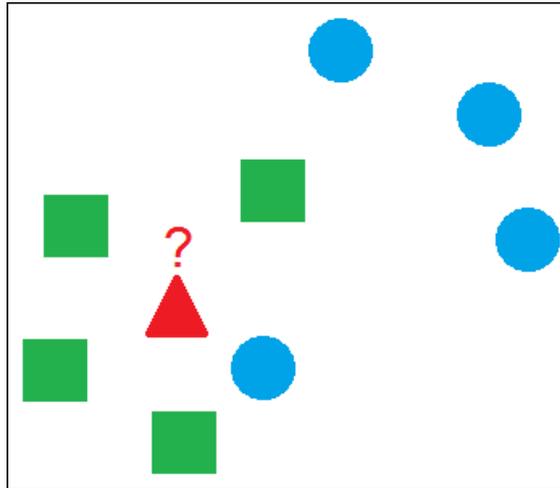


Abbildung 20: Beispiel einer KNN-Klassifikation mit zwei Klassen.
Rot: zu klassifizierende Instanz.

Die nächste Umgebung der zu klassifizierenden Instanz (rot) wird klar von grünen Instanzen dominiert, während ein Großteil der blauen Distanzen sich in deutlicher Entfernung befindet. Lediglich ein blauer Ausreißer befindet sich in der Nähe der zu klassifizierenden Instanz. Dieser reicht allerdings aus, um bei einer Klassifikation mit $K=1$ die rote Instanz der blauen Klasse zuzuordnen.

Eine Erhöhung von K führt dazu, dass der unbekannten Instanz nicht mehr automatisch die Klasse des nächsten Nachbarn zugewiesen wird. Vielmehr werden die K nächsten Nachbarn bestimmt und eine Mehrheitsentscheidung getroffen. Dabei kann es vorkommen, dass eine Pattsituation entsteht, also z. B. bei $K=4$ je zwei Instanzen der Klasse A und zwei der Klasse B der unbekannten Instanz am nächsten sind. Deswegen bietet sich eine Gewichtung der Distanzen an. Dabei wird die Wahl jeder Instanz nicht mehr gleich stark mit 1 gewichtet, sondern abhängig von ihrer Distanz. Häufig wird dabei als Gewicht der Kehrwert der Distanz verwendet. [Witten 2011]

2.2.2 Die Support Vector Machine

In modernen Anwendungen gehören Support Vector Machines (englisch für: Stützvektormaschine) zu den am häufigsten verwendeten Klassifikatoren. Dies lässt sich vor allem auf ihre hohe Anpassungsfähigkeit an verschiedene Domänen, sowie auf ihre hohe Effizienz und meist sehr guten Klassifikationsraten zurückführen.

Im Folgenden sollen zunächst das Grundprinzip von Support Vector Machines und die mathematischen Grundlagen erläutert werden. Anschließend werden Verfeinerungen vorgestellt, mit deren Hilfe häufig auftretende Probleme in der Praxis umgangen werden können.

2.2.2.1 Grundprinzip

Ziel des Trainings einer Support Vector Machine ist das Finden eines linearen Modells, mit dessen Hilfe Instanzen zweier oder mehrerer Klassen möglichst sicher unterschieden werden können. Abbildung 21 stellt ein einfaches Szenario mit zwei Klassen schematisch dar.

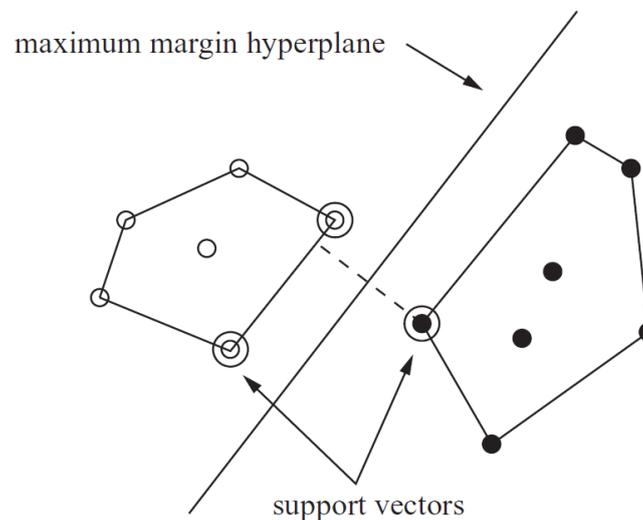


Abbildung 21: Grundprinzip einer Support Vector Machine. [Witten 2011]

Zu sehen sind zwei Klassen von Trainingsbeispielen, weiß und schwarz, die mittels einer Hyperebene bestmöglich separiert werden sollen. In diesem einfachen, zweidimensionalen Fall genügt als Hyperebene eine simple Gerade. Sie repräsentiert das lineare Modell und wird durch sogenannte Supportvektoren beschrieben. Dies sind die eingekreisten Instanzen, die der Hyperebene am nächsten liegen. Für jede Klasse existiert immer mindestens ein Supportvektor. Sind diese erst einmal identifiziert, sind alle weiteren Trainingsbeispiele irrelevant, da die Hyperebene eindeutig durch die Supportvektoren beschrieben wird. [Alpaydin 2010]

Bei Betrachtung von Abbildung 21 wird schnell klar, dass es eine Vielzahl von Möglichkeiten gibt, die separierende Gerade so zu platzieren, dass die Instanzen unterschiedlicher Klassenzugehörigkeit voneinander getrennt werden. Ein mögliches Beispiel ist eine vertikal verlaufende Gerade, die sich rechts vom rechts gelegenen, weißen Supportvektor und links vom schwarzen Supportvektor befindet. Es ist offensichtlich, dass eine solche Gerade ein schlechteres Modell repräsentiert, da der Abstand zu den Supportvektoren deutlich geringer ist als bei der hier zu sehenden Variante. Ziel des Trainingsschrittes der Support Vector Machine ist es, eine sogenannte Maximal Separierende Hyperebene zu finden, die nicht nur die Trainingsbeispiele korrekt separiert, sondern dabei auch einen maximalen Abstand zwischen Supportvektoren und Hyperebene garantiert. Dieser Abstand wird als *Margin* (englisch *margin*: Abstand, Spielraum) bezeichnet. Je größer er ist, desto besser lässt sich mit dem Modell generalisieren. [Alpaydin 2010]

2.2.2.2 Mathematische Definition

Gegeben seien linear separierbare Beispiele $(x_1, y_1), \dots, (x_n, y_n)$, wobei x^i der die jeweilige Instanz beschreibende Attributvektor ist und y_i die Klassenzugehörigkeit angibt (1, falls Beispiel positiv; -1, falls negativ).

Eine Hyperebene wird beschrieben durch:

$$f(x) := \langle w \cdot x \rangle + b = 0 \quad (2.15)$$

w ist hierbei der Normalenvektor der Ebene, der senkrecht auf allen Vektoren der Hyperebene steht. $\langle w \cdot x \rangle$ beschreibt das Skalarprodukt zwischen w und x ; $\frac{b}{\|w\|}$ gibt den Abstand zwischen der Geraden und dem Ursprung an.

Ziel ist es, im Folgenden w und b so zu bestimmen, dass gilt:

$$\langle w \cdot b \rangle = \begin{cases} > 0, \text{ falls } x \text{ im positiven Halbraum} \\ = 0, \text{ falls } x \text{ auf der Hyperebene} \\ < 0, \text{ falls } x \text{ im negativen Halbraum} \end{cases} \quad (2.16)$$

und

$$|\langle w \cdot x \rangle + b| = 1 \quad (2.17)$$

für alle Beispiele, die am nächsten zur Hyperebene liegen.

Um den maximalen Margin zu bestimmen, wird der Margin M zunächst als die Summe der Abstände d_+ und d_- definiert, die jeweils den Abstand zum nächsten positiven (d_+) bzw. negativen (d_-) Beispiel angeben:

$$M = d_+ + d_- \quad (2.18)$$

Des Weiteren werden zwei zusätzliche Hyperebenen H_1 und H_2 definiert, die parallel zu H im Abstand d_+ und d_- verlaufen und damit die Grenzen zum positiven und negativen Bereich darstellen:

$$y_i \cdot (\langle w \cdot x_i \rangle) - 1 \geq 0 \quad \begin{cases} H_1: \langle w \cdot x_i \rangle + b \geq 1 \text{ bei } y = +1 \\ H_2: \langle w \cdot x_i \rangle + b \leq -1 \text{ bei } y = -1 \end{cases} \quad (2.19)$$

Abbildung 22 zeigt den beispielhaften Verlauf einer maximal separierenden Hyperebene H und ihrer zugehörigen Hyperebenen H_1 und H_2 .

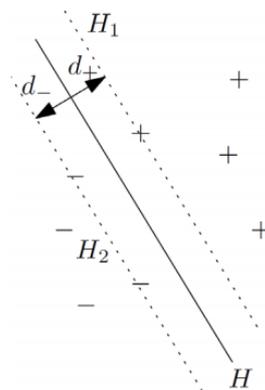


Abbildung 22: Maximal separierende Hyperebene H mit zugehörigen Hyperebenen H_1 und H_2 . [Hotho 2014]

Im Folgenden werden Hyperebenen betrachtet, für die $d_+ = d_-$ ist und daher gilt:

$$M = d_1 + d_2 = \frac{1 - b}{\|w\|} + \frac{1 + b}{\|w\|} = \frac{2}{\|w\|} \quad (2.20)$$

Um die Hyperebene mit dem maximalen Margin zu finden, muss $\frac{2}{\|w\|}$ maximiert und damit $\|w\|$ minimiert werden, wobei folgende Nebenbedingung für alle Beispiele $i = 1, \dots, n$ erfüllt sein muss:

$$y_i \cdot (\langle w \cdot x_i \rangle) - 1 \geq 0 \quad (2.21)$$

Dies ist ein konvexes, quadratisches Optimierungsproblem und damit eindeutig lösbar in $\mathcal{O}(n^3)$. Das Ergebnis ist eine Hyperebene, die die beiden Halbräume optimal trennt.

Aus der Nebenbedingung lässt sich direkt folgender Zusammenhang ableiten:

$$y_i > 0 \Leftrightarrow \langle w \cdot x_i \rangle + b > 0 \quad (2.22)$$

Daraus ergibt sich wiederum der letztendliche Klassifikator:

$$f(x) = \langle w \cdot x \rangle + b \quad (2.23)$$

Die Klassenzugehörigkeit wird direkt über das Vorzeichen von $f(x)$ bestimmt. [Hotho 2014]

2.2.2.3 Erweiterungen in der Praxis

Das in 2.2.2.2 vorgestellte Vorgehen ist nur unter gewissen Bedingungen und unter starken Einschränkungen möglich. Zum einen werden lediglich Klassifikationsaufgaben mit zwei Klassen behandelt, zum anderen wird angenommen, dass die gegebenen Trainingsbeispiele linear separierbar sind. Beides ist in der Praxis meist nicht der Fall. Deswegen sollen im Folgenden drei Erweiterungen vorgestellt werden. Zunächst wird der Umgang mit Ausreißern, die eine lineare Separierung verhindern, erläutert. Anschließend wird der sogenannte Kerneltrick behandelt, der durch geschicktes Abbilden der Attribute stets eine Separierbarkeit der Beispiele garantiert. Abschließend werden Multi-class-Support Vector Machines vorgestellt, mit deren Hilfe eine Einteilung in beliebig viele Klassen möglich ist.

2.2.2.3.1 Ignorieren einzelner Ausreißer beim Training

Die Bedeutung von Ausreißern wurde bereits beim K-Nearest-Neighbours-Klassifikator deutlich. In Abbildung 23 wird die Problematik bei der Verwendung einer Support Vector Machine erkennbar. Die Trainingsbeispiele sind zum größten Teil klar separierbar, nämlich in die Kreise rechts oben und die Dreiecke links unten. Lediglich das am weitesten rechts gelegene Dreieck fällt aus diesem Schema. Durch diese eine Beispielinstantz ist es nicht möglich, eine Hyperebene mit großem Margin zu bestimmen. Dies führt dazu, dass das gelernte Modell auf dem Testset schlechter generalisiert und Fehldetektionen häufiger auftreten. Wird allerdings der Ausreißer ignoriert, so lässt sich eine Hyperebene finden, die die beiden Klassen mit einem deutlichen Margin voneinander abtrennt. Dies führt wiederum auch zu besseren Klassifikationsergebnissen.

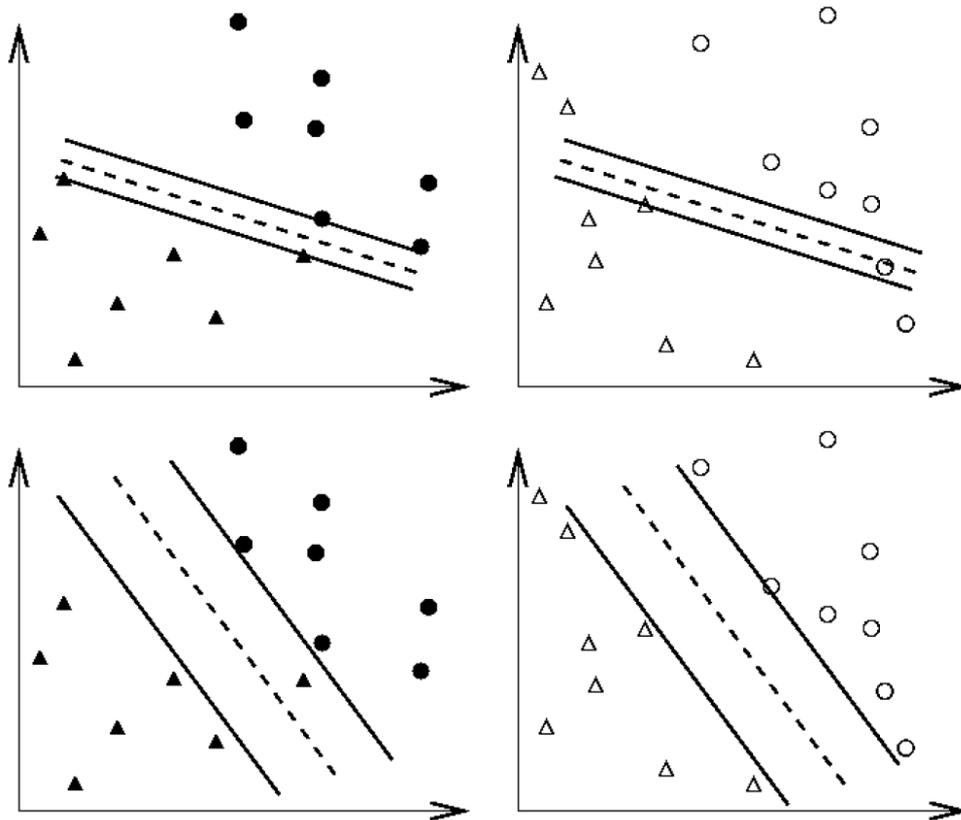


Abbildung 23: Anwendung zweier auf einem Testset (links) gelernter Modelle auf ein Trainingsset (rechts). Oben: Berücksichtigung aller Instanzen mit daraus resultierendem Overfitting, unten: Ignorieren eines Ausreißers. [Chih-Wei 2003]

Nach dieser Anpassung spricht man von einer weich trennenden Hyperebene. Das oben vorgestellte mathematische Modell wird wie folgt erweitert:

Es werden zwei zusätzliche Parameter eingeführt: zum einen die Schlupfvariablen ξ_i , die angeben, welche Fehler bei der Trennung zugelassen werden; zum anderen der Parameter C , der bestimmt, wie stark Fehler gewichtet werden. Dabei gilt $C \in \mathbb{R}_{>0}$. Weiterhin ist ein Optimierungsproblem zu lösen. Es soll folgender Ausdruck minimiert werden:

$$\|w\|^2 + C \cdot \sum_{i=1}^n \xi_i \quad (2.24)$$

sodass für alle i gilt:

$$\text{für } y_i = 1: f(x_i) = \langle w \cdot x_i \rangle + b \geq 1 - \xi_i \quad (2.25)$$

$$\text{für } y_i = -1: f(x_i) = \langle w \cdot x_i \rangle + b \leq \xi_i - 1 \quad (2.26)$$

ξ_i definiert dabei einen Bereich um die Hyperebene, in dem Fehler ignoriert werden. [Hotho 2014] [Witten 2011]

2.2.2.3.2 Der Kerneltrick

Selbst bei Verwendung einer weich trennenden Hyperebene ist eine lineare Trennung nicht garantiert. In solchen Fällen ist es mit Hilfe einer Transformation ϕ möglich, die Attributvektoren x_i der Beispiele auf eine höhere Dimension abzubilden. In diesem

höherdimensionalen „Feature-Space“ wird anschließend die optimal trennende Hyperebene berechnet. Abschließend erfolgt die Rücktransformation. Abbildung 24 stellt das Mapping in eine höhere Dimension schematisch dar.

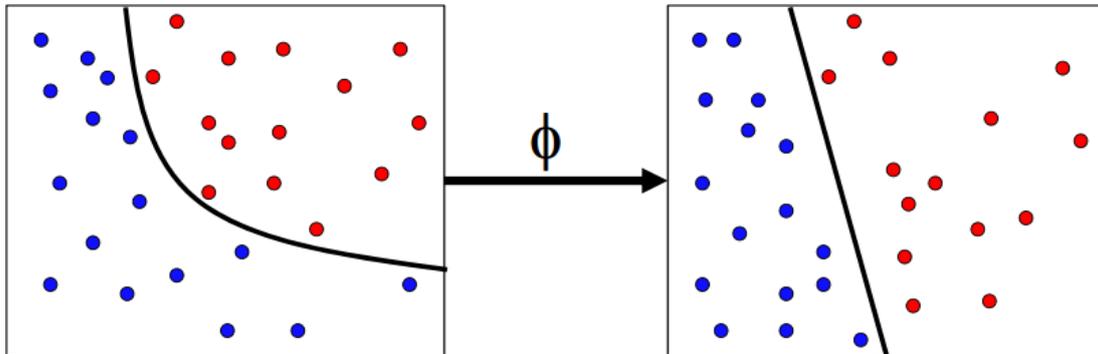


Abbildung 24: Lineare Separation einer vorher nicht linear trennbaren Menge an Beispielen durch das Mapping in eine höhere Dimension.
Links: Input-Space, rechts: Feature-Space. [Hotho 2014]

Da theoretisch bis in eine unendlich hohe Dimension gemappt werden kann, lässt sich dadurch jede beliebige Anordnung von Instanzen linear trennen. Der Nachteil ist, dass Berechnungen im Feature-Space durch die sehr große Anzahl von Dimensionen sehr aufwendig werden können. Die Lösung dieses Problems bietet der sogenannte Kerneltrick. Dabei werden die Datenpunkte im hochdimensionalen Raum nicht explizit berechnet. Zur Lösung des dualen Problems wird lediglich das Skalarprodukt zweier Punkte und nicht die Datenpunkte selbst benötigt. Eine Kernelfunktion K entspricht demnach dem Skalarprodukt zweier Punkte, die zuvor in einem höherdimensionalen Raum abgebildet wurden:

$$K: K(x, y) = \langle \phi(x) \cdot \phi(y) \rangle \quad (2.27)$$

[Hotho 2014]

2.2.2.3.3 Multi-class-Support Vector Machines

In der Realität reicht eine binäre Klassifikation im Normalfall nicht aus. Die zwei am häufigsten verwendeten Herangehensweisen an eine Klassifikation mit mehr als zwei Klassen sind die Einer-gegen-Alle- und die Jeder-gegen-Jeden-Methode.

Die Einer-gegen-Alle-Methode ist der direkteste Ansatz. Angenommen ein Datensatz von Beispielen soll in m Klassen unterteilt werden, dann werden m binäre Support Vector Machine-Klassifikatoren kreiert. Jeder dieser Klassifikatoren wird darauf trainiert, eine Klasse von allen der verbleibenden $m-1$ Klassen zu unterscheiden. Es werden in der Trainingsphase des i -ten Klassifikators alle Datenpunkte aus Klasse i als positive und alle anderen als negative Beispiele angenommen. In der Anwendungsphase wird für jeden Klassifikator der Abstand des zu bestimmenden Beispiels von der separierenden Hyperebene bestimmt. Die letztendliche Ergebnisklasse ist die, bei deren Einer-gegen-Alle-Klassifikation der größte Abstand erreicht werden konnte.

Bei der Jeder-gegen-Jeden-Methode wird je ein binärer Klassifikator für jede mögliche Paarung zweier unterschiedlicher Klassen kreiert. Bei m Klassen werden also $m(m-1)$ Klassifikatoren benötigt. In der Anwendungsphase weist jeder Klassifikator das zu bestimmende Beispiel einer Klasse zu. Die „Gewinner“-Klassen werden gespeichert und das finale Ergebnis ist die Klasse, der das Beispiel am häufigsten zugewiesen wurde.

Es lässt sich nicht allgemein sagen, welche Methode die bessere ist. Für die Jeder-gegen-Jeden-Methode müssen deutlich mehr Klassifikatoren trainiert werden. Allerdings lässt sich, aufgrund des simplen Aufbaus, jeder dieser Klassifikatoren deutlich unkomplizierter entwerfen als die der Einer-gegen-Alle-Methode. In den meisten Anwendungen liefern beide Methoden ähnlich gute Resultate. Bei der Verwendung von Support Vector Machines wird die Jeder-gegen-Jeden-Methode empfohlen. Dies ist auch der Ansatz, den die in dieser Arbeit verwendete Support Vector Machine-Bibliothek *LibSVM* (vgl. [LibSVM]) nutzt. [Mahesh 2008]

2.2.3 Evaluationsmethoden

Zum Testen der Performanz entwickelter Klassifikatoren ist es einerseits wichtig, über eine ausreichende Menge von Trainings- und Testbeispielen zu verfügen. Andererseits muss stets ein hohes Maß an Vergleichbarkeit vorliegen, damit verschiedene Klassifikatoren bzw. gleiche Klassifikatoren mit verschiedenen Parametern oder unter Verwendung unterschiedlicher Features fair evaluiert werden können. Im Folgenden sollen einige Standard-Evaluationsmethoden erläutert werden, die auch in dieser Arbeit Verwendung finden.

2.2.3.1 Verwendung eines Trainings- und Testsets

Die simpelste Methode ist die fixe Aufteilung der Beispiele in ein Trainings- und in ein Testset. Der Klassifikator wird anhand der Trainingsbeispiele trainiert und anschließend auf dem Testset evaluiert. Der Vorteil dieser Methode liegt in der Vergleichbarkeit. Jeder Klassifikator wird anhand exakt derselben Beispiele trainiert und getestet. Ein Nachteil ist, dass die Ergebnisse unter Umständen nicht ausreichend aussagekräftig sind. So ist es z. B. möglich, dass die Aufteilung einen Klassifikator oder eine bestimmte Featuregruppe stark bevorzugt, während bei vielen anderen Aufteilungen ein anderes System im Vorteil wäre. [Russel & Norvig 2010]

2.2.3.2 K-Fold-Cross-Validation

Die verbreitetste Methode ist die sogenannte *k-fold-cross-validation*, also die Über-Kreuz-Validierung mittels k Teil-Datensets. Die Idee dahinter ist, dass jedes Beispiel sowohl zum Training als auch zum Testen verwendet wird. Zunächst werden die Trainingsdaten zufällig in k gleichgroße Teilsets gegliedert. Anschließend erfolgen k Durchläufe, in denen jeweils $k-1$ Sets zum Training und das verbleibende Set zum Testen verwendet werden. Die so erhaltenen Ergebnisse werden gemittelt. Typische Werte für k sind 5 oder 10, da sie zum einen mit hoher Wahrscheinlichkeit ein brauchbares Ergebnis liefern und sich zum anderen die Bearbeitungszeit in Grenzen hält. Der Vorteil ist, dass sich leicht auch mit relativ wenigen Beispielen ein aussagekräftiges Ergebnis mit hoher statistischer Signifikanz erzielen lässt. Der Nachteil besteht im erhöhten Zeitaufwand. [Russel & Norvig 2010]

2.2.3.3 Leave-one-out

Eine extreme Form der *k-fold-cross-validation* ist die 1 x all-Evaluation, auch *Leave-one-out-Evaluation* genannt. Dabei wird bei N vorhandenen Beispielen eine *N-fold-cross-validation* durchgeführt. Das bedeutet, es wird jeweils nur ein Beispiel ausgenommen, anschließend wird auf dem Restdatensatz trainiert und abschließend wird das zunächst

entfernte Beispiel mit dem so entstandenen Modell klassifiziert. Der große Vorteil dieser Methode ist die maximale Vergleichbarkeit, da Ergebnisse nicht mehr durch zufällige Aufteilungen verfälscht werden können. Der Nachteil besteht wiederum im hohen Zeitaufwand. Bei einer großen Menge an Beispielen müssen ebenso viele Modelle kreiert und getestet werden. Deshalb eignet sich die 1 x all-Evaluation hauptsächlich für vergleichende Klassifikatoren, wie den K-Nearest-Neighbours-Ansatz. [Russel & Norvig 2010]

2.2.4 Weka

Weka ist ein frei verfügbares Data-Mining-Tool, das von der University of Waikato in Neuseeland entwickelt wurde. Die Abkürzung Weka steht für *Waikato Environment for Knowledge Analysis*. Es handelt sich um eine umfassende Sammlung verschiedenster Machine Learning Algorithmen aus sämtlichen Bereichen des Data-Minings wie Daten-Vorverarbeitung, Klassifikation, Regression, Clustering, Assoziationsregeln und Visualisierung.

Weka kann sowohl als GUI verwendet als auch in eigene Java-Projekte eingebunden werden. Das System wurde in Java programmiert und kann unter Windows, Linux und Mac OS verwendet werden. [Weka]

2.2.4.1 Klassifikatoren

In dieser Arbeit wird das Tool hauptsächlich zur Klassifikation eingesetzt. Dazu stellt Weka in der Standardversion knapp 70 Klassifikatoren entweder direkt zur Verfügung oder bietet Schnittstellen zu anderen Bibliotheken, wie z. B. der Support Vector Machine-Bibliothek LibSVM.

In der aktuellen Version 3.7 (Stand: 18. September 2015) sind die Klassifikatoren in Weka in sieben Gruppen untergliedert:

- Bayes: 7 Ausprägungen des Naive Bayes- bzw. BayesNet-Klassifikators.
- Functions: 12 funktionenbasierte Klassifikatoren wie Support Vector Machines und Multilayer Perceptron.
- Lazy: 3 Variationen des K-Nearest-Neighbours-Klassifikators.
- Meta: 19 Hilfs-Klassifikatoren, die es erlauben, Klassifikatoren aus den anderen Klassen unter definierten Umständen zu verwenden. Dies beinhaltet u. a. das Ignorieren von String-Attributen.
- Misc: 2 Hilfs-Klassifikatoren, die es z. B. trotz fehlender Attributwerte ermöglichen, eine sinnvolle Abbildung und somit eine erfolgreiche Klassifikation durchzuführen.
- Rules: 6 regelbasierte Klassifikatoren wie Entscheidungstabellen oder den ZeroR-Klassifikator, der stets die im jeweiligen Datensatz am häufigsten auftretende Klasse vorhersagt und damit die Baseline berechnet.
- Trees: 8 baumbasierte Klassifikatoren wie Entscheidungsbäume und der J48- oder Random Forest-Klassifikator.

Theofilis (vgl. [Theofilis 2013]) gibt eine genauere Übersicht über die verschiedenen Klassifikatoren.

2.2.4.2 ARFF-Dateien

Die Instanzen und ihre Attribute werden in sogenannten .ARFF (Attribute-Relation File Format)-Dateien gespeichert. Deren sehr intuitiver Aufbau ist in Abbildung 25 zu sehen.

```
% ARFF file for the weather data with some numeric features
%
@relation weather

@attribute outlook { sunny, overcast, rainy }
@attribute temperature numeric
@attribute humidity numeric
@attribute windy { true, false }
@attribute play? { yes, no }

@data
%
% 14 instances
%
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
rainy, 70, 96, false, yes
rainy, 68, 80, false, yes
rainy, 65, 70, true, no
overcast, 64, 65, true, yes
sunny, 72, 95, false, no
sunny, 69, 70, false, yes
rainy, 75, 80, false, yes
sunny, 75, 70, true, yes
overcast, 72, 90, true, yes
overcast, 81, 75, false, yes
rainy, 71, 91, true, no
```

Abbildung 25: Beispielhafter Aufbau einer .ARFF-Datei. [Witten 2011]

Jede ARFF-Datei ist unterteilt in zwei Abschnitte:

Im Header werden, neben dem Namen der Relation, alle Attribute deklariert. Dabei wird sowohl deren Name als auch Datentyp angegeben. Bei nominalen Attributen müssen zusätzlich alle möglichen Attributausprägungen angegeben werden. Bei Bedarf wird ein nominales Attribut als Klassenattribut definiert. Dies kann allerdings auch nachträglich oder erst direkt vor der Klassifikation erfolgen.

Im zweiten Teil, dem Datenabschnitt, werden die einzelnen Instanzen definiert. Dabei muss zu jedem im Header definierten Attribut ein passender Wert angegeben werden.

[Weka] [Witten 2011]

2.2.4.3 Alternative: RapidMiner

Der RapidMiner ist eine Open-Source Umgebung für Data-Mining und Machine Learning. Dieser wird seit 2001 am Lehrstuhl für Künstliche Intelligenz der TU Dortmund entwickelt. Er stellt mehr als 500 Operatoren zur Verfügung, u. a. für Datenvorverarbeitung und aus den Bereichen Machine Learning und Data-Mining. [Wiki: RapidMiner]

Dank seiner Flexibilität, der übersichtlichen GUI und der Möglichkeit einer simplen Integration in ein Java-Programm, stellt der RapidMiner eine echte Alternative zu Weka dar. Nach Meinung dieser Arbeit bewegen sich beide Programme auf einem ähnlichen, sehr hohen Niveau, und es ist letztendlich eine Frage der persönlichen Vorliebe, welches den Vorrang erhält.

2.2.5 Die LibSVM-Bibliothek

Zur Klassifikation mittels einer SVM wird in dieser Arbeit die Open-Source SVM-Bibliothek LibSVM verwendet (vgl. [LibSVM]). Diese wird seit dem Jahr 2000 stetig weiterentwickelt und ist nach eigenen Angaben eine der am meisten verwendeten Softwares für SVM-Klassifikation. In der Zeitspanne zwischen 2000 und 2010 wurde die Bibliothek mehr als 250.000 mal heruntergeladen und es existieren zahlreiche Veröffentlichungen aus einer Vielzahl von Bereichen, u. a. der Computer Vision, dem Natural Language Processing oder der Bioinformatik, in denen LibSVM erfolgreich zum Einsatz gekommen ist.

Die Bibliothek implementiert neben der Standard-SVM-Aufgabe, der binären Klassifikation, eine Vielzahl von weiteren Funktionen, wie eine effiziente Multi-class-Klassifikation, Wahrscheinlichkeitsschätzungen und zahlreiche Kernelfunktionen. Der Sourcecode ist sowohl in Java als auch in C++ verfügbar. Des Weiteren existieren Schnittstellen zu fast allen modernen Programmiersprachen, u. a. Python, R, Matlab, PHP, Perl und Ruby.

[Chang 2013]

Der Hauptgrund, wieso in dieser Arbeit LibSVM verwendet wurde, ist die Schnittstelle zum bereits vorgestellten Data-Mining-Tool Weka. Dadurch kann LibSVM direkt über die Weka-GUI genutzt werden. Dies erleichtert die Handhabung ungemein, da Vorverarbeitungsschritte, wie die Normalisierung der Attributwerte, sowie die Einstellung von Parametern oder die Auswertung der Klassifikationsergebnisse direkt in der GUI vorgenommen werden können.

2.3 Framework zur Bildverarbeitung, Feature-Extraktion und Klassifikation

Zur Durchführung dieser Arbeit ist die Kombination von Methoden aus der Bildverarbeitung, der Feature-Extraktion und der Klassifikation nötig. Zu diesem Zweck wäre ein Framework, das zumindest Teile dieser Anwendungen verknüpft, sehr hilfreich. Im besten Fall müsste der Nutzer lediglich einige Vorverarbeitungsschritte angeben, die gewünschten Features auswählen und könnte anschließend die erzielten Resultate mittels eines Klassifikators direkt testen.

Leider existiert nach Wissen dieser Arbeit ein Framework bzw. eine Pipeline-Architektur dieser Art nicht einmal im Ansatz. Bibliotheken und Tools wie die OpenCV- oder LibSVM-Library oder Weka erleichtern viele Aufgaben zwar ungemein, aber letztendlich muss ein Großteil des Systems von Grund auf zusammengestellt werden.

2.4 Blattmerkmale

Bei der Bestimmung von Baumspesies spielen die Blätter die mit Abstand wichtigste Rolle. Zwar liefern auch Blüten, Früchte, Rinde, Knospen oder der Habitus gute Anhaltspunkte, jedoch ist es häufig schwierig oder sogar unmöglich, mit diesen als Einzelmerkmal einen Baum sicher zu bestimmen. Optimal ist natürlich eine Kombination mehrerer Baumbestandteile. Da es in der Praxis nicht praktikabel ist, zahlreiche Fotos mehrerer Baumorgane aufzunehmen und zu analysieren, liegt der Fokus dieser Arbeit allein auf den Blättern. Es wird angenommen, dass eine einzelne Aufnahme eines Blattes ausreichend Möglichkeiten bietet, eine Baumspesies zu identifizieren.

Im Folgenden soll zunächst der grobe Aufbau eines Blattes erläutert werden. Anschließend werden die klassischen Blattmerkmale vorgestellt.

2.4.1 Aufbau eines Blattes

Abbildung 26 zeigt schematisch den Aufbau eines Blattes.

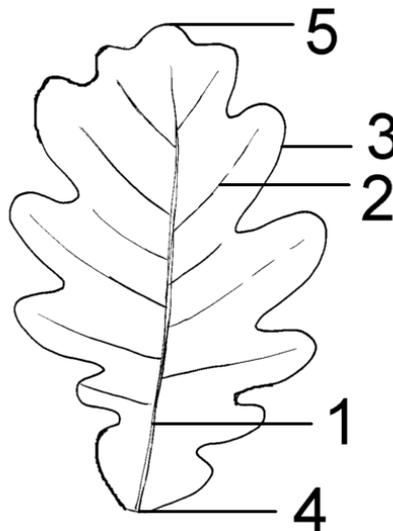


Abbildung 26: Der Aufbau eines Blattes. 1: Mittelrippe, 2: Seitenrippe, 3: Blattrand, 4: Spreitengrund, 5: Spreitenspitze. [Wiki: Blatt]

In der Mitte des Blattes verläuft die Mittelrippe, in dieser Arbeit auch Hauptvene genannt. Diese verläuft vom Spreitengrund, der Stelle des Blattes, an der der Stiel ansetzt, zur Spreitenspitze, der Spitze des Blattes. Die kleineren Venen, die von der Mittelrippe nach außen verlaufen, nennt man Seitenrippen.

2.4.2 Zusammenfassung der klassischen Blattmerkmale

Möchte man mittels eines Bestimmungsbuches oder einer Webseite wie <http://www.baumkunde.de> (vgl. [Baumkunde]) eine Baumspesies anhand von Blattmerkmalen bestimmen, ist üblicherweise der erste Schritt, die Blattform zu ermitteln. Anschließend wird der Blattrand betrachtet. Reicht dies noch nicht zur eindeutigen Identifikation der Baumspesies, werden weitere Merkmale untersucht.

2.4.2.1 Die Blattform

Die Blattform beinhaltet zum einen die Anordnung der Blätter am Zweig. Des Weiteren wird zwischen einfachen und zusammengesetzten Blättern unterschieden. In Abbildung 27 sind die jeweiligen Kombinationen zu sehen.

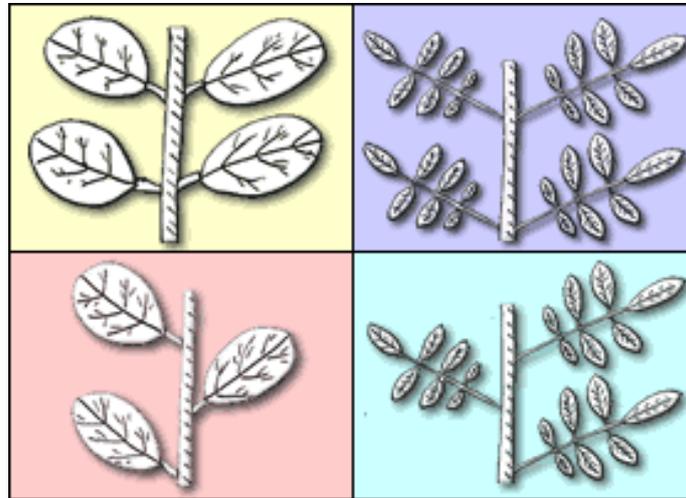


Abbildung 27: Mögliche Blattstellungen. Oben: gegenständig: einfach (links), zusammengesetzt (rechts); unten: wechselständig: einfach (links), zusammengesetzt (rechts). [Baumkunde]

Als gegenständig bezeichnet man Blätter, deren Ansatzstellen sich genau gegenüberstehen. Wechselständig hingegen sind Blätter dann, wenn an den Knoten jeweils nur ein einziges Blatt sitzt.

Einfache Blätter zeichnen sich dadurch aus, dass sie eine einheitliche, nicht in Teilportionen, sogenannte Fiedern, untergliederte Blattspreite besitzen. Als zusammengesetzt bezeichnet man demgegenüber Blätter, die aus mehreren Teilpartien, sogenannten Blättchen, bestehen. Hierbei wird zwischen paarig und unpaarig gefiederten Blättern unterschieden. Bei paarig gefiederten Blättern stehen sich dabei jeweils zwei Blättchen gegenüber. Unpaarig gefiederte Blätter besitzen ein zusätzliches Blättchen an der Spitze des Blattes, wie es z. B. rechts in Abbildung 27 zu sehen ist. [Kremer 1989]

2.4.2.2 Der Blattrand

Ist die Blattform bestimmt, wird normalerweise der Blattrand untersucht. Abbildung 28 zeigt verschiedene Blattränder und deren Bezeichnungen.

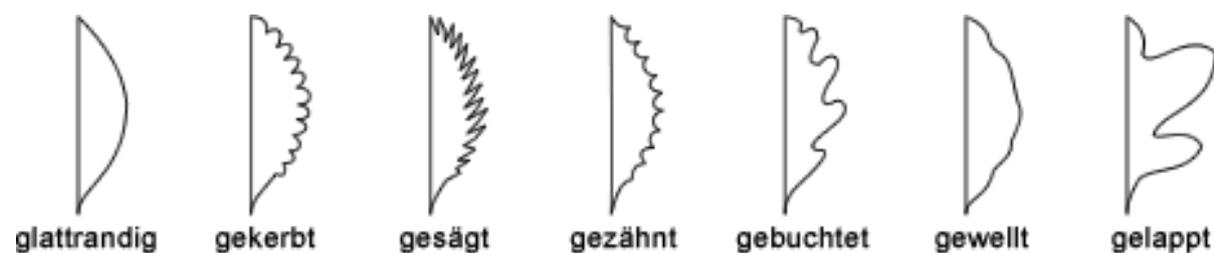


Abbildung 28: Mögliche Beschaffenheiten des Blattrandes. [Baumkunde]

In der Literatur werden häufig die Klassen unterschiedlich definiert. So wird z. B. nicht immer zwischen gesägt und gezähnt oder gekerbt, gebuchtet und gelappt unterschieden.

Dies ist dadurch zu begründen, dass der Übergang zwischen den einzelnen Klassen oft fließend ist. [Hecker 2012]

2.4.2.3 Weitere Merkmale

Es ist offensichtlich, dass die Kombination der bisher vorgestellten Merkmale bei weitem nicht ausreichen kann, um alle Blattspezies eindeutig zu bestimmen. Zur weiteren Unterteilung bieten sich zusätzliche Merkmale bzw. eine feinere Unterteilung der bisher vorgestellten Merkmale an. Einige Beispiele sind in Abbildung 29 zu sehen.

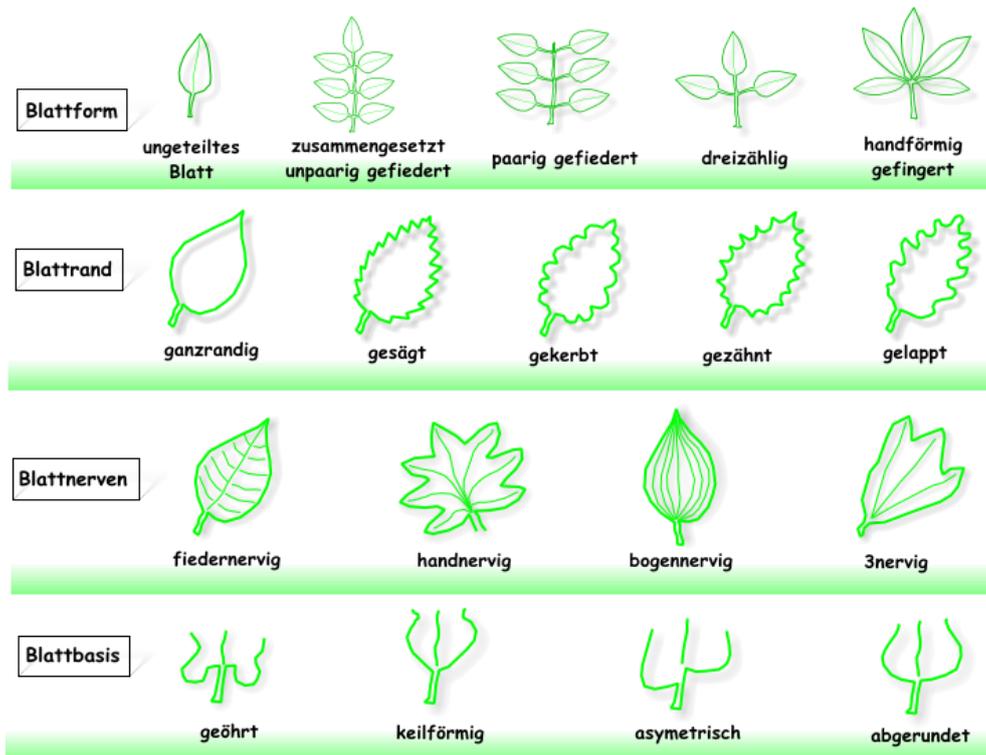


Abbildung 29: Übersicht über die Blattmerkmale. [Baumkunde]

Die Blattform lässt sich weiter unterteilen in z. B. dreizählige oder handförmig gefingerte Blätter. Als weiteres Merkmal kann die Venierung, also der Verlauf der Blattnerven dienen. Auch die Form der Blattbasis kann dazu beitragen, die in Frage kommenden Spezies weiter einzuschränken. Selbiges gilt für die Form der Blattspitze. Ein weiteres Merkmal ist die Färbung der Blätter. Hierbei stehen vor allem starke Kontraste, z. B. zwischen Vorder- und Hinterseite des Blattes im Vordergrund. Ebenfalls typisch für einige Spezies sind bestimmte Eigenschaften der Blattoberfläche. Diese kann sich glatt, rau oder ledrig anfühlen oder mit kleinen Härchen überzogen sein. [Kremer 1989]

3 Datensets

Zur Überprüfung der implementierten Methoden wurden verschiedene Blatt-Datensets zusammengetragen, die in diesem Kapitel vorgestellt werden sollen. Dabei handelt es sich zum einen um online frei verfügbare Datensets, zum anderen wurde eigens für diese Arbeit ein neues Datenset angelegt.

Bei den folgenden Abbildungen ist anzumerken, dass die Größe der Blätter angepasst wurde und somit die ursprünglichen Größenverhältnisse nicht mehr gegeben sind. Zwei in der Abbildung ähnlich große Blätter müssen in der Realität also keineswegs die gleiche Größe besitzen.

3.1 Flavia

Das Flavia-Datenset (vgl. [Flavia]) ist das bekannteste Blatt-Datenset und wird in den meisten Publikationen zum Vergleich der Resultate herangezogen. Es besteht aus 1907 Bildern von insgesamt 32 Spezies. Die verwendeten Spezies stammen zum größten Teil aus dem Yangtze-Delta in China. Jedes Bild ist ein Scan eines einzelnen Blattes und hat eine Auflösung von 1600x1200 Pixeln. Die Bilder sind vorsegmentiert. Das bedeutet einerseits, dass der Stiel bereits entfernt wurde. Andererseits wurde das Blatt bereits vom Hintergrund freigestellt, d. h. jeder Hintergrundpixel hat den Wert RGB(255, 255, 255) und somit den Graustufenwert 255, also das hellste Weiß. Dies erleichtert die Segmentierung bzw. die Lokalisation des Blattes erheblich.

Abbildung 30 zeigt einen Überblick über die verschiedenen Spezies.

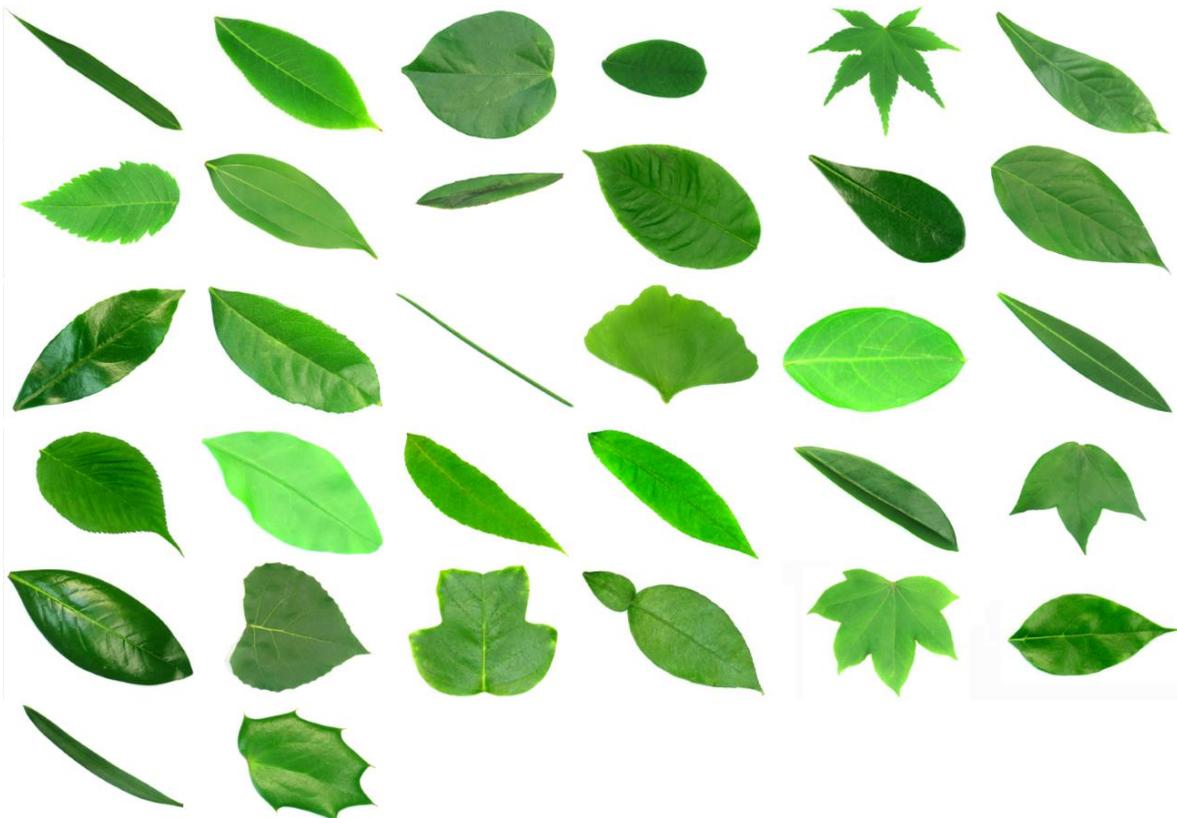


Abbildung 30: Je ein Beispielblatt pro Spezies aus dem Flavia-Datenset.

3.2 Foliage

Ebenfalls häufige Verwendung findet das Foliage-Datenset (vgl. [Foliage]). Es beinhaltet 60 Spezies mit jeweils 120 Bildern. Genau wie das Flavia-Datenset ist es vorsegmentiert. Eine Besonderheit ist die fixe Aufteilung in ein Trainingsset und ein Testset. Hierbei entfallen pro Spezies je 100 Blätter auf das Trainingsset und die verbleibenden 20 Blätter bilden das Testset. Dies erhöht die Vergleichbarkeit der erzielten Ergebnisse ungemein.



Abbildung 31: Je ein Beispielblatt pro Spezies aus dem Foliage-Datenset.

Wie in Abbildung 31 zu sehen ist, ist die Farbvielfalt im Vergleich zum Flavia-Datenset deutlich größer. Dabei ist anzumerken, dass es sich bei den hier in der Übersicht gezeigten Blättern nicht um Ausreißer handelt, sondern in den meisten Fällen auch die anderen Vertreter derselben Spezies eine ähnliche Färbung aufweisen.

3.3 Swedish Leaf Dataset (SLD)

Wie sich aus dem Namen bereits ableiten lässt, beinhaltet dieses Datenset Blätter von in Schweden heimischen Bäumen. Es wurden von 15 Spezies je 75 Blätter hochauflösend gescannt. Im Gegensatz zu den ersten beiden vorgestellten Datensets wurden beim Swedish Leaf Dataset keine weiteren Vorverarbeitungsschritte durchgeführt. Es erfolgte weder eine Entfernung der Stiele, noch wurden die Blätter vom Hintergrund freigestellt. Eine vollständige Segmentierung der Blätter lässt sich dennoch in den meisten Fällen problemlos erreichen, da die Blätter sehr sorgfältig gescannt wurden und kaum Schatten auftreten. Durch den hellen Hintergrund liefert Otsus Methode äußerst ansprechende Ergebnisse. Auch die automatische Stielentfernung (s. Abschnitt 4.6) weist nur eine sehr geringe Fehlerquote auf.

Abbildung 32 zeigt sowohl eine Übersicht über die 15 Spezies des Swedish Leaf Dataset als auch ein Beispielblatt vor der Segmentierung und der Stielentfernung.



Abbildung 32: Links: Je ein Beispielblatt pro Spezies aus dem Swedish Leaf Datenset, rechts: Beispielblatt als Farbscan.

3.4 Middle European Woody Plants (MEW)

Von den im Internet frei verfügbaren Datensets ist das Middle European Woody Plants mit 153 Spezies das zweitgrößte. Pro Spezies wurden im Schnitt etwa 63 Blätter in Zentraleuropa, hauptsächlich in Tschechien, gesammelt und dann in einer 300-DPI Auflösung gescannt. Um die richtige Zuordnung der Blätter zur jeweiligen Spezies sicher zu gewährleisten, wurde die Erstellung des Datensets von ausgebildeten Botanikern überwacht. Ähnlich wie beim Swedish Leaf Dataset sind die Bilder nicht vorsegmentiert und alle Blätter besitzen noch ihren Stiel. Allerdings liegt auch in diesem Fall ein sehr heller Hintergrund vor, sodass Otsus Methode sehr gute Segmentierungsergebnisse liefert. Eine Besonderheit des Datensets ist, dass die Bilder nicht nur als die ursprünglichen Scans vorliegen, sondern auch als bereits segmentierte Binärbilder. Auch in den Binärbildern sind alle Stiele noch vorhanden.

Zur Segmentierung wurde ebenfalls Otsus Methode verwendet. Anhand eines Subsets von 109 Blättern testeten Suk et al. (vgl. [Suk 2013]) die Wirksamkeit von Otsus Methode. Dazu wurden diese 109 Blätter auch manuell segmentiert und die Ergebnisse verglichen. Die Abweichung betrug 1,12%. Genauere Angaben über das Testverfahren bzw. die verwendete Metrik sind nicht bekannt.

In dieser Arbeit wird die Ansicht vertreten, dass das Middle European Woody Plants Datenset den aussagekräftigsten Vergleich verschiedener Klassifikationssysteme zulässt. Dies liegt vor allem daran, dass es, gemessen an seiner Größe, sehr sorgfältig zusammengestellt wurde und eine große Anzahl teils gut unterscheidbarer, aber auch zum Teil sehr ähnlicher Spezies liefert. Der Schwierigkeitsgrad liegt deutlich höher als bei den vorher genannten Datensets.

Sowohl eine Auswahl der Spezies als auch ein Beispielblatt als Farbscan und als Binärbild sind in Abbildung 33 zu sehen.

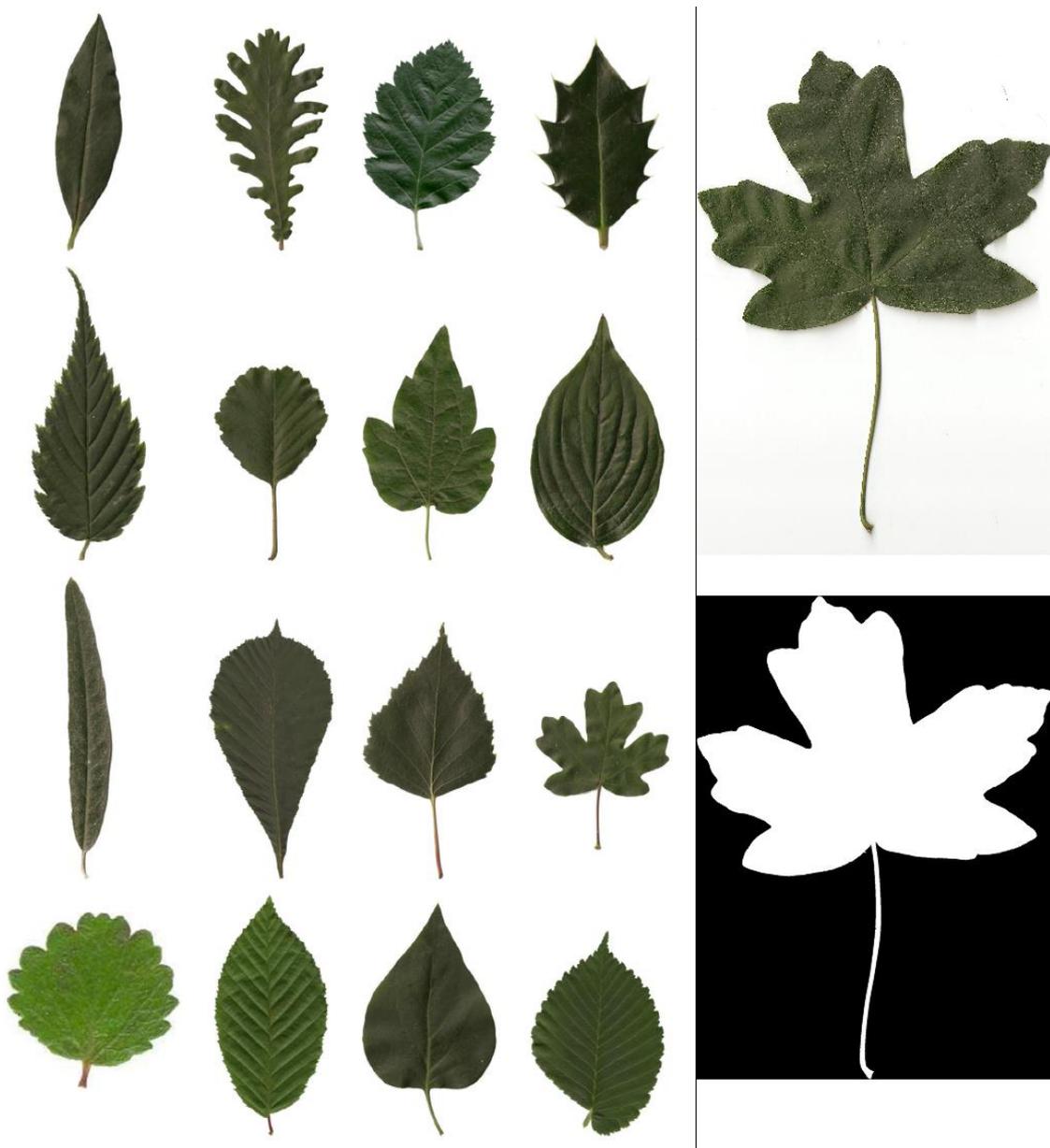


Abbildung 33: Links: Das Middle European Woody Plants Datenset (Auswahl) (vgl. [MEW]), rechts: Beispielblatt als Farbscan (oben), zugehöriges Binärbild (unten).

3.5 Intelligent Computing Laboratory (ICL)

Das mit Abstand größte in dieser Arbeit verwendete Datenset ist das des Intelligent Computing Laboratory der Chinese Academy of Sciences (vgl. [ICL]). Es beinhaltet insgesamt 16.851 Scans von 220 in China heimischen Spezies. Dabei streut die Anzahl der Bilder pro Spezies sehr stark, nämlich von 26 bis hin zu 1078. Ähnlich dem SLD und dem MEW wurden die Blätter vor hellem Hintergrund gescannt, ohne vorher die Stiele zu entfernen. Wiederum ermöglicht Otsus Methode bei geringem Aufwand eine qualitativ hochwertige Segmentierung. Abbildung 34 zeigt eine Auswahl an Spezies aus dem ICL.

Ein Nachteil des ICL ist das Vorkommen einiger unbekannter bzw. nicht benannter Spezies, die auch nachträglich nicht mehr eindeutig identifiziert werden konnten. Um die Vergleichbarkeit zu erhalten, wurden diese bei den Tests dennoch verwendet. Außerdem wurde bei vielen Blättern die Rückseite gescannt. Dies kann bei der Verwendung mancher Features, z. B. Farb- und Venierungs-Features, zu Problemen führen.



Abbildung 34: Links: Das Intelligent Computing Laboratory Datenset (Auswahl), rechts: Beispielblatt als Farbscan.

3.6 Bavaria Leaf Dataset (BLD)

Das eigens für diese Arbeit angelegte Datenset besteht aus 10 Spezies. Gesammelt wurde es in Süddeutschland, genauer gesagt in der Oberpfalz, Bayern. Angelehnt an das Swedish Leaf Dataset leitet sich auch hier der Name aus dem Ort der Zusammenstellung des Datensets ab: *Bavaria* ist die englische Bezeichnung für das Bundesland Bayern.

Anders als bei den vorhergehenden Datensets wurden die Blätter nicht gescannt, sondern fotografiert. Dabei kamen verschiedene Kameras unterschiedlicher Qualität zum Einsatz, darunter sowohl mehrere Handykameras, eine Standard-Digitalkamera, als auch eine hochwertige Spiegelreflexkamera.

Eine weitere Besonderheit ist die Zweiteilung des Datensets. So wurden die Blätter zum einem direkt am Baum, also vor unkontrolliertem Hintergrund fotografiert. Zum anderen

erfolgte auch eine Aufnahme vor kontrolliertem Hintergrund, nämlich einem Blatt Papier. Der Grund dafür ist, dass das später vorgestellte System unter realistischen Bedingungen getestet werden soll. So ist es für einen potentiellen Nutzer einfacher, ein Blatt direkt am Baum oder auf einem Stück Papier mit seinem Handy abzufotografieren, als es einzusammeln und zu Hause mit einem Scanner (falls vorhanden) einzulesen.

Die verwendeten Spezies sind in Abbildung 35 zu sehen.



Abbildung 35: Je ein Beispielblatt pro Spezies aus dem BLD.

Bei der Erstellung des Datensets wurde darauf geachtet, eine möglichst repräsentative Auswahl an Blättern und Aufnahmesituation zu erreichen. So wurden zum einen auch von der Norm abweichende Blätter verwendet. Diese Herangehensweise unterscheidet sich deutlich von der der meisten anderen Datensets, speziell Flavia und Foliage. Darunter fallen u. a. leicht deformierte Blätter (z. B. mit kleinen ein- oder abgerissenen Stellen) oder Blätter, die durch Krankheiten oder Umwelteinflüsse deutlich in ihrer Erscheinung verändert wurden (z. B. schwarze Flecken oder Löcher).

Zum anderen wurde, bei Aufnahmen am Baum, keine Rücksicht auf den vorherrschenden Hintergrund genommen. Dies beinhaltet auch, dass die Blätter zum größten Teil bei der Aufnahme des Fotos nicht vor den Stamm des Baumes oder einen unauffälligen Hintergrund (z. B. einheitlicher Boden) in der Umgebung gehalten wurden. Daraus ergaben sich zum Teil Aufnahmen mit sehr komplexen Hintergründen, wie beispielsweise weitere Blätter derselben Spezies, Buschwerk oder sehr uneinheitlicher Boden.

Die Aufnahmen auf Papier wurden im Freien und zu unterschiedlichen Tageszeiten durchgeführt. Durch den daraus resultierenden, wechselnden Lichteinfall ergaben sich teilweise sehr unterschiedliche und auch in der Intensität stark variierende Schattenverläufe. Dies erschwert die Vorverarbeitung beträchtlich, spiegelt jedoch einen realistischen Anwendungsverlauf wider.

Des Weiteren wurde Stielen keine Aufmerksamkeit gewidmet. Es wurden weder Stiele entfernt noch wurde darauf geachtet, dass jedes fotografierte Blatt noch über einen Stiel verfügt. Auch die Ausrichtung des Blattes auf dem Papier erfolgte weitestgehend willkürlich. Die einzige Vorgabe war, dass stets die Vorderseite des Blattes fotografiert wurde.

Das BLD ist mit nur zehn Spezies relativ klein, eignet sich aber sehr gut zum Testen der in späteren Kapiteln vorgestellten Segmentierungs- und Klassifikationsmethoden.

Abbildung 36 zeigt Beispielaufnahmen sowohl auf Papier, als auch am Baum.



Abbildung 36: Beispielaufnahmen von Blättern aus dem BLD. Oben: auf Papier, unten: am Baum.

3.7 Überblick

In Tabelle 1 sind die wichtigsten Eckdaten aller verwendeten Datensets noch einmal zusammengefasst:

Tabelle 1: Verwendete Datensets.

Datenset	# Spezies	# Blätter pro Spezies	Art der Segmentierung
Flavia	32	ca. 60	vorsegmentiert
Foliage	60	120	vorsegmentiert
Swedish Leaf Dataset	15	75	Scan
Middle European Woody	153	ca. 63	Scan
Intelligent Computing Laboratory	220	ca. 76	Scan
BLD	10	ca. 90	Foto

3.8 Weitere, nicht verwendete Datensets

Es gibt noch eine Vielzahl weiterer Datensets, die in dieser Arbeit keine Anwendung finden. In den meisten Fällen ist die Begründung, dass das betreffende Datenset entweder eine zu geringe Anzahl an Spezies oder zu wenig Bilder pro Spezies enthält. Ein weiteres Ausschlusskriterium war es, wenn ein Datenset lediglich Binärbilder oder sogar nur Bilder mit der äußeren Kontur eines Blattes zur Verfügung stellte, da dies sofort einige mögliche Methoden zur Klassifikation, wie z. B. Farb-Features oder Venierungsfeatures, ausschloss.

Zwei der nicht verwendeten Datensets sollen im Folgenden noch näher vorgestellt werden, da sie aufgrund ihres Umfangs durchaus geeignet gewesen wären, jedoch aufgrund anderer Kriterien aussortiert wurden.

3.8.1 Leafsnap

Leafsnap ist die bekannteste, kostenlos verfügbare Baumerkennungs-App für iOS. Zum Zeitpunkt des Verfassens dieser Arbeit wurde die App bereits über eine Million Mal heruntergeladen. Mit ihr ist es möglich, Blätter, die auf weißem, texturlosem Hintergrund fotografiert wurden, automatisch zu segmentieren und zu klassifizieren.

Der Fokus liegt dabei hauptsächlich auf Baumspezies, die in Nordamerika heimisch sind. Insgesamt befinden sich im Leafsnap Datenset aktuell 184 verschiedene Spezies. Das Datenset ist unterteilt in zwei Teilssets, die beide bei der Klassifikation zum Einsatz kommen. Das erste Teilsset besteht aus 23.915 Bildern, die jeweils unter Laborbedingungen mit immer gleichen Lichtverhältnissen und einer qualitativ hochwertigen Kamera aufgenommen wurden. Im zweiten Set befinden sich insgesamt 5.192 Bilder, die unter realistischen Bedingungen und mit verschiedenen Handykameras aufgenommen wurden. [Kumar 2012]

In Abbildung 37 ist je ein Exemplar aller 184 Spezies zu sehen.



Abbildung 37: Je ein Beispielblatt pro Spezies aus dem Leafsnap Datenset. [Kumar 2012]

Obwohl das Leafsnap Datenset die Ansprüche sowohl an die Spezies-Vielfalt als auch an die Blattanzahl pro Spezies klar erfüllt, kommt das Datenset in dieser Arbeit nicht zum Einsatz. Der Grund dafür ist, dass das Datenset wenig Vergleichsmöglichkeiten bietet. In aktuellen Veröffentlichungen werden die jeweiligen Systeme fast ausnahmslos auf einem oder mehreren der oben genannten Datensets evaluiert. Selbst der direkte Vergleich mit der Originalarbeit (vgl. [Kumar 2012]) wurde in dieser Arbeit nicht in Betracht gezogen, da sich aufgrund mangelnder Angaben zum Evaluationsverfahren keine aussagekräftigen Vergleiche ziehen lassen.

Eine zukünftige Integration des Leafsnap Datensets in das finale Trainings- bzw. Vergleichsdatsenset ist allerdings durchaus denkbar, vor allem da die in Nordamerika heimischen Spezies durch die anderen Datensets nur unzureichend abgedeckt werden.

3.8.2 LifeCLEF

Die *LifeCLEF 2015 Plant Task* (CLEF=Cross Language Evaluation Forum) ist Teil der ImageCLEF 2015-Evaluationskampagne, deren Ziel der Vergleich verschiedenster Klassifikations- und Evaluationsaufgaben ist. So wurden neben der Plant-Task, bei der die Bestimmung von Pflanzen im Vordergrund stand, auch ähnliche Aufgaben für die Klassifikation von Fischen und Vögeln gestellt. Überdies können sich die registrierten

Teams u. a. in den Bereichen Allgemeine Bild-Annotation, Medizinische Klassifikation und Clustering messen.

Das zur Verfügung gestellte LifeCLEF 2015 Plant Dataset unterscheidet sich deutlich von den bereits vorgestellten Datensets. Zunächst beschränken sich die Fotos nicht nur auf die Blätter, sondern zeigen auch Blüten, Früchte, Zweige, Stämme oder den gesamten Habitus des Baumes. Des Weiteren sollen neben Bäumen auch Sträucher und Blumen klassifiziert werden. Jedem Bild ist eine XML-Datei zugeordnet, die neben dem wissenschaftlichen Namen der Spezies weitere Informationen, wie das Entstehungsdatum und den Ort der Aufnahme, speichert. Außerdem ist zu jedem Bild ein sogenannter Vote-Wert abrufbar. Dieser bewertet die Bildqualität und wurde über Jahre hinweg von Mitgliedern der *Tela Botanica* (URL: <http://www.tela-botanica.org/>), einer Art Facebook für Botaniker, kreiert. Einige Bewertungen verschiedener Spezies sind in Abbildung 38 zu sehen. Dabei beschreibt eine Bewertung nahe der 5-Sterne-Marke ein „gutes Foto“, ausgezeichnet dadurch, dass das betreffende Organ der Pflanze klar im Mittelpunkt des Bildes steht und möglichst viel Fläche im Bild einnimmt. Überdies soll die Aufnahme fokussiert sein, also den Vordergrund scharf darstellen und den Hintergrund bereits etwas verschleiern.

Stars	★★★★★	★★★★☆	★★★☆☆	★★☆☆☆	★☆☆☆☆
Branch <i>Cercis siliquastrum</i> L.					
Entire <i>Quercus ilex</i> L.					
Leaf (photo) <i>Pittosporum tobira</i> L.					
Leaf (scan & scan-like) <i>Hedera helix</i> L.					
Flower <i>Papaver rhoeas</i> L.					
Fruit <i>Crataegus monogyna</i> L.					
Stem <i>Betula pendula</i> L.					

Abbildung 38: Das LifeCLEF 2015 Plant Dataset (Auswahl) inklusive Vote-Bewertung. [LifeCLEF 2015]

Insgesamt beinhaltet das LifeCLEF 2015 Plant Dataset 113.205 Fotos und Scans von 1000 Spezies, die hauptsächlich in Frankreich oder in den an Frankreich angrenzenden Ländern heimisch sind.

Das LifeCLEF Datenset erfüllt offensichtlich die geforderten Kriterien an Vielfalt und Umfang und besticht durch seine professionelle Zusammenstellung und Struktur. Trotzdem wird es in dieser Arbeit nicht verwendet, da die Vergleichbarkeit mit dem hier vorgestellten System nur sehr eingeschränkt gegeben ist. Zum einen beschäftigt sich diese Arbeit ausschließlich mit der Klassifikation anhand von Blättern und vernachlässigt weitere Organe wie Blüte, Frucht, Stamm, Zweig und Habitus. Zum anderen steht beim vorgestellten System die Klassifikation nur anhand eines Fotos und ohne weitere Hilfsmittel im Vordergrund. So ist z. B. die Sterne-Bewertung, die Hinweise auf die Art der benötigten Vorverarbeitung geben kann, nicht erwünscht.

Wie beim Leafsnap Datenset gilt auch für das LifeCLEF Datenset, dass es durchaus Sinn machen kann, zumindest den Leaf (scan & scan-like)-Teil, siehe Zeile 4 in Abbildung 38, in das letztendliche Trainings- bzw. Vergleichsdatenset zu integrieren.

3.9 Begriffsklärung

Zwei der vorgestellten Datensets, Foliage und ICL, enthalten Bilder von Blättern, die nicht von Bäumen, sondern von Sträuchern, Blumen oder anderen Gewächsen stammen. Da es für das Vorgehen in dieser Arbeit keinen Unterschied macht, wurde darauf verzichtet, derartige Spezies auszusortieren. Für Teile der späteren Evaluation ist es sogar von Vorteil, auf eine möglichst umfangreiche Menge verschiedener Arten zurückgreifen zu können. Das Grundthema bleibt jedoch weiterhin die Baum-klassifikation, da diese auch in den meisten vergleichbaren Systemen im Vordergrund steht. Ist im weiteren Verlauf die Rede von „Bäumen“ oder deren Blättern, schließt dies, soweit nicht ausdrücklich anders erwähnt, alle Spezies ein, die in den eben vorgestellten Datensets vorkommen.

4 Vorverarbeitung und Segmentierung

Als Eingabe für die Feature-Extraktion wird ein Bild erwartet, auf dem alle Pixel, die nicht zum Blatt gehören, vollkommen weiß sind, also den Graustufenwert 255 und damit RGB(255, 255, 255) besitzen. Bei den in dieser Arbeit verwendeten Datensets erfüllen diese Voraussetzung lediglich das Flavia- und das Foliage-Datenset. Bei allen anderen Datensets müssen die Blätter noch segmentiert, d. h. vom Hintergrund freigestellt werden. Abbildung 39 zeigt Beispiele von erfolgreichen Segmentierungen sowohl am Baum als auch auf Papier.



Abbildung 39: Beispiele für eine erfolgreiche Segmentierung.
Oben: auf Papier, unten: am Baum.

Im Folgenden sollen zunächst verschiedene Standard-Ansätze aus der Bildverarbeitung erläutert werden. Anschließend werden ein eigener Segmentierungsalgorithmus und eine Methode zur manuellen Segmentierung vorgestellt. Vor dem abschließenden Vergleich wird die Herangehensweise zur Entfernung der Blattstiele erläutert.

Bei allen Ansätzen wird dabei vorausgesetzt, dass sich das freizuschneidende Blatt vor einem hellen, einheitlichen Hintergrund befindet. Dies wird entweder durch Scans oder durch das Fotografieren des Blattes auf Papier gewährleistet. Fast alle verwendeten Datensets erfüllen diese Voraussetzung. Lediglich die am Baum fotografierten Blätter stellen eine Ausnahme dar. Bei ihnen ist es nach Auffassung dieser Arbeit nicht möglich, eine zuverlässige automatische Segmentierung vorzunehmen, was vor allem an den auftretenden komplexen Hintergründen liegt. Daher muss bei am Baum aufgenommenen Bildern die Segmentierung stets mit Unterstützung des Nutzers erfolgen.

4.1 Standardverfahren zur automatischen Segmentierung

Da die Segmentierung in der Bildverarbeitung eine wichtige und ständig wiederkehrende Aufgabe darstellt, existieren dazu einige Standardansätze, die im Folgenden kurz erläutert werden sollen. Geordnet nach ansteigender Komplexität werden dabei zunächst Otsus Methode, das K-Means-Verfahren und abschließend der Watershed-Algorithmus vorgestellt.

4.1.1 Otsus Methode

Die einfachste Segmentierungsmethode stellt eine simple Grenzwertoperation dar. Das Eingangsbild wird zunächst in ein Graustufenbild umgewandelt. Anschließend wird ein Grenzwert bestimmt und alle Pixel unterhalb dieses Wertes als Vordergrund und die restlichen Pixel als Hintergrund definiert oder umgekehrt. Ein manuell festgelegter, fixer Grenzwert macht aufgrund wechselnder Belichtungsverhältnisse nur wenig Sinn. Die bereits in 2.1.1.3 vorgestellte Methode von Otsu (vgl. [Otsu 1979]) ermöglicht eine simple Bestimmung eines statistisch optimalen Grenzwertes.

Der restliche Segmentierungsablauf gestaltet sich sehr einfach. Zunächst wird, wie in Abbildung 40 zu sehen, die Binärdarstellung des Eingangsbildes berechnet. Anschließend werden anstelle der schwarzen Vordergrundpixel wieder die ursprünglichen Farbpixel eingefügt, da für die folgende Feature-Extraktion ein Farbbild erwartet wird. Die Hintergrundpixel, also die Pixel mit einem Graustufenwert von 255, werden lediglich auf die neue Farbdarstellung angepasst und erhalten einen Wert von RGB(255, 255, 255).



Abbildung 40: Otsus Methode am Beispiel eines Blattes. Links: Original, Mitte: Binärdarstellung, rechts: Ergebnis der Segmentierung.

Leicht lässt sich die Problematik dieser Methode erkennen: Sie ist sehr anfällig gegenüber Schatten, da dieser im Vergleich zum hellen Hintergrund sehr dunkel ist und daher größtenteils als Vordergrund klassifiziert wird. Im Verlauf dieses Kapitels wird sich zeigen, dass der Schatten die mit Abstand größte Hürde bei der Segmentierung darstellt. Die Identifikation und Entfernung des Schattens ist keinesfalls trivial. Simple Annahmen, wie beispielsweise die, dass Schattenpixel stets dunkler sind als Blattpixel, lassen sich schnell widerlegen. Ebenso ist es nicht hilfreich, sich auf einzelne Farbkanäle des RGB-Modells zu stützen. Die Vermutung, dass Blattpixel höhere Grünwerte besitzen als Schattenpixel, trifft zwar in der Mehrheit der Fälle, jedoch bei Weitem nicht immer zu. Dies wird z. B. bei dem in 3.2 vorgestellten Foliage-Datenset deutlich. Nicht wenige Blätter haben nicht die typische Grünfärbung, sondern sind rot oder gelb. Es ist demnach sehr schwierig allgemein gültige Annahmen zu treffen.

4.1.2 K-Means-Clustering

Das K-Means-Clustering ist die einfachste und am häufigsten benutzte Clustering-Methode und kommt oft im Bereich des Data-Minings zum Einsatz. Die Grundidee ist, eine gegebene Menge von N Instanzen, die jeweils über eine Menge von Eigenschaften $V(x_i)$ verfügen, K Clustern zuzuordnen. Die Anzahl K an Clustern wird dabei vom Nutzer vorgegeben. Ziel ist es, dass nach Anwendung des Clusterings die Instanzen innerhalb eines Cluster sich möglichst stark ähneln und Instanzen aus verschiedenen Clustern sich möglichst stark unterscheiden.

Der eigentliche Algorithmus besteht aus drei einfachen Schritten:

1. Initialisierung: Zu Beginn müssen die Cluster bzw. deren Mittelwerte initialisiert werden. Beim klassischen K-Means-Algorithmus erfolgt die Initialisierung durch die zufällige Auswahl von K Instanzen. Verschiedene Initialisierungen können durchaus zu unterschiedlichen Endergebnissen führen.
2. Zuweisen der Instanzen zu den Clustern: Jedes Cluster C_K besitzt einen Mittelwert μ_K . Für alle Instanzen lässt sich nun mittels einer Distanzfunktion die Distanz zwischen der jeweiligen Instanz und den K Clustern, gegeben durch ihren Mittelwert μ_K , bestimmen. Die Instanzen werden dem Cluster zugewiesen, dem sie am ähnlichsten sind, also dem, zu dem sie die geringste Distanz haben.
3. Neuberechnung der Parameter: Nach Zuweisung der Instanzen müssen die Mittelwerte der Cluster neu berechnet werden. Anschließend wird die in Schritt 2 beschriebene Zuweisung wiederholt. Dies erfolgt so lange, bis keine Änderungen bei den Zuordnungen mehr auftreten.

[Liu 2012]

Der eben vorgestellte Ansatz lässt sich leicht auf die hier vorliegende Segmentierungsaufgabe übertragen. Jeder Pixel entspricht einer Instanz. Sein Eigenschaftensvektor enthält die Farbwerte des Pixels: Rot-, Grün- und Blau-Wert. K ist trivialerweise gleich 2: Ein Cluster enthält die Vordergrundpixel, das zweite die Hintergrundpixel. Abbildung 41 verdeutlicht die Verteilung verschiedenfarbiger Pixel im RGB-Raum.

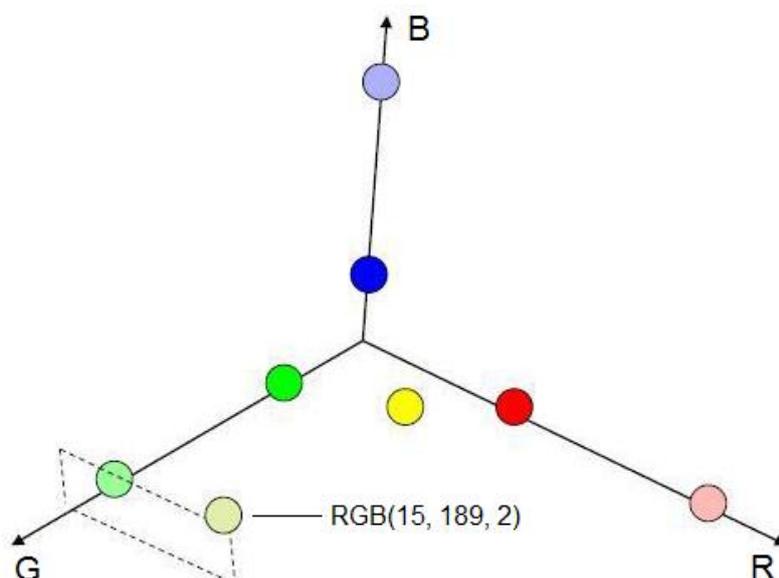


Abbildung 41: Pixel im dreidimensionalen RGB-Raum.

Die Initialisierung erfolgt zufällig. Wie oben erwähnt, können unterschiedliche Initialisierungen durchaus zu verschiedenen Clusterungen führen. Im vorliegenden Fall lohnt es sich allerdings nicht, spezielle Ausgangscluster festzulegen. Das liegt daran, dass sich die beiden Cluster für Vordergrund und Hintergrund massiv unterscheiden und auch sehr unterschiedliche Ausgangslagen zum selben Segmentierungsergebnis führen.

Als Distanzfunktion dient die Euklidische Distanz. Sei der Mittelwert eines Cluster gegeben durch die drei Mittelwerte (μ_R, μ_G, μ_B) des Rot-, Grün- und Blau-Kanals (R, G, B) all seiner zugehörigen Pixel:

$$\mu_R = \frac{1}{N} \cdot \sum_{i=1}^N R_i \quad (4.1)$$

$$\mu_G = \frac{1}{N} \cdot \sum_{i=1}^N G_i \quad (4.2)$$

$$\mu_B = \frac{1}{N} \cdot \sum_{i=1}^N B_i \quad (4.3)$$

Dabei beschreibt N die Anzahl der zum Cluster gehörigen Pixel.

Die Distanz d eines Pixels P mit den Werten (R, G, B) zu einem Zentrum $C(\mu_R, \mu_G, \mu_B)$ ist demnach:

$$d(P, C) = \sqrt{(\mu_R - R)^2 + (\mu_G - G)^2 + (\mu_B - B)^2} \quad (4.4)$$

Abbildung 42 zeigt das Ergebnis einer Segmentierung mit Hilfe des K-Means-Algorithmus. Um zu ermitteln, welches Cluster dem Vordergrund und welches dem Hintergrund entspricht, werden die Randpixel betrachtet. Das sind die äußersten Pixel des Bildes, also die in der ersten und letzten Spalte und Reihe. Da sich die Blätter stets mittig befinden, kann davon ausgegangen werden, dass das Cluster, das mehr Randpixel enthält, dem Hintergrund entspricht. Es lassen sich Fälle konstruieren, in denen diese Methode fehlschlägt, allerdings trat ein solcher bei allen in dieser Arbeit getesteten Bildern niemals auf.

Es lässt sich feststellen, dass das Ergebnis beinahe identisch mit dem durch Otsus Methode erzielten ist. Auch hier zeigt sich wieder die Schattenproblematik. Selbst unter Beachtung aller drei Farbkanäle sind die Schattenpixel im Schnitt den Blattpixeln ähnlicher als den Papierpixeln im Hintergrund.



Abbildung 42: K-Means-Segmentierung am Beispiel eines Blattes. Links: Original, Mitte: Finale Clusterung, rechts: Ergebnis der Segmentierung.

Auch ein Versuch, das Clustering mit $K=3$ durchzuführen, d. h. ein Cluster für den Hintergrund, eines für den Vordergrund und eines für den Schatten, blieb ohne Erfolg. Vor allem bei ungleichmäßig gefärbten Blättern kam häufig das Problem zum Vorschein, dass große Teile des Blattes falsch zugeordnet wurden und die Segmentierung unbrauchbar war.

4.1.3 Watershed

Der Watershed-Algorithmus ist in der Bildverarbeitung eine der verbreitetsten Segmentierungsmethoden. Im Folgenden soll zunächst kurz das Grundprinzip erläutert und anschließend eine Erweiterung vorgestellt werden. Abschließend erfolgt die Übertragung auf den in dieser Arbeit vorliegenden Fall der Blattsegmentierung.

In der Geografie bezeichnen *Watersheds* (englisch für: Wasserscheide, Wassereinzugsgebiet) den Grenzverlauf zwischen zwei benachbarten Flusssystemen oder Seen. Diese trennen die Einzugsgebiete der angrenzenden Gewässer.

4.1.3.1 Grundprinzip

Übertragen auf die Bildverarbeitung bietet der Watershed-Algorithmus eine Möglichkeit Bilder zu segmentieren, indem er Graustufenbilder als topologische Oberfläche betrachtet. Der Graustufenwert eines jeden Pixels gibt dessen Höhenlage an. Durch diese Betrachtung entstehen Berge, d. h. Regionen mit hohen Graustufenwerten, und Täler, d. h. Regionen mit niedrigen Graustufenwerten. Bildlich gesprochen erfolgt die Segmentierung durch das Auffüllen der Täler mit Wasser. Dabei darf zu jedem Zeitpunkt das Wasser überall nur gleich hoch stehen. Trifft Wasser aus zwei verschiedenen Becken aufeinander, entsteht an dieser Stelle eine Wasserscheide, die ein Mischen des Wassers verhindert. Ein Beispiel mit zwei Becken, dem ansteigenden Wasser und der später entstehenden Wasserscheide ist in Abbildung 43 zu sehen. Die Wasserscheiden kennzeichnen in der erhaltenen Segmentierung die Grenzen zwischen verschiedenen Bereichen des Bildes. [Liu 2012] [OpenCV 2008]

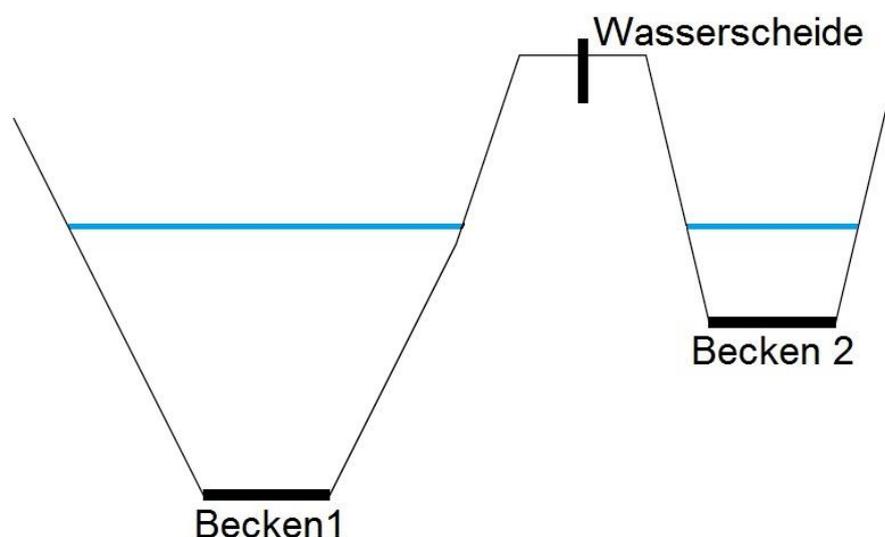


Abbildung 43: Bildliche Darstellung des Watershed-Algorithmus. [Liu 2012]

4.1.3.2 Marker-basierte Segmentierung

Im Normalfall besitzt ein zu segmentierendes Bild eine riesige Anzahl an unterschiedlichen Graustufenwerten und demnach zu viele Berge und Täler, um eine brauchbare Segmentierung zu erreichen. Diese sogenannte „Übersegmentierung“ ist in Abbildung 44 oben zu sehen. Als Beispiel wurde die Konturierung von Proteinen in elektrophoretischem Gel gewählt, da hierbei die Effekte deutlich besser zu sehen sind als bei der Segmentierung eines einzelnen Blattes. Das Problem der Übersegmentierung ist klar zu erkennen: Es entsteht eine hohe Anzahl von kleinen Auffangbecken und damit Segmentierungen, da es zu viele lokale Minima im Ausgangsbild gibt. Der Grund für diese lokalen Minima sind kleine Variationen der Graustufenwerte, hauptsächlich durch Rauschen. Eine mögliche Herangehensweise zur Verbesserung der Segmentierung wäre eine Filterung des Originalbildes, die das Rauschen größtenteils unterdrückt. Dies gestaltet sich allerdings schwierig, da zum einen bei zu schwacher Filterung nicht alle lokalen Minima entfernt werden können, zum anderen würde möglicherweise eine zu starke Filterung dazu führen, dass wichtige Informationen verloren gehen.

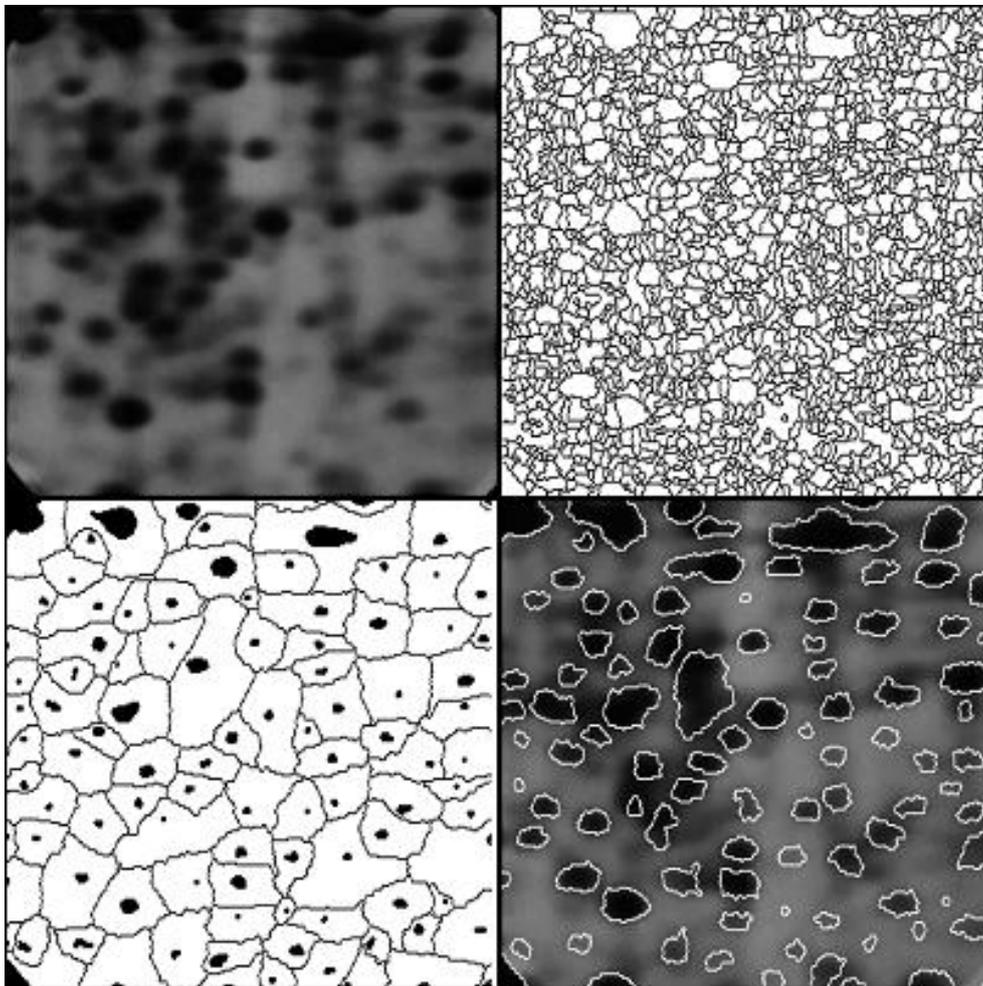


Abbildung 44: Auswirkung von Markern bei der Watershed-Segmentierung.
Oben: Originalbild (links), Ergebnis der Segmentierung ohne Marker (rechts);
unten: gewählte Marker (schwarz, links), Ergebnis der Segmentierung mit
Markern (rechts). [Beucher 1991]

Eine Lösung für dieses Problem liefert eine weit verbreitete Variante des Watershed-Algorithmus, bei der sogenannte *Marker* (englisch für: Markierung, Kennzeichnung) verwendet werden. Diese haben die Aufgabe, die Größe und die Anzahl der verschiedenen

Regionen zu begrenzen und zu kontrollieren (vgl. Abbildung 44 unten). Der grundsätzliche Ablauf der Segmentierung verändert sich nicht. Es wird lediglich die Oberfläche nicht bei jedem lokalen Minimum durchstoßen, sondern nur an den Stellen, die als Marker gekennzeichnet sind. Auf diese Weise kann das Wasser nur dort eindringen. In der Folge entstehen exakt so viele Auffangbecken wie die entsprechende Anzahl der zu Beginn festgelegten Marker. [Beucher 1991]

Die Zuweisung der Marker kann manuell oder automatisch erfolgen. Bei der automatischen Zuweisung wird häufig Otsus Methode verwendet. Um zu garantieren, dass als Marker nur sicherer Vordergrund und sicherer Hintergrund dienen, wird auf das Ergebnis im Normalfall eine Kombination aus Dilatation und Erosion oder eine Distanztransformation angewendet. Erstere ist in Abbildung 45 (oben) zu sehen. Die Erosion trägt den äußeren Rand der Münzen ab und gewährleistet damit, dass unsichere Bereiche nicht als Marker für Vordergrundbereiche verwendet werden. Analog verfährt die Dilatation mit den Hintergrundbereichen. Im hier gewählten Beispiel erschwert die Tatsache, dass sich die Münzen untereinander berühren, die Marker-Zuweisung erheblich, da die Erosion an den Verbindungsstellen nicht den gewünschten Effekt liefert.

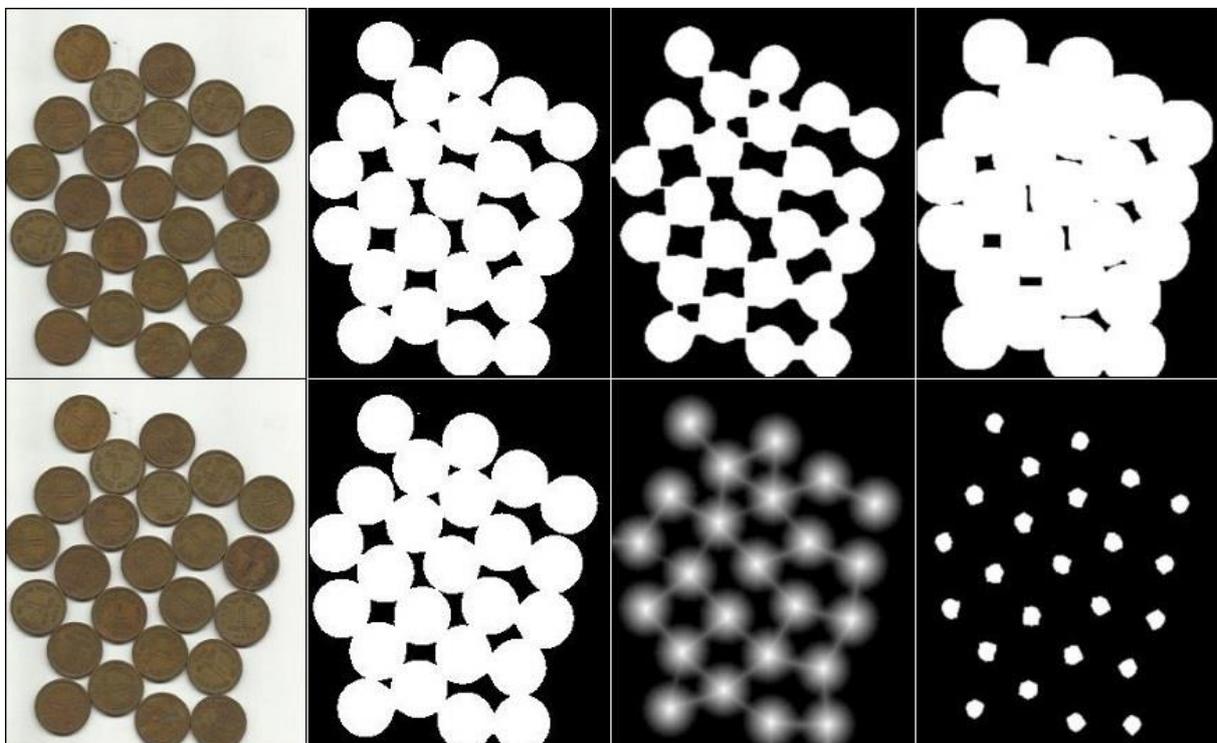


Abbildung 45: Marker-Generierung am Beispiel von Münzen. Jeweils von links nach rechts: Oben: Originalbild, Binärdarstellung nach Otsu, Ergebnis der Erosion, Ergebnis der Dilatation. Unten: Originalbild, Binärdarstellung nach Otsu, Ergebnis der Distanztransformation, Pixel mit einem Distanztransformations-Wert von mindesten 70% des maximalen Wertes. [OpenCV: Watershed]

In diesem Fall liefert die Distanztransformation ein besseres Ergebnis (vgl. Abbildung 45 unten). Dabei werden als sicherer Vordergrund lediglich die Stellen angesehen, die einen gewissen Mindestwert in der Distanztransformation-Darstellung und damit einen Mindestabstand zu den nächsten Hintergrundpixeln enthalten. Im hier zu sehenden Beispiel wurde als Grenzwert 70% des maximalen auftretenden Wertes gewählt. [OpenCV: Watershed]

4.1.3.3 Anwendung auf die Blatt-Segmentierung

Da die Schattenbereiche sehr groß sein können, bietet sich bei der Segmentierung von auf Papier fotografierten Blättern keine der eben vorgestellten Methoden zur Positionierung der Marker an. Stattdessen werden sich die jeweiligen Eigenschaften der Blatt-, Papier- und Schattenpixel in verschiedenen Darstellungsformen zunutze gemacht und daraus der sichere Vordergrund und Hintergrund und die zugehörigen Marker extrahiert. Dieser Vorgang wird in 4.3.2 ausführlich erläutert.

Abbildung 46 zeigt das Ergebnis einer Marker-gestützten Watershed-Segmentierung am Beispiel eines Blattes.

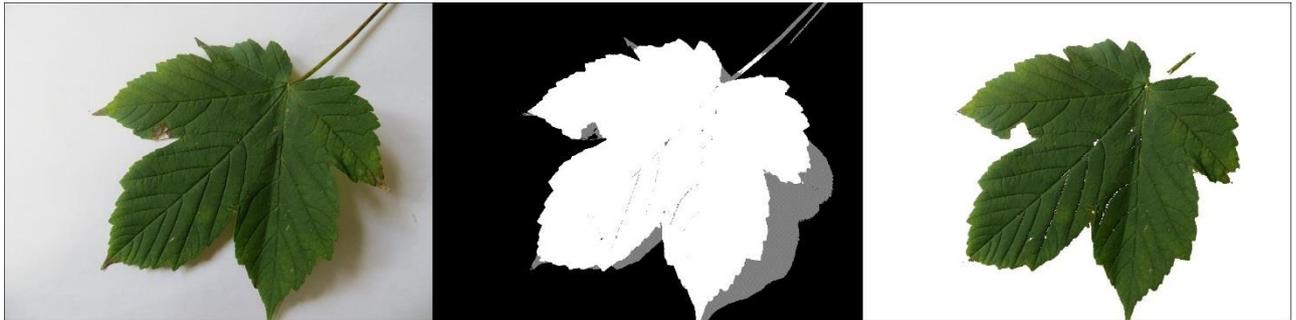


Abbildung 46: Watershed-Segmentierung am Beispiel eines Blattes. Links: Original, Mitte: zugewiesene Marker für sicheren Vordergrund (weiß) und sicheren Hintergrund (schwarz), rechts: Segmentierungsergebnis.

4.2 Der GrabCut-Algorithmus

Die bisher vorgestellten Methoden sind für einfache Segmentierungsaufgaben zumeist ausreichend. Je simpler der Hintergrund gehalten ist und je deutlicher sich dieser von dem zu extrahierenden Vordergrund unterscheidet, desto exaktere Ergebnisse sind möglich. In Situationen mit unkontrolliertem und komplexem Hintergrund stoßen automatische Methoden allerdings schnell an ihre Grenzen. So ist es z. B. unmöglich, mit Otsus Methode, dem K-Means- oder dem Watershed-Algorithmus zuverlässig am Baum fotografierte Blätter vom Hintergrund freizustellen, da dieser völlig unkontrolliert ist und sowohl andere Blätter, Bäume, Sträucher, Teile des Himmels oder sonstige Objekte enthalten kann.

Für die Segmentierung solch komplexer Bilder ist meist eine vom Nutzer überwachte Methode unerlässlich. Eine dieser Methoden, der sogenannte GrabCut-Algorithmus (vgl. [Rother 2004]), soll im Folgenden vorgestellt werden. Dazu werden zunächst detailliert die Grundlagen erläutert, auf denen der Algorithmus basiert. Diese gehen in erster Linie auf das Prinzip von iterierten Graph Cuts (vgl. [Boykov 2001]) zurück. Anschließend werden kurz einige Verbesserungen vorgestellt, die den GrabCut-Algorithmus von der ursprünglichen Version unterscheiden.

4.2.1 GraphCut

Graphen finden häufig Verwendung im Bereich der Bildverarbeitung, da sich die Pixel-basierte Darstellung fast ohne Aufwand direkt in einen Graphen umwandeln lässt. Dies wird im Folgenden näher erläutert. Zunächst müssen aber einige Grundbegriffe aus der Graphentheorie geklärt werden.

4.2.1.1 Grundbegriffe

Ein Graph $G = \langle V, E \rangle$ ist definiert durch eine Menge von Knoten V (englisch, *vertice*: Knoten) und Kanten E (englisch, *edge*: Kante), die diese Knoten verbinden. Ist für die Kanten keine Richtung angegeben, spricht man von einem ungerichteten Graphen.

Des Weiteren lässt sich jeder Kante $e \in E$ ein Kantengewicht w_e zuweisen, oft auch als Kosten bezeichnet.

Ein Schnitt ist eine Teilmenge von Kanten $C \subset E$. Die Kosten eines Schnittes berechnen sich aus der Summe der Einzelgewichte der geschnittenen Kanten:

$$|C| = \sum_{e \in E} w_e \quad (4.5)$$

4.2.1.2 Überführung des Bildes in einen Graphen

Mit Hilfe dieser Grundlagen lässt sich ein Bild wie folgt in einen Graphen überführen:

Pixel sind die Knoten des Graphen. Horizontal und vertikal direkt benachbarte Pixel sind durch Kanten verbunden. Das Kantengewicht beschreibt, basierend auf den Farbwerten, die Ähnlichkeit der zugehörigen Pixel. Dabei steht ein niedriger Wert für eine schwache und ein hoher Wert für eine starke Ähnlichkeit der Pixel. Außerdem werden zusätzlich zwei spezielle Knoten, sogenannte Terminals bzw. Endknoten, eingeführt. Diese definieren im Verlaufe des Algorithmus das jeweils aktuelle Modell für den Hintergrund und den Vordergrund. Jeder normale Knoten, d. h. jeder Pixel des Bildes, ist durch eine Kante mit beiden Endknoten verbunden. Analog zu den eben beschriebenen Beziehungen zwischen aneinander angrenzenden Pixeln gibt auch hier das Kantengewicht die Ähnlichkeit zwischen den jeweiligen Knoten an.

Im Folgenden werden Kanten, die Nachbarknoten verbinden, als Nachbarkanten bezeichnet. Analog sind Endkanten die Kanten, die einen Knoten mit einem der beiden Endknoten verbinden.

Zusätzlich zu der eben beschriebenen Ausgangsdarstellung ist es bei der Initialisierung des Algorithmus notwendig, dass der Nutzer manuell sogenannte Seeds (englisch, *seed*: Same, Keim) setzt, indem er Pixel markiert, die sich sicher dem Vordergrund bzw. dem Hintergrund zuordnen lassen. Dadurch entstehen die beiden Teilmengen $O \subset P$ und $B \subset P$, die die Menge von als Vordergrund bzw. als Hintergrund markierten Pixeln beschreiben. Dabei dürfen Pixel nicht gleichzeitig beiden Klassen zugeordnet sein. Es gilt demnach $O \cap B = \emptyset$.

Die eben erwähnte manuelle Zuweisung ist in Abbildung 47 links zu sehen. Der Pixel p links oben ist als sicherer Hintergrund markiert, der Pixel v rechts unten als sicherer Vordergrund.

Die Dicke der Kanten gibt deren Gewicht an. p und v wurden fix zugeordnet, somit werden deren Verbindungskanten zu den beiden Terminals mit speziellen Extremwerten versehen. Da p maximal mit dem Hintergrund-Terminal T verbunden ist, erhält deren Kante den maximal möglichen Wert. Dementsprechend erhält die Kante zwischen p und dem Vordergrund- oder Objekt-Terminal den minimalen Wert 0. Dies ist gleichbedeutend mit einem Entfernen dieser Kante aus dem Graphen.

Analog bekommen die Kanten zwischen v und S bzw. T den maximalen bzw. den minimalen Wert 0 zugewiesen.

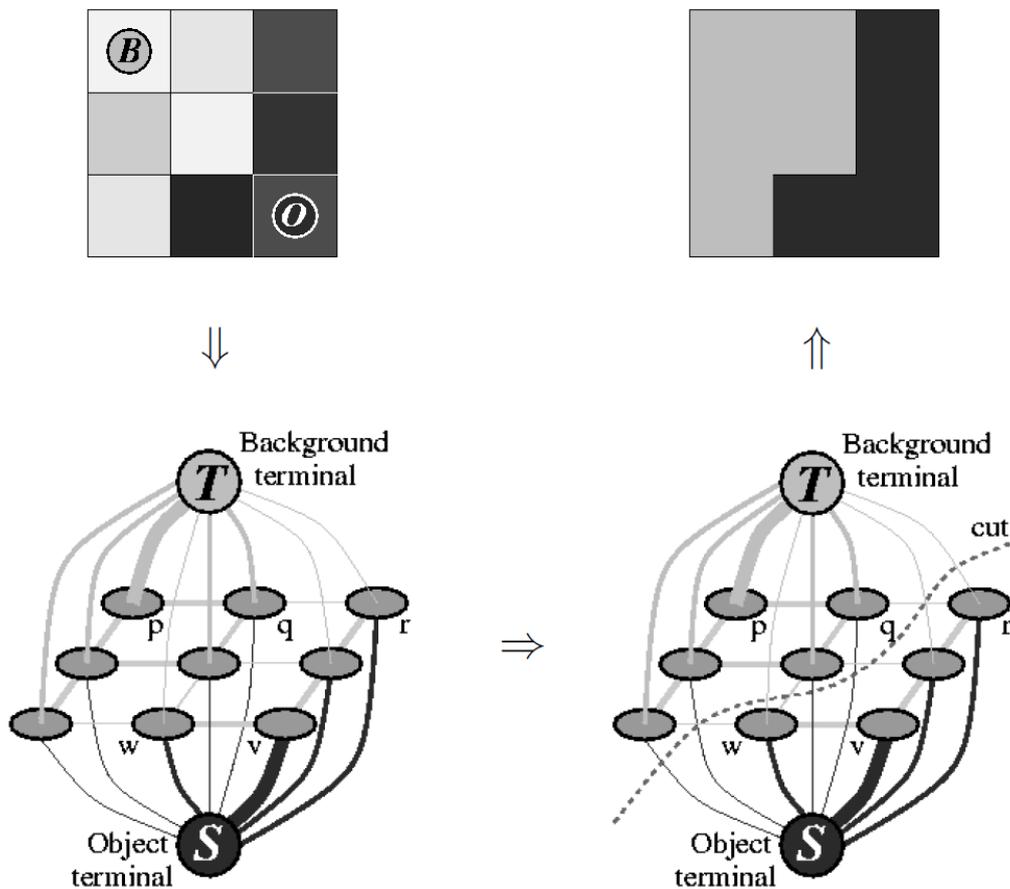


Abbildung 47: Schematische Darstellung einer GraphCut-Iteration. Links: Ausgangsbild mit Seeds (oben), Ausgangsbild als Graph (unten), rechts: Minimaler Schnitt (unten), Ergebnisbild (oben). [Boykov 2001]

Der maximale Wert wird bei einer theoretischen Betrachtung im Normalfall mit $+\infty$ angegeben. Da sich dies in einer realen Implementierung schwierig gestaltet, wird den Endkanten von Pixeln, die vom Nutzer als sicherer Vordergrund bzw. sicherer Hintergrund markiert wurden, ein Wert K zugewiesen, der sich wie folgt berechnet:

$$K = 1 + \max_{p \in P} \sum_{q: \{p,q\} \in N} B_{\{p,q\}} \quad (4.6)$$

Dabei bezeichnet $B_{\{p,q\}}$ das Kantengewicht der Kante zwischen den Knoten p und q und N die Menge der Kanten benachbarter Knoten. In diesem Zusammenhang dürfen weder p noch q Endknoten sein. Dadurch wird sichergestellt, dass vom Nutzer markierte Knoten beim nachfolgenden *Minimalen Schnitt* nicht abgetrennt werden, da alle Nachbarkanten ein geringeres Gewicht besitzen.

4.2.1.3 Segmentierungsschritt

Der eigentliche Segmentierungsschritt erfolgt durch das Berechnen des eben erwähnten minimalen Schnitts. Ziel ist es, dabei Kanten so zu durchschneiden, dass keine Verbindung

mehr zwischen den Endknoten B und O besteht. Dies entspricht der Zuordnung jedes Knotens, also jedes Pixels, zum Vordergrund oder zum Hintergrund. Aus allen möglichen Schnitten wird der ausgewählt, bei dem die Summe der Gewichte der durchtrennten Kanten minimal ist. In Abbildung 47 ist rechts schematisch ein minimaler Schnitt und die daraus resultierende Segmentierung dargestellt.

Der Vorgang ist allerdings an dieser Stelle noch nicht beendet. Durch die Neuzuweisung von Pixeln zum Vordergrund und Hintergrund ändern sich die Werte der Endknoten und damit auch die Kantengewichte der Endkanten. Mit den neuen Werten wird wiederum ein minimaler Schnitt berechnet. Ähnlich wie beim K-Means-Algorithmus wird dieses Vorgehen entweder solange wiederholt, bis das Ergebnis konvergiert oder bis eine vorgegebene Maximalzahl an Iterationen erreicht wurde.

Ist der Nutzer mit dem so erhaltenen Ergebnis nicht zufrieden, kann er, wie zu Beginn der Segmentierung, manuell Pixel als sicheren Vordergrund oder sicheren Hintergrund klassifizieren und die Berechnung erneut starten.

[Boykov 2001]

4.2.2 Erweiterung zu GrabCut

Der in OpenCV zum Einsatz kommende Algorithmus implementiert zusätzlich zu dem eben vorgestellten GraphCut-Algorithmus zwei Erweiterungen, die Rother et al. (vgl. [Rother 2004]) entwickelt haben.

Zum einen wird dem Nutzer eine einfachere Möglichkeit zur Festlegung der ursprünglichen Vorder- bzw. Hintergrundpixel zu Verfügung gestellt. Dies ist in Abbildung 48 zu sehen.

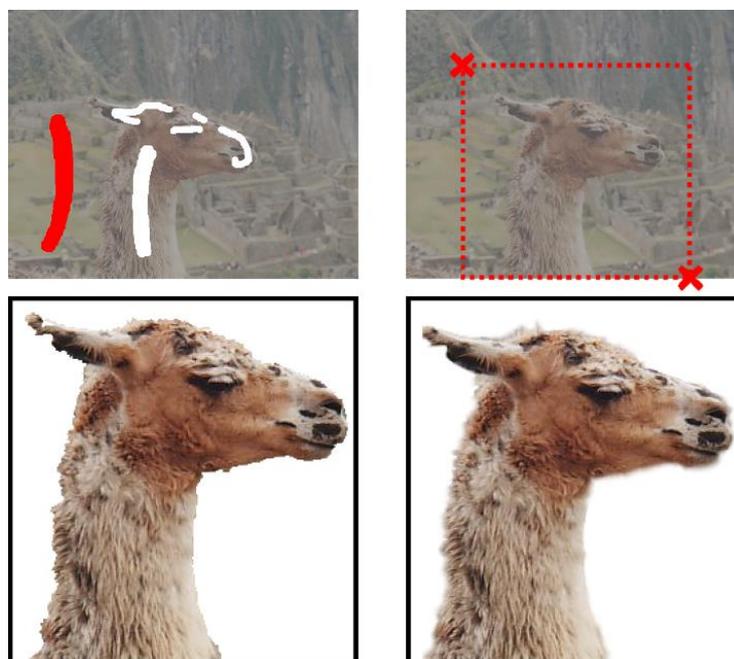


Abbildung 48: Erweiterungen des GrabCut-Algorithmus anhand eines Beispiels. Links: GraphCut, rechts: GrabCut; oben: Initialisierung durch den Nutzer, unten: Ergebnis der Segmentierung. [Rother 2004]

Beim ursprünglichen GraphCut-Algorithmus muss der Nutzer sowohl Hintergrund (rot) als auch Vordergrund (weiß) manuell markieren. Gerade in Randbereichen, wie hier z. B.

an den Ohren oder der Schnauze, sollte dies durchaus präzise erfolgen. Demgegenüber genügt es beim GrabCut-Algorithmus, ein Rechteck um das zu segmentierende Objekt aufzuziehen.

Ein weiterer großer Vorteil ist die Pixelzuweisung in den Randbereichen. Wie in Abbildung 48 unten zu sehen, liefert der GrabCut-Algorithmus, trotz der deutlich weniger aufwendigen Initialisierung, sichtbar bessere Ergebnisse in den Übergangsbereichen zwischen Vorder- und Hintergrund. Dies wird besonders im Halsbereich erkennbar, rechts erfolgt der Übergang deutlich weicher. Verantwortlich dafür ist das sogenannte *border matting* (englisch *border*: Grenze, Rand; *matting*: Verfilzung). Dabei wird zunächst eine Trimap erstellt, die die Pixel des Bildes in Vordergrund, Hintergrund und einen Übergangsbereich unterteilt. Durch Energieminimierung werden die Pixel im genannten Bereich einer der beiden Klassen zugewiesen. Auf eine exakte Beschreibung wird hier bewusst verzichtet, da es den Rahmen dieser Arbeit sprengen würde und es sich um einen hoch mathematischen und wenig anschaulichen Vorgang handelt. [Rother 2004]

4.2.3 Praktische Umsetzung der manuellen Segmentierung mittels GrabCut

Abbildung 49 zeigt den beispielhaften Ablauf einer Segmentierung durch den User mit Hilfe von GrabCut. In diesem Fall soll der Vordergrund ausschließlich aus dem großen Blatt in der Mitte des Bildes bestehen. Der Rest soll als Hintergrund definiert und ausgeblendet werden.

Der erste Schritt erfordert direkt ein Eingreifen des Nutzers. Mittels eines Rechtecks (hier: rot; s. oben rechts) wird zunächst grob zwischen Vordergrund und Hintergrund unterschieden. Dies geschieht, indem erstmalig eine GrabCut-Maske erstellt wird, die wie folgt initialisiert wird: Alle Pixel, die innerhalb des roten Rechtecks liegen, werden als wahrscheinlicher Vordergrund klassifiziert, die Pixel außerhalb des Rechtecks dagegen als sicherer Hintergrund. Sind Pixel erst einmal als sicherer Vorder- oder Hintergrund markiert, lässt sich dies nicht mehr rückgängig machen. Es ist daher wichtig, dass beim Aufspannen des Rechtecks der komplette Vordergrund im Inneren liegt. Je enger das Rechteck an den Vordergrund anliegt, desto besser wird im Normalfall der erste Segmentierungsvorschlag.

Mit Hilfe der Eingabemaske wird unter Verwendung des GrabCut-Algorithmus ein erstes Segmentierungsergebnis (s. Mitte links) berechnet. In den meisten Fällen liefert dieser Schritt bereits ein recht brauchbares Ergebnis. Allerdings sind häufig noch einige Regionen falsch zugeordnet. Ähnlich wie im hier gezeigten Beispiel handelt es sich dabei meist um fälschlicherweise als Hintergrund klassifizierte Randbereiche des Blattes oder um an das Blatt anschließende Teile des Hintergrundes, die als Vordergrund erkannt wurden. Um einen besseren Überblick zu bekommen, kann der Nutzer sich das Ausgangsbild als Graustufenbild anzeigen lassen, in dem lediglich der Rot-Kanal der Pixel, die bisher als Vordergrund markiert sind, auf den maximalen Wert gesetzt wird (s. Mitte rechts).

Ist der Nutzer mit der erhaltenen Segmentierung nicht zufrieden, hat er die Möglichkeit, manuell Pixel als sicheren Vordergrund oder sicheren Hintergrund zu markieren (s. unten links). Zu diesem Zweck kann im Bild der Vordergrund grün und der Hintergrund blau markiert werden. Dabei ist es nicht nötig, alle falsch klassifizierte Pixel zu kennzeichnen. Wie im Beispiel zu sehen, genügt es meist kleinere Pixelgruppen zu markieren.

Die GrabCut-Eingabemaske wird entsprechend der neu gewonnenen Informationen angepasst und es erfolgt eine weitere Zuweisung der Pixel. Dieser Schritt wird solange wiederholt, bis eine Segmentierung erreicht wird, die den Ansprüchen des Nutzers genügt (s. rechts unten).

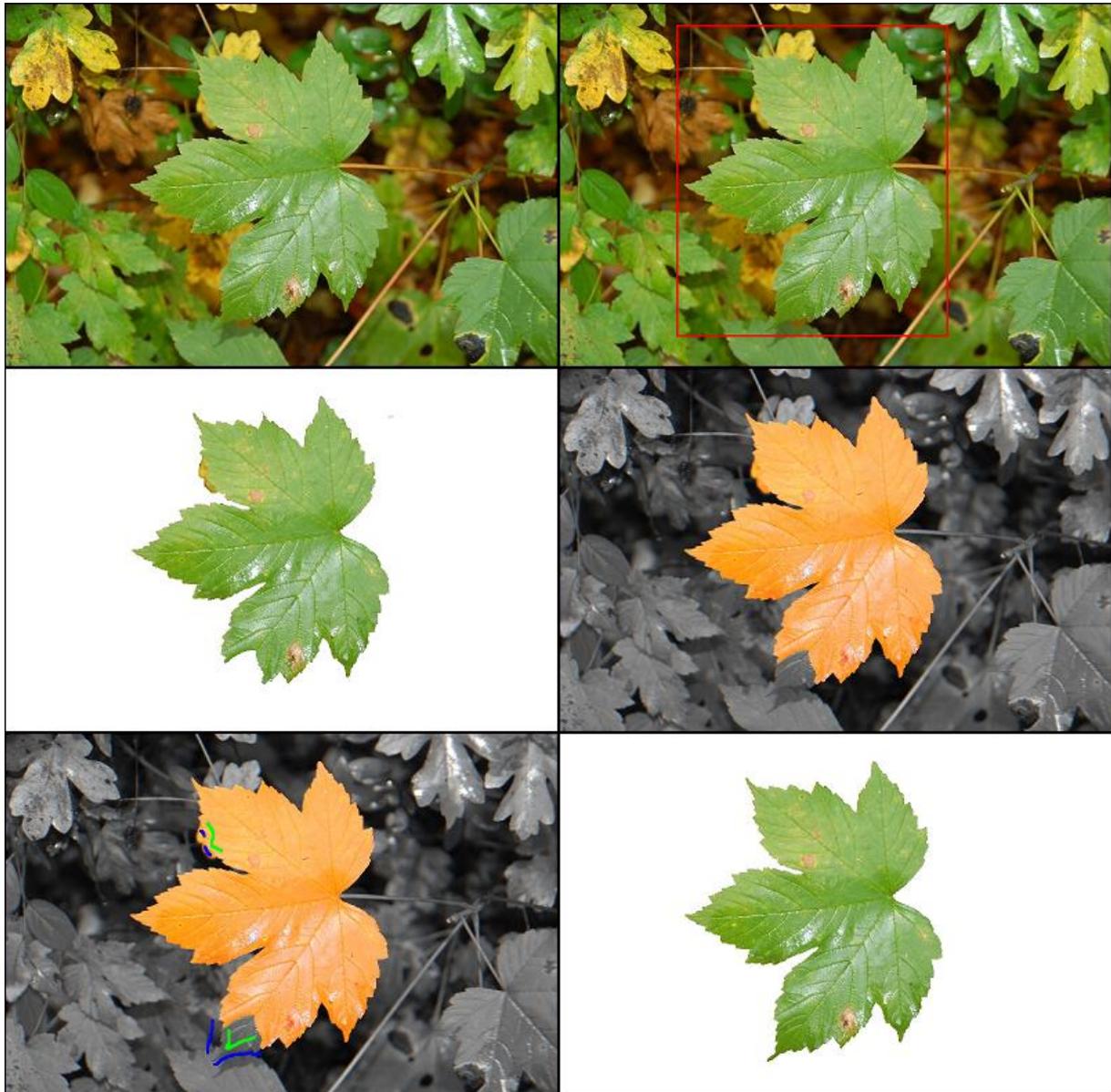


Abbildung 49: Manuelle Segmentierung mittels GrabCut. Oben: Originalbild (links), initialisierendes Rechteck (rechts), Mitte: Zwischenergebnis (links), Überlagerung mit Originalbild (rechts), unten: manuell vorgenommene Korrekturen (links), Endergebnis (rechts).

4.2.4 Diskussion der praktischen Umsetzung

Der in Abbildung 49 vorgestellte, beispielhafte Verlauf einer manuellen Segmentierung mit dem GrabCut-Algorithmus deutet bereits an, was das System zu leisten imstande ist. Ein solcher Verlauf ist bei weitem kein Einzelfall. Häufig lassen sich Blätter nur mit Hilfe des initialisierenden Rechtecks fast einwandfrei segmentieren. Gelingt dies nicht, reicht im Normalfall eine weitere Interaktion mit dem Nutzer, um das gewünschte Ergebnis zu erreichen.

Es gibt einige Faktoren, die großen Einfluss auf den Schwierigkeitsgrad der Segmentierung haben. Diese sollen im Folgenden mit Hilfe der in Abbildung 50 zu sehenden Beispiele erläutert werden.

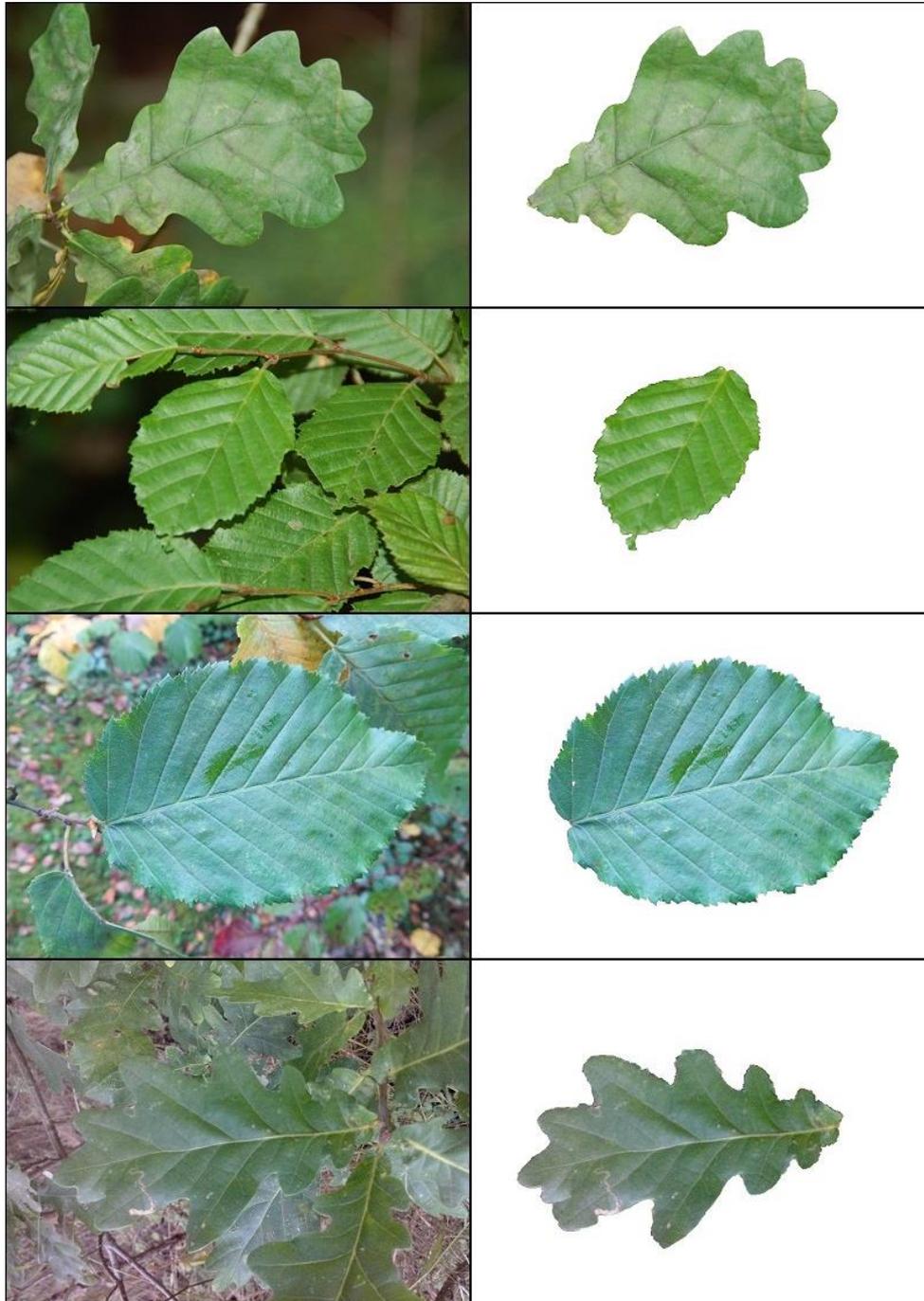


Abbildung 50: *Beispielsegmentierungen mit dem GrabCut-Algorithmus.
Jeweils links: Original, rechts: Ergebnis der Segmentierung.*

Ein wichtiger Faktor ist die Qualität der verwendeten Kamera. Die Bilder in den oberen beiden Reihen wurden mit einer hochwertigen Spiegelreflexkamera aufgenommen. Es ist deutlich zu erkennen, dass das jeweilige Blatt sich klar im Fokus befindet und der Hintergrund bereits automatisch geblurt wurde. Dies hilft bei der Segmentierung ungemein, da sich in den meisten Fällen das Blatt besser vom Hintergrund abhebt und es mehr einheitliche Flächen gibt, bei denen der Algorithmus weniger fehlerhaft agiert.

Selbst bei sehr komplexem Hintergrund (s. zweite Reihe) kann das gewünschte Blatt leicht segmentiert werden.

Die Bilder in den unteren beiden Reihen wurden mit Kameras handelsüblicher Mobiltelefone aufgenommen. Es fällt sofort auf, dass der Hintergrund deutlich weniger verwaschen ist und somit mehr Details zu sehen sind, die die Segmentierung erschweren.

Ein weiterer Faktor ist, wie bereits erwähnt, die Beschaffenheit und Komplexität des Hintergrundes. Bei den Blättern in den Reihen 2 bis 4 erhöht der Hintergrund den Schwierigkeitsgrad der Segmentierung jeweils deutlich. Dort befinden sich zum Teil eine Vielzahl gleichfarbiger Blätter (s. Reihe 2 und 4) und/oder ein wirrer Hintergrund, dessen Farbe ständig wechselt (s. Reihe 3 und 4).

Offensichtlich spielt auch die Beschaffenheit des zu segmentierenden Blattes eine tragende Rolle. Ein rundes, völlig glattes Blatt lässt sich deutlich einfacher vom Hintergrund trennen als ein Blatt mit einem Rand aus vielen kleinen Zacken, zwischen denen jeweils der Hintergrund zu sehen ist.

Zuletzt ist der Faktor der Auflösung zu erwähnen. Prinzipiell funktioniert der Algorithmus umso besser, je niedriger die Auflösung ist. Die Berechnungen verlaufen deutlich schneller und es gibt im Normalfall weniger Gebiete, in denen sich die Farbe von Pixel zu Pixel stark ändert. Andererseits muss beachtet werden, dass eine geringere Auflösung auch einen Informationsverlust bedeutet. Wird das Bild zu stark komprimiert, können Details verloren gehen, wie z. B. ob die äußere Kontur eines Blattes glatt oder leicht gesägt ist. Eine Auflösung von 800x600 Pixeln hat sich als guter Kompromiss erwiesen.

Abschließend sei erwähnt, dass sich die Blätter in den Reihen 1 bis 3 mit sehr geringem Aufwand segmentieren ließen. Nach dem Aufziehen des initialisierenden Rechtecks entstand in allen drei Fällen eine Segmentierung, die lediglich noch minimal korrigiert werden musste. Dazu war jeweils nur eine weitere Iteration nötig. Dies zeigt, dass sich selbst sehr anspruchsvolle Segmentierungsprobleme in der Mehrheit der Fälle ohne Probleme lösen lassen. Lediglich bei dem Blatt in Reihe 4 war ein größerer Aufwand mit mehreren Korrekturen von Seiten des Nutzers nötig. Auch das Ergebnis ist nicht optimal, da an einigen Stellen die Ränder des Blattes nicht sauber vom Hintergrund getrennt wurden. Derartige Fälle traten während dieser Arbeit allerdings nur sehr selten auf, da dafür nicht nur ein komplexer Hintergrund vorhanden sein muss, sondern sich auch das Blatt nicht im Fokus der Kamera befinden darf.

4.3 Entwickeltes System

Um die Nachteile der bisher vorgestellten, automatischen Segmentierungs-Methoden weitestgehend zu vermeiden, wird aufbauend auf den vorausgegangenen Ansätzen ein eigenes System vorgestellt, das sich speziell mit der vorliegenden Problematik beschäftigt. Im Besonderen wird sich damit auseinandergesetzt, wie der Schatten möglichst exakt identifiziert und entfernt werden kann.

4.3.1 Grundidee

Mit Hilfe des bereits vorgestellten GrabCut-Algorithmus lassen sich sehr gute Ergebnisse erzielen, jedoch nur mit manueller Unterstützung. Bei Bildern mit simplem Hintergrund, wie z. B. einem Blatt Papier, reicht oft schon die Initialisierung durch das Aufziehen des Rechtecks aus, um sehr ansprechende Ergebnisse zu erhalten. Wird zusätzlich noch ein Teil des Schattens als Hintergrund markiert, wird fast immer eine Segmentierung erreicht, die auch hohen Ansprüchen genügt. Dies soll sich im Folgenden zunutze gemacht werden. Die Grundidee ist, automatisch eine möglichst exakte Eingabemaske für den GrabCut-Algorithmus erstellen zu lassen. Die Maske muss so viele Pixel wie möglich als sicheren Vordergrund oder Hintergrund klassifizieren und bei den übrigen Pixeln zumindest Tendenzen, ob Vordergrund oder Hintergrund, angeben.

4.3.2 Sichere Vordergrund-/Hintergrundpixel

Es werden zunächst die verschiedenen Farbkanäle der RGB-, HSV- und Lab-Darstellung betrachtet, die in Abbildung 51 zu sehen sind. Das rote Rechteck markiert die besonders interessanten Farbkanäle, nämlich zum einen den a-Kanal (oben) und zum anderen den Sättigungskanal (unten). Diese werden dazu verwendet, die Pixel zu finden, die sicher zum Vordergrund oder zum Hintergrund gehören.

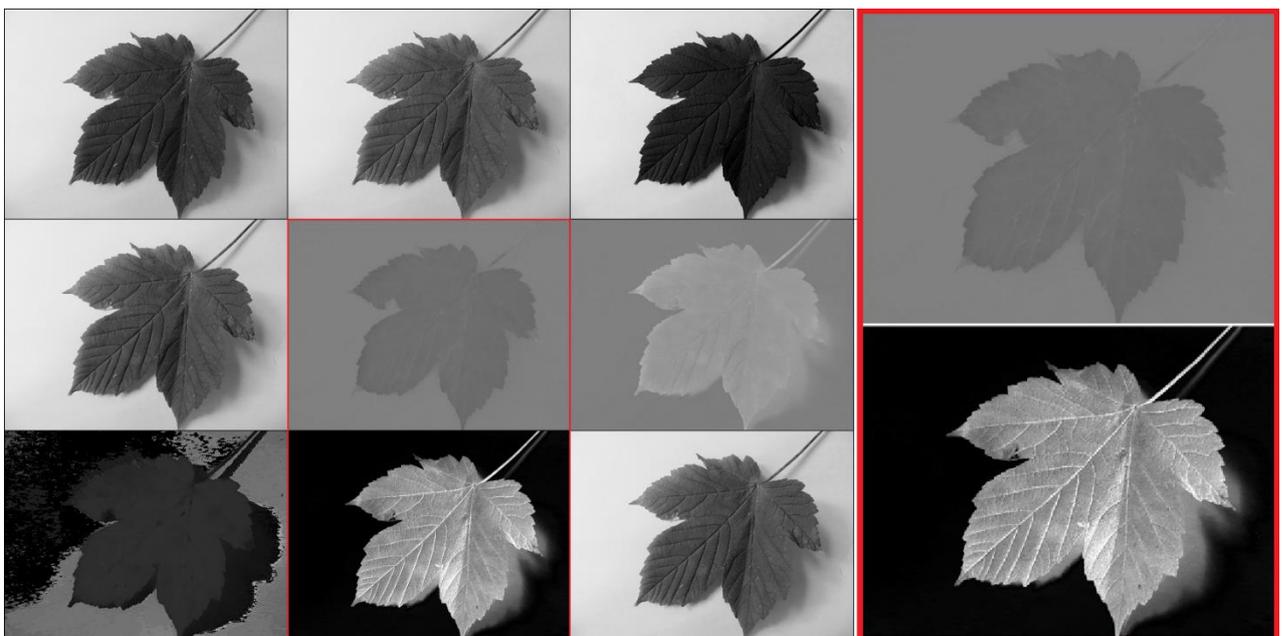


Abbildung 51: Farbkanäle eines Blattes. Jeweils von links nach rechts: Oben: Rot-, Grün- und Blaukanal. Mitte: Lumineszenz-, a und b-Kanal. Unten: Hue-, Sättigungs- und Farbwertkanal.

Beim a-Kanal fällt auf, dass er, anders als alle anderen Kanäle, den Schatten kaum darstellt. Des Weiteren ist in ihm der Großteil des Blattes klar dunkler als das Papier im Hintergrund. Der Nachteil ist, dass sich einige braune Stellen an den Spitzen des Blattes im a-Kanal fast nicht mehr vom Hintergrund unterscheiden lassen. Auffällig am Sättigungs-Kanal ist, dass in ihm der helle Hintergrund sehr zuverlässig in tiefstem Schwarz erscheint. Allerdings lässt sich das Blatt nur schwer vom Schatten unterscheiden.

Mit Hilfe dieser Informationen kann bereits eine klare Vereinfachung des Segmentierungsproblems erfolgen. Wird auf beide Kanäle Otsus Methode angewendet, entstehen Binärbilder, wie sie in Abbildung 52 zu sehen sind.

Das Ergebnisbild des a-Kanals stellt den sicheren Vordergrund in Schwarz dar, das des Sättigungskanals den sicheren Hintergrund in Weiß.



Abbildung 52: Extrahierung des sicheren Vordergrunds und Hintergrunds durch Anwendung von Otsus Methode auf den a-Kanal und den Sättigungskanal. Links: a-Kanal, sicherer Vordergrund (schwarz), Mitte: Sättigungskanal, sicherer Hintergrund (weiß), rechts: noch nicht sicher zugeordnete Pixel (weiß).

Pixel, die weder im linken Bild schwarz noch im mittleren Bild weiß sind, konnten noch nicht endgültig zugeordnet werden. Dabei handelt es sich zum überwiegenden Teil um Schattenpixel. Jedoch können auch Blattpixel oft nicht auf Anhieb als sicherer Vordergrund klassifiziert werden. Beim hier gezeigten Beispielblatt ergibt sich eine sehr brauchbare Segmentierung, wenn lediglich der sichere Vordergrund als Blatt klassifiziert wird und der Rest als Hintergrund. Dies ist allerdings häufig nicht der Fall. Im Folgenden besteht die Aufgabe nun darin, die noch unbekannt Pixel so sicher wie möglich zu klassifizieren.

4.3.3 Zuordnung der noch unbekannt Pixel

Die Klassifikation der bislang noch nicht zugeordneten Pixel gestaltet sich als sehr schwierig. Bestimmte Beobachtungen helfen allerdings, der Eingabemaske für den GrabCut-Algorithmus zumindest einige grobe Anhaltspunkte zu verschaffen, welche Pixel wahrscheinlich zum Vordergrund und welche wahrscheinlich zum Hintergrund gehören.

4.3.3.1 Intensität der Pixel im Sättigungskanal

Zum einen lässt sich feststellen, dass die bislang als unbekannt klassifizierten Blattpixel im Sättigungskanal zum größten Teil merklich heller sind als die Schattenpixel. Um diese zu trennen, wird zunächst der Mittelwert im Sättigungskanal gebildet. Dabei werden lediglich die Werte der Pixel verwendet, die bisher noch keiner Klasse zugeordnet wurden. Dies liefert einen aussagekräftigeren Mittelwert, da die Anzahl sicherer Vordergrund- und Hintergrundpixel, die stark von der Größe des Blattes im Verhältnis

zum Hintergrund abhängt, keine Rolle bei der Berechnung spielt. Anschließend wird eine einfache Grenzwertoperation durchgeführt. Alle Pixel, deren Sättigungswert über dem eben berechneten Mittelwert liegen, werden als wahrscheinlicher Vordergrund klassifiziert, alle anderen als wahrscheinlicher Hintergrund.

4.3.3.2 Kantendetektion im a-Kanal

Zum anderen fällt auf, dass es in den noch unbekanntem Bereichen im a-Kanal einen weiteren Unterschied zwischen den Blatt- und den Schattenbereichen gibt: Es treten im Blattbereich häufig Kanten auf, während der Schatten meist eine durchgehende Fläche ohne Kanten ist. Dies wird sich wie folgt zunutze gemacht: Zunächst wird der a-Kanal geblurt. Dabei handelt es sich um ein Standardverfahren, das in der Bildverarbeitung vor fast jeder Kantendetektion durchgeführt wird. Ziel ist es, im Anschluss nur aussagekräftige Kanten zu detektieren und nicht jeden Bereich, in dem sich nur zwei bis drei Pixel minimal unterscheiden, als Kante zu markieren.

Anschließend wird die Canny-Kantendetektion ausgeführt. Dies ist im hier vorgestellten Segmentierungsalgorithmus der einzige Punkt, an dem Grenzwerte vorgegeben und nicht aus dem Bild selbst bestimmt werden. Als Werte für den oberen und unteren Grenzwert wurden 7 und 10 gewählt. Da die Intensitätsunterschiede teilweise sehr gering sind, müssen die Werte entsprechend klein festgelegt werden. Mit 7 und 10 wurden gute Ergebnisse erzielt, allerdings sind keine nennenswerten Abweichungen zu erkennen, solange sich die Werte in einem sehr niedrigen Bereich bewegen. Abbildung 53 zeigt das Ergebnis der Kantendetektion. Die roten Kreise markieren Stellen, an denen zuvor unbekannte Pixel nun als wahrscheinlicher Vordergrund klassifiziert werden können, da sie Kantenpixel sind.

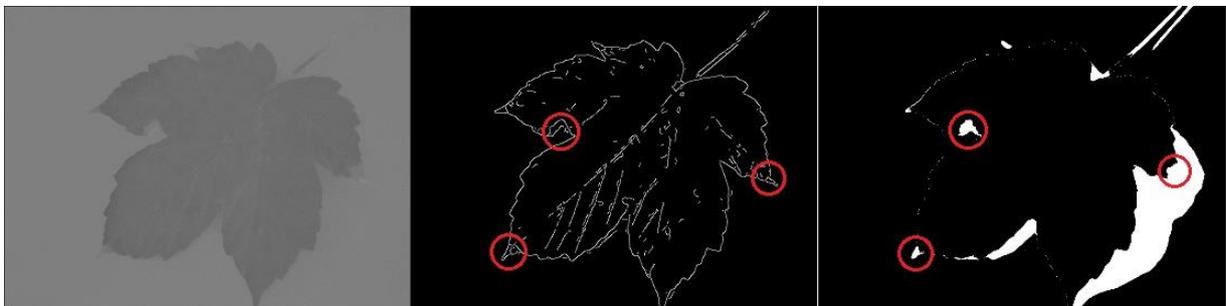


Abbildung 53: Bestimmung wahrscheinlicher Hintergrundpixel durch Kantendetektion.
Links: a-Kanal, Mitte: Ergebnis der Kantendetektion,
rechts: zuvor bestimmte unbekannte Pixel (weiß).

Es ist anzumerken, dass durch die Kantendetektion in den meisten Fällen nur eine geringe Anzahl von Pixeln näher klassifiziert werden kann. Jedoch ist es möglich, dass sich selbst eine kleine Menge an zusätzlicher Information stark auf das Segmentierungsergebnis auswirkt. Des Weiteren ist festzuhalten, dass lediglich Kantenpixel als wahrscheinlicher Vordergrund klassifiziert werden. Dies bedeutet aber keinesfalls, dass gleichzeitig Nicht-Kantenpixel als wahrscheinlicher Hintergrund festgelegt werden. Offensichtlich ist das nicht der Fall, da auch im Blattbereich größere Flächen ohne signifikante Intensitätsunterschiede auftreten.

4.3.4 Erstellung der finalen GrabCut-Eingabemaske

Die bislang gesammelten Informationen müssen nun zusammengefasst werden, damit die Eingabemaske für den GrabCut-Algorithmus erstellt werden kann. Eine Übersicht des gesamten Ablaufs ist in Abbildung 54 zu sehen.

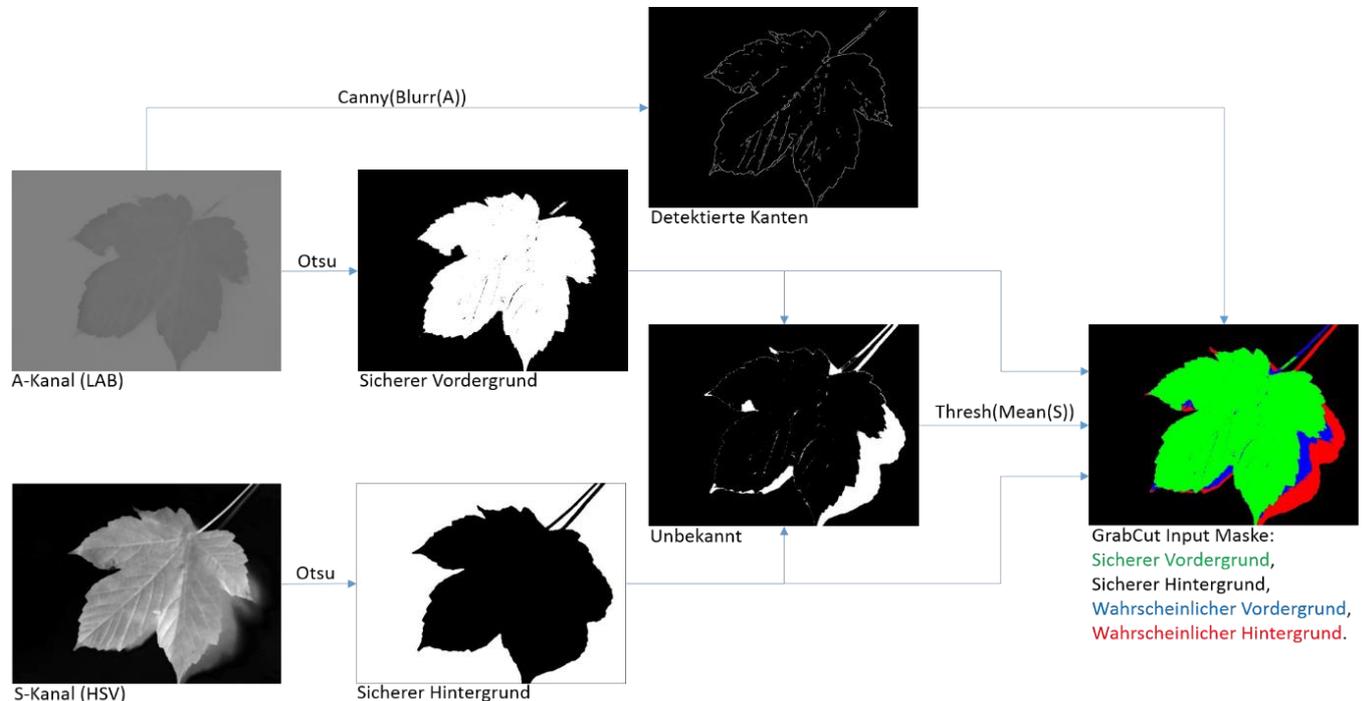


Abbildung 54: Übersicht über die Erstellung der GrabCut-Eingabemaske.

Bei der Erstellung der Eingabemaske ist zu beachten, dass der in OpenCV implementierte GrabCut-Algorithmus für jeden Pixel eine Klassifikation benötigt. Es ist dabei nicht möglich, einen Pixel als „unbekannt“ zu beschreiben. Jeder Pixel muss einen der vier Werte für sicheren Vordergrund, sicheren Hintergrund, wahrscheinlichen Vordergrund oder wahrscheinlichen Hintergrund zugewiesen bekommen.

Die Eingabemaske wird wie folgt kreiert:

1. Initialisierung aller Pixel als wahrscheinlicher Hintergrund.
2. Zuweisung der entsprechenden Werte für sicheren Vordergrund und sicheren Hintergrund.
3. Unbekannte Pixel werden als wahrscheinlicher Vordergrund klassifiziert, wenn ihre Zugehörigkeit bisher unbekannt ist und sie entweder ein Kantenpixel im a-Kanal sind oder oberhalb des Mittelwertes im Sättigungskanal liegen.
4. Alle nicht geänderten Pixel bleiben wahrscheinlicher Hintergrund.

Ist die Eingabemaske fertig initialisiert, kann der GrabCut-Algorithmus mit ihr gestartet werden.

Abbildung 55 zeigt ein Beispielergebnis der hier vorgestellten Methode.

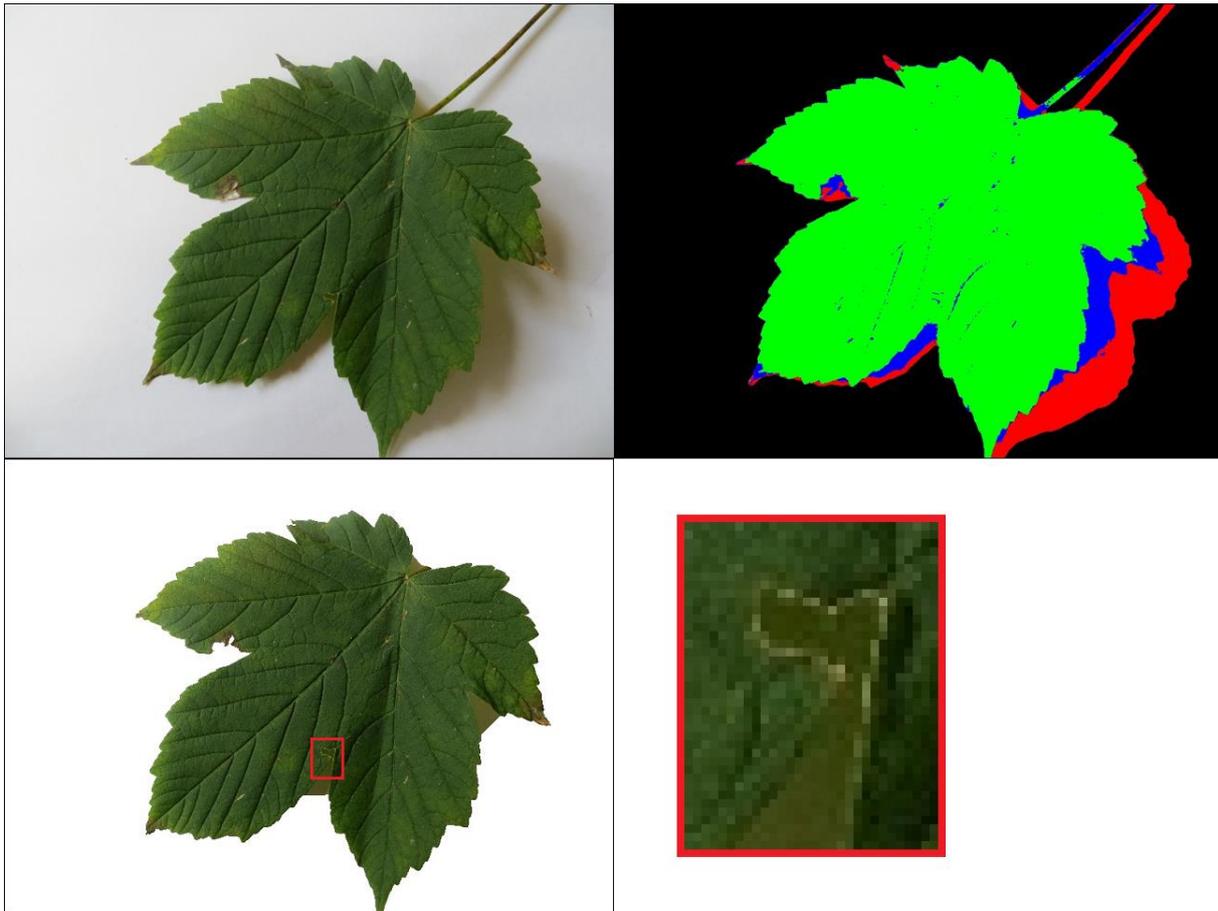


Abbildung 55: Eigene Segmentierung am Beispiel eines Blattes. Oben: Original (links), automatisch erstellte Eingabemaske für den GrabCut-Algorithmus (rechts); unten: Ergebnis der Segmentierung.

Es ist zu sehen, dass zwar auch hier Teile des Schattens fälschlicherweise als Vordergrund klassifiziert wurden. Allerdings ist der Fehler wesentlich kleiner als bei der Otsu- oder K-Means-Segmentierung. Der Watershed-Algorithmus liefert bei diesem Beispielblatt zwar eine ähnlich gute Segmentierung, ist aber im Schnitt der vorgestellten Methode unterlegen. Dies wird in Abschnitt 4.7.2 ausführlich evaluiert.

Des Weiteren ist zu erwähnen, dass der Fehler im hier gewählten Beispielbild um fast 50% höher liegt als der durchschnittliche Fehler aller Bilder.

Ein Problem bei der hier vorgestellten Methode ist, dass Pixel nicht als „unbekannt“ klassifiziert werden können. Es muss daher teilweise mit nur sehr wenig Information entschieden werden, ob ein Pixel eher zum wahrscheinlichen Vordergrund oder zum wahrscheinlichen Hintergrund gehört. In der abgebildeten Eingabemaske ist dieses Problem deutlich zu erkennen. Während der sichere Vordergrund und der sichere Hintergrund eindeutig richtig zugewiesen werden können, ist ein nicht unerheblicher Teil des Schattens als wahrscheinlicher Vordergrund erkannt worden. Dies bedeutet keinesfalls, dass derartige Bereiche, wie in obigem Beispiel, größtenteils auch letztendlich als Vordergrund klassifiziert werden. Es heißt lediglich, dass der GrabCut-Algorithmus diese Pixel zunächst als wahrscheinlichen Vordergrund betrachtet und die weiteren Berechnungen dementsprechend durchführt.

In Abbildung 55 (unten) wird außerdem nochmals der hohe Schwierigkeitsgrad der hier vorliegenden Segmentierungsaufgabe deutlich. Wie im rot umrandeten Bereich zu sehen ist, ähneln sich einige Blatt- und Schattenbereiche nicht nur in ihrer Intensität, sondern es ist auch eine deutliche Grünfärbung des Schattens zu erkennen.

4.4 Vorläufiger, qualitativer Vergleich der Segmentierungsergebnisse

In Abbildung 56 sind die Resultate der verschiedenen Methoden noch einmal zusammengefasst. Otsus Methode und der K-Means-Ansatz liefern ein ähnlich schlechtes Ergebnis. Beide klassifizieren fast den kompletten Schatten als Vordergrund. Am wenigsten Schatten bleibt beim Watershed-Algorithmus zurück, allerdings entfernt dieser auch fälschlicherweise einige Bereiche an den Blattspitzen. Der modifizierte GrabCut-Algorithmus klassifiziert lediglich einige Schattenpixel an den Stellen, an denen die Blattlappen zusammenlaufen, als Vordergrund und fast keine Blattpixel als Hintergrund.

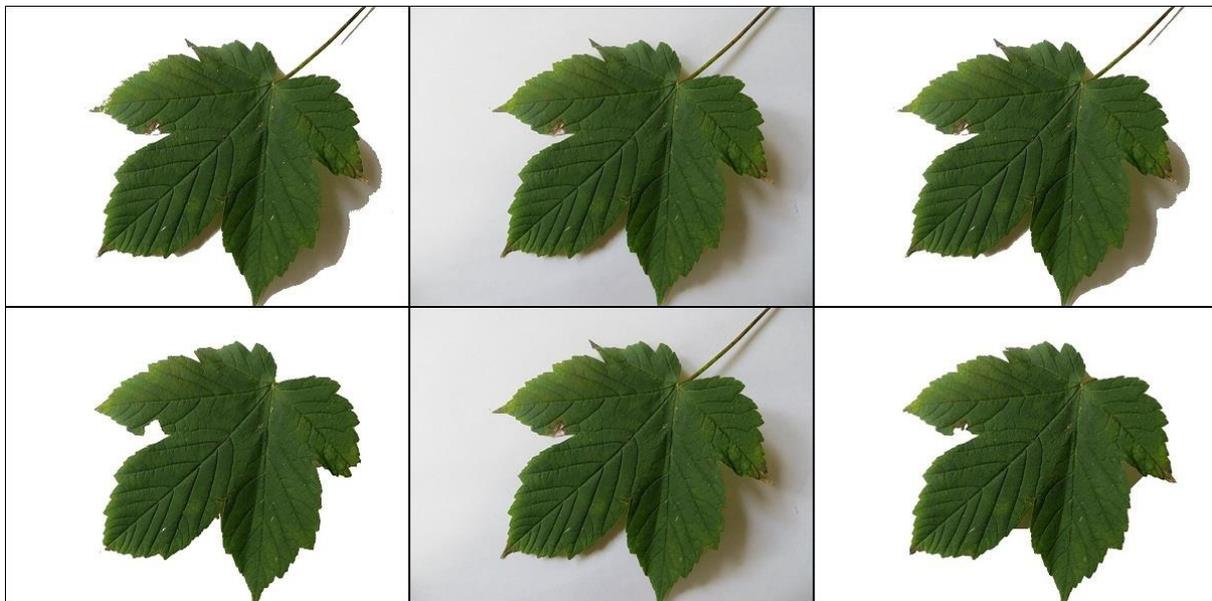


Abbildung 56: Vergleich der Segmentierungsergebnisse. Oben: Otsus Methode (links), Original (Mitte), K-Means (rechts); unten: Watershed (links), Original (Mitte), modifizierter GrabCut (rechts).

4.5 Nachbearbeitungsschritt: Füllen der Kontur

Blätter enthalten nicht selten eine Vielzahl von Verunreinigungen oder braunen Stellen, die bei der Segmentierung manchmal als Hintergrund klassifiziert werden und zu weißen Aussparungen innerhalb des Blattes führen. Diese können bei der späteren Feature-Extraktion, speziell bei der Berechnung der Farb-Features, zu starken Verzerrungen führen. Deshalb erfolgt bei allen Methoden nach der Segmentierung ein Nachbearbeitungsschritt, in dem die entstandenen Löcher geschlossen werden.

Dazu wird zunächst die Kontur des Blattes bzw. des Vordergrundes detektiert. Anschließend wird allen Pixeln innerhalb der Kontur der Wert zugewiesen, den der

jeweilige Pixel im Originalbild hatte. Alle anderen Pixel erhalten den Wert RGB(255, 255, 255).

Abbildung 57 zeigt das Ergebnis des Nachbearbeitungsschrittes.

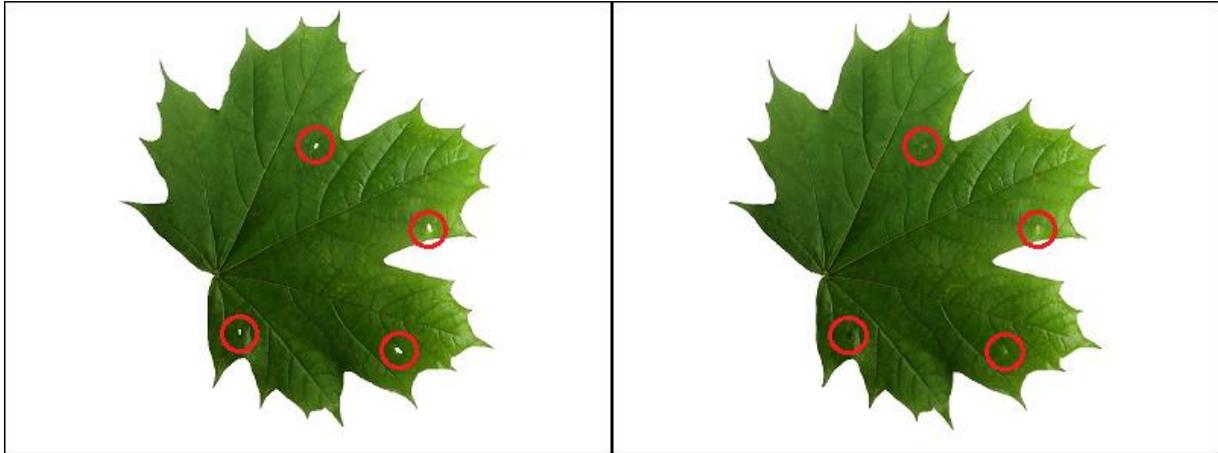


Abbildung 57: *Das Schließen von Löchern am Beispiel eines Blattes.
Links: Ergebnis der Segmentierung, rechts: Ergebnis der Nachbearbeitung.*

Dieser Schritt führt auch dazu, dass beispielsweise Stielüberbleibsel, die von der Kontur abgetrennt wurden, verschwinden.

4.6 Detektion und Entfernen des Stiels

Wie bereits in Kapitel 3 erwähnt, wurden bei einigen Datensets vor dem Scannen bzw. dem Fotografieren der Blätter die Stiele nicht entfernt. Da deren manuelle Beseitigung wiederum einen enormen menschlichen Aufwand darstellen würde, wurde ein Algorithmus entwickelt, der die Stiele automatisch entfernt. Dieser soll zunächst erläutert und anschließend kurz evaluiert werden.

4.6.1 Algorithmus

Zunächst ist zu klären, welche Eigenschaften Stiele im Allgemeinen ausmachen. Zum einen schließen sie stets an die äußere Kontur des Blattes an. Zum anderen ist ihre Form länglich und in den meisten Fällen sind sie dünner als andere Teile des Blattes. Unter Beachtung dieser Eigenschaften lässt sich ein Algorithmus entwickeln, der die Stiele mit hoher Genauigkeit identifiziert und entfernt.

Eine wichtige Rolle bei der Stieldetektion spielt die in 2.1.5.3 beschriebene Black-Hat-Transformation. Bei der Wahl des Strukturelements ist zu beachten, dass dessen Durchmesser größer ist als die Breite jedes zu erwartenden Stiels (vgl. [Kumar 2012]). Ein Wert von 51 erwies sich bei Bildern mit einer Auflösung von 800x600 Pixeln als optimal. Abbildung 58 zeigt das Ergebnis der Top-Hat-Transformation. Alle Teile des Blattes, die dünner sind als der gegebene Durchmesser des Strukturelements, werden weiß eingefärbt. Der Rest des Blattes und der Hintergrund sind schwarz.

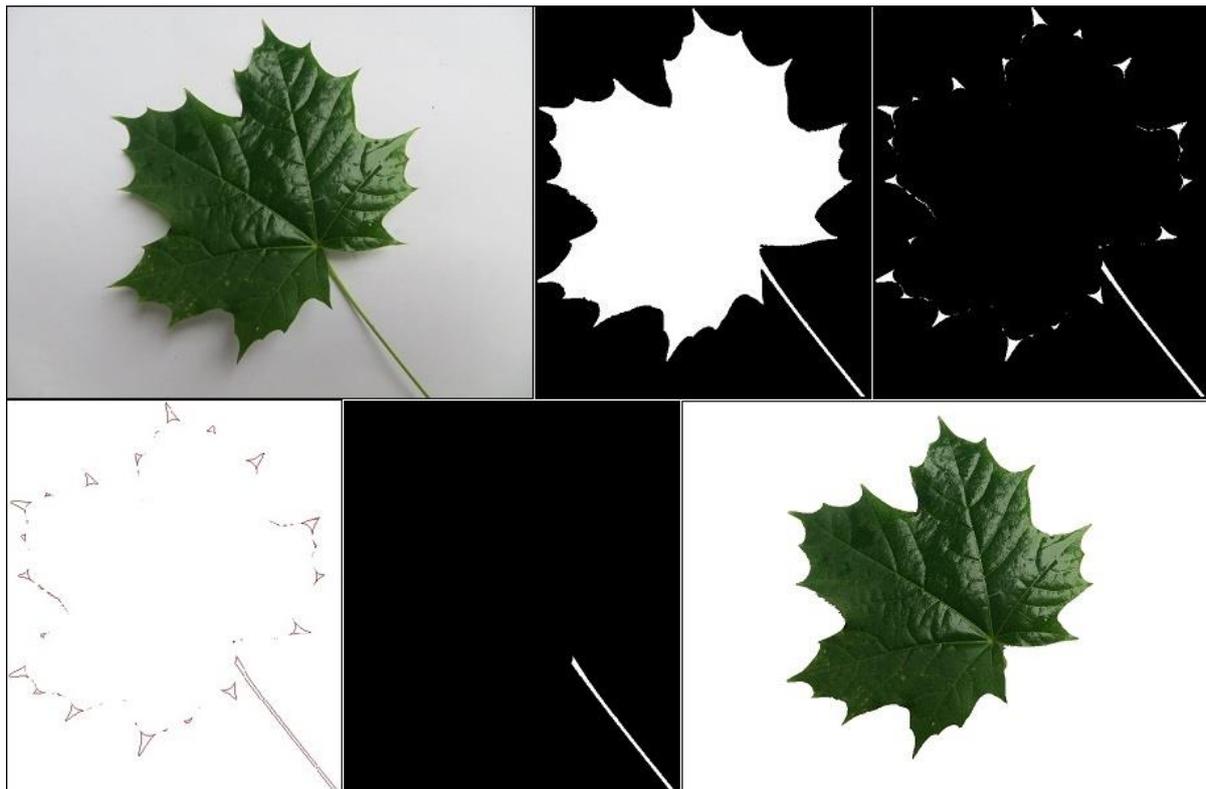


Abbildung 58: *Top-Hat-Transformation am Beispiel eines Blattes. Oben: Original (links), Binärdarstellung der vorläufigen Segmentierung (Mitte), Ergebnis der Top-Hat-Transformation (rechts); unten: Konturen der Stielkandidaten (links), detektierter Stiel (Mitte), Ergebnis der Segmentierung (rechts).*

Jedes weiße Element im Ergebnis der Top-Hat-Transformation wird als potentieller Stiel betrachtet. Zunächst wird die Kandidatenliste ausgedünnt, indem die Elemente aussortiert werden, die keine längliche Form besitzen. Dazu wird jeweils die äußere Kontur berechnet und das *MinimumBoundingRect* bestimmt. Anschließend werden die Konturen verworfen, bei denen das Verhältnis der längeren zur kürzeren Seite des *MinimumBoundingRects* kleiner ist als ein bestimmter Grenzwert; diesen zu optimieren ist schwierig. Wird er zu klein gewählt, können beispielsweise auch spitz zulaufende Enden des Blattes fälschlicherweise als mögliche Stiele beibehalten werden; wird er dagegen zu groß gewählt, kann es dazu führen, dass kurze Stiele verworfen werden. Ein Wert von 3,5 hat sich als zufriedenstellender Kompromiss erwiesen.

Zuletzt wird von den verbliebenen Kandidaten der ausgewählt, der den größten Umfang besitzt. Im Segmentierungsergebnis werden alle Stielpixel als Hintergrund klassifiziert und mit RGB(255, 255, 255) ersetzt.

4.6.2 Evaluation

Eine numerische Bewertung der Performanz der Stielentfernung gestaltet sich schwierig, da die Grenze zwischen erlaubter bzw. erwünschter Ungenauigkeit und der fehlerhaften Nicht-Entfernung des Stiels meist fließend ist.

Dieses Phänomen soll mit Hilfe einiger typischer Problemfälle in Abbildung 59 erläutert werden.

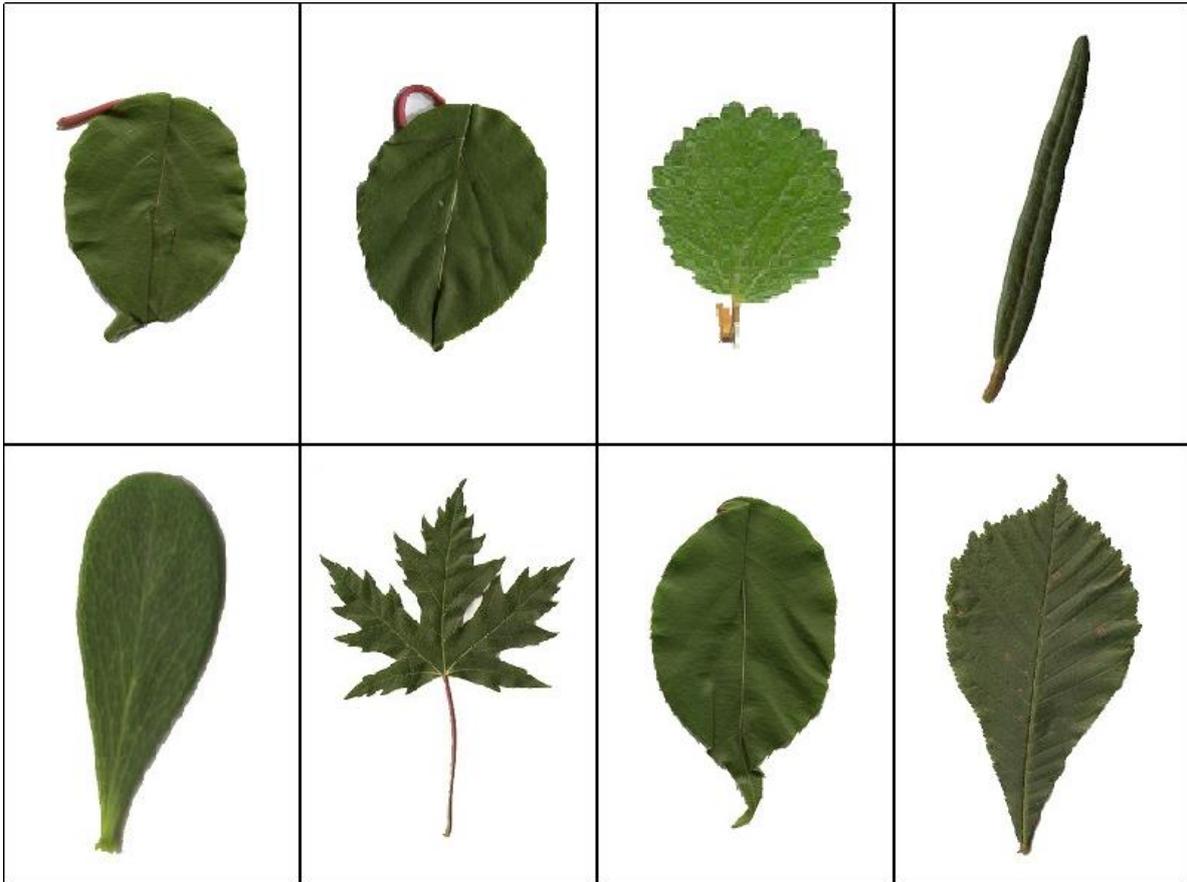


Abbildung 59: Typische Problemfälle bei der Stielentfernung.

In den ersten drei Bildern der oberen Zeile sind Blätter mit Stielen zu erkennen, die aufgrund ihrer Lage oder Form nicht als Stiel detektiert und damit auch nicht entfernt werden konnten. Im ersten Bild liegt der Stiel zu nahe am Blatt an. Der freistehende Teil, der bei der Top-Hat-Transformation hervorgehoben werden soll, ist zu kurz bzw. nicht elongiert genug und wird daher nicht als möglicher Kandidat erkannt. Ein ähnliches Problem tritt im zweiten Bild auf. Im dritten Bild ist der Stiel offenbar durch ein abgetrenntes Stück Rinde zusätzlich verdickt und kommt ebenfalls nicht als Stiel in Frage. Bild 4 zeigt ein stark elongiertes Blatt, dessen Stiel nur unwesentlich dünner ist als das Blatt selbst. In solchen Fällen ist die erfolgreiche Detektion und Entfernung des Stiels nur schwer möglich, da aufgrund seiner geringen Breite nicht selten das gesamte Blatt inklusive Stiel als Stielkandidat erkannt wird. Eine Entfernung ist in diesem Fall natürlich nicht möglich, da es die Anzahl der im Bild zu sehenden äußeren Kontur auf 0 reduzieren würde. In der zweiten Zeile ist im ersten Bild ein Blatt zu sehen, dessen Stiel selbst für einen Menschen schwer zu identifizieren ist, da der Übergang zwischen Blatt und Stiel fließend ist.

Bild 2 der zweiten Zeile zeigt eine schwere Nicht-Detektion eines klar zu erkennenden Stiels. Fehler dieser Art treten in weniger als 0,05% der Fälle, also extrem selten auf und lassen sich nur sehr schwer nachvollziehen. Die Stiele werden, ähnlich wie in Abbildung 58 rechts oben zu sehen, korrekt mit Hilfe der Top-Hat-Transformation markiert, jedoch bei der anschließenden Konturextraktion ignoriert. Eine endgültige Erklärung für dieses Phänomen kann in dieser Arbeit nicht geliefert werden.

Viele der oben angesprochenen Problemfälle ließen sich vermeiden, wenn die verwendeten Parameter dementsprechend angepasst würden. So könnte beispielsweise eine Verminderung des geforderten Längenverhältnisses der Stiele die Fehler der Art beheben, wie sie in der oberen Zeile in Bild 1 und 3 zu sehen sind. Allerdings würde dies wiederum eine andere Art von Fehler generieren. Beispiele dafür sind in der unteren Zeile in den Bildern 3 und 4 zu sehen. Beide Blätter weisen Spitzen auf, die in der Top-Hat-Transformation hervorgehoben werden und somit als mögliche Stielkandidaten gelten. Jedoch erfüllen sie nicht die Anforderung eines Längenverhältnisses von mindestens 3,5 und werden damit abgewiesen. Bei einem geforderten Verhältnis von etwa 2,5 oder weniger würden beide Spitzen als Stiel detektiert und entfernt werden.

Abschließend lässt sich festhalten, dass die Stieldetektion in den allermeisten Fällen sehr gute Ergebnisse liefert. Allerdings gestaltet sich die Parameteroptimierung schwierig. In dieser Arbeit wurde sich dazu entschieden, die Parameter so zu wählen, dass im Zweifelsfall ein tatsächlicher Stiel lieber nicht entfernt wird, als dass ein anderer Teil des Blattes fälschlicherweise als Stiel identifiziert und abgetrennt wird. Der Grund dafür liegt in der späteren Klassifikation der Blätter. Bei den im nächsten Kapitel vorgestellten verwendeten Features wird deutlich werden, dass ein kleiner Stielansatz keinen großen, negativen Einfluss auf eine erfolgreiche Bestimmung der Spezies hat. Dies gilt vor allem für den K-Nearest-Neighbours-Klassifikator, wenn auch in der Vergleichsmenge Instanzen existieren, bei denen der Stiel nicht entfernt wurde. Ein komplett fehlendes charakteristisches Merkmal, wie beispielsweise eine ausgeprägte Spitze, kann dagegen leicht zu einer Fehlklassifikation führen.

4.7 Vergleich der vorgestellten Methoden

Zum objektiven Vergleich der vorgestellten Methoden ist es nötig, ihre Segmentierungsergebnisse unter denselben Umständen ausführlich zu evaluieren. Im Folgenden soll zunächst der Evaluationsmodus erläutert werden. Anschließend werden die Ergebnisse vorgestellt und ausführlich diskutiert.

4.7.1 Evaluationsmodus

Um die Qualität der Segmentierung objektiv und qualitativ bewerten zu können, ist eine hohe Anzahl von Blattfotos nötig, zu denen auch jeweils eine „perfekte“ Segmentierung als Vergleich vorhanden sein muss. Zu diesem Zweck wurde das BLD verwendet. Alle Aufnahmen auf Papier wurden sorgfältig per Hand segmentiert. Dazu wurde der GrabCut-Algorithmus genutzt. Die so entstandenen Bilder werden als Goldstandard verwendet. Es wird also angenommen, dass die abgebildeten Blätter perfekt vom Hintergrund freigestellt wurden. Diese Annahme entspricht selbstverständlich nicht der Realität, da auch bei manueller Segmentierung nicht stets 100% der Pixel korrekt klassifiziert werden können. Allerdings ist es mit Hilfe der manuellen Segmentierung durchaus gelungen, Ergebnisse zu erzielen, die auch höchsten Ansprüchen genügen und bei denen mit bloßem Auge in den allermeisten Fällen keine Fehler zu erkennen sind.

Dies ist in Abbildung 60 zu sehen.

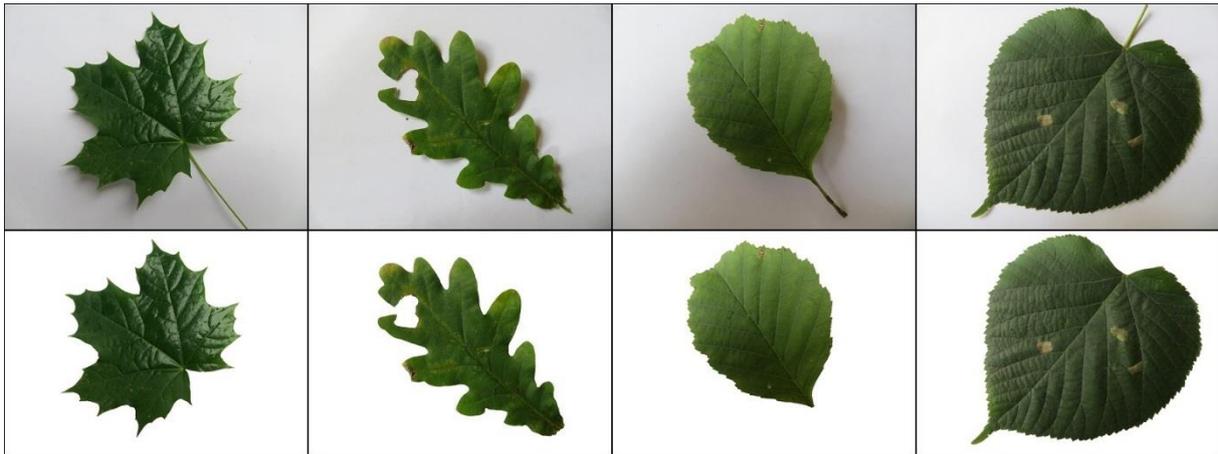


Abbildung 60: *Ergebnisse der manuellen Segmentierung ausgewählter Blätter aus dem BLD. Oben: Original, unten: Ergebnis der Segmentierung.*

Mit Hilfe dieses Goldstandards ist es nun möglich, die verschiedenen Segmentierungsergebnisse qualitativ zu vergleichen. Dazu wird folgende Metrik verwendet:

Für jedes Bild wird jeder Pixel des automatisch segmentierten Bildes mit dem entsprechenden Pixel des manuell segmentierten Bildes verglichen. Dabei wird lediglich zwischen Vordergrund und Hintergrund unterschieden. Weist ein Pixel unterschiedliche Werte auf, so wird dies als Fehler gezählt. Hierbei ist zu erwähnen, dass sich Fehler nicht gegenseitig aufheben können. Ist also ein Vordergrundpixel fälschlicherweise als Hintergrund und ein anderer Pixel fälschlicherweise als Vordergrund klassifiziert, so werden zwei Fehler gezählt.

Sind alle Pixel durchlaufen, wird das Endergebnis berechnet, indem die Anzahl der Fehler durch die Anzahl der manuell bestimmten Vordergrundpixel dividiert wird. Es wird also nicht die Gesamtzahl der Pixel verwendet, da dieses das Ergebnis verfälschen würde. Ein im Verhältnis zum Bild groß aufgenommenes Blatt, also eines, das aus vielen Pixeln besteht, bietet naturgemäß mehr Möglichkeiten, Pixel fehlerhaft zuzuordnen als ein kleines.

4.7.2 Ergebnisse

Tabelle 2 zeigt eine Gegenüberstellung der Ergebnisse der einzelnen Segmentierungsmethoden Otsu, K-Means, Watershed und die eigens implementierte Methode mit Unterstützung des GrabCut-Algorithmus. Zum Vergleich ist außerdem noch die manuelle Segmentierung, also der Goldstandard angegeben.

Die Spalten geben einerseits die Zeit an, die im Durchschnitt benötigt wurde, um ein Bild zu segmentieren. Es ist anzumerken, dass etwa 0,25s auf die Entfernung des Stiels entfallen. Andererseits wird der durchschnittliche Fehler aufgezeigt, der wie im vorherigen Abschnitt beschrieben berechnet wurde. Des Weiteren kann abgelesen werden, wieviel Prozent der Bilder mit einem Fehler von weniger als 6% segmentiert wurden. Diese 6%-Grenze wurde gewählt, da sie in etwa dem 1,5-fachen des durchschnittlichen Fehlers der besten Segmentierung entspricht.

Es wurde folgende Hardware verwendet:

Prozessor: Intel(R) Core(TM) i7-4770 CPU @3.40 GHz

Arbeitsspeicher: 8,00 GB

Betriebssystem: Windows 7 (64 Bit)

Tabelle 2: Ergebnisse der automatischen Segmentierung.

Methode	Ø Zeit	Ø Fehler	Fehler < 6%
Otsu	0,59s	10,37%	33,78%
K-Means	0,97s	8,75%	38,14%
Watershed	0,84s	5,06%	86,69%
 Mod. GrabCut 	 1,74s 	 4,10% 	 92,06%
Manuell	ca. 15s	0%	100%

Bei der durchschnittlich benötigten Zeit wird der Nachteil der eigenen Implementierung deutlich. Die im GrabCut-Algorithmus wiederholt auftretenden GraphCuts sind rechenintensiv und führen dazu, dass die Segmentierung mit dieser Methode fast doppelt so viel Zeit in Anspruch nimmt wie mit der K-Means- oder Watershed-Methode. Die simple Otsu-Methode ist sogar fast dreimal schneller.

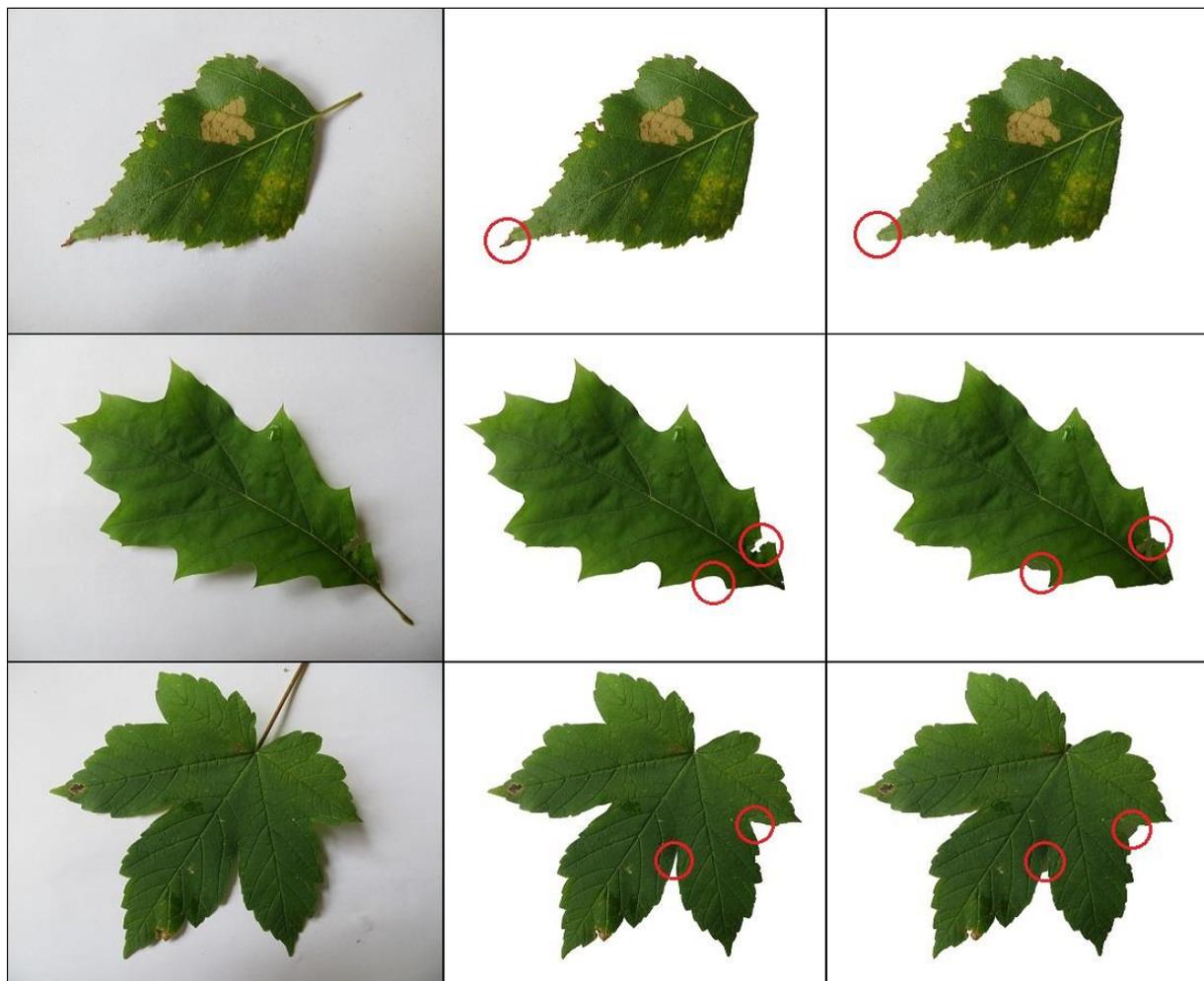


Abbildung 61: Segmentierte Blätter, bei denen der Segmentierungsfehler etwa dem durchschnittlichen Fehler von 4,5% entspricht. Links: Original, Mitte: Ergebnis der manuellen Segmentierung, rechts: Ergebnis der automatischen Segmentierung.

Dieser Mehraufwand macht sich allerdings bezahlt, wie beim durchschnittlichen Fehler zu erkennen ist. Dieser liegt mit 4,10% deutlich niedriger als der der Vergleichsmethoden.

Lediglich der Watershed-Algorithmus erzielt mit 5,06% noch ein annähernd so gutes Ergebnis. Auch beim Anteil der Segmentierungen mit einem Fehler von weniger als 6% liegt die eigene Implementierung mit 92,06% klar vorne. Der hohe Wert zeigt die Zuverlässigkeit der Methode. In den allermeisten Fällen treten nur sehr geringe Fehler bei der Segmentierung auf.

Um den durchschnittlichen Fehler besser einschätzen zu können, sind in Abbildung 61 Beispiele von segmentierten Blättern aufgeführt, deren Segmentierungsfehler in etwa 4,5% beträgt und somit etwas höher liegt als der durchschnittliche Fehler.

Es fällt sofort auf, dass der Unterschied zwischen den Ergebnissen der automatischen und der manuellen Segmentierung nur minimal ist. Typische Fehler sind kleine, braune Stellen an Blattspitzen (s. rote Kreise in der oberen Reihe) oder nicht entfernte Schatten, gerade an Stellen, an denen Teile des Blattes zusammenlaufen (s. rote Kreise in der mittleren und unteren Reihe). Wie durch die Klassifikationsergebnisse in Tabelle 2 belegt wird, lassen diese minimalen Fehler in den allermeisten Fällen trotzdem noch eine korrekte Klassifikation der vorliegenden Spezies zu.

5 Verwendete Features

Im Verlauf dieses Kapitels sollen die getesteten Features vorgestellt werden. Es ist zu erwähnen, dass in der Literatur noch deutlich mehr Feature-Klassen Verwendung finden. Eine ausführliche Vorstellung jeder dieser Klassen ist allerdings weder sinnvoll noch im Umfang dieser Arbeit möglich. Es wurde daher versucht, die wichtigsten Klassen herauszufiltern.

Ein weiteres Auswahlkriterium war es Feature-Klassen zu wählen, die sich möglichst gut ergänzen. Beispielsweise erscheint es wenig sinnvoll, auf eine Vielzahl von Herangehensweisen zurückzugreifen, die alle darauf abzielen, die feine Beschaffenheit der äußeren Kontur zu bestimmen, d. h. festzustellen ob der Blattrand glatt oder gezackt/gesägt ist.

Die nun folgenden Feature-Klassen sind aufsteigend nach Komplexität sortiert.

5.1 Kontur-Features

Die wohl offensichtlichste Feature-Klasse bilden die Kontur-Features. Sie beschreiben relativ grob die äußere Kontur des Blattes und kommen in einem Großteil der aktuellen Systeme zur Blattklassifikation zum Einsatz.

Diese Features beschreiben ausschließlich geometrische Verhältnisse. Einfache Längen- oder Flächenmaße kommen nicht vor, da sie sich je nach der vorherrschenden Entfernung bei Aufnahme des Bildes stark unterscheiden und daher nutzlos sind. Selbst bei durch Scans gewonnenen Bildern wäre der Nutzen dieser einfachen Maße höchst zweifelhaft, da sich z. B. die Fläche des größten ausgewachsenen Blattes von der des kleinsten ausgewachsenen Blattes am selben Baum um ca. Faktor 2 unterscheidet [Suk 2013].

Zur effizienten Bestimmung der geometrischen Verhältnisse werden einfache, an die Blattkontur angepasste Hilfsformen verwendet. Diese sind in Abbildung 62 zu sehen und können mit Hilfe der OpenCV-Library berechnet werden.

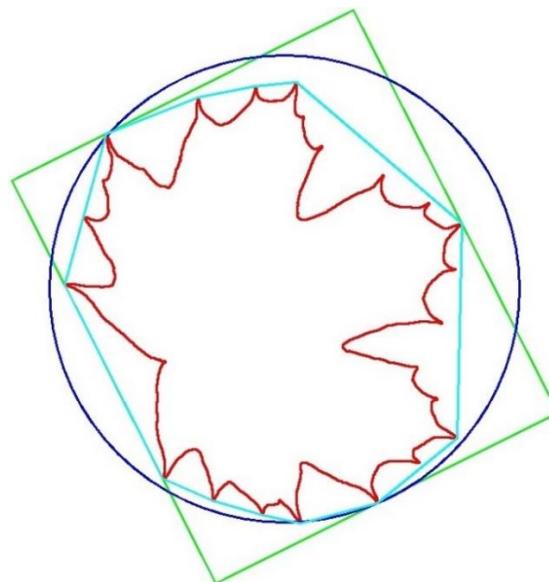


Abbildung 62: Eine Blattkontur (rot) mit zugehörigem `MinimumBoundingRect` (grün), `MinimumEnclosingCircle` (dunkelblau) und Konvexer Hülle (hellblau).

Das *MinimumBoundingRect* ist ein Rechteck minimaler Fläche, das die Kontur komplett einschließt.

Analog ist der *MinimumEnclosingCircle* der Kreis, der mit minimaler Fläche die gesamte Kontur bedeckt.

Die *Konvexe Hülle* einer Punktmenge P ist definiert als das kleinste konvexe Set, das P enthält. Das bedeutet, dass keine konkaven, d. h. zur Punktmenge hin gewölbten, Elemente vorkommen dürfen. Bildlich gesprochen entspricht die *Konvexe Hülle* einer Kontur einem Gummiband, das um die Kontur gespannt wurde. [OpenCV 2008]

Unter Verwendung dieser Hilfsformen werden folgende Features extrahiert:

$$\textit{AspectRatio} = \frac{\textit{Width}}{\textit{Length}} \quad (5.1)$$

Dabei ist die Länge als die längere Seite des *MinimumBoundingRect* definiert. Die *AspectRatio* besitzt also stets einen Wert ≤ 1 .

$$\textit{Rectangularity} = \frac{\textit{ContourArea}}{\textit{RectangleArea}} \quad (5.2)$$

$$\textit{ConvexHullAreaRatio} = \frac{\textit{ContourArea}}{\textit{ConvexHullArea}} \quad (5.3)$$

$$\textit{ConvexHullPerimeterRatio} = \frac{\textit{ContourPerimeter}}{\textit{ConvexHullPerimeter}} \quad (5.4)$$

$$\textit{PerimeterLengthWidthRatio} = \frac{\textit{ContourPerimeter}}{\textit{Length} + \textit{Width}} \quad (5.5)$$

[Wu 2012]

In der Literatur werden häufig noch weitere Verhältnisse als Features genutzt. Diese sind allerdings meist hochredundant und liefern keinerlei oder nur einen vernachlässigbar kleinen Mehrwert für die Klassifikation. In dieser Arbeit findet z. B. der *MinimumEnclosingCircle* keinerlei Verwendung und wurde lediglich der Vollständigkeit halber mit angegeben.

Abbildung 63 zeigt die Auswirkungen der verwendeten Features. Am Beispiel des Flavia-Datensets sind für jedes Feature die beiden Blätter gegenübergestellt, die innerhalb des Datensets den höchsten bzw. niedrigsten Wert aufweisen.

Bei der *AspectRatio* ist dabei anzumerken, dass der Wert natürlich Verzerrungen aufweisen kann, da sich das Länge-zu-Breite-Verhältnis aus dem *MinimumBoundingRect* berechnet, das wiederum in seltenen Fällen auch bei ähnlichen Blättern sehr unterschiedlich an die Kontur angepasst werden kann. Für optimale Ergebnisse würde es sich anbieten, die Länge als den Abstand zwischen der Spitze des Blattes und der Basis, also dem Ansatzpunkt des Stiels, zu definieren. Die Breite ergibt sich dann aus der Länge der längsten dazu senkrecht stehenden Strecke. Das Problem bei diesem Ansatz ist die exakte Bestimmung der Position der Spitze und der Basis. Es existieren bereits Arbeiten, die sich mit diesem Bereich beschäftigten und auch durchaus gute Ergebnisse lieferten. So erreichten Mzoughi et al. (vgl. [Mzoughi 2012]) eine Detektionsrate von 96% für die

Spitze und sogar 98% für die Basis. Allerdings ist das Verfahren recht aufwendig und bei den Fehldetektionen ist der Abstand zur wahren Position nicht selten sehr groß.

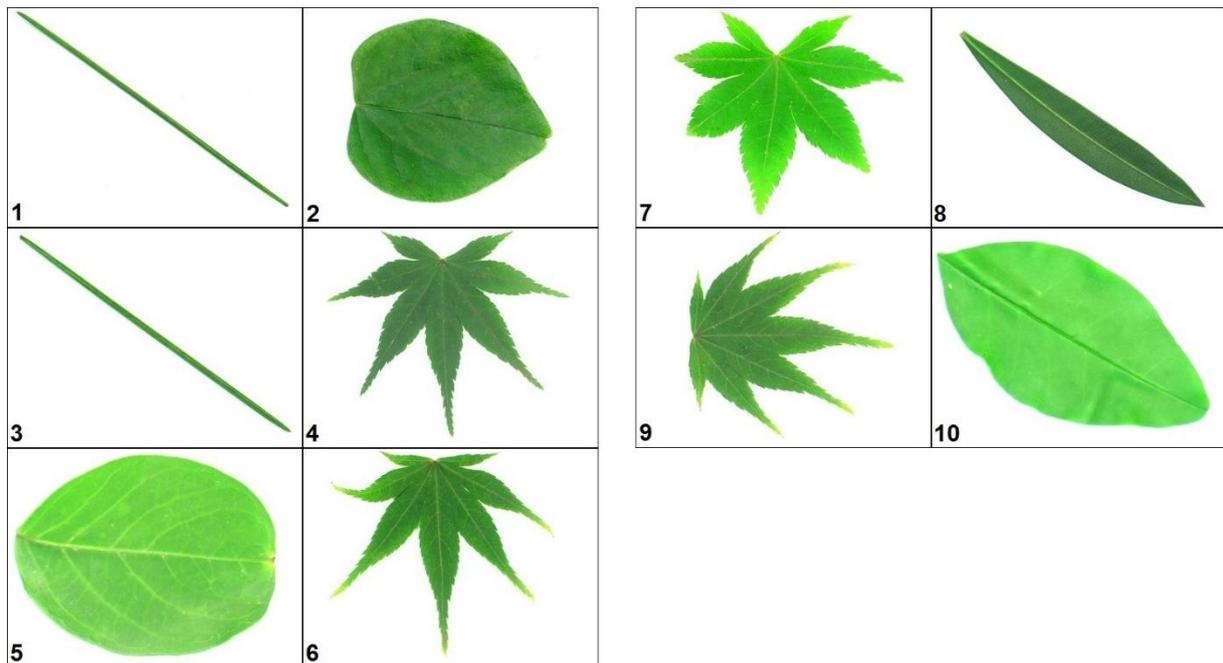


Abbildung 63: Blätter aus dem Flavia-Datenset, die die maximalen und minimalen Werte für AspectRatio (1: max, 2: min), Rectangularity (3: max, 4: min), ConvexHullAreaRatio (5: max, 6: min), ConvexHullPerimeterRatio (7: max, 8: min) und PerimeterLengthWidthRatio (9: max, 10: min) liefern.

Ein weiterer Ansatzpunkt ist die manuelle Kennzeichnung der Spitze und der Basis, wie es beispielsweise in [Wu 2012] Anwendung findet. Während das Markieren dieser Punkte bei einem einzelnen Blatt lediglich wenige Sekunden in Anspruch nimmt, wäre bei der Menge der in dieser Arbeit verwendeten Blätter ein erheblicher menschlicher Aufwand nötig gewesen, der in keinem Verhältnis zur möglichen Verbesserung der Klassifikationsrate steht. Außerdem gestaltet sich die Markierung der Spitze und der Basis nicht immer trivial. Beispiele dazu sind in Abbildung 64 zu sehen. Während bei dem Blatt auf der linken Seite ein Nutzer ohne Botanik-Kenntnisse lediglich raten könnte, wo sich die Spitze des Blattes befindet, gibt bei den Blättern in der Mitte und rechts lediglich die Hauptvene Aufschluss über die Position der Spitze. Gerade auf Fotos, die am Baum aufgenommen werden, ist jedoch die Hauptvene oft nicht klar zu sehen. Ohne diesen Anhaltspunkt ist es sehr schwer nachzuvollziehen, wieso sich bei zwei Blättern derselben Spezies die Spitze einmal genau auf einer lokalen Spitze (Mitte) und einmal genau zwischen zwei lokalen Spitzen (rechts) befindet.

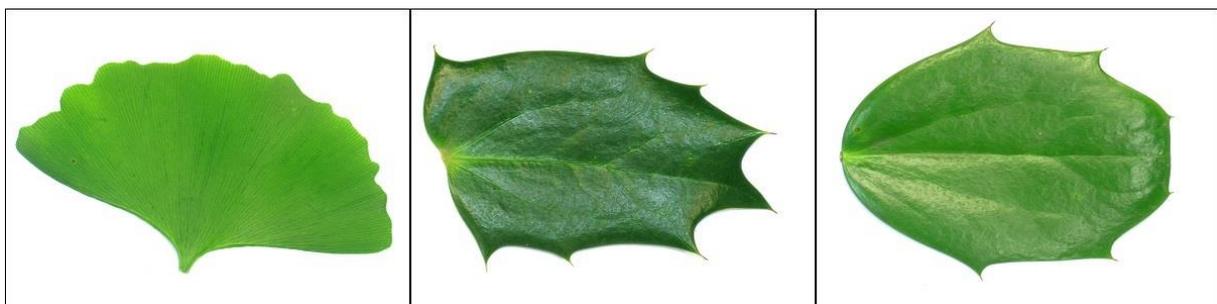


Abbildung 64: Problemfälle aus dem Flavia-Datenset. Links: *Ginkgo biloba*, Mitte und rechts: *Mahonia bealei*.

5.2 Farb-Features

Da als Input Farbbilder verwendet werden, bietet es sich an, die Aussagekraft der Farbgebung der Blätter zu untersuchen. Dazu werden die vier statistischen Momente Mittelwert μ , Standardabweichung σ , Schiefe v und Wölbung γ wie folgt berechnet:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.6)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (5.7)$$

$$v = \frac{\sum_{i=1}^n (x_i - \mu)^3}{n\sigma^3} \quad (5.8)$$

$$\gamma = \frac{\sum_{i=1}^n (x_i - \mu)^4}{n\sigma^4} - 3 \quad (5.9)$$

[Kadir 2014]

Abbildung 65 stellt die Bedeutung der vier statistischen Momente grafisch dar. Die Standardabweichung gibt die Streuung der Messwerte vom Mittelwert aus an. Wie stark asymmetrisch eine Häufigkeitsverteilung ist, wird durch die Schiefe angegeben. Wie steil oder spitz sie verläuft, gibt die Wölbung an.

[Wiki: {StdDev, Schiefe, Wölbung}]

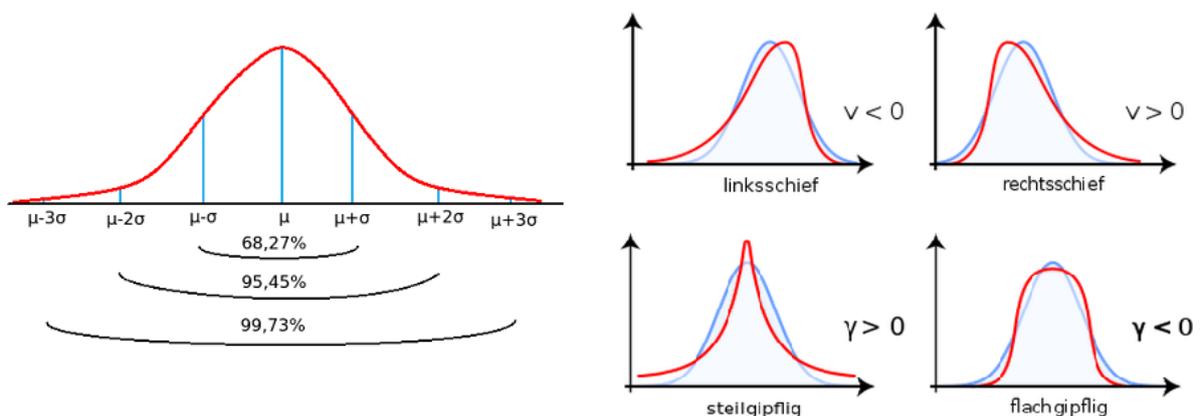


Abbildung 65: Die vier statistischen Momente. Links: Mittelwert und Standardabweichung, rechts: Schiefe (Mitte) und Wölbung (unten). [Wiki: {StdDev, Schiefe, Wölbung}]

Die Berechnung der vier Momente erfolgt für jeden Farbkanal (Rot, Grün, Blau) und das Graustufenbild. Insgesamt werden also 16 Farb-Features bestimmt. Um das Ergebnis durch die Größe des Blattes nicht zu verzerren, werden jeweils nur die Vordergrund-, also die Blattpixel, berücksichtigt.

Es ist bekannt, dass die Färbung von Blättern stark von äußeren Einflüssen, z. B. Umwelteinflüssen, abhängt. Diese Problematik wird in Abschnitt 7.6 noch ausführlich diskutiert.

5.3 Kurvatur-Features

Die in 5.1 vorgestellten Kontur-Features beschreiben effizient und zuverlässig die grobe Form eines Blattes. Um genauere Aussagen über die Beschaffenheit des Blattrandes treffen zu können, wird eine weitere Feature-Klasse, die Kurvatur-Features, eingeführt.

Um diese Features aus einer gegebenen Kontur zu extrahieren, muss zunächst das Zentrum $C(x, y)$ der Kontur ermittelt werden:

$$C(x, y) = C \left(\frac{1}{N} \cdot \sum_{i=1}^N x_i, \frac{1}{N} \cdot \sum_{i=1}^N y_i \right) \quad (5.10)$$

wobei N die Anzahl der Konturpunkte und x_n bzw. y_n die x - bzw. y -Koordinaten der Konturpixel beschreiben.

Vom Zentrum aus werden dann die interessanten Punkte (*Points of Interest*, kurz *PoI*) auf der Kontur des Blattes identifiziert. Dies ist in Abbildung 66 zu sehen. Der erste PoI liegt stets senkrecht über dem vorher bestimmten Zentrum. Von dort aus werden die restlichen Punkte abhängig vom vorgegebenen Winkel bestimmt. Die Kontur wird im Uhrzeigersinn durchlaufen und die Konturpunkte, die sich im Abstand eines ganzzahligen Vielfachen des Eingabewinkels vom Startpunkt befinden, gespeichert. Im unten gezeigten Beispiel betragen die Winkel 5° bzw. 15° .

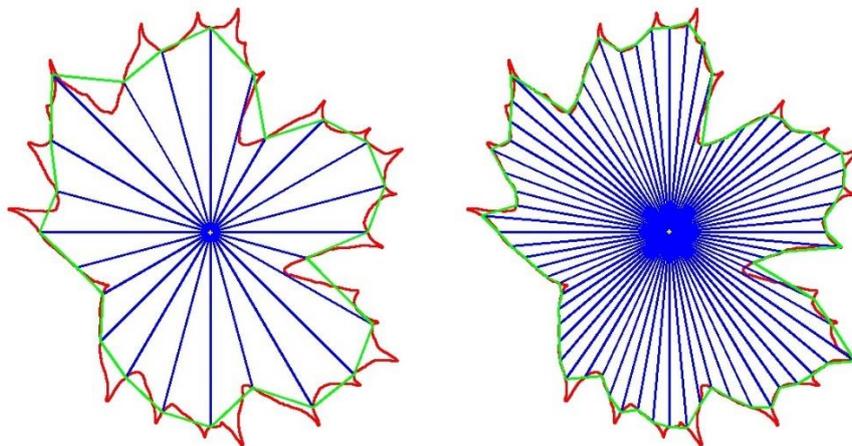


Abbildung 66: Bestimmung der Kurvatur-Features. Links: 15° , rechts: 5° . Kontur (rot), Zentrum (gelb), Distanzen zu den Pols vom Zentrum aus (blau) und Distanzen zwischen aufeinanderfolgenden Pols (grün).

Anschließend wird für alle gespeicherten Punkte die Distanz d zum Zentrum berechnet:

$$d(P(x_P, y_P), C(x_C, y_C)) = \sqrt{(x_C - x_P)^2 + (y_C - y_P)^2} \quad (5.11)$$

Um Größeninvarianz zu erreichen, werden alle Distanzen normiert, indem sie durch die maximal auftretende Distanz dividiert werden. Die Ergebnisse werden anschließend als Histogramm gespeichert.

Analog wird mit den Distanzen zwischen den aufeinanderfolgenden Punkten verfahren.

Zusätzlich zu den Verteilungen in den Histogrammen werden noch folgende Features gespeichert:

$$\text{MinDistanceRatio} = \frac{\text{Minimale Distanz zum Zentrum}}{\text{Durchschnittliche Distanz zum Zentrum}} \quad (5.12)$$

$$\text{MaxDistanceRatio} = \frac{\text{Maximale Distanz zum Zentrum}}{\text{Durchschnittliche Distanz zum Zentrum}} \quad (5.13)$$

$$\text{StandardDevRatio} = \frac{\text{Standardabweichung}}{\text{Durchschnittliche Distanz zum Zentrum}} \quad (5.14)$$

ZeroCrossingRate: Anzahl der auftretenden Vorzeichenwechsel bei Durchlauf der Punkte. Dabei beschreibt „+“ einen Wert, der größer als der Durchschnitt ist, „-“ einen kleineren.

TopPeaks: Anzahl der Punkte, die weiter als die durchschnittliche Distanz vom Zentrum entfernt liegen.

Die in Abbildung 66 grün markierten Strecken zwischen aufeinanderfolgenden PoIs finden im weiteren Verlauf keine Verwendung mehr. Kurze Tests der Features, die analog zu den eben vorgestellten, aus den Distanzen gewonnenen Features erstellt wurden, lieferten keine brauchbaren Ergebnisse.

5.4 Bildmomente

Wie bereits erwähnt, ist die äußere Kontur eines Blattes der wichtigste Anhaltspunkt für einen Vergleich bzw. für eine Klassifikation. Einer der simpelsten Wege, um zwei Konturen zu vergleichen, ist die Berechnung sogenannter Konturmomente. Im Folgenden sollen die Standard- oder auch Räumlichen Momente zunächst allgemein definiert werden. Anschließend wird eine für die Bildverarbeitung besonders nützliche Klasse von Momenten, die Hu-Momente, genauer vorgestellt und diskutiert. Zum Abschluss wird grob eine weitere Momenten-Klasse, die Zernike-Momente, besprochen.

5.4.1 Räumliche Momente

Betrachtet wird eine Kontur in einem Binärbild. Diese besteht aus n Pixeln, die jeweils eine x - und eine y - Koordinate sowie einen Intensitätswert I besitzen. Allgemein ist das (p, q) Moment $m_{p,q}$ einer solchen Kontur definiert als:

$$m_{p,q} = \sum_{i=1}^n I(x, y) \cdot x^p \cdot y^q \quad (5.15)$$

Dabei legen p und q jeweils fest, mit welcher Potenz die x - bzw. y -Koordinaten eines Pixels in die Summe eingehen. Es fällt auf, dass m_{00} lediglich die Länge der Kontur in Pixeln angibt, wenn jeder Konturpixel den Wert 1 besitzt und die Kontur nicht gefüllt ist.

Werden Momente für höhere Ordnungen berechnet, können damit zwei Konturen grob verglichen werden. In der OpenCV-Library werden dazu die sogenannten Räumlichen Momente m_{00} , m_{10} , m_{01} , m_{20} , m_{11} , m_{02} , m_{30} , m_{21} , m_{12} und m_{03} verwendet.

[Bradski 2008]

5.4.2 Hu-Momente

Bei den meisten praktischen Anwendungen erweisen sich die eben vorgestellten Standard-Momente als nutzlos, da sie weder translations-, noch skalierungs-, noch rotationsinvariant sind. Für zwei Konturen gleicher Form, aber verschiedener Größe ergeben sich bei der Berechnung deutlich unterschiedliche Momente. Gleiches gilt für zwei ähnliche oder sogar identische Konturen, die zueinander verdreht sind. Abhilfe schaffen die sogenannten Hu-Momente (vgl. [Hu 1962]), die sich wie folgt berechnen lassen.

5.4.2.1 Definition

Zunächst werden Zentrale Momente definiert, die invariant gegenüber Translation sind. Der einzige Unterschied zu den oben vorgestellten Räumlichen Momenten ist, dass von den x - und y -Koordinaten jeweils der zugehörige Mittelwert subtrahiert wird:

$$\mu_{p,q} = \sum_{i=1}^n I(x,y) \cdot (x - x_{avg})^p \cdot (y - y_{avg})^q \quad (5.16)$$

wobei

$$x_{avg} = \frac{m_{10}}{m_{00}}, \quad y_{avg} = \frac{m_{01}}{m_{00}} \quad (5.17)$$

Mit Hilfe der Zentralen Momente lassen sich die Normalisierten Momente ableiten, die sowohl invariant gegenüber Translation als auch gegenüber Skalierung sind:

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{00}^{\frac{p+q}{2}+1}} \quad (5.18)$$

Durch lineare Kombination der Zentralen Momente entstehen die Hu-Momente. Die Grundidee ist, durch die Kombination verschiedener Normalisierter Momente invariante Funktionen zu kreieren, die unterschiedliche Aspekte des Bildes unabhängig von Translation, Skalierung und Rotation darstellen. Hu definierte 1962 sieben dieser Momente:

$$h_1 = \eta_{20} + \eta_{02} \quad (5.19)$$

$$h_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \quad (5.20)$$

$$h_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (5.21)$$

$$h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (5.22)$$

$$\begin{aligned} h_5 = & (\eta_{30} - 3\eta_{12})^2 \cdot (\eta_{30} + \eta_{12}) \cdot \left((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03}) \right) \\ & + (3\eta_{21} - \eta_{03}) \cdot (\eta_{21} + \eta_{03}) \\ & \cdot \left(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right) \end{aligned} \quad (5.23)$$

$$\begin{aligned} h_6 = & (\eta_{20} - \eta_{02}) \cdot \left((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right) + 4\eta_{11} \\ & \cdot (\eta_{30} + \eta_{12}) \cdot (\eta_{21} + \eta_{03}) \end{aligned} \quad (5.24)$$

$$\begin{aligned}
 h_7 = & (3\eta_{21} - \eta_{03}) \cdot (\eta_{21} + \eta_{03}) \cdot (3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \\
 & - (\eta_{30} - 3\eta_{12}) \cdot (\eta_{21} + \eta_{03}) \\
 & \cdot 3((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)
 \end{aligned} \tag{5.25}$$

Die Momente lassen sich mit Hilfe der OpenCV-Library direkt aus einer detektierten Kontur extrahieren. Jedes Moment bildet ein Feature.

Des Weiteren bietet OpenCV die Möglichkeit, die Ähnlichkeit I zweier Konturen A und B über die Funktion `matchShapes` zu bestimmen. Die Berechnung verwendet dabei die Hu-Momente der Konturen und erfolgt mittels folgender Formel:

$$I(A, B) = \sum_{i=1}^7 \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \tag{5.26}$$

Für zwei identische Konturen ergibt sich der Wert 0. Je stärker die Konturen sich unterscheiden, desto größer ist der berechnete Wert. [Bradski 2008] [Hu 1962]

Im nächsten Abschnitt wird die Anwendung der Hu-Momente anhand eines Beispiels näher erläutert.

5.4.2.2 Beispiel

Zur Verdeutlichung werden die in Abbildung 67 zu sehenden Blätter hinsichtlich ihrer Hu-Momente und ihrer Ähnlichkeit untersucht.

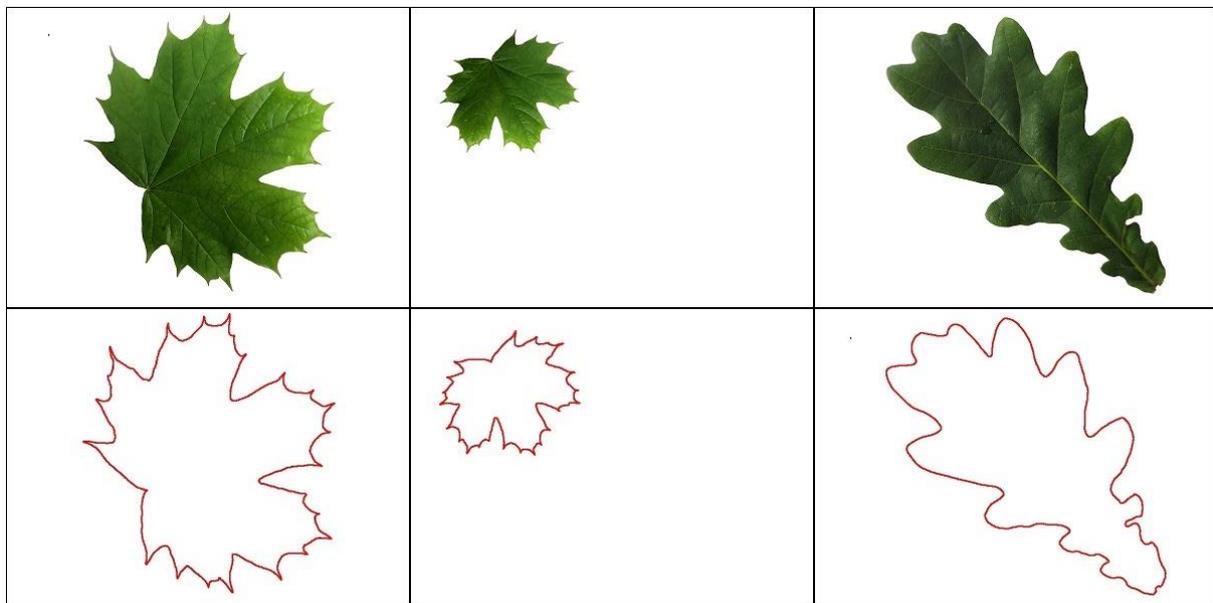


Abbildung 67: Drei Blätter (oben) mit ihren jeweiligen Konturen (unten). Links: Blatt 1, Mitte: Blatt 2: Blatt 1 um 75% verkleinert, um 90° im Uhrzeigersinn rotiert und in den Randbereich des Bildes verschoben, rechts: Blatt 3, dessen Kontur sich deutlich von der von Blatt 1 und 2 unterscheidet.

Zunächst wird die Kontur der Blätter extrahiert, danach werden die jeweiligen Hu-Momente berechnet. Die Ergebnisse sind in Tabelle 3 zu sehen.

Tabelle 3: Hu-Momente der drei Beispielblätter.

Blatt	Momente						
	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7
1	$6,9 \cdot 10^{-4}$	$3,9 \cdot 10^{-8}$	$1,3 \cdot 10^{-16}$	$6,1 \cdot 10^{-16}$	$1,5 \cdot 10^{-31}$	$1,0 \cdot 10^{-19}$	$7,2 \cdot 10^{-32}$
2	$6,9 \cdot 10^{-4}$	$3,7 \cdot 10^{-8}$	$4,7 \cdot 10^{-15}$	$1,3 \cdot 10^{-14}$	$2,0 \cdot 10^{-29}$	$1,9 \cdot 10^{-18}$	$-9,9 \cdot 10^{-29}$
3	$6,9 \cdot 10^{-4}$	$3,8 \cdot 10^{-8}$	$3,8 \cdot 10^{-15}$	$1,7 \cdot 10^{-15}$	$3,7 \cdot 10^{-30}$	$2,0 \cdot 10^{-19}$	$1,9 \cdot 10^{-30}$

Es fällt auf, dass sich aus den einzelnen Momenten nur schwer für den Menschen nachvollziehbare Informationen über das jeweilige Blatt ableiten lassen. Dem gegenüber sprechen die Werte, die bei einer Ähnlichkeitsberechnung erzielt werden können, eine deutliche Sprache:

Die Ähnlichkeit zwischen Blatt 1 und 2 beträgt $I(1,2) = 0,0012$, die zwischen Blatt 1 und 3 $I(1,3) = 0,5768$. Wie erwartet, ist das Ergebnis für Blatt 1 und 2 beinahe Null, da es sich, abgesehen von Translation, Rotation und Skalierung, um identische Konturen handelt. Ein Wert von 0 wird nicht erreicht, da bei der Skalierung kleine Änderungen an der Kontur auftreten. Das Ergebnis für Blatt 2 und Blatt 3 liegt ungleich höher, da es sich um zwei sehr unähnliche Konturen handelt.

5.4.3 Zernike-Momente

Aus Gründen der Vollständigkeit soll noch eine andere, weit verbreitete Klasse von Momenten, die sogenannten Zernike Momente, zumindest kurz vorgestellt werden. Diese sind zwar rotations-, aber nicht translations- und skalierungsinvariant. Alle Konturen müssen vor dem Vergleich auf eine einheitliche Größe normiert und innerhalb des Bildes zentriert werden.

Der grobe Ablauf der Berechnung der Zernike-Momente gestaltet sich wie folgt: Zunächst werden radiale Polynome berechnet und die Zernike-Basisfunktion bestimmt. Abschließend werden die Momente abgeleitet, indem das Bild auf die Basisfunktion projiziert wird. [Tsolakidis 2014]

Der große Vorteil der Zernike- gegenüber den Hu-Momenten ist, dass sich mit ihnen Konturen rekonstruieren lassen. Zu sehen ist dies in Abbildung 68.

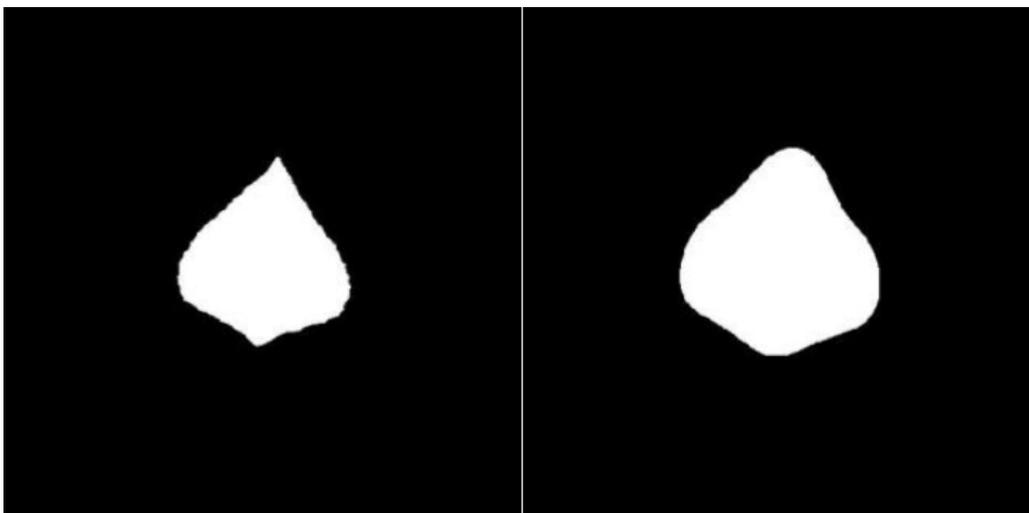


Abbildung 68: Rekonstruktion eines Blattes mit Hilfe von Zernike-Momenten.
Links: Original in Binärdarstellung, rechts: Rekonstruktion. [Tsolakidis 2014]

Diese Eigenschaft kann in praktischen Anwendungen sehr nützlich sein, im hier behandelten Anwendungsfall ist sie allerdings völlig wertlos. Da Hu- und Zernike-Momente ansonsten Konturen ähnlich exakt beschreiben (vgl. [Sabhara 2013]), wurden in dieser Arbeit die Hu-Momente verwendet, da sie sich sehr viel einfacher berechnen und handhaben lassen.

5.5 HOCS-Features

Die HOCS-Features (*Histogram Of Curvature over Scale*) ermöglichen es, sowohl die grobe Grundform eines Blattes als auch die feine Beschaffenheit des Randes exakt zu beschreiben. Sie wurden von Kumar et al. (vgl. [Kumar 2012]) entwickelt und sind die einzigen Features, die in der in 3.8.1 vorgestellten Leafsnap-App Verwendung finden.

Im Folgenden sollen zunächst die Features vorgestellt und die Grundidee ihrer Berechnung erläutert werden. In einem nächsten Schritt wird die Aussagekraft der HOCS-Features noch anhand eines Beispiels dargestellt. Abschließend sollen Möglichkeiten vorgestellt werden, den aufgrund der aufwendigen Berechnung sehr langsamen Feature-Gewinnungsprozess deutlich zu verkürzen, sowie deren praktische Umsetzung erläutert werden.

5.5.1 Grundidee

Die Grundidee hinter HOCS ist, Kreise einer bestimmten Größe um Konturpunkte zu legen und deren Eigenschaften zu betrachten. Dabei wird zum einen untersucht, welcher Anteil des Kreisbogens innerhalb der Kontur liegt und damit Blattpixel überdeckt. Zum anderen wird derselbe Ansatz mit der Kreisfläche wiederholt.

Abbildung 69 verdeutlicht den prinzipiellen Ansatz.

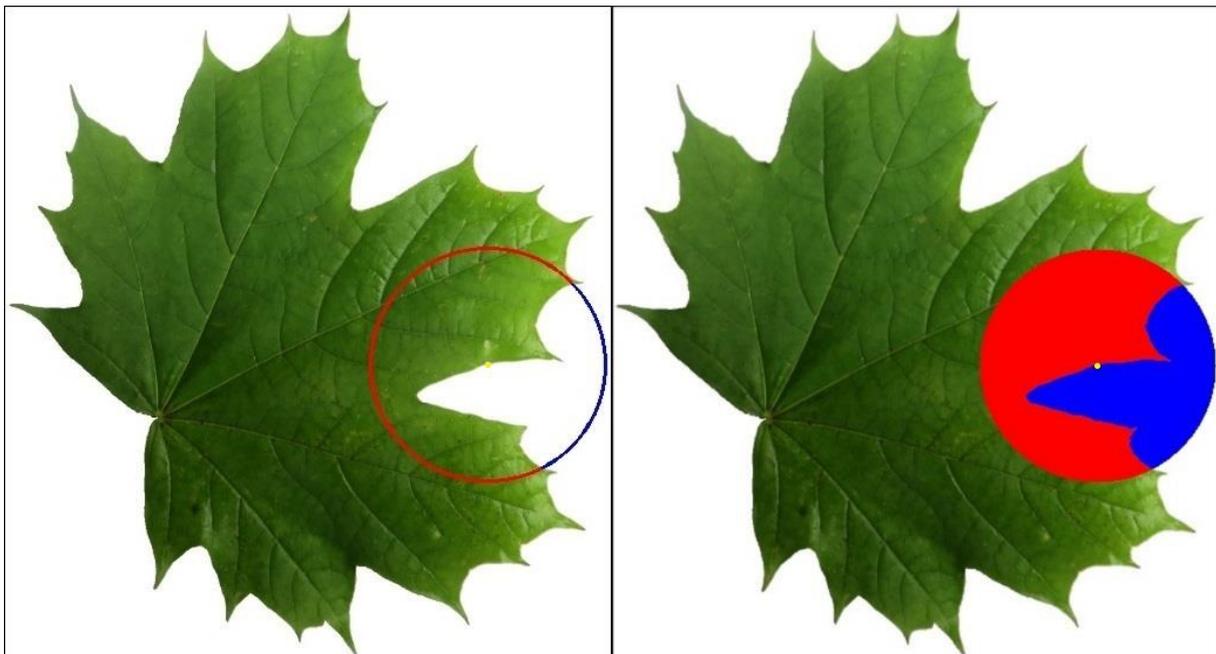


Abbildung 69: Die HOCS-Features. Links: Bogen, rechts: Fläche. Rot: Vom Kreisumfang bzw. der Kreisfläche überdeckte Vordergrundpixel, Blau: Hintergrundpixel analog, Gelb: Kreismittelpunkt.

Der zu betrachtende Konturpixel ist gelb markiert. Um diesen wird ein Kreis mit vorgegebenem Radius gezogen. Anschließend wird ermittelt, wie viele Vordergrund- bzw. Blattpixel sich auf dem Kreisbogen bzw. innerhalb der Kreisfläche befinden (rot). Diese Anzahl wird im Falle des Bogens durch die Gesamtzahl der Bogenpixel, im Falle der Fläche durch die Gesamtzahl der Flächenpixel des Kreises dividiert. Die erhaltenen Werte befinden sich alle im Intervall $]0;1[$ und lassen sich somit gut in Histogrammen darstellen. Es werden dabei pro Radius zwei Histogramme, je eines für Bogen und Fläche, erstellt.

5.5.2 Beispiel

Die Wirkungsweise der HOCS-Features lässt sich am besten an einem Beispiel veranschaulichen. Dazu werden aus zwei Bildern von Blättern, wie oben beschrieben, die Bogen- und Flächen-Features extrahiert. Als Beispielblätter werden Vertreter zweier Spezies gewählt, deren Blätter sich in der groben Grundform sehr ähnlich sind, bei denen aber die Feinbeschaffenheit des Randes völlig unterschiedlich ist. So sind die Blätter der Hain-Buche (*Carpinus betulus*, links) fein gezackt, während der Rand der Rot-Buche (*Fagus sylvatica*, rechts) glatt ist. Um Größeninvarianz zu erreichen, werden beide Blattkonturen auf 30.000 Pixel normiert.

Die beiden Beispielblätter sind in Abbildung 70 zu sehen.

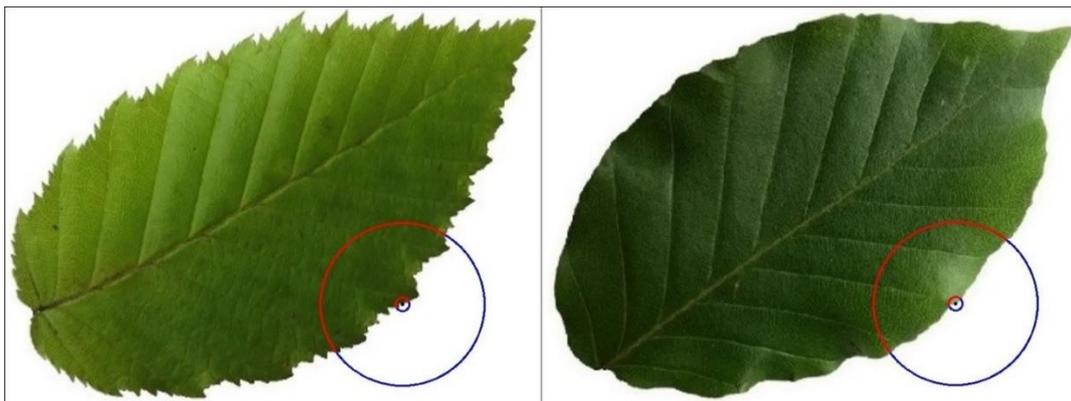


Abbildung 70: Beispielblätter zur Erläuterung der HOCS-Features.
Links: Hain-Buche, rechts: Rot-Buche.

Die verwendeten Radien sind zum einen ein sehr kleiner Radius (5 Pixel) und zum anderen ein sehr großer Radius (60 Pixel). Abbildung 71 und Abbildung 72 zeigen die Ergebnisse.

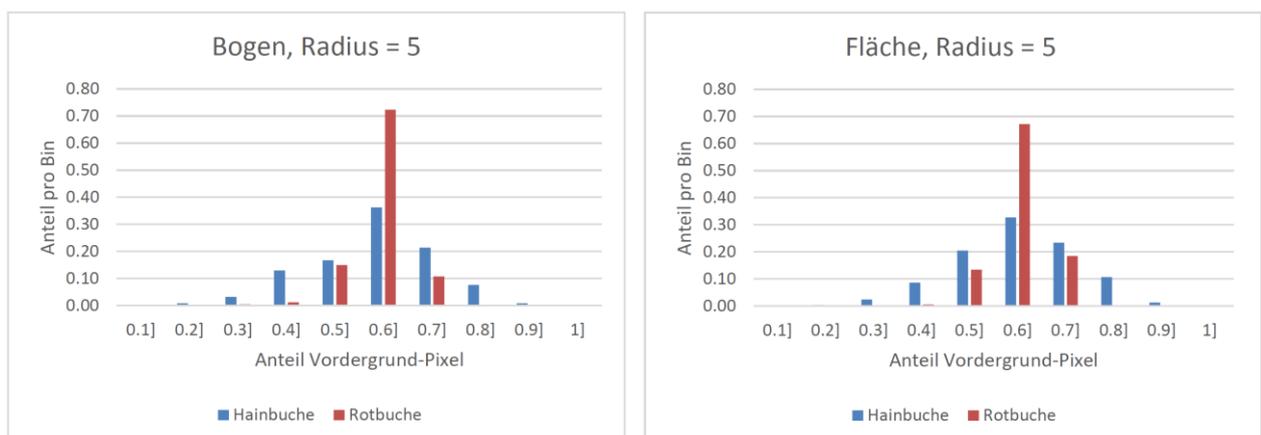


Abbildung 71: Die extrahierten HOCS-Features ($R=5$) der Beispielblätter in Histogramm-Form.

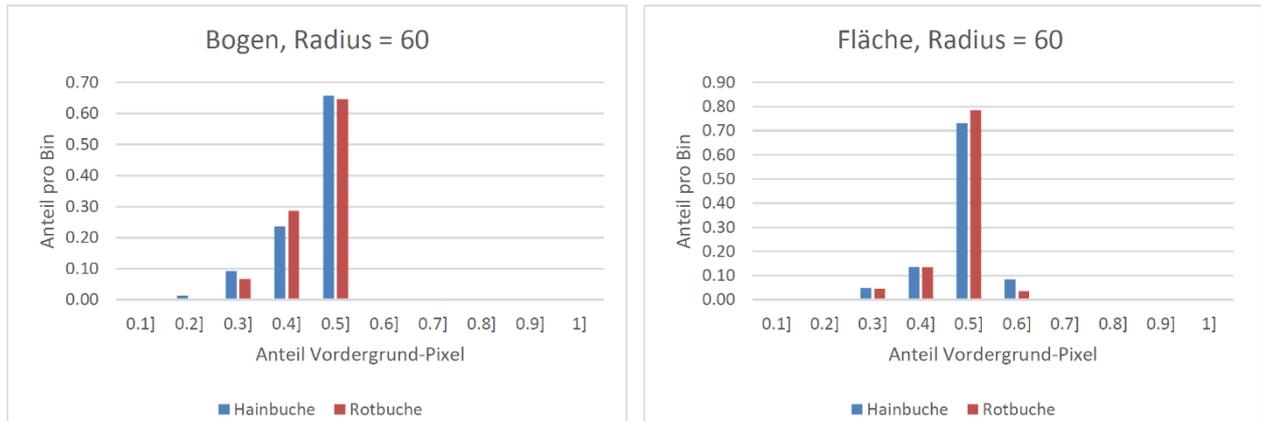


Abbildung 72: Die extrahierten HOCS-Features ($R=60$) der Beispielblätter in Histogramm-Form.

Betrachtet man zunächst die untere Reihe (Radius=60), so fällt auf, dass die Histogramme der beiden Spezies sowohl in Hinblick auf den Radius als auch in Hinblick auf die Fläche fast identisch sind. Dies erklärt sich dadurch, dass ein großer Radius vor allem dazu dient, die grobe Grundform des Blattes darzustellen. Wie oben bereits erwähnt, sind sich die beiden Beispielblätter in dieser Hinsicht sehr ähnlich.

Ein ganz anderes Bild zeigt sich in der oberen Reihe (Radius=5). Die beiden Histogramme sind völlig unterschiedlich. Während bei der Rot-Buche eine starke Häufung im mittleren Bereich auftritt, streuen die Werte bei der Hain-Buche deutlich mehr.

5.5.3 Speed-Ups

Bevor überprüft werden kann, ob es sich bei einem Pixel um Vordergrund oder Hintergrund handelt, muss zunächst sichergestellt werden, dass sich der Pixel überhaupt innerhalb des Kreises bzw. auf dem Kreisbogen befindet. Dazu wird im Normalfall die Kreisgleichung verwendet. Ein Punkt $P(P_x, P_y)$ liegt auf einem Kreis mit Zentrum $C(C_x, C_y)$ mit Radius r genau dann, wenn folgende Gleichung erfüllt ist, die sich aus einfachen geometrischen Überlegungen ergibt:

$$(P_x - C_x)^2 - (P_y - C_y)^2 = r^2 \quad (5.27)$$

Analog liegt ein Punkt innerhalb des Kreises, wenn gilt:

$$(P_x - C_x)^2 - (P_y - C_y)^2 < r^2 \quad (5.28)$$

Das Lösen dieser Gleichungen ist, gerade wenn es sich um eine große Anzahl von Pixeln handelt, sehr zeitintensiv. Im nächsten Abschnitt sollen daher zwei Methoden vorgestellt werden, die die benötigte Rechenzeit erheblich reduzieren.

5.5.3.1 Berechnung der Bogen-Features

Anstatt für jeden Pixel die zugehörigen Kreispunkte neu zu berechnen, werden diese zu Beginn der Feature-Extraktion einmalig ermittelt, gespeichert und im weiteren Verlauf lediglich nachgeschlagen. Die Vorgehensweise ist dabei wie folgt:

Für jeden verwendeten Radius müssen zunächst die Kreispunkte ermittelt werden. Dazu wird angenommen, dass der Kreis um das Zentrum $C(C_x, C_y)$ gezeichnet wird. Die Kreispixel sind durch die Punkte $P(P_x, P_y)$ gegeben.

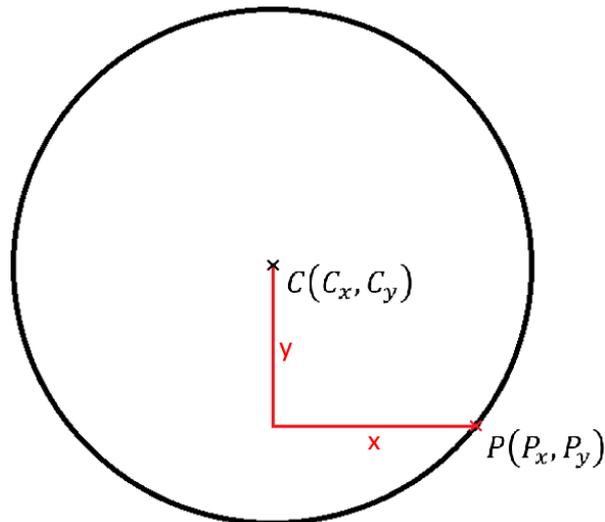


Abbildung 73: Speed-Up der Berechnung der Bogenfeatures.

Wie in Abbildung 73 zu sehen, lassen sich die Koordinaten der Kreispunkte jeweils durch die Summe aus den Koordinaten des Zentrums und dem Offset in horizontaler (x) und vertikaler Richtung (y) angeben:

$$P(P_x, P_y) = P(C_x + x, C_y + y) \quad (5.29)$$

Es bietet sich daher an, die Kreispunkte jeweils nur durch ihre Offsetwerte x und y zu speichern. Bei der späteren Bestimmung der Kreispunkte um einen Konturpunkt müssen dann lediglich die Offsets für den jeweiligen Radius nachgeschlagen und auf die Koordinaten des Konturpunktes aufaddiert werden. Dies reduziert den Rechenaufwand erheblich, da die aufwendige Bestimmung der zu einem Zentrum gehörigen Kreispixel nur einmalig zu Beginn erfolgt. Für die spätere Identifikation der Kreispixel sind lediglich einfache Additionen vonnöten. [Kumar 2012]

5.5.3.2 Berechnung der Flächen-Features

Der eben erläuterte Ansatz lässt sich natürlich auch bei Flächen-Features anwenden. Im Gegensatz zu den Bogen-Features besteht bei diesen das Problem, dass für jeden Konturpunkt eine enorme Anzahl von Pixeln darauf überprüft werden muss, ob es sich um Vordergrund oder Hintergrund handelt. Diese Überprüfung ist an sich zwar nicht rechenintensiv, muss allerdings extrem häufig durchgeführt werden. Dies soll anhand des in 5.5.2 vorgestellten Beispiels erläutert werden:

Betrachtet wird lediglich eines der beiden Blätter und nur der größere der beiden Radien ($r=60$ Pixel). Bei der Berechnung der Flächenfeatures muss die Vordergrund/Hintergrund-Überprüfung pro Konturpunkt für jeden Kreispixel durchgeführt werden, also für etwa 11.300 Pixel ($\pi \cdot 60^2$). Die Kontur jedes der beiden Blätter besteht aus etwa 3.000 Pixeln. Daraus ergibt sich eine Gesamtzahl von fast 34.000.000 (11.300 Kreispixel \cdot 3.000 Konturpixel) Überprüfungen. Werden weitere Radien verwendet, steigt diese Zahl natürlich stark an.

Die Anzahl der Vergleiche kann durch eine einfache Überlegung stark eingeschränkt werden. Dazu ist lediglich die Annahme nötig, dass die Punkte einer Kontur in einer sinnvollen Reihenfolge vorliegen, also der Konturpunkt $i + 1$ direkt an den Konturpunkt i anschließt. Unter dieser Annahme und aufgrund der diskreten Anordnung der Pixel lässt

sich leicht feststellen, dass sich der nächste Konturpixel nur in acht Richtungen vom letzten Konturpixel befinden kann.

Eine weitere Beobachtung wird in Abbildung 74 deutlich. Bei der Verschiebung des Zentrums um nur einen Pixel bleibt ein Großteil der Kreispixel erhalten. Ausschließlich in der Nähe des Kreisbogens treten leichte Veränderungen auf. So fallen bei der Verschiebung des Kreises nach rechts unten einige Pixel im linken, oberen Bereich des Kreises weg, während dieselbe Anzahl Pixel im rechten, unteren Bereich hinzukommt.

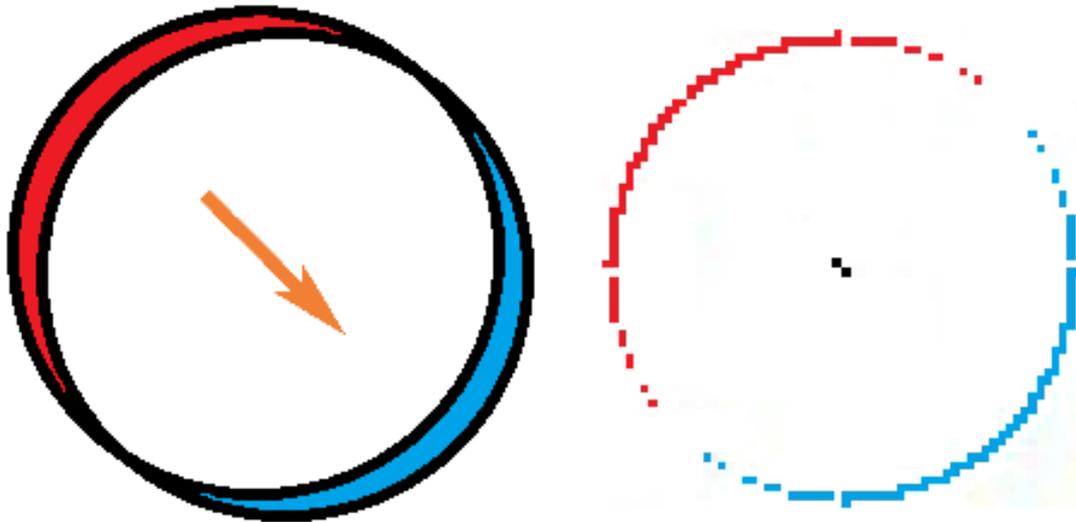


Abbildung 74: Wegfallende (rot) und hinzukommende (blau) Pixel bei einer Verschiebung des Zentrums nach rechts unten. Links: schematische Darstellung, rechts: realistische Darstellung.

Das wiederum bedeutet, dass sich der Überprüfungsaufwand deutlich reduzieren lässt, indem jeweils die Anzahl der Vordergrundpixel des letzten Konturpunkts gespeichert wird. Bei einer Verschiebung müssen nur die weggefallenen und neu hinzugekommenen Pixel überprüft werden. Der Wert errechnet sich wie folgt:

$$\begin{aligned} \text{Ergebnis}[i] &= \text{Ergebnis}[i - 1] + \text{neue Vordergrundpixel} \quad (5.30) \\ &\quad - \text{alte Vordergrundpixel} \end{aligned}$$

Dies verringert die benötigte Anzahl an Vergleichen von πr^2 bei der naiven Methode auf ca. $2\pi r$. Bei obigem Beispiel entspricht das einer Reduktion von 34.000.000 Überprüfungen auf 1.131.000.

5.5.3.3 Praktische Umsetzung

Wegen der pixelbasierten Darstellung ist es in der Praxis unmöglich, einen vollkommen runden Kreis zu kreieren. Es zeigt sich auch, dass es schwierig ist, mit Hilfe von Gleichung (5.27) exakt die Pixel zu bestimmen, die mathematisch gesehen auf dem Kreis liegen.

Dies wird in Abbildung 75 deutlich.

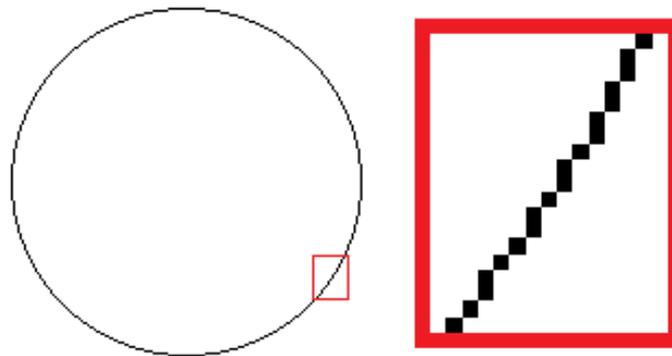


Abbildung 75: Darstellung eines Kreises.

In vorliegender Arbeit wird dieses Problem wie folgt umgangen:

Die Entscheidung, welche Pixel auf einem Kreis mit einem bestimmten Radius liegen, wird nicht bzw. nicht direkt mit Hilfe der Formel berechnet. Vielmehr wird für jeden Radius ein weißer Kreis in die Mitte eines vollkommen schwarzen Bildes gezeichnet. Dazu dient die OpenCV-Methode `circle`, die einen optimierten Kreis mit vorgegebener Dicke des Randes, in diesem Fall 1, zeichnet. Im Anschluss wird das Bild durchlaufen und die Offsets aller weißen Pixel vom Kreismittelpunkt gespeichert.

Analog wird mit den Pixeln innerhalb des Kreises verfahren bzw. mit den Pixeln, die bei einer Verschiebung des Kreismittelpunktes wegfallen oder hinzukommen. In diesem Fall wird wiederum ein ausgefüllter, weißer Kreis in ein schwarzes Bild gezeichnet. Anschließend wird derselbe Kreis in ein zweites Bild gezeichnet, allerdings mit einem um einen Pixel verschobenen Zentrum. Dies wird für jeden Radius und für jede der acht möglichen Verschiebungen durchgeführt, und die jeweils wegfallenden und neu hinzukommenden Pixel werden anhand ihrer Offset-Koordinaten gespeichert.

5.5.4 Bilaterale Filterung

Ein typisches Problem bei der Arbeit mit Histogrammen sind auftretende Ungenauigkeiten, wenn sich Werte im Grenzbereich der Bins befinden. Dies lässt sich anhand eines simplen Beispiels leicht nachvollziehen: Angenommen ein Histogramm teilt sich in zehn Bins auf: Bin 1: $[0;0,1[$, Bin 2: $[0,1;0,2[$ usw.. So werden alle Werte zwischen 0,1 und 0,2 exakt gleich behandelt, was u. a. dazu führt, dass ein Wert von 0,1999 in Bin 1 wandert und der nur minimal größere Wert 0,2 in Bin 2. Dies kann, gerade wenn Werte gehäuft in diesen Grenzbereichen auftreten, starke Verschiebungen in den Histogrammen hervorrufen.

Abhilfe schafft die sogenannte bilaterale Filterung. Mit ihr ist es möglich, einen Wert auf zwei Bins zu verteilen. Umgesetzt wird dies mit Hilfe einer Gewichtung, die vom Abstand des Wertes von den beiden nächsten Bin-Mittelwerten abhängt.

Der gesamte Vorgang soll anhand eines Beispiels erläutert werden: Sei der einzusortierende Wert 0,19 und das Histogramm verfüge über zehn Bins, wie oben beschrieben. Zunächst werden die beiden nächstgelegenen Mittelwerte bestimmt, in diesem Fall 0,15 und 0,25. Der Abstand zum ersten Mittelwert beträgt 0,04, der zum zweiten 0,06. Im Verhältnis zum Gesamtintervall 0,1 entspricht dies 0,4 und 0,6. Der Counter der zugehörigen Bins wird demnach um 0,4 (Bin 2: $[0,1;0,2[$) und 0,6 (Bin 3: $[0,2;0,3[$) erhöht.

5.6 Binary Pattern-Features

Die letzte in dieser Arbeit verwendete Feature-Klasse sind die sogenannten Binary Patterns. Diese liefern eine weitere Möglichkeit, sowohl die grobe Form der äußeren Kontur eines Blattes als auch deren feine Beschaffenheit zu beschreiben.

Im Folgenden soll zunächst der prinzipielle Ansatz der Binary Patterns erläutert werden. Anschließend werden einige Anpassungen vorgestellt, die eine leichtere Handhabung und eine bessere Generalisierbarkeit der Binary Patterns ermöglichen.

5.6.1 Grundidee

Genau wie die HOCS-Features werden die Binary Patterns zunächst für jeden Konturpunkt berechnet und anschließend in Histogrammen zusammengefasst. Zur Berechnung des zugehörigen Pattern eines Konturpixels werden Pixel betrachtet, die sich auf einem Kreis um eben diesen Konturpunkt befinden. Dies wird in Abbildung 76 deutlich. Es wird dabei folgende Notation verwendet: Auf einem Kreis mit dem Radius R befinden sich P Konturpunkte, dies wird mit (P,R) bezeichnet. $(8,1)$ beschreibt demnach einen Kreis mit einem Radius von einem Pixel, auf dem acht Punkte betrachtet werden.

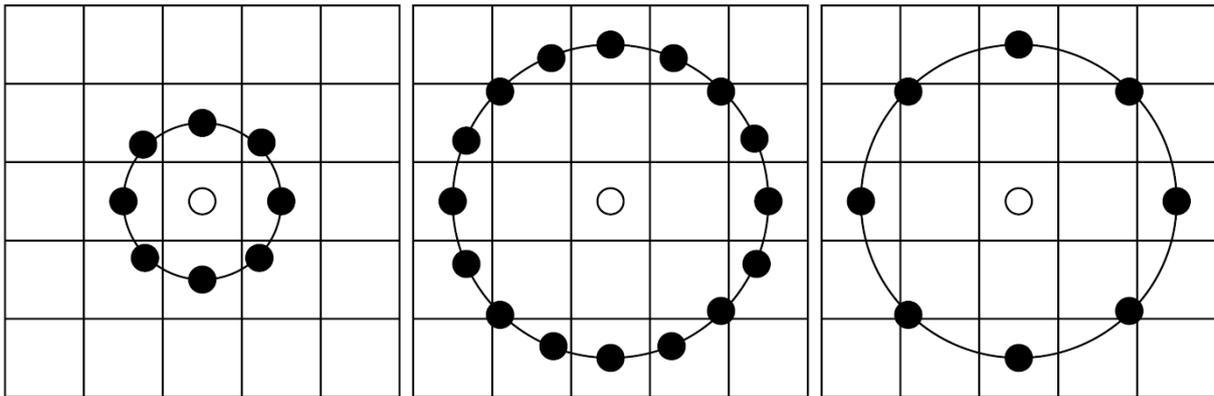


Abbildung 76: Zu betrachtende Konturpunkte für drei Fälle.
Von links nach rechts: $(8,1)$, $(16,2)$ und $(8,2)$. [Pietikäinen 2011]

Um einen Pattern zu generieren, werden diese Punkte darauf untersucht, ob sie einem Vordergrundpixel mit dem Wert 1 oder einem Hintergrundpixel mit dem Wert 0 entsprechen. Für jeden Punkt und für jeden Radius R ergibt sich danach ein Pattern der Länge P . Beispiele für den Fall $(8,1)$ sind in Abbildung 77 zu sehen.

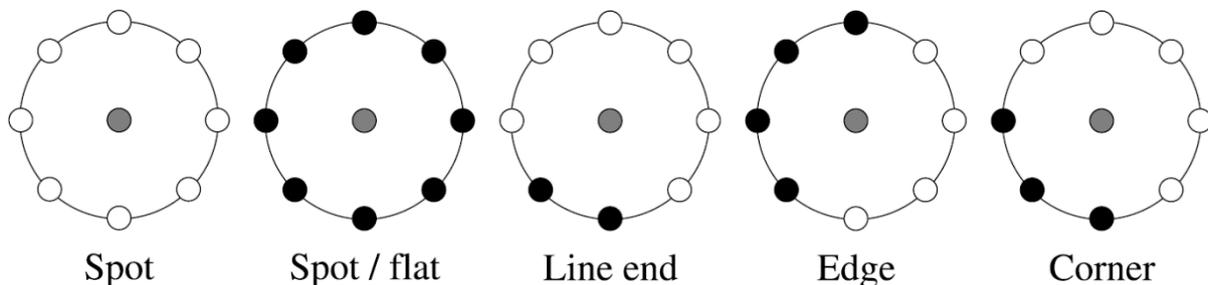


Abbildung 77: Beispiele für primitive Strukturen für den Fall $(8,1)$. Hintergrundpixel (schwarz, Wert 0 im Binärbild) und Vordergrundpixel (weiß, Wert 1).

Jedes Pattern lässt sich ausschreiben, indem man von einem fixen Startpunkt aus die jeweiligen Werte abliest. Startet man beispielsweise bei dem Pattern, der eine Ecke

beschreibt (Abbildung 77, rechts), bei dem Pixel, der sich senkrecht über dem Kreismittelpunkt befindet, und bewegt sich anschließend im Uhrzeigersinn um den Kreis, erhält man den Binary Pattern 00001110. [Pietikäinen 2011]

5.6.2 Anpassungen

Das eben vorgestellte Vorgehen erlaubt eine sehr exakte Beschreibung der Umgebung eines Konturpixels. Allerdings besitzt dieser primitive Ansatz einige Schwächen. Beispielsweise bietet er nur eine sehr geringe Generalisierbarkeit und ist somit vor allem nicht rotationsinvariant. Im Folgenden sollen deshalb einige Verbesserungen aufgezeigt werden.

5.6.2.1 Uniforme Patterns

Werden Patterns der Länge 8 betrachtet, sind insgesamt 256 ($=2^8$) unterschiedliche Patterns möglich, da jede der acht Stellen entweder den Wert 0 oder 1 annehmen kann. Dies entspricht demnach in einer realen Anwendung 256 Features pro Radius. Um eine solch große Anzahl erheblich zu reduzieren, wird zwischen sogenannten uniformen und nicht-uniformen Patterns unterschieden.

Als uniform werden dabei die Patterns bezeichnet, die maximal zwei Übergänge von 0 auf 1 oder umgekehrt aufweisen. Beispielsweise sind die Patterns 00000001 (1 Übergang) und 00111100 (2 Übergänge) uniform, während die Patterns 10010001 (4 Übergänge) und 10101010 (7 Übergänge) nicht-uniform sind. Im Falle von acht verwendeten Punkten pro Radius führt das zu 58 unterschiedlichen uniformen Patterns, die in Abbildung 78 zu sehen sind.

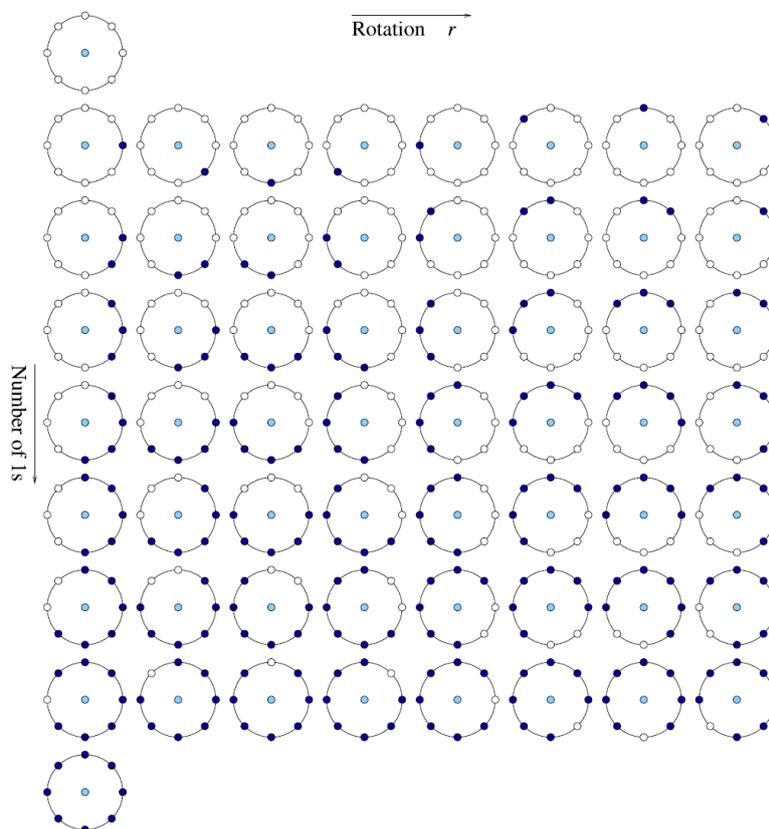


Abbildung 78: Die 58 unterschiedlichen, uniformen Patterns in einer $(8,R)$ -Nachbarschaft. [Pietikäinen 2011]

Ahonen et al. (vgl. [Ahonen 2006]) stellten bei der Untersuchung der Verwendung von Binary Patterns zur Gesichtswiedererkennung fest, dass diese uniformen Patterns den überwiegenden Teil der in einem Bild enthaltenen Information wiedergeben. So sind bei einer Nachbarschaft von (8,1) 90,6% der vorkommenden Patterns uniform. Im Fall (8,2) sind es immer noch 85,2%. [Ahonen 2006]

Bei der realen Anwendung reduziert sich demnach bei der Verwendung von acht Punkten pro Radius die Anzahl der Bins von 256 auf 59, d. h. 58 Bins für die unterschiedlichen uniformen Patterns, 1 Bin für alle nicht-uniformen Patterns. Neben einer Reduktion des Zeitaufwandes bei der Klassifikation führt dies auch zu einer erhöhten Robustheit gegenüber Rauschen. [Pietikäinen 2011]

5.6.2.2 Rotationsinvarianz

Während die zuvor vorgestellten HOCS-Features von Natur aus rotationsinvariant sind, ist dies bei den Binary Patterns nicht der Fall. Betrachtet man beispielsweise in Abbildung 78 das Pattern ganz links, lässt sich erkennen, dass es eine Kante beschreibt, die steil von links unten nach rechts oben oder umgekehrt verläuft. Das zweite Pattern dagegen entspricht derselben Kante, die jedoch wesentlich flacher verläuft. In einer realen Anwendung kann diese zusätzliche Information von großer Bedeutung sein, solange sichergestellt werden kann, dass die zu vergleichenden Objekte stets gleich ausgerichtet sind, wie es z. B. bei der Gesichtserkennung meist vorausgesetzt werden kann. In dieser Arbeit ist das allerdings nicht der Fall. Die Blätter können auf den Bildern beliebig ausgerichtet sein.

Bei erneuter Betrachtung von Abbildung 78 fällt auf, dass sich die einzelnen Patterns einer Zeile lediglich durch ihre Ausrichtung, also durch ihre Rotation, unterscheiden. Dieser Umstand wird durch das Beispiel in Abbildung 79 deutlich.

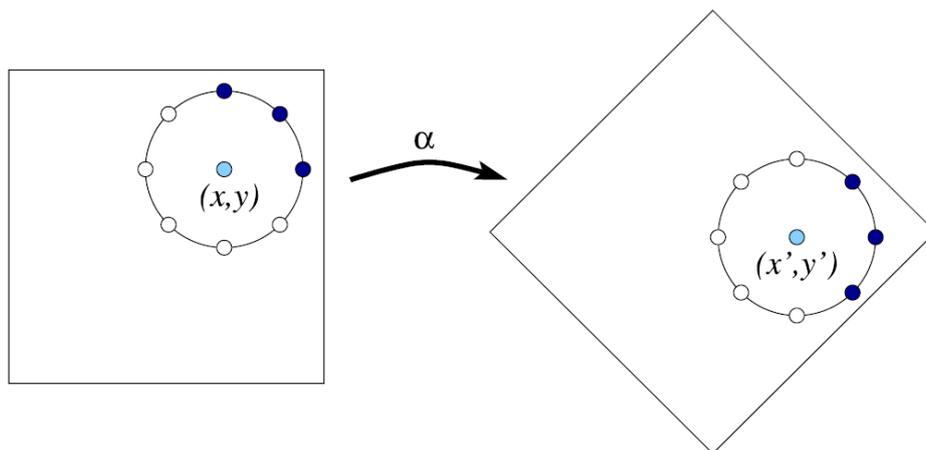


Abbildung 79: Einfluss der Rotation eines Bildes auf ein Binary Pattern. [Pietikäinen 2011]

Um Rotationsinvarianz zu erreichen, müssen daher lediglich alle Patterns einer Zeile, das bedeutet alle uniformen Patterns mit derselben Anzahl an Einsen, zusammengefasst werden. Bei acht Punkten pro Radius führt dies zu lediglich zehn Bins pro Radius, neun für die Anzahl der Einsen von 0 bis 8 und ein Bin für die uniformen Patterns.

Neben der Invarianz gegenüber Rotation wird durch diese Anpassung eine hohe Robustheit und Generalisierbarkeit gewährleistet. Außerdem sinkt der Zeitaufwand bei der Klassifikation durch die erhebliche Reduktion der Bins und damit der Features pro Radius von 59 auf 10.

Auf ein abschließendes Beispiel wird an dieser Stelle bewusst verzichtet, da für den Menschen die Binary Patterns in Histogramm-Form deutlich weniger anschaulich sind als die HOCS-Features

5.7 Nicht verwendete Features

Wie zu Beginn dieses Kapitels erwähnt, lässt sich in der Literatur noch eine Vielzahl weiterer Features und Feature-Klassen finden, die in dieser Arbeit allerdings keine Verwendung finden. Im Folgenden soll ein kurzer Überblick über die drei wichtigsten gegeben werden.

5.7.1 Venierungs-Features

Die Venierung eines Blattes, also der Verlauf der Blattnerven, ist ein wichtiges Bestimmungsmerkmal und findet in mehreren Publikationen zur Blattklassifikation Verwendung. Die Herangehensweise ist dabei sehr simpel:

Zunächst wird auf der Graustufenansicht des Blattes ein morphologisches Opening mit Hilfe eines scheibenförmigen Strukturelements durchgeführt. Dies führt zu einer deutlicheren und breiteren Darstellung der Venierung des Blattes. Durch eine einfache Grenzwertoperation wird das Graustufenbild in die binäre Darstellung überführt (vgl. Abbildung 80). Diese Schritte werden mit mehreren Strukturelementen verschiedener, kleiner Radien durchgeführt. Die eigentliche Feature-Extraktion erfolgt durch das Bilden von Verhältnissen der Anzahl der entstandenen weißen Pixel unter Verwendung der verschiedenen Radien. [Wu 2007]

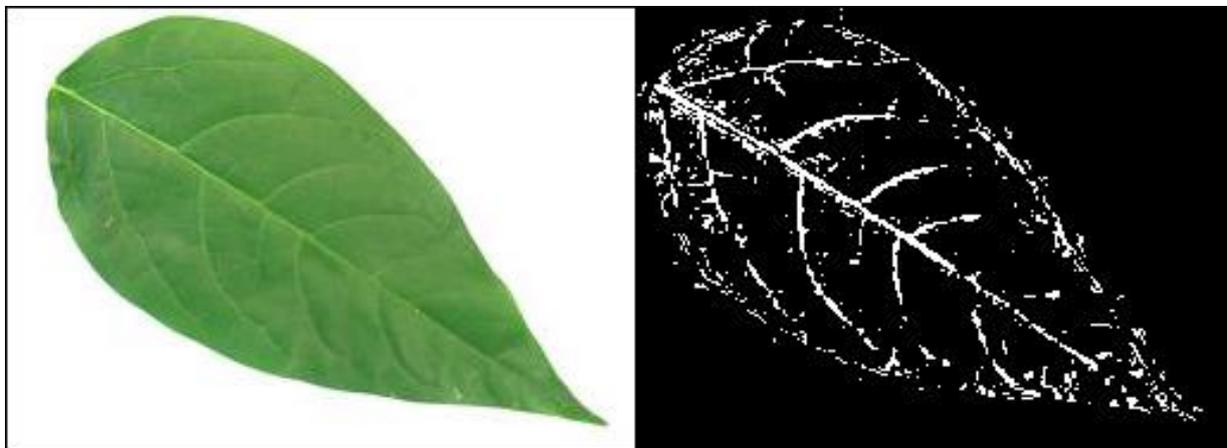


Abbildung 80: *Beispiel einer erfolgreichen Venenextraktion.
Links: Originalbild, rechts: Ergebnis als Binärdarstellung. [Wu 2007]*

Das vorgestellte Vorgehen konnte zwar das Klassifikationsergebnis in einigen Systemen leicht steigern (vgl. [Wu 2007] [Kadir 2014]), jedoch ist es nach Ansicht dieser Arbeit höchst zweifelhaft:

Die Qualität der Extraktion der Venen ist stark abhängig von den gewählten Parametern beim Opening und der Überführung in die Binärform. Überdies wird das Ergebnis erheblich von der Qualität des ursprünglichen Farbbildes beeinflusst. Durch Testen vieler Parameterkombinationen ist es zwar möglich, für einen Großteil der Blätter eines einzelnen Datensets recht ansprechende Ergebnisse zu erhalten, allerdings liefern

dieselben Parameter bei Verwendung eines anderen Datensets nur unzureichende Ergebnisse.

Eine Parameteroptimierung auf jedem einzelnen Datenset widerspricht dem Grundgedanken dieser Arbeit. Überdies ist es nicht möglich, bei der Aufnahme eines Blattes durch einen Nutzer stets zu gewährleisten, dass die Qualität des Bildes hoch genug ist, um eine zuverlässige Extraktion der Venierung zu sichern, auch nicht mit ggf. komplexeren Methoden als der eben vorgestellten.

Abschließend sei erwähnt, dass eine zuverlässige und saubere Extraktion der Venierung durchaus zu sehr brauchbaren Ergebnissen führen könnte. Allerdings müssten dazu wohl komplexere Features verwendet werden als die bereits erwähnten Flächenverhältnisse, z. B. die Lage der Ansatzpunkte der Venierung zweiten Grades an die Hauptvene.

5.7.2 Gray-Level-Co-Occurrence-Matrix (GLCM)-Features

Die GLCM ist ein wichtiges Hilfsmittel in der Bildverarbeitung. Sie gibt Auskunft über den Kontrastreichtum der Oberflächenstruktur eines Objektes. Dazu werden für jeden Pixel die Nachbarpixel betrachtet und die Häufigkeit auftretender Patterns für jede Richtung gespeichert. Eine Zelle der GLCM gibt beispielsweise an, wie oft der rechte Nachbarpixel eines Pixels mit dem Wert X den Wert Y besitzt. Dies wird für alle Wertepaare und Richtungen durchgeführt. Meist werden die ursprünglichen Pixelwerte zunächst auf einen kleineren Bereich gemappt, z. B. Werte von 0 bis 3 anstatt von 0 bis 255, um eine aussagekräftigere Matrix zu erhalten. Als Features dienen anschließend aus der GLCM extrahierte statistische Momente (vgl. [Kadir 2014]).

Die Vorgehensweise soll anhand des in Abbildung 81 zu sehenden Beispielbildes kurz erläutert werden. Die Pixelwerte bewegen sich im Bereich von 0 bis 3; je höher der Wert desto heller der Pixel.

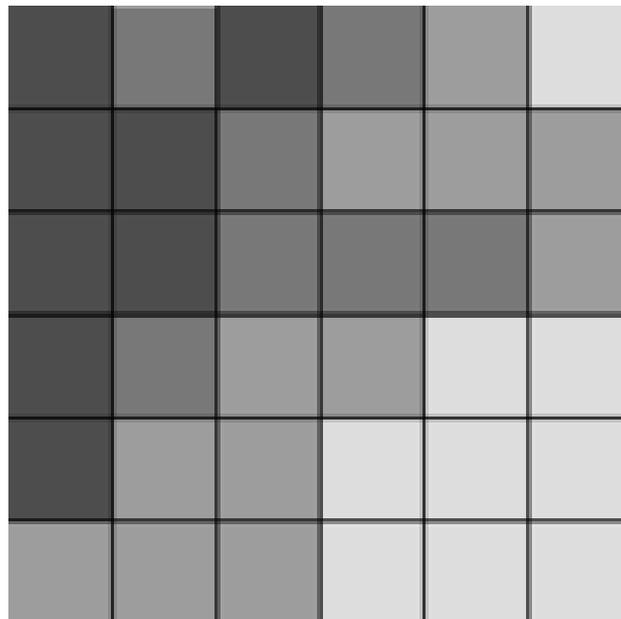


Abbildung 81: Ausgangsbild für das GLCM-Beispiel. [Wiki: GLCM]

Die Matrixdarstellung des Bildes G lautet demnach:

$$G = \begin{bmatrix} 0 & 1 & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & 2 & 3 & 3 \\ 0 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix} \quad (5.31)$$

Betrachtet man anschließend die Grauwertematrix $W(g_1, g_2)$, in der g_1 den Wert des aktuell betrachteten Pixels angibt und g_2 den Wert seines rechten Nachbarn:

$$W(g_1, g_2) = \begin{bmatrix} (0,0) & (0,1) & (0,2) & (0,3) \\ (1,0) & (1,1) & (1,2) & (1,3) \\ (2,0) & (2,1) & (2,2) & (2,3) \\ (3,0) & (3,1) & (3,2) & (3,3) \end{bmatrix} \quad (5.32)$$

Daraus folgt:

$$W(g_1, g_2) = \begin{bmatrix} 2 & 5 & 1 & 0 \\ 1 & 2 & 4 & 0 \\ 0 & 0 & 6 & 4 \\ 0 & 0 & 0 & 5 \end{bmatrix} \quad (5.33)$$

Der Wert $W(1,2)$ ergibt sich beispielsweise daraus, wie oft in G der rechte Nachbar einer Eins eine Zwei ist. [Wiki: GLCM]

In dieser Arbeit findet die GLCM keine Anwendung, da die durch sie gelieferte Information bereits sehr gut von den Binary Pattern-Features abgedeckt wird. Außerdem ist es trotz einiger Anpassung und Verallgemeinerungen nicht möglich, eine ausreichende Rotationsunabhängigkeit zu erreichen.

5.7.3 Histogramm Of Orientated Gradients (HOG)-Features

Die HOG-Features sind in der Lage, sowohl die Kontur als auch die Venierung eines Blattes zu beschreiben. Zunächst wird ein Graustufenbild in kleine Zellen, beispielsweise 4x4-Pixel-Zellen, unterteilt. Anschließend werden diese hinsichtlich ihrer Helligkeit normiert und jeweils ein Histogramm der auftretenden Gradienten-Richtungen erstellt. Der allgemeine Ablauf ist in Abbildung 82 zu sehen.

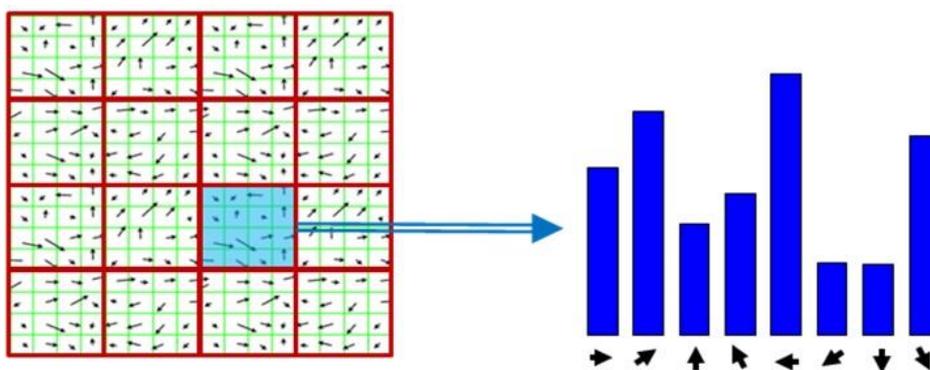


Abbildung 82: Prinzipielle Vorgehensweise der HOG-Features. Links: In Zellen unterteiltes Bild mit Richtung und Stärke aller Gradienten, rechts: zugehöriges Histogramm. [Gil]

Jeder Gradient geht dabei mit einem Gewicht ein, das von der Stärke des Gradienten abhängt. Die einzelnen Histogramme werden konkateniert und bilden den Feature-Vektor. [Gil]

Systeme, die HOG-Features verwenden, konnten sehr gute Ergebnisse in verschiedensten Bereichen erzielen, speziell bei der Gesichts- und Personenerkennung. Für die Blattklassifikation sind sie nach Auffassung dieser Arbeit jedoch eher weniger geeignet, da sie weder skalierungs- noch rotationsinvariant sind.

6 Konzept

In diesem Kapitel soll zum einen ein Überblick über den allgemeinen Ablauf gegeben werden, mit dem aus dem Bild eines Blattes eine abstrakte Feature-Repräsentation entsteht, die schließlich zur Klassifikation genutzt werden kann.

Außerdem sollen verschiedene konzeptionelle Entscheidungen diskutiert werden, die im Verlauf dieser Arbeit getroffen wurden. Dies beinhaltet beispielsweise das allgemeine Klassifikationsverfahren mittels Machine Learning-Techniken. Es soll erläutert werden, warum sich die Herangehensweise, wie sie z. B. in Bestimmungsbüchern vorgeschlagen wird, nicht für die Computer-gestützte Klassifikation eignet.

Zum Abschluss wird erläutert, warum sich das in dieser Arbeit entwickelte System auf die Klassifikation von Laubblättern beschränkt und wie zusammengesetzte Blätter behandelt werden.

6.1 Überblick

In Abbildung 83 ist der schematische Ablauf zu sehen, der von einem Eingabebild über verschiedene Vorverarbeitungsschritte und die Feature-Extraktion zu einer Repräsentation einer Instanz führt, die anschließend für verschiedene Klassifikationsaufgaben genutzt werden kann.

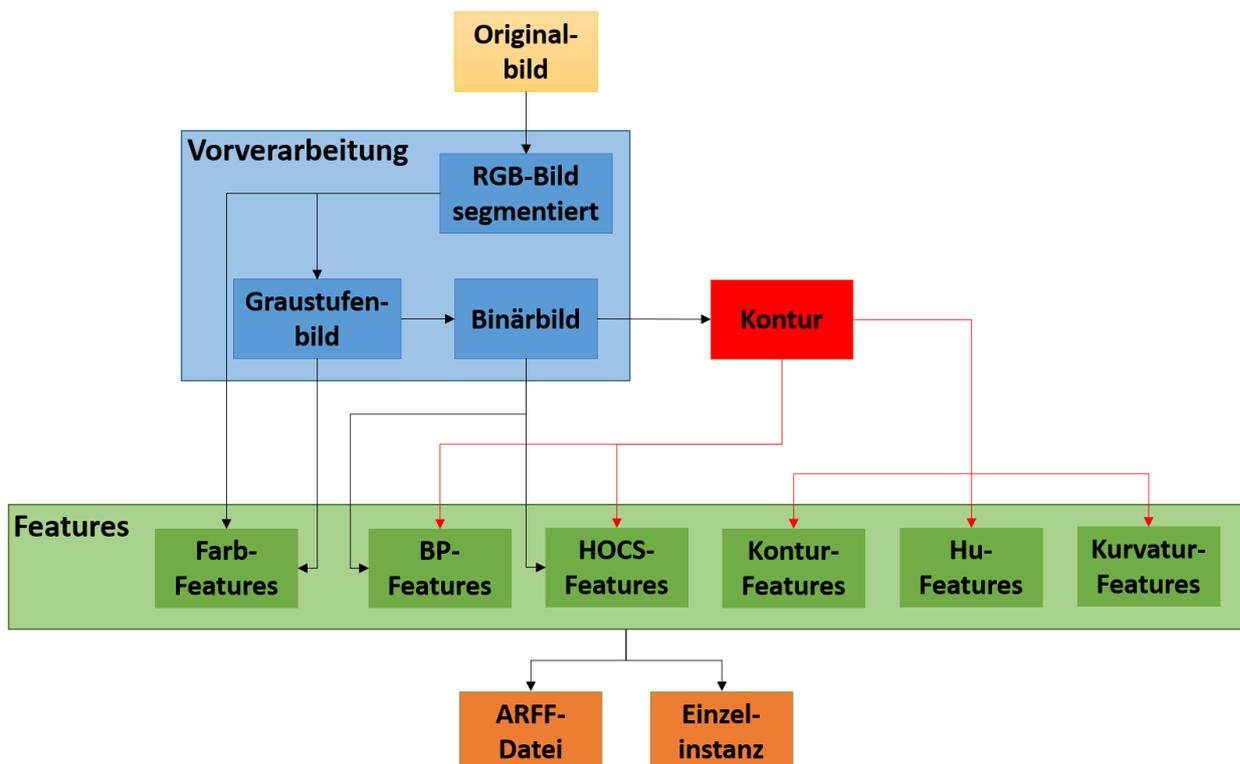


Abbildung 83: Übersicht über die einzelnen Vorverarbeitungs- und Feature-Extraktionsschritte.

Es ist zu erwähnen, dass in diesem Kapitel lediglich das prinzipielle Konzept und der grobe Ablauf im Vordergrund stehen sollen. Auf Einzelheiten und Besonderheiten bei der Implementierung wird bzw. wurde in den entsprechenden Kapiteln genauer eingegangen.

Zu Beginn der Verarbeitung steht lediglich ein Bild eines Blattes. Im ersten Schritt gilt es, dieses Blatt vom Hintergrund freizustellen, also zu segmentieren. Je nach Art der Aufnahme kann dies manuell erfolgen oder mittels einer der in Kapitel 4 vorgestellten Methoden automatisch. Liegt das zu verarbeitende Bild bereits in segmentierter Form vor, wie es beispielsweise bei einigen der verwendeten Datensets der Fall ist, entfällt dieser erste Schritt. Eine qualitativ hochwertige Segmentierung ist eine wichtige Voraussetzung für eine reibungslose und erfolgreiche Weiterverarbeitung des Bildes.

Im Anschluss wird das so vorsegmentierte Bild in die Graustufen- und anschließend in die Binärsicht umgewandelt. Ersteres dient dabei nicht nur als Zwischenverarbeitungsschritt, sondern wird im weiteren Verlauf noch zur Berechnung der Farb-Features benötigt.

Aus dem erhaltenen Binärbild kann das zentrale Element der Feature-Extraktion berechnet werden, die äußere Kontur des Blattes. Diese wird bei fünf von sechs Feature-Klassen als Eingabe benötigt.

Anschließend kann der eigentliche Prozess der Feature-Extraktion beginnen. Wie bereits erwähnt, spielt dabei die Kontur eine entscheidende Rolle. Lediglich bei der Berechnung der Farb-Features findet sie keine Verwendung. Dort genügen als Eingabeparameter das segmentierte RGB- und das Graustufenbild. Ersteres wird vor Bestimmung der Features noch in seine drei Farbkanäle R, G und B gesplittet.

Die Kontur-, Kurvatur- und Hu-Features lassen sich direkt aus der Kontur ableiten. Im Falle der Kontur-Features müssen lediglich zunächst das `MinimumBoundingRect` und die Konvexe Hülle berechnet werden. Bei den Kurvatur-Features genügt die Bestimmung des Schwerpunktes der Kontur. Die Hu-Features werden ohne weiteren Zwischenschritt aus der Kontur berechnet.

Zusätzlich zur Kontur benötigen die beiden letzten verbleibenden Feature-Klassen, die HOCS- und Binary Pattern-Features, das Binärbild als Eingabe. Bei deren Berechnung wird die Kontur Punkt für Punkt durchlaufen und die jeweilige Umgebung im Binärbild betrachtet.

Zum Abschluss können die extrahierten Features entweder in einer ARFF-Datei abgelegt oder als neue Einzelinstanz gespeichert werden. Ersteres dient zur späteren Verwendung, beispielsweise als Trainings- oder Vergleichsdatenset. Letzteres bietet sich an, wenn direkt eine Klassifikation der neu erstellten Instanz vorgenommen werden soll.

Abbildung 84 zeigt die einzelnen Schritte anhand eines Beispielblattes.

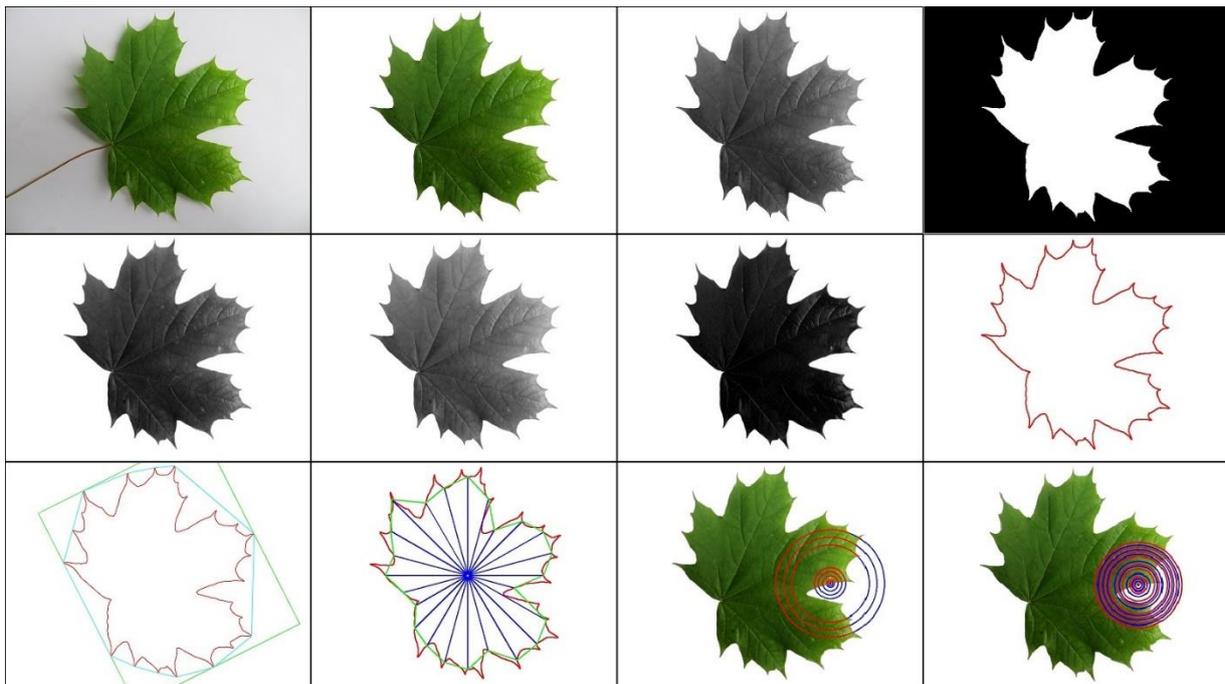


Abbildung 84: Verarbeitungsschritte am Beispiel eines Blattes. Jeweils von links nach rechts: Originalbild, segmentiertes RGB-Bild, Graustufenbild, Binärbild (Zeile 1); Rot-, Grün-, Blaukanal, äußere Kontur des Blattes (Zeile 2); Kontur mit MinimumBoundingRect (grün) und Konvexer Hülle (hellblau), schematische Darstellungen der Kontur-, Kurvatur-, HOCS-, Binary Pattern-Features (Zeile 3).

In der letzten Zeile fehlen die Farb- und Hu-Features, da diese sich offensichtlich nicht anschaulich darstellen lassen. Die eingezeichneten Radien bei den HOCS- und Binary Pattern-Features entsprechen denen des finalen Systems. Ihre Auswahl wird in Abschnitt 7.3 näher erläutert.

6.2 Allgemeine Herangehensweise

Die in dieser Arbeit gewählte Vorgehensweise unterscheidet sich stark von der Methode, die ein Mensch normalerweise anwenden würde, um eine Baumart anhand eines Blattes, beispielsweise mit Hilfe eines Bestimmungsbuches, zu klassifizieren. Es werden abstrakte, für den Menschen oft eher wenig anschauliche Blattmerkmale verwendet, die dann wiederum mit Hilfe eines Klassifikators zusammengeführt und evaluiert werden. Im Folgenden soll zunächst die menschliche Herangehensweise erläutert und anschließend deren Schwächen im Hinblick auf das in dieser Arbeit vorgestellte Klassifikationsszenario diskutiert werden.

6.2.1 Blattklassifikation durch einen Menschen

Die menschliche Herangehensweise entspricht im Allgemeinen eher einem Regelsystem. Es werden der Reihe nach einzelne Merkmale betrachtet. Im Normalfall wird dabei mit eher groben Merkmalen begonnen und anschließend Schritt für Schritt verfeinert. Jede weitere untersuchte Eigenschaft des Blattes führt dazu, dass die Menge der in Frage

kommenden Spezies immer weiter schrumpft, bis im Idealfall nur noch eine Spezies übrig ist.

Warum sich dieser Ansatz für eine automatische, computergestützte Klassifikation nicht eignet, soll im Folgenden mit Hilfe eines Beispiels erläutert werden. Dazu wird ein Ausschnitt aus einem Baum-Klassifikationsbuch betrachtet. Im Kosmos Naturführer „Welcher Baum ist das“ (vgl. [Mayer 2009]) wird zunächst zwischen Nadel- und Laubbäumen unterschieden. Die Laubbäume sind wiederum untergliedert in Spezies mit einfachen und zusammengesetzten Blättern. Erstere werden erneut unterteilt in ganzrandige, gekerbte bis gesägte und gelappte Blätter, letztere in gegen- und wechselständige. Ist es gelungen die Auswahl dementsprechend einzugrenzen, steht in den meisten Fällen dennoch eine Vielzahl von Spezies zur Auswahl, die weiterhin in Textform beschrieben sind. So wird beispielsweise die aus dieser Arbeit bereits bekannte Rot-Buche (*Fagus sylvatica*), Blätter einfach, gekerbt bis gesägt, wechselständig, wie folgt charakterisiert:

„Merkmale: 25-30m hoher Baum, dicht verzweigt, im freien Stand ausladend; Blätter eiförmig, 5-10cm lang, dunkelgrün glänzend, Rand schwach gekerbt oder gezähnt, teils ganzrandig; Blüte im April/Mai, männliche Blüten in Büscheln, weibliche in behaarten, später verholzenden Fruchtblättern, darin ab September je 2 dreikantige Nüsse (Bucheckern).“ [Mayer 2009]

6.2.2 Auftretende Probleme bei der computergestützten Klassifikation

Betrachtet man die aufgezählten Eigenschaften, wird schnell deutlich, warum sich bei der Klassifikation die Verwendung von abstrakten, numerischen Merkmalen anbietet. Zunächst fällt auf, dass ein Großteil der Eigenschaften sich nicht auf das Blatt des Baumes bezieht. Es wird die Höhe und das Aussehen des Habitus charakterisiert und Blüten bzw. Früchte beschrieben. Überdies werden Eigenschaften der Blätter genannt, die sich anhand eines Fotos nur schwer überprüfen lassen. So lässt sich beispielsweise die Größe eines Blattes auf einem Bild meist nicht exakt bestimmen. Zuverlässige Aussagen darüber, ob Blätter glänzen oder nicht, lassen sich ebenfalls nur sehr schwierig treffen.

Des Weiteren ist es unmöglich, alle benötigten Informationen auf nur einem Foto festzuhalten. Es werden Aufnahmen vom Habitus, den einzelnen Blättern, der Baumrinde und eventuellen Blüten oder Früchten benötigt. Dies erhöht den Nutzeraufwand stark und gestaltet die Klassifikation erheblich komplexer. Sicher ist die Verknüpfung mehrerer Aufnahmen und der darin enthaltenen Informationen eine interessante Herausforderung, jedoch ist der Ansatz in dieser Arbeit ein völlig anderer. Ziel ist es, Baumspezies ausschließlich anhand eines einzigen Fotos eines einzelnen Blattes zu identifizieren.

Dies reduziert die Anzahl verwendbarer und nützlicher Merkmale auf ein Minimum. Es bleiben lediglich die Blattform (einfach oder zusammengesetzt und anschließend rund, eiförmig etc.) und der Blattrand (glattrandig, gekerbt, gesägt, gezähnt, gebuchtet, gewellt oder gelappt). Bei letzterem ist zu erwähnen, dass selbst in Bestimmungsbüchern eine exakte Beschreibung des Blattrandes nur schwer möglich ist. Dies zeigt sich beispielsweise in obigem Zitat, bei dem der Blattrand als „schwach gekerbt, gezähnt, teils ganzrandig“ charakterisiert wird. Betrachtet man diese Merkmale und deren Ausprägungen und führt sich gleichzeitig die große Menge an existierenden Baumspezies vor Augen, wird schnell deutlich, dass ein starres regelbasiertes System, wie es in Bestimmungsbüchern Verwendung findet, nur wenig Sinn ergibt.

6.2.3 Verwendeter Ansatz

Aus den zuvor diskutierten Gründen wurde sich in dieser Arbeit dazu entschieden, auf einen Machine Learning-Ansatz mit zahlreichen abstrakten, numerischen Merkmalen zu setzen. Dies ermöglicht eine deutlich exaktere Charakterisierung der Blätter. So kann beispielsweise bei der Beschreibung der Beschaffenheit des Randes nicht nur zwischen groben Klassen unterschieden werden, sondern für jedes Blatt und somit jede Spezies individuell Tiefe und Frequenz eventueller Einkerbungen als Features dargestellt werden.

6.3 Handhabung zusammengesetzter Blätter

Wie in Abschnitt 2.4.2.1 beschrieben, wird in der Botanik zwischen einfachen und zusammengesetzten Blättern unterschieden. Zusammengesetzte Blätter bestehen dabei aus mehreren sogenannten Blättchen. Einige Beispiele aus den in dieser Arbeit verwendeten Datensets sind in Abbildung 85 zu sehen.



Abbildung 85: Drei Beispiele zusammengesetzter Blätter. Links: Gewöhnliche Rosskastanie (*Aesculus hippocastanum*), Mitte: Eberesche (*Sorbus aucuparia*), rechts: Schwedische Mehlbeere (*Sorbus hybrida*).

Im Folgenden sollen zunächst mögliche Schwierigkeiten bei der Arbeit mit zusammengesetzten Blättern vorgestellt werden. Anschließend wird erläutert, wie in dieser Arbeit damit umgegangen wird und warum andere Ansätze nicht in Frage kommen.

6.3.1 Problematik zusammengesetzter Blätter

Bei einer realen Anwendung können diese Blätter zu Schwierigkeiten führen, da z. B. ein Nutzer mit nur geringen Kenntnissen im Bereich der Botanik leicht einfache und zusammengesetzte Blätter verwechseln kann. Dies würde dazu führen, dass statt dem ganzen Blatt lediglich ein einzelnes Blättchen ausgewählt wird. Erwartet allerdings das Klassifikationssystem ein Foto des vollständigen Blattes, ist eine korrekte Bestimmung ausgeschlossen.

Ein weiteres Problem, das bei der Arbeit mit zusammengesetzten Blättern auftreten kann, wird in Abbildung 86 deutlich: Die Lage der Blättchen kann einen großen Einfluss auf die äußere Kontur des Blattes und somit auf die extrahierten Features haben. Je mehr Überlappungen auftreten, desto kompakter wirkt das Blatt.

Das Hauptproblem ist allerdings die erschwerte Aufnahme und Segmentierung zusammengesetzter Blätter. Gerade bei der Fotografie am Baum kann es sehr fordernd sein, ein gut verwertbares Bild zu erhalten. Dies ist zum einen durch das teils starke Durchhängen zusammengesetzter Blätter zu begründen. Zum anderen kann komplexer

Hintergrund, der zwischen den zahlreichen Blättchen zu sehen ist, zu großen Problemen bei der manuellen Segmentierung führen.

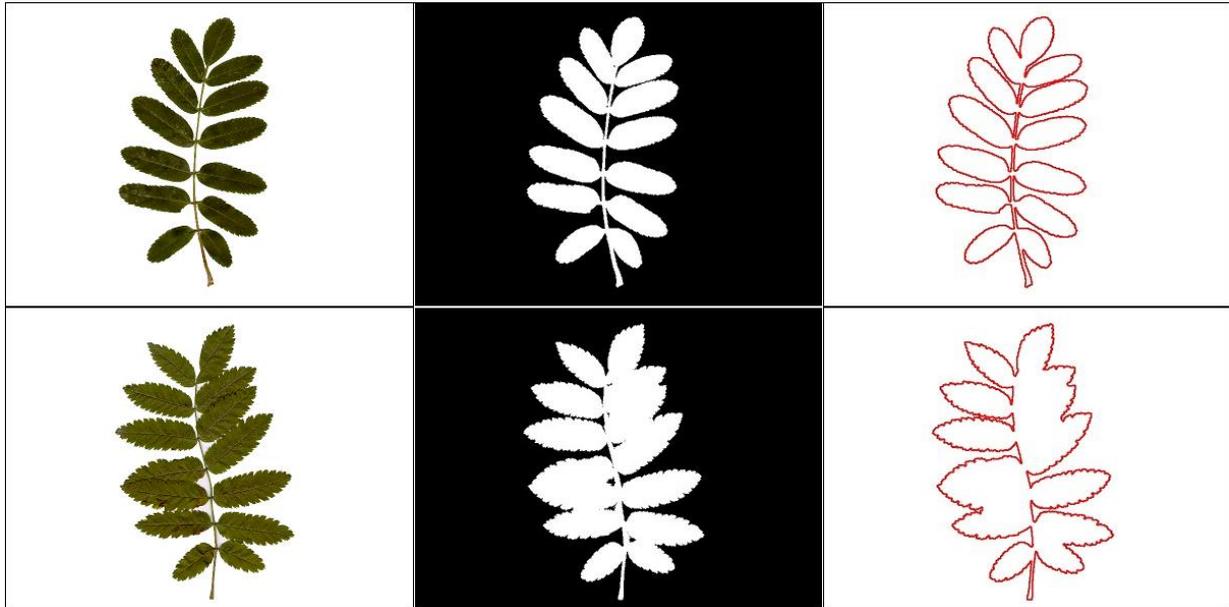


Abbildung 86: *Problematik durch überlappende Blättchen.*
 Links: Originalbild, Mitte: Binärdarstellung, rechts: extrahierte äußere Kontur.

6.3.2 Mögliche Ansätze

Im Folgenden sollen drei Ansätze vorgestellt und ihre Vor- und Nachteile kurz diskutiert werden.

6.3.2.1 Aufteilung derselben Spezies in zusammengesetzte Blätter und Blättchen

Ein möglicher zukünftiger Ansatz ist es, sowohl zusammengesetzte Blätter als Ganzes, als auch deren Blättchen als zwei separate Klassen zu behandeln und somit dem Nutzer freie Hand zu lassen, was das Eingabeformat angeht. Dieser könnte sich somit sogar zusätzliche Sicherheit verschaffen, indem er zunächst das Blatt als Ganzes und anschließend ein Blättchen bestimmen lässt. Sauber umgesetzt liefert diese Methode mit Sicherheit im Schnitt die besten Ergebnisse. Problematisch ist allerdings nach wie vor der große Einfluss der Lage der Blättchen.

6.3.2.2 Automatische Extraktion von Blättchen

Eine Weiterentwicklung des eben vorgestellten Ansatzes könnte bestmögliche Klassifikationsergebnisse mit maximalem Nutzungskomfort verbinden. Dazu muss bei Betrachtung eines Blattes durch das System zunächst entschieden werden, ob es sich um ein einfaches oder zusammengesetztes Blatt handelt. Ist letzteres der Fall, erfolgt automatisch eine Extraktion eines Blättchens und anschließend dessen Klassifikation. Zusätzlich kann eine Klassifikation des gesamten Blattes erfolgen.

Die Schwierigkeit bei diesem Ansatz liegt in der Verarbeitung des zusammengesetzten Blattes. Zum einen muss sichergestellt werden, dass Blätter höchst zuverlässig richtig als einfaches oder zusammengesetztes Blatt klassifiziert werden. In den meisten Fällen sollte

dies problemlos möglich sein. Allerdings besteht gerade bei Blättern mit starken Überlappungen der Blättchen (vgl. Abbildung 85 (rechts) auf Seite 99) die Gefahr der Verwechslung mit einer anderen Spezies, die über einfache Blätter verfügt.

Zum anderen muss bei zusammengesetzten Blättern die Extraktion eines einzelnen Blättchens sauber und zuverlässig erfolgen. Ein möglicher Ansatz dafür ist ein modifizierter Ansatz der Stieldetektion und -entfernung. Wendet man die dort eingesetzte Distanztransformation auf ein zusammengesetztes Blatt an, werden die freiliegenden Stellen des Stiels zwischen den Blättchen ebenfalls hervorgehoben. Diesen Teil des Blattes bezeichnet man als Rhachis. Nach dessen Entfernung liegen die Blättchen frei und können mittels Konturdetektion extrahiert werden.

Dieses Vorgehen dürfte in den meisten Fällen gute Ergebnisse liefern. Allerdings ergeben sich einige Schwierigkeiten: Zunächst wird mindestens ein freiliegendes Blättchen benötigt, also eines ohne Überlappungen, was nicht immer der Fall ist. Zum anderen wird ein Auswahlkriterium benötigt, nach dem ein Blättchen ausgewählt wird, sollte es mehrere Kandidaten geben. Es könnte beispielsweise das sein, dessen Kontur am deutlichsten einer Ellipse gleicht. Während diese Herangehensweise meist das für die Klassifikation am besten geeignete Blättchen liefern sollte, wäre das z. B. bei vielen Kastanienarten (vgl. Abbildung 85 (links) auf Seite 99) nicht der Fall. Hier würde vermutlich eines der kleineren Blätter ausgewählt werden, obwohl die charakteristische Form, nämlich an einem Ende dünn und spitz zulaufend und am anderen Ende stark verdickt mit anschließender Spitze, gerade bei den großen Blättern auftritt.

6.3.2.3 Behandlung der Blättchen als einfache Blätter

Aufgrund der beschriebenen Problematik der beiden vorherigen Ansätze, werden in dieser Arbeit keine zusammengesetzten Blätter behandelt. Das System erwartet als Eingabe entweder einfache Blätter oder ein einzelnes Blättchen. Diese Herangehensweise verzichtet dabei auf die Verwendung einer wichtigen Information, stellt aber sicher, dass jeder Nutzer ohne zusätzliches Wissen das System optimal verwendet. Eine konstant zuverlässige Klassifikation wird allein dadurch zwar nicht garantiert. Es wird allerdings sichergestellt, dass grobe Fehler bei der Eingabe weitestgehend vermieden werden können.

6.4 Fehlende Klassifikation von Nadelbäumen

In dieser Arbeit wird sich ausschließlich auf die Klassifikation von Laubbäumen beschränkt. Dafür gibt es zahlreiche Gründe:

- Der Aufbau und die Eigenschaften von Blättern und Nadeln unterscheiden sich sehr stark, sodass zur Bestimmung von Nadelbäumen ein Großteil der vorgestellten Features verworfen werden und ein völlig neuer Ansatz erstellt werden müsste, was den Umfang dieser Arbeit sprengen würde.
- Zur erfolgreichen Klassifikation von Nadeln werden sehr hoch auflösende Bilder benötigt (vgl. [Fiel 2011]). Allerdings ist der Grundgedanke dieser Arbeit, eine zuverlässige Bestimmung der Klasse auch mit Hilfe einer durchschnittlichen Handykamera zu gewährleisten.
- Nach Wissen dieser Arbeit gibt es keinerlei öffentlich zugängliche Datensets, die Bilder von Nadeln enthalten. Eine Erstellung eines eigenen Sets ist zwar möglich, wäre aber mit hohem Aufwand verbunden und würde keinerlei Vergleichbarkeit mit anderen Systemen ermöglichen.
- Aufbauend auf dem letzten Punkt ist zu erwähnen, dass nach Kenntnisstand dieser Arbeit die Klassifikation von Nadeln noch ganz am Anfang steht. Es existiert lediglich eine Veröffentlichung (vgl. [Fiel 2011]), die diese Thematik kurz behandelt.

Aus oben genannten Gründen werden Nadelbäume in dieser Arbeit nicht behandelt.

7 Evaluation

In diesem Kapitel wird das finale System Schritt für Schritt erstellt und ausführlich evaluiert. Dabei steht neben den verwendeten Klassifikatoren, Features und deren Parametern vor allem der Unterschied zwischen der Klassifikation innerhalb eines Datensets und der Datenset-übergreifenden Klassifikation im Vordergrund.

Des Weiteren wird der Teil des BLD, der die auf Papier fotografierten Blätter enthält, mit „Papier“, der mit den am Baum erstellten Aufnahmen mit „Baum“ bezeichnet.

7.1 Allgemeines Vorgehen

Bei der Evaluation von Klassifikationssystemen spielt das sogenannte *Overfitting* eine wichtige Rolle. Darunter versteht man eine Überanpassung des Klassifikators an den Trainingsdatensatz. Das führt auf diesem zwar häufig zu sehr guten Ergebnissen, die jedoch auf dem Testdatensatz nicht immer repliziert werden können. Um aussagekräftige Ergebnisse zu erhalten, ist es daher wichtig, z. B. die verwendeten Parameter nicht auf jedem Datensatz neu zu optimieren.

In dieser Arbeit wurde folgendes Vorgehen gewählt: Alle Parameter, sowohl die der Features als auch der Klassifikatoren, werden stets nur auf einem Datensatz optimiert und dann unverändert für die Tests auf anderen Datensets übernommen. Zur Optimierung wird, wenn nicht ausdrücklich anders erwähnt, immer das Flavia-Datenset verwendet.

7.2 Verwendete Klassifikatoren

In dieser Arbeit kommen der K-Nearest-Neighbours-Klassifikator und die Support Vector Machine zum Einsatz. Die SVM stellt für viele Klassifikationsaufgaben den State of the Art dar. Der KNN bietet eine einfache Klassifikation, hohe Anschaulichkeit und lieferte bei vergleichbaren Systemen gute Ergebnisse (vgl. [Kumar 2012]). Im nächsten Abschnitt sollen die beiden Klassifikatoren und ihre Parameter näher erläutert werden.

7.2.1 Der K-Nearest-Neighbours-Klassifikator

In dieser Arbeit wird die KNN-Implementierung der Weka Bibliothek verwendet. Im Folgenden soll kurz auf die gewählte Distanzfunktion und den Parameter K eingegangen werden.

7.2.1.1 Distanzfunktion

Weka stellt standardmäßig vier Distanzfunktionen zur Verfügung, um die Distanz zwischen zwei Datenpunkten p und q zu berechnen:

Manhattan Distanz:

$$d(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (7.1)$$

Euklidische Distanz:

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (7.2)$$

Chebyshev Distanz:

$$d(p, q) = \max_i (|p_i - q_i|) \quad (7.3)$$

Minkowski Distanz p -ter Ordnung:

$$d(p, q) = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{\frac{1}{p}} \quad (7.4)$$

[Wiki: {Chebyshev, Euklid, Manhattan, Minkowski}]

Diese vier Distanzfunktionen wurden auf dem Flavia-Datensatz im Verfahren 1 x all und mit $K=1$ getestet. Die Ergebnisse sind in Tabelle 4 zu sehen.

Tabelle 4: Vergleich verschiedener Distanzfunktionen beim KNN-Klassifikator.

Flavia; 1 x all; 1NN	
Distanz	Ergebnis
Manhattan	99,42%
Euklidische	99,69%
Chebyshev	96,80%
Minkowski	99,69%

Die Minkowski Distanz wurde mit $p=2$ berechnet. Eine weitere Erhöhung von p führt zu schlechteren Ergebnissen. Das beste Ergebnis konnte mit der Euklidischen und der Minkowski Distanz erzielt werden. Die Manhattan Metrik fällt leicht ab, die Chebyshev Distanz sogar deutlich. Im weiteren Verlauf dieser Arbeit wird stets die Euklidische Metrik verwendet, da sie zusammen mit der Minkowski Metrik das beste Ergebnis liefert und nicht von einem weiteren Parameter abhängt.

7.2.1.2 Der Parameter K

Wie bereits in Abschnitt 2.2.1.2 beschrieben, gibt der Parameter K beim KNN-Klassifikator die Anzahl der nächsten Nachbarn an, die bei der Klassifikation berücksichtigt werden. Ist $K=1$, wird die unbekannte Instanz demnach der Klasse zugeordnet, deren Vertreter die minimale Distanz aufweist. Für $K>1$ ist es sinnvoll, die Distanzen der einzelnen nächsten Nachbarn in die Klassifikation mit einzubeziehen. Weka bietet hierfür die bereits vorgestellten Methoden $1/dist$ und $1-dist$ an.

Zur Bestimmung der besten Werte für K wurden auf dem Flavia-Datensatz Testläufe mit insgesamt sechs verschiedenen K durchgeführt. Zum Vergleich diente das MEW-Datenset.

Tabelle 5 zeigt die Ergebnisse.

Tabelle 5: Einfluss des Parameters K und der Gewichtungsfunktion.

Flavia; 1 x all; 1NN				MEW; 1 x all; 1NN			
K	-	1/dist	1-dist	K	-	1/dist	1-dist
1	99,69%	99,69%	99,69%	1	95,66%	95,66%	95,66%
2	99,11%	99,69%	99,69%	2	94,57%	95,66%	95,66%
3	99,42%	99,42%	99,42%	3	95,20%	95,60%	95,60%
4	98,95%	99,42%	99,42%	4	95,08%	95,74%	95,73%
5	98,90%	98,90%	98,90%	5	95,16%	95,38%	95,38%
10	98,37%	98,95%	98,74%	10	94,17%	94,67%	94,60%
20	98,06%	98,32%	98,27%	20	91,92%	92,66%	92,35%

Offensichtlich spielt die verwendete Gewichtungsfunktion bei $K=1$ keine Rolle. Ebenfalls klar ist, dass eine Gewichtung bei $K=2$ effektiv einer Klassifikation mit $K=1$ entspricht, da der nächste Nachbar am stärksten gewichtet wird und somit alleinig ausschlaggebend für die Zuweisung zu einer Klasse ist.

Beim Flavia-Datensatz liefert $K=1$ das beste Ergebnis. In den meisten Fällen verschlechtern sich die erzielten Klassifikationsergebnisse mit steigendem K . Ein etwas anderes Bild zeigt sich beim MEW-Datensatz. Hier konnte die höchste Klassifikationsrate mit $K=4$ und der Gewichtungsfunktion $1/dist$ erzielt werden.

Allgemein lässt sich sagen, dass für $K>1$ der Klassifikator unter Verwendung einer Distanzfunktion stets bessere Ergebnisse liefert als ohne. Des Weiteren ist die Klassifikationsrate mit $1/dist$ immer mindestens genauso hoch wie bei $1-dist$ und übertrifft sie bei Werten für K von 10 oder höher sogar.

Aufgrund der Vorgabe, zur Optimierung der Werte lediglich das Flavia-Datensatz zu benutzen, wird im weiteren Verlauf dieser Arbeit, soweit nicht ausdrücklich anders angegeben, stets $K=1$ gewählt. Allerdings ist aus Gründen der Robustheit und der Varianzreduktion, gerade bei einer realen Anwendung, die Verwendung eines höheren K durchaus möglich. Diese Problematik wird im weiteren Verlauf dieser Arbeit noch behandelt.

7.2.2 Die Support Vector Machine

Weka erlaubt es sowohl in der GUI als auch im Java-Code, die SVM Bibliothek LibSVM direkt zu integrieren. Im Folgenden sollen zunächst die zur Verfügung stehenden Kernel vorgestellt werden. Anschließend erfolgt die Normalisierung der einzelnen Features und die Optimierung der Parameter.

7.2.2.1 Wahl der Kernel-Funktion

Weka und LibSVM unterstützen standardmäßig folgende vier Kernelfunktionen:

Linear:

$$K(x_i, x_j) = x_i^T x_j \quad (7.5)$$

Polynomial:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad (7.6)$$

Radial Basis Function (RBF):

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (7.7)$$

Sigmoid:

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (7.8)$$

γ , r und d bezeichnen dabei die Kernelparameter.

Chih-Wei et al. empfehlen für die meisten Anwendungen den RBF-Kernel. Dieser bildet Beispiele nicht-linear in eine höhere Dimension ab und kann somit, anders als der lineare Kernel, nicht-lineare Beziehungen zwischen Attributen und Klassenzugehörigkeit handhaben. Zudem lässt sich mit dem RBF-Kernel im Normalfall ein weniger komplexes Modell erstellen als mit dem polynomialen Kernel. Die Verwendung des RBF-Kernels bietet sich nicht an, wenn die Anzahl der Features sehr hoch ist. Hier sollte ein linearer Kernel verwendet werden, was allerdings in dieser Arbeit nicht der Fall ist. [Chih-Wei 2003]

7.2.2.2 Normalisierung der Attributwerte

Um brauchbare Klassifikationsergebnisse zu erhalten, müssen die Attribute zunächst skaliert werden. So wird verhindert, dass Attribute, die oft große Werte annehmen, andere Attribute mit deutlich kleineren Werten dominieren. In dieser Arbeit werden die Attribute linear auf den Bereich $[0, 1]$ skaliert. [Chih-Wei 2003]

7.2.2.3 Optimierung der Parameter γ und C

Gerade bei mehreren Parametern kann deren manuelle Optimierung oft sehr schwierig und zeitaufwendig sein. Aus diesem Grund bietet Weka eine Funktion mit dem Namen Grid-Search (englisch *grid*: Gitter, Raster) an. Sie ermöglicht es, für beide Parameter γ und C je einen zu überprüfenden Wertebereich und eine Schrittweite anzugeben. Wird beispielsweise für γ der Bereich 0,2 bis 0,5 mit der Schrittweite 0,05 gewählt, so werden die Werte 0,20; 0,25; ... ; 0,45; 0,50 evaluiert. Diese Evaluation erfolgt mit Hilfe einer Cross-Validation, um Overfitting weitestgehend zu vermeiden und zuverlässige Ergebnisse zu generieren.

Zur Bestimmung der besten Parameter werden verschiedene Wertepaare für γ und C gewählt, getestet und das Paar übernommen, das die besten Ergebnisse bei der Cross-Validation liefert. Die Grid-Search nutzt einen naiven Ansatz, bei dem gegebenenfalls eine große Anzahl von Parameterkombinationen getestet werden muss. Dies erfordert einen hohen Zeitaufwand. Der Vorteil ist allerdings, dass nach einmaliger Angabe der Suchkriterien kein weiteres menschliches Eingreifen mehr notwendig ist, bis der Algorithmus die optimalen Werte bestimmt hat.

Als zu testende Werte haben sich exponentiell steigende Sequenzen wie beispielsweise $C = 2^{-5}, 2^{-3}, \dots, 2^{15}$ und $\gamma = 2^{15}, 2^{-13}, \dots, 2^3$ bewährt.

In dieser Arbeit wurde die Grid-Search auf dem Flavia-Datenset durchgeführt. Der beste Klassifikationswert wurde mit den Parametern $\gamma = 0,335$ und $C = 20$ und unter der Verwendung einer 10-Fold-Cross-Validation erzielt. Dieser betrug 99,42%.

7.2.3 Vergleich der beiden Klassifikatoren

Die beiden vorgestellten Klassifikatoren wurden auf dem Flavia-Datenset verglichen. Als Evaluationsmethode wurde eine 10-Fold-Cross-Validation gewählt, da der 1 x all-Standardansatz sich für die Klassifikation mit einer SVM nicht eignet. Es wurden jeweils zehn Läufe durchgeführt, die Ergebnisse gemittelt und der Standardfehler berechnet. Die erhaltenen Ergebnisse sind in Tabelle 6 zu sehen.

Tabelle 6: Klassifikationsergebnisse einer 10-Fold-Cross-Validation unter Verwendung des 1NN-Ansatzes und einer SVM auf dem Flavia-Datensatz.

Flavia; 10FCV	
Klassifikator	Ergebnis
1NN	99,61%±0,02%
SVM	99,41%±0,02%

Der 1NN-Ansatz liefert ein etwas besseres Ergebnis. Dies reicht jedoch bei Weitem nicht aus, um mit Sicherheit sagen zu können, dass der 1NN-Klassifikator für die in dieser Arbeit geforderten Klassifikationsaufgaben besser geeignet ist als die SVM. Dennoch wird im weiteren Verlauf der Evaluation zunächst der 1NN-Ansatz verwendet, da dieser, abgesehen von der minimal besseren Klassifikationsrate, einige wichtige Vorteile besitzt:

Zum einen bietet der 1NN-Klassifikator eine extrem hohe Erklärbarkeit. Wird beispielsweise eine Instanz, also ein einzelnes Blatt, falsch klassifiziert, steht bei der Verwendung einer SVM lediglich ein komplexes und sehr unanschauliches Modell zur Verfügung, um den Fehler näher zu untersuchen. Beim 1NN-Ansatz hingegen kann nicht nur der nächste Nachbar, also die Instanz, die für die Fehlklassifikation verantwortlich war, betrachtet werden, sondern auch der nächste Vertreter der korrekten Spezies. Diese können anschließend wiederum mit bloßem Auge oder unter Einbeziehung ihrer Features verglichen werden.

Zum anderen ermöglicht der 1NN-Ansatz problemlos die Verwendung von 1 x all als Evaluationsmethode. Dies liefert beispielsweise bei der folgenden Optimierung der Feature-Parameter zuverlässige, varianzfreie Ergebnisse nach nur einem Durchlauf. Mit einer SVM macht der 1 x all-Ansatz in den allermeisten Fällen keinen Sinn, da damit z. B. beim Flavia-Datensatz 1907mal ein neues Modell aus 1906 Instanzen berechnet werden müsste, um anschließend eine einzige Instanz zu klassifizieren. Der Zeitaufwand wäre viel zu hoch.

Es ist außerdem stark davon auszugehen, dass die mittels des 1NN-Klassifikators ermittelten Optimierungen für die Feature-Parameter auch unter der Verwendung einer SVM gelten.

Bei der Evaluation des optimierten Gesamtsystems in Abschnitt 7.5 wird neben dem 1NN-Ansatz auch die SVM verwendet werden.

7.3 Optimierung der Feature-Parameter

Im folgenden Abschnitt sollen die Parameter für die verschiedenen Feature-Klassen evaluiert werden. Dies betrifft den Winkelabstand der Kurvatur-Features, die Radien und Anzahl der Bins der HOCS-Features und die Radien und Anzahl der Punkte der Binary Pattern-Features.

Die Optimierung des Winkelabstandes erfolgt unter ausschließlicher Verwendung der Kontur-Features, da ihr Einfluss auf das Gesamtsystem nur sehr gering ist. Die Parameter HOCS- und Binary Pattern-Features werden unter Verwendung aller Feature-Klassen optimiert. Grund dafür ist, dass aufgrund der Überschneidungen bei den in den Features gespeicherten Informationen eine Verknüpfung der besten Teilsysteme nicht zwingend das beste Gesamtsystem ergeben muss.

Des Weiteren ist zu erwähnen, dass gerade bei den HOCS- und Binary Pattern-Features eine erschöpfende Suche nach den besten Parametern nicht möglich ist, da theoretisch jede beliebige Kombination an Parametern getestet werden müsste. Es wurde daher versucht, durch geschicktes Hinzufügen, Entfernen und Abändern von Parametern das bestmögliche System zu erreichen. Dabei wurde sich zum einen eng an den bisher erzielten Ergebnissen orientiert und zum anderen gesunder Menschenverstand eingesetzt. Überdies wurden tendenziell eher „runde“ Parameter verwendet, also ganzzahlige Vielfache von 5 bzw. 10.

7.3.1 Kurvatur-Features

Bei den Kurvatur-Features bestimmt der Winkelabstand die Anzahl der verwendeten Konturpunkte. Offensichtlich bieten sich als Parameter Werte an, die 360° ganzzahlig teilen. Mit Hilfe des Flavia-Datensatzes wurden verschiedene Parameter evaluiert. Die Ergebnisse sind in Tabelle 7 zu sehen und wurden unter alleiniger Verwendung der Kurvatur-Features erzielt.

Tabelle 7: Einfluss des Winkelabstandes bei den Kurvatur-Features.

Flavia; 1 x all; 1NN		
Winkelabstand	# Konturpunkte	Ergebnis
5°	71	57,79%
2°	179	61,77%
1°	359	61,98%
-	50%	65,34%
-	100%	72,78%

Es fällt auf, dass sich das Klassifikationsergebnis mit steigender Punktzahl verbessert. Das mit Abstand beste Ergebnis wird unter Verwendung aller Konturpunkte erzielt. Dies war zwar zu erwarten, aber nicht unbedingt als sicher anzunehmen, da die Abstraktion der Kontur und somit die Reduktion auf weniger Punkte auch Vorteile haben kann. Im weiteren Verlauf dieser Arbeit werden stets alle Konturpunkte zur Bildung der Kurvatur-Features verwendet.

7.3.2 HOCS-Features

Bei den HOCS-Features gilt es zwei Parameterklassen zu optimieren: zum einen die verwendeten Radien und zum anderen die Anzahl der Bins, die in den Histogrammen zum Einsatz kommen. Zuvor muss allerdings festgelegt werden, auf welche Fläche die Blattkonturen normiert werden, bevor die Features extrahiert werden können.

7.3.2.1 Fläche der Blattkonturen

Um Vergleichbarkeit zu gewährleisten müssen alle Blattkonturen vor der Extraktion der HOCS-Features auf eine einheitliche Fläche normiert werden. In dieser Arbeit wurde eine Fläche von 30.000 Pixeln verwendet. Zur Festlegung dieses Wertes wurden zunächst die Flächen und das Länge-zu-Breite-Verhältnis aller Blätter des Flavia-Datensets betrachtet. Ziel war es, die Spezies zu finden, die am stärksten elongiert ist und die geringste Fläche besitzt. Diese Spezies ist *Cedrus deodara*. Drei zufällig ausgewählte Blätter sind in Abbildung 87 zu sehen.

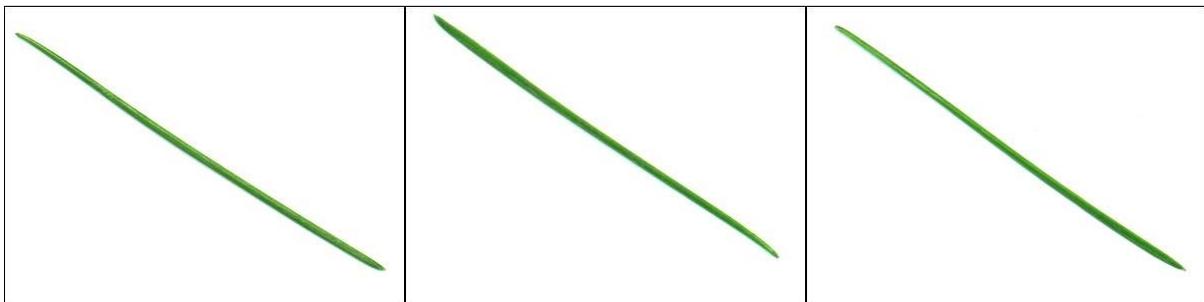


Abbildung 87: Drei Vertreter der Spezies *Cedrus deodara* aus dem Flavia-Datenset.

Das Auffinden dieser Spezies war deswegen vonnöten, da bei der Normierung auf eine einheitliche Fläche sichergestellt werden muss, dass das normierte Blatt vollständig auf dem neu entstandenen Bild zu sehen ist. Dazu wurde die geringste Fläche der Vertreter der Spezies *Cedrus deodara* ausgewählt und für den Fall, dass in einem anderen Datensatz noch stärker elongierte Spezies auftreten, mit einem Puffer versehen. Dies lieferte den bereits erwähnten Wert von 30.000 Pixeln.

Natürlich wäre es möglich gewesen, zur internen Verarbeitung größere Bilder zu verwenden und die Blätter auf eine entsprechende Fläche zu normieren. Allerdings führte eine Vergrößerung der Fläche und eine damit verbundene Anpassung der verwendeten Radien nicht nur zu deutlich längeren Berechnungszeiten, sondern auch zu einer Verschlechterung der Klassifikationsergebnisse.

7.3.2.2 Verwendete Radien

Die Auswahl der aussagekräftigsten Radien für die Bestimmung der HOCS-Features spielt eine wichtige Rolle, da jeder Radiusbereich verschiedene Informationen der Kontur repräsentiert. Kleine Radien beschreiben vor allem die feine Beschaffenheit des Randes, während große die grobe Form eines Blattes wiedergeben. Es muss daher sichergestellt werden, dass kein wichtiger Radiusbereich ausgelassen wird, da sonst Informationen verloren gehen können. Andererseits dürfen auch nicht zu viele Radien gewählt werden, deren redundante Information zu einer Verschlechterung des Klassifikationsergebnisses führen kann.

Zunächst soll gezeigt werden, wie groß die Aussagekraft der HOCS-Features ist. Dazu wurde die Klassifikation jeweils nur mit einem Radius durchgeführt. Es wurden keine weiteren Feature-Klassen verwendet. In Tabelle 8 sind die Ergebnisse zusammengefasst.

Tabelle 8: *Klassifikationsergebnisse unter Verwendung der HOCS-Features eines einzelnen Radius.*

Flavia; 1 x all; 1NN								
Radius	1	2	3	4	5	10	15	20
Ergebnis	54,06%	65,55%	75,41%	75,62%	79,65%	85,63%	86,84%	86,84%
Radius	25	30	35	40	45	50	60	70
Ergebnis	84,84%	85,63%	84,90%	84,85%	84,53%	83,80%	82,85%	85,11%

Auffällig ist, dass die Ergebnisse bei sehr niedrigen Radien noch stark von denen bei mittleren und hohen Radien abweichen. Der Grund dafür ist, dass Radien von 1 oder 2 Pixeln noch nicht in der Lage sind, die feine Kontur eines Blattes ausreichend zu beschreiben, und keinerlei Informationen über die Blattform als Ganzes geben. Ab einem Radius von etwa 10 bis 70 Pixeln sind die Klassifikationsergebnisse konstant hoch. Das Maximum wird mit fast 87% bei Radien von 15 und 20 erreicht. Zum Vergleich: Ein System bestehend aus allen Kontur-, Kurvatur- und Hu-Features erreicht lediglich knapp über 89%.

Der exakte Ablauf des Auswahlprozesses der Radien soll an dieser Stelle nicht näher vorgestellt werden. Im Prinzip wurden sukzessive Radien, die das Ergebnis verbesserten, hinzugefügt und Radien, die das Ergebnis verschlechterten, entfernt. Dies führte zu einer Menge von acht Radien mit 3, 5, 10, 15, 20, 50, 60, 70 Pixeln. Zur besseren Einordnung sind die verwendeten Radien in Abbildung 88 dargestellt.

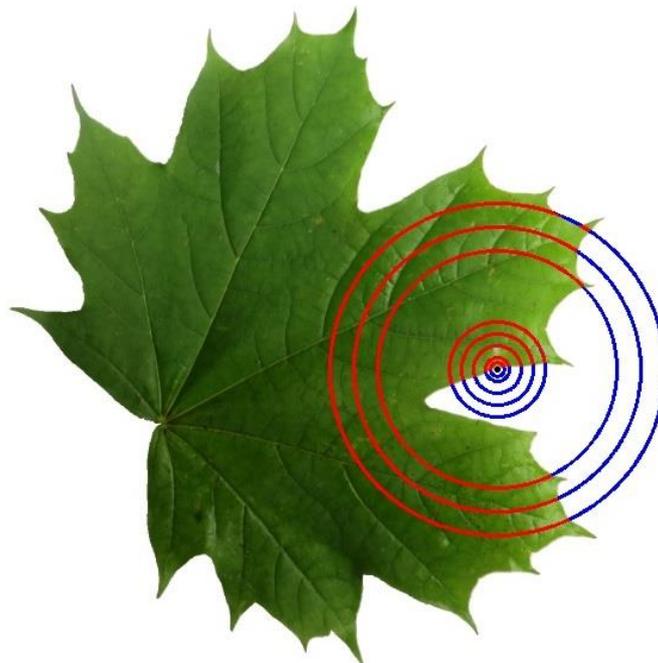


Abbildung 88: *Zur HOCS-Feature-Gewinnung verwendete Radien.*

Sofort fällt die große Lücke im mittleren Bereich der Radien auf. Im Test erwiesen sich z. B. Radien im Bereich von 25 bis 55 Pixeln nicht nur als nutzlos, sondern sogar als schädlich, da sie das Klassifikationsergebnis verschlechterten.

Im weiteren Verlauf werden für die HOCS-Features stets die eben vorgestellten Radien verwendet. Es ist allerdings nicht auszuschließen, dass andere Radien auf weiteren, speziell auf größeren Datensets bessere Ergebnisse erzielen würden.

7.3.2.3 Anzahl Bins

Bei der Darstellung der HOCS-Features in Histogramm-Form spielt die Anzahl der Bins eine wichtige Rolle. Sie gibt an, wie stark die Werte in den Histogrammen zusammengefasst werden. Bei der Festlegung der Bin-Anzahl ist es nötig zu beachten, dass eine zu hohe Anzahl zu einer Überanpassung an den Datensatz führt und somit die Anpassbarkeit des Systems eingeschränkt wird, eine zu niedrige führt dagegen zum Informationsverlust.

Zur Untersuchung der besten Bin-Breite wurden die im vorherigen Abschnitt bestimmten aussagekräftigsten Radien zusammen mit allen anderen Feature-Klassen, mit Ausnahme der Binary Patterns, verwendet. Die Ergebnisse sind in Tabelle 9 zu sehen. Dabei bedeutet ein Wert von 10 für die Anzahl der Bins, dass die Werte für den Prozentsatz des Bogens und der Fläche innerhalb der Kontur wie folgt zusammengefasst werden: Bin 1: [0, 0.1], Bin 2: [0.1, 0.2], ... Bin 10: [0.9, 1].

Tabelle 9: Einfluss der bei den HOCS-Features verwendeten Bins.

Flavia; 1 x all; 1NN						
# Bins	5	6	7	8	9	10
Ergebnis	98,43%	98,58%	98,58%	98,84%	98,69%	98,95%
# Bins	11	12	13	14	15	20
Ergebnis	98,58%	98,85%	98,79%	98,90%	98,85%	98,74%

In dem untersuchten Bereich von 5 bis 20 Bins pro Histogramm variieren die erzielten Klassifikationsergebnisse nur wenig. Das beste Ergebnis wurde mit 98,95% unter Verwendung von 10 Bins erzielt. Dies scheint einen guten Kompromiss zwischen Überanpassung und Informationsverlust darzustellen. Ähnlich wie bei den Radien ist es auch hier nicht auszuschließen, dass sich zur Klassifikation einer größeren Menge von Spezies eine andere, größere Anzahl von Bins besser eignen würde.

7.3.3 Binary Pattern-Features

Genau wie die HOCS-Features verfügen auch die Binary Patterns über zwei zu optimierende Parameterklassen. Wiederum gilt es, zunächst die Radien möglichst gut aufeinander abzustimmen. Anschließend muss geklärt werden, wie viele Punkte pro Radius betrachtet werden sollen. Die Anzahl der Punkte hat direkten Einfluss auf die Anzahl der Bins, die das Histogramm des jeweiligen Radius enthält. Die Anzahl der Bins ist stets um 2 höher als die Anzahl der Punkte des jeweiligen Radius. Dies ist dadurch begründet, dass noch je ein weiterer Bin für die nicht uniformen Patterns und für die Patterns, die vollständig aus Nullen bestehen, zur Verfügung gestellt werden muss.

Die Extraktion der Binary Patterns erfolgt auf denselben Bildern, die zuvor für die HOCS-Features verwendet wurden, d. h. alle Blattkonturen sind auf eine einheitliche Fläche von 30.000 Pixeln normiert.

7.3.3.1 Verwendete Radien

Die Problematik beim Auswählen der Radien ist dieselbe wie bei den HOCS-Features. Zu wenige Radien führen zu Informationsverlust, zu viele führen zu Redundanzen und zu Überanpassung. Auch das Vorgehen ist dasselbe.

Zunächst soll der Einfluss einzelner Radien, ohne die Verwendung zusätzlicher Feature-Klassen, auf das Klassifikationsergebnis getestet werden. Anders als bei den HOCS-Features wurde hier bereits die finale Auswahl an Radien verwendet. Diese sind in Abbildung 89 zu sehen. Die Kreise wurden lediglich aus Gründen der Übersicht abwechselnd rot und blau gefärbt.



Abbildung 89: Zur Binary Pattern-Feature-Gewinnung verwendete Radien.

Der Auswahlprozess erfolgte analog zu dem der HOCS-Features. Die Ergebnisse können in Tabelle 10 abgelesen werden. Zur Klassifikation wurden acht Punkte pro Radius verwendet. Diese Wahl wird im nächsten Abschnitt näher erläutert.

Tabelle 10: Klassifikationsergebnisse unter Verwendung der Binary Pattern-Features eines einzelnen Radius.

Flavia; 1 x all; 1NN						
Radius	1	4	6	7	9	12
Ergebnis	51,28%	58,00%	64,39%	66,28%	64,60%	65,01%
Radius	17	20	22	25	30	32
Ergebnis	69,11%	71,00%	63,61%	63,61%	59,99%	56,63%
Radius	35	37	40	42	45	
Ergebnis	50,29%	50,55%	48,87%	47,67%	44,47%	

Es ist augenfällig, dass die erzielten Ergebnisse jeweils deutlich niedriger ausfallen als jene, die mit Hilfe der HOCS-Features erzielt werden konnten (vgl. Tabelle 9). Dort betrug der maximale Wert 86,84% und wurde bei den Radien der Größe 15 und 20 Pixeln erzielt. Bei den Binary Patterns liegt der Bestwert lediglich bei 71,00% und trat ebenfalls bei einem Radius von 20 auf.

Des Weiteren fällt auf, dass die Ergebnisse ab einem Radius von etwa 35 Pixeln deutlich abfallen, während die HOCS-Features bei Radien zwischen 10 und 70 Pixeln relativ konstante Ergebnisse lieferten. Bei den Binary Patterns erwiesen sich Radien von mehr als 45 Pixeln für das Gesamtsystem nicht mehr als nützlich und wurden demnach entfernt.

7.3.3.2 Anzahl Punkte pro Radius

Im vorherigen Abschnitt wurde für jeden Radius dieselbe Anzahl an Punkten, nämlich 8, verwendet. Diese Wahl mag fragwürdig erscheinen, da Kreise mit größerem Radius damit nur äußerst spärlich besetzt sind. Im Folgenden soll die Wahl dieses Parameters erläutert werden.

7.3.3.2.1 Manuelle Optimierung der Anzahl der Punkte für jeden Radius

Um den Einfluss der Anzahl der Punkte auf verschiedenen Radien besser verstehen zu lernen, wurden zahlreiche Experimente durchgeführt, bei denen eben diese Anzahl variiert wurde und die Klassifikationsergebnisse verglichen wurden. Es wird sofort klar, dass der Aufwand dieser Methode sehr groß ist und die Suche nach den Parametern keinesfalls erschöpfend erfolgen kann. Tabelle 11 zeigt einen Ausschnitt der erzielten Ergebnisse. Für jeden aufgeführten Radius ist die Anzahl der Punkte angegeben, mit der der beste Wert bei der Klassifikation erreicht werden konnte. Zur leichteren Umsetzung und zur Einschränkung des Suchraums wurden lediglich ganzzahlige Vielfache der Zahl 4 als mögliche Anzahl von Punkten verwendet. Der optimale Wert für einen Radius von 1 Pixel ist trivialerweise 8.

Tabelle 11: Auswahl der „optimalen“ Anzahl von Punkten pro Radius.

Flavia; 1 x all; 1NN				
Radius	4	6	9	12
# Punkte	16	20	24	28
Ergebnis	64,24%	72,05%	77,71%	79,02%
Radius	17	22	32	45
# Punkte	28	28	60	80
Ergebnis	76,71%	71,47%	71,42%	72,94%

Beim Vergleich dieser Ergebnisse mit denen aus Tabelle 10 ist festzustellen, dass sich zum größten Teil deutliche Verbesserungen erzielen ließen. Gerade bei den großen Radien ist ein enormer Anstieg zu beobachten. So verbessert sich das Klassifikationsergebnis bei einem Radius von 45 Pixeln beispielsweise um fast 30%. Ein Ergebnis dieser Art war auch durchaus zu erwarten, da es nur schwer vorstellbar ist, dass acht Punkte auf einem Radius dieser Größe genügen sollen, um eine Blattform ausreichend zu beschreiben.

7.3.3.2.2 Integration in das Gesamtsystem

Im letzten Abschnitt wurde eindeutig nachgewiesen, dass sich die mit einzelnen Radien erzielten Ergebnisse durch Optimierung der jeweiligen Anzahl von Punkten stark verbessern lassen. Werden allerdings all diese Radien kombiniert und in das Gesamtsystem integriert, zeigt sich ein ganz anderes Bild, wie in Tabelle 12 deutlich wird.

Tabelle 12: Vergleich der Klassifikationsergebnisse mit acht Punkten pro Radius mit denen mit einer optimierten Anzahl pro Radius.

Flavia; 1 x all; 1NN		
# Punkte / Radius	8	optimiert
# Radien	17	17
# BP-Features	80	742
# Features gesamt	364	935
Ergebnis BP	88,46%	89,93%
Gesamtergebnis	99,69%	98,06%

Sofort sticht das signifikant schlechtere Gesamtergebnis ins Auge, das man unter Verwendung der Binary Patterns mit vermeintlich optimierten Parametern erhält. Bereits beim Ergebnis der Binary Pattern-Features alleine zeigt sich, dass der Einfluss der Optimierung bei weitem nicht mehr so groß ist wie bei den einzelnen Radien. Grund hierfür ist die stark angestiegene Anzahl an Features. Dadurch, dass sich die Menge an Binary Pattern-Features fast verzehnfacht hat, steigt die Zahl aller Features um mehr als das 2,5-fache. Bei der Klassifikation mit Hilfe des KNN-Klassifikators führt dies vermutlich dazu, dass die restlichen Features, besonders die extrem aussagekräftigeren HOCS-Features, einen geringeren Einfluss auf die Distanzberechnung haben.

Aufgrund dieser Ergebnisse wurde bei der Extraktion der Binary Pattern-Features eine fixe Anzahl von acht Punkten pro Radius verwendet.

7.3.4 Übersicht

Werden alle Feature-Klassen zusammengenommen, erhält man für jede Instanz, also jedes Blatt, einen Feature-Vektor mit 366 Attributen. Diese setzen sich wie folgt zusammen:

- 5 Kontur-Features.
- 16 Farb-Features.
- 5 Krümmung-Features.
- 6 Hu-Features.
- 160 HOCS-Features: 8 Radien mit je einem Histogramm für Bogen und Fläche, 10 Bins pro Histogramm.
- 170 Binary Pattern-Features: 17 Radien mit je 10 Bins pro Radius.
- 1 ID-Feature: Dieses Feature wird bei der Klassifikation ausgeblendet. Es dient dazu, jede Instanz eindeutig zu identifizieren und dem ursprünglichen Blatt zuzuordnen zu können. In dieser Arbeit wurde dazu ein String der Form *[Datenset]Speziesname_Blattnummer* verwendet. So verweist z. B. die ID „[MEW]Acer platanoides_0“ auf das erste Blatt der Spezies Acer platanoides im MEW-Datenset.
- 1 Class-Feature: Gibt die Klasse an, der eine Instanz angehört.

7.4 Evaluation der Feature-Klassen

In diesem Abschnitt werden verschiedene Kombinationen von Feature-Klassen getestet, um das bestmögliche System zu kreieren. Des Weiteren soll der Einfluss einzelner Feature-Klassen auf das Gesamtsystem untersucht werden.

Als Klassifikator wird der KNN-Klassifikator mit $K=1$ verwendet. Des Weiteren erfolgt die Evaluation gemäß der 1 x all-Methode, da diese nach nur einem Lauf ein varianzloses Ergebnis liefert. Die Evaluation erfolgt auf dem Flavia- und dem MEW-Datenset, d. h. sowohl auf einem mittelgroßen Datenset mit 32 Spezies (Flavia) als auch auf einem großen Datenset mit 153 Spezies (MEW).

7.4.1 Klassifikationsergebnisse der einzelnen Feature-Klassen

Zunächst wird der Einfluss jeder einzelnen Feature-Klasse überprüft. Dazu erfolgt jeweils ein Klassifikationsdurchlauf, bei dem alle anderen Features ausgeschlossen werden. Die Ergebnisse sind in Tabelle 13 zu sehen.

Tabelle 13: Klassifikationsergebnisse der einzelnen Feature-Klassen.

Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN	
Features	Ergebnis	Features	Ergebnis
Kontur	77,45%	Kontur	40,55%
Farbe	78,81%	Farbe	62,34%
Kurvatur	72,78%	Kurvatur	35,76%
Hu	43,84%	Hu	31,08%
HOCS	95,86%	HOCS	84,99%
BP	88,46%	BP	51,11%
Alle	99,69%	Alle	95,66%

Zunächst fällt auf, dass die Klassifikationsergebnisse auf dem MEW-Datenset durchwegs schlechter ausfallen als auf dem Flavia-Datenset. Dies lässt sich durch eine deutlich höhere Anzahl an verschiedenen Spezies (153, Flavia: 32) und den daraus entstehenden höheren Schwierigkeitsgrad des MEW-Datensets begründen.

Des Weiteren zeigt sich, dass die HOCS-Features mit Abstand den größten Einfluss auf das Gesamtergebnis haben. Sie sind auch ohne die anderen Feature-Klassen in der Lage, ein respektables Ergebnis von 95,86% (Flavia) bzw. 84,99% (MEW) zu erzielen. Dies ist nicht weiter verwunderlich, da sie fähig sind, sowohl feine Unterschiede in der Beschaffenheit der Kontur als auch die grobe Form eines Blattes sehr exakt zu beschreiben.

Beim Flavia-Datensatz liefern die Binary Pattern-Features das zweitbeste Ergebnis. Knapp 10% dahinter liegen fast gleichauf die Farb- und Kontur-Features. Es folgen die Kurvatur- und weit abgeschlagen die Hu-Features.

Dem gegenüber zeigen sich beim MEW-Datenset einige deutliche Abweichungen. Besonders auffällig ist das vergleichsweise sehr schlechte Klassifikationsergebnis der Binary Patterns. Dies lässt sich durch die stark erhöhte Anzahl von Spezies und die damit verbundene große Menge an ähnlich geformten Blättern erklären. Gerade bei den größeren Radien genügen acht Punkte pro Radius nicht mehr, um die Form des Blattes ausreichend genau zu beschreiben. Eine mögliche Verbesserung wäre eine Anhebung der Punkteanzahl. Dies ist allerdings mit erheblichem Testaufwand verbunden und ein Overfitting wird sehr wahrscheinlich.

Ebenso sind die Kontur-Features bei einer so hohen Anzahl von Spezies nicht mehr in der Lage, ähnliche Blattformen ausreichend zu unterscheiden. Auch ihr Ergebnis fällt stark ab.

7.4.2 Hinzufügen einer weiteren Feature-Klasse

Ausgehend von den Ergebnissen aus Tabelle 13 wird die jeweils beste Feature-Klasse, in diesem Fall die HOCS-Features, einzeln mit allen anderen Klassen kombiniert. Tabelle 14 zeigt die Ergebnisse.

Tabelle 14: Klassifikationsergebnisse der einzelnen Feature-Klassen in Kombination mit den HOCS-Features.

Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN	
Features	Ergebnis	Features	Ergebnis
HOCS +	95,86%	HOCS +	84,99%
Kontur	96,43%	Kontur	85,49%
Farbe	98,74%	Farbe	94,19%
Kurvatur	96,70%	Kurvatur	86,83%
Hu	96,28%	Hu	88,13%
BP	97,17%	BP	87,08%

Es fällt auf, dass bei beiden Datensets die Farb-Features das Ergebnis stärker verbessern als alle anderen Features, was vor allem beim MEW-Datenset deutlich wird. Hier steigern die Farb-Features das Klassifikationsergebnis um fast 10%, während alle anderen Feature-Klassen sichtbar abfallen. Dies lässt sich durch die Einzigartigkeit der Farb-Features erklären. Bei den Klassen gibt es Überschneidungen und Redundanzen, da sie alle, wenn auch auf unterschiedliche Art und Weise, die äußere Kontur und damit die Form des Blattes charakterisieren. Die Farb-Features beschreiben dagegen Eigenschaften des Blattes, die von den anderen Klassen völlig ignoriert werden. Der große Einfluss der Farb-Features ist dennoch bemerkenswert und nur schwer nachzuvollziehen. Zum einen handelt es sich bei ihnen lediglich um einfache, statistische Merkmalsbeschreibungen. Zum anderen besteht das MEW-Datenset aus 153 Spezies, die sich für das menschliche Auge zum größten Teil in ihrer Färbung nur wenig unterscheiden. Anders als z. B. beim Foliage-Datenset kommen keine Spezies mit einer auffälligen, durchgehenden Rot- oder Gelbfärbung vor. Alle Blätter weisen vor allem einen Grünton auf. Abbildung 90 zeigt ein Beispiel zweier Spezies, die ohne Farb-Features eine deutlich höhere Fehlerquote aufweisen.

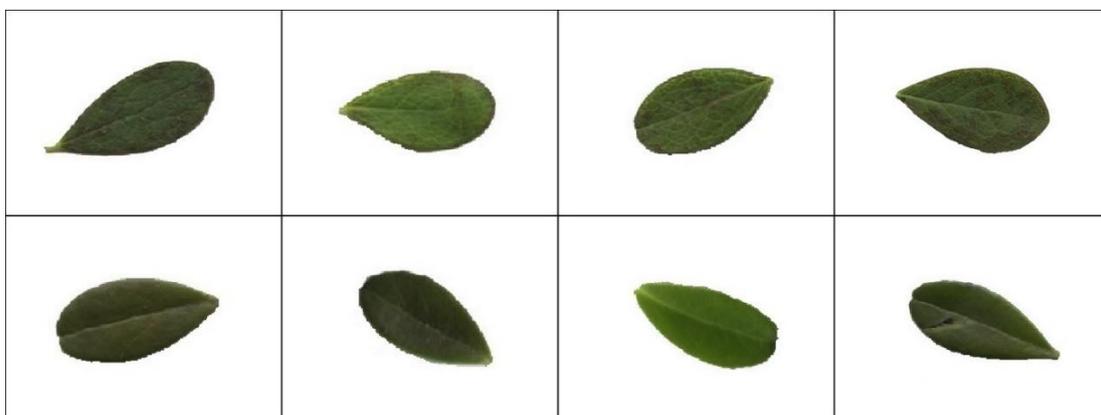


Abbildung 90: Je vier Beispielblätter aus dem MEW-Datenset.
Oben: *Vaccinium uliginosum*, unten: *Vaccinium vitisidaea*.

Es wurden je vier Bilder zufällig ausgewählt. Die Blätter weisen eine sehr ähnliche Form auf und auch bei der Farbe lässt sich mit dem bloßen Auge kein zuverlässiges Unterscheidungsmerkmal erkennen. Trotzdem wurden bei der Klassifikation ohne Farb-Features zehn Blätter der Spezies *Vaccinium uliginosum* (oben) fälschlicherweise als *Vaccinium vitisidaea* (unten) klassifiziert, mit Farb-Features dagegen kein einziges. Andersherum sank die Zahl der Fehlklassifikationen sogar von 16 auf 1. Das Klassifikationsergebnis für diese beiden Spezies verbesserte sich von etwa 49% und 43% auf etwa 94% und 99%. Diese Ergebnisse sind mit dem menschlichen Auge keineswegs nachzuvollziehen. Allerdings lässt sich der Einfluss der Farb-Features aufgrund der Ergebnisse nicht bestreiten. Dieses Thema wird in 7.6 noch näher diskutiert.

Überraschend ist der Einfluss der Hu-Features, speziell auf dem MEW-Datenset. Sie sorgen nach den Farb-Features für den größten Anstieg. Das ist verwunderlich, da diese Feature-Klasse alleine den niedrigsten Wert erzielt und nur eine grobe Beschreibung der Form des Blattes zulässt. Daher war zu erwarten, dass die Hu-Features lediglich Informationen liefern, die bereits durch die HOCS-Features abgedeckt sind. Das ist offensichtlich nicht der Fall. Eine logische Erklärung dafür kann in dieser Arbeit nicht geliefert werden.

7.4.3 Schrittweiser Aufbau des Systems

Im Folgenden wird das System weiter schrittweise aufgebaut. Analog zu der Vorgehensweise aus Tabelle 14 wird das bisher beste System einzeln mit jeder Feature-Klasse kombiniert und die Ergebnisse bestimmt. Daraus wird wiederum das System mit dem besten Klassifikationsergebnis ausgewählt und erweitert. Dieser Vorgang wird solange wiederholt, bis das komplette System erreicht wird. Tabelle 15 zeigt diesen Vorgang und die dabei erzielten Ergebnisse.

Tabelle 15: Schrittweiser Aufbau des Systems.

Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN	
Features	Ergebnis	Features	Ergebnis
HOCS + Farbe +	98,74%	HOCS + Farbe +	94,19%
Kontur	98,90%	Kontur	94,33%
Kurvatur	98,79%	Kurvatur	94,84%
Hu	98,85%	Hu	95,21%
BP	99,63%	BP	94,51%
HOCS + Farbe + BP	99,63%	HOCS + Farbe + Hu	95,21%
Kontur	99,63%	Kontur	95,48%
Kurvatur	99,69%	Kurvatur	95,52%
Hu	99,63%	BP	95,63%
HOCS + Farbe + BP + Kurvatur	99,69%	HOCS + Farbe + Hu + BP	95,63%
Kontur	99,69%	Kontur	95,63%
Hu	99,69%	Kurvatur	95,68%
Alle	99,69%	Alle	95,66%

Auf dem Flavia-Datensatz führt lediglich das Hinzufügen der Binary Patterns noch zu einer nennenswerten Verbesserung des Klassifikationsergebnisses. Die Berücksichtigung der Kontur- und Hu-Features steigert das Ergebnis lediglich von 99,63% auf 99,69%. Dies entspricht, bei 1907 im Flavia-Datensatz enthaltenen Blättern, einer einzigen zusätzlichen korrekten Klassifikation und kann deshalb nicht als signifikante Verbesserung angesehen werden.

Beim MEW-Datensatz liefert eine Hinzunahme der Hu-Features als dritte Feature-Klasse das beste Ergebnis. Dies war, wie zuvor bereits erwähnt, nicht zu erwarten und nicht logisch zu begründen. Es folgen die Binary Patterns, Kurvatur- und Kontur-Features. Auffällig ist, dass mit einem System ohne Kontur-Features ein minimal besseres Ergebnis erreicht werden kann. Allerdings darf auch diese Änderung nicht als signifikant angesehen werden. Da die Kontur-Features sich zuvor als durchaus aussagekräftig erwiesen haben und sich mit sehr geringem Aufwand berechnen lassen, werden sie nicht aus dem System entfernt.

Abschließend sei erwähnt, dass, genau wie bei der Parameteroptimierung, nicht jede Kombination an Feature-Klassen getestet werden konnte bzw. der Aufwand zu hoch erschien. Dass eine nicht getestete Kombination ein signifikant höheres Ergebnis erreicht, ist fast ausgeschlossen.

7.4.4 Einfluss der Wegnahme einzelner Feature-Klassen

In diesem Abschnitt soll untersucht werden, wie stark sich die Wegnahme einzelner Feature-Klassen auf das Klassifikationsergebnis auswirkt. Dies wird wiederum auf dem Flavia- und dem MEW-Datensatz evaluiert. Die Ergebnisse sind in Tabelle 16 zu sehen.

Tabelle 16: Einfluss der Wegnahme einzelner Feature-Klassen auf das Klassifikationsergebnis.

Flavia; 1 x all; 1NN			MEW; 1 x all; 1NN		
Features	Ergebnis	Einfluss	Features	Ergebnis	Einfluss
Alle	99,69%	-	Alle	95,66%	-
Kontur	99,69%	0,00%	Kontur	95,68%	+0,02%
Farbe	97,22%	-2,47%	Farbe	90,23%	-5,43%
Kurvatur	99,63%	-0,06%	Kurvatur	95,61%	-0,05%
Hu	99,69%	0,00%	Hu	95,04%	-0,62%
HOCS	97,90%	-1,79%	HOCS	90,97%	-4,69%
BP	99,06%	-0,63%	BP	95,65%	-0,01%

Den stärksten Einfluss haben, wie erwartet, die Farb-Features, da sie im Gegensatz zu allen anderen Klassen keinerlei Redundanz aufweisen. Es folgen die HOCS-Features. Der Einfluss der Kontur- und Kurvatur-Features ist bei beiden Datensets fast vernachlässigbar.

Auffällig ist wiederum, dass beim kleineren Flavia-Datensatz die Binary Patterns den drittgrößten Einfluss haben, während sie beim größeren MEW-Datensatz keine nennenswerte Verbesserung liefern. Genau entgegengesetzt verhält es sich mit den Hu-Features. Der höhere Einfluss der Binary Patterns auf dem Flavia-Datensatz lässt sich zum einen durch die geringe Anzahl an Punkten, nämlich acht pro Radius, erklären. Zum anderen wurden die gewählten Radien auf dem Flavia-Datensatz optimiert. Dasselbe gilt allerdings auch für die HOCS-Features, die beim MEW-Datensatz einen ganz erheblichen Einfluss aufweisen.

Des Weiteren fällt auf, dass der absolute Wert des Einflusses der Farb- und HOCS-Features beim MEW-Datenset deutlich höher ist als beim Flavia-Datenset. Speziell bei den HOCS-Features deutet dies darauf hin, dass sie in der Lage sind, die Form eines Blattes deutlich genauer zu beschreiben als die übrigen Feature-Klassen, die die äußere Kontur des Blattes charakterisieren. Der Unterschied beim Einfluss ergibt sich daraus, dass bei einer geringeren Anzahl von Spezies und somit sich ähnelnden Blattformen die weniger exakten Kontur-, Krümmung-, Hu- und Binary Pattern-Features zusammen leichter die Informationen aufwiegen können, die in den sehr komplexen HOCS-Features gespeichert sind.

7.5 Evaluation des Gesamtsystems innerhalb eines Datensets

In diesem Abschnitt soll das entwickelte System auf den verwendeten Datensets ausführlich getestet werden. Davor erfolgt zunächst ein Überblick über den Ablauf der Evaluation, vor allem über die verwendeten Evaluationsmethoden. Anschließend werden die Ergebnisse, die auf dem zur Parameteroptimierung verwendeten Flavia-Datenset erzielt wurden, näher betrachtet.

7.5.1 Ablauf der Evaluation

Für jedes Datenset werden zunächst die Standard-Evaluationsmethoden 10-Fold-Cross-Validation (10FCV) und Leave-one-out (1 x all) verwendet. Des Weiteren werden je nach Datenset weitere Tests durchgeführt, um eine bessere Vergleichbarkeit mit anderen Systemen zu erreichen. Es hat sich z. B. beim Swedish Leaf Dataset durchgesetzt, 25 Blätter pro Spezies als Trainingsset und die restlichen 50 als Testset festzulegen (50 x 25).

Bei den Evaluationsmethoden, die nicht varianzfrei sind, wurden jeweils zehn Läufe durchgeführt und der Mittelwert inklusive Standardfehler berechnet. Dies betrifft, abgesehen von der fixen Aufteilung in Trainings- und Testset beim Foliage-Datenset und den 1 x all-Ansatz, alle anderen Methoden. Als Seed für den Weka-Zufallsgenerator werden jeweils die Zahlen von 1 bis 10 verwendet.

7.5.2 Übersicht

Bevor der in dieser Arbeit entstandene Ansatz mit anderen Systemen verglichen wird, soll zunächst eine Übersicht über die erzielten Ergebnisse gegeben werden. Dazu wurden Testläufe auf den frei verfügbaren Datensets Flavia, Foliage, MEW, ICL und SLD durchgeführt. Bei jedem Datenset kamen die Standardevaluationsmethoden 1 x all, 10-Fold-Cross-Validation (10FCV) und $\frac{1}{2} \times \frac{1}{2}$ zum Einsatz. Des Weiteren werden bei einigen Datensets spezielle Evaluationsmethoden verwendet, die sich durch diverse Veröffentlichungen etabliert haben. Die Ergebnisse sind in Tabelle 17 zu sehen. Das Foliage-Trainings- und das Testdatenset werden zur weiteren Evaluation zusammengefasst und lediglich für die Test x Train-Methode getrennt betrachtet. Nach dem zur Parameteroptimierung verwendeten Flavia-Datensatz sind die übrigen Datensets nach den darin enthaltenen Spezies aufsteigend sortiert. Alle Ergebnisse sind in Prozent angegeben.

Tabelle 17: Übersicht über die erzielten Ergebnisse auf den frei verfügbaren Datensets.

1NN-Klassifikator		Evaluationsmethode					
Datenset	# Spezies	1 x all	10FCV	½ x ½	Te x Tr	10 x 40	25 x 50
Flavia	32	99,69	99,61 ±0,02	98,93 ±0,10	-	99,37 ±0,08	-
SLD	15	98,74	98,68 ±0,04	98,24 ±0,24	-	-	97,81 ±0,15
Foliage	60	96,45	96,32 ±0,03	95,20 ±0,12	95,64	-	-
MEW	153	95,66	95,50 ±0,01	93,80 ±0,09	-	-	-
ICL	220	93,48	93,20 ±0,02	90,19 ±0,07	-	-	-

Absolut gesehen wurde das beste Ergebnis beim Flavia-Datenset erzielt. Dies war zu erwarten, da es zum einen nach dem SLD die zweitwenigsten Spezies enthält und es zum anderen zur Optimierung der Parameter verwendet wurde. Bei den übrigen Datensets nehmen die Ergebnisse in der Regel mit steigender Anzahl der Spezies ab. Eine Ausnahme bildet das SLD. Diese und andere Auffälligkeiten sollen im Folgenden näher erläutert werden, wenn einige der Datensets noch einmal einzeln betrachtet werden.

7.5.3 Evaluation auf dem Flavia-Datensatz

Wie in vorhergehenden Abschnitten bereits mehrfach zu sehen war, beträgt die Klassifikationsrate auf dem Flavia-Datenset 99,69%, wenn als Klassifikator der K-Nearest-Neighbours-Klassifikator mit $K=1$ und als Evaluationsmethode der 1 x all-Ansatz verwendet wird. Im Folgenden soll dieses Ergebnis näher untersucht werden.

Zunächst sei erwähnt, dass das Gesamtergebnis mit 99,69% deutlich besser ausfiel als erwartet. Wie in Abschnitt 3.1 zu sehen war, ist die Klassifikation auf dem Flavia-Datenset alles andere als trivial. Viele Spezies weisen eine starke Ähnlichkeit auf, und für einen Menschen ohne entsprechenden Hintergrund wäre es vollkommen unmöglich, eine ähnliche Klassifikationsrate zu erreichen.

Allerdings muss auch beachtet werden, dass alle Feature-Parameter auf dem Flavia-Datenset optimiert wurden. Es ist demnach zu erwarten, dass das System auf anderen Datensätzen zumindest etwas schlechter arbeitet.

Das Flavia-Datenset enthält 1907 Instanzen, somit entspricht ein Klassifikationsergebnis von 99,69% 1901 korrekten Klassifikationen. Die Blätter, denen eine unzutreffende Klasse zugewiesen wurde, sind in Abbildung 91 mittig zu sehen. Zudem ist links jeweils der nächste Nachbar abgebildet, also der ähnlichste Vertreter der Spezies, der das zu klassifizierende Blatt fälschlicherweise zugeordnet wurde. Außerdem ist rechts der jeweilige nächste Nachbar der korrekten Spezies zu sehen, also das Blatt, das die größte Chance hatte, die zu klassifizierende Instanz korrekt zuzuordnen. Abbildung 92 zeigt analog die Binärdarstellungen.

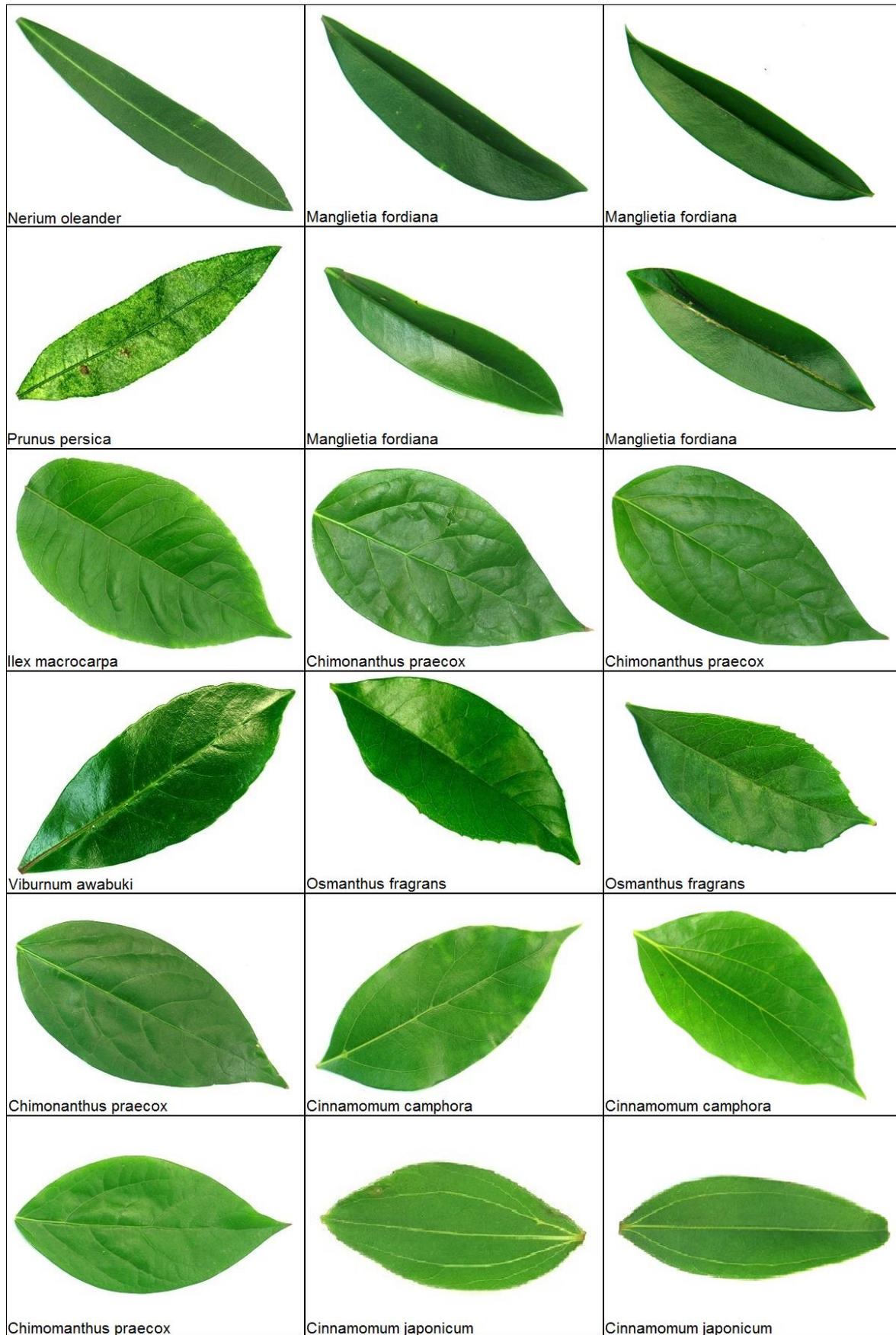


Abbildung 91: Fehlerhaft klassifizierte Instanzen aus dem Flavia-Datenset.
Links: Nächster Nachbar, Mitte: zu klassifizierende Instanz,
rechts: nächster Nachbar der korrekten Spezies.

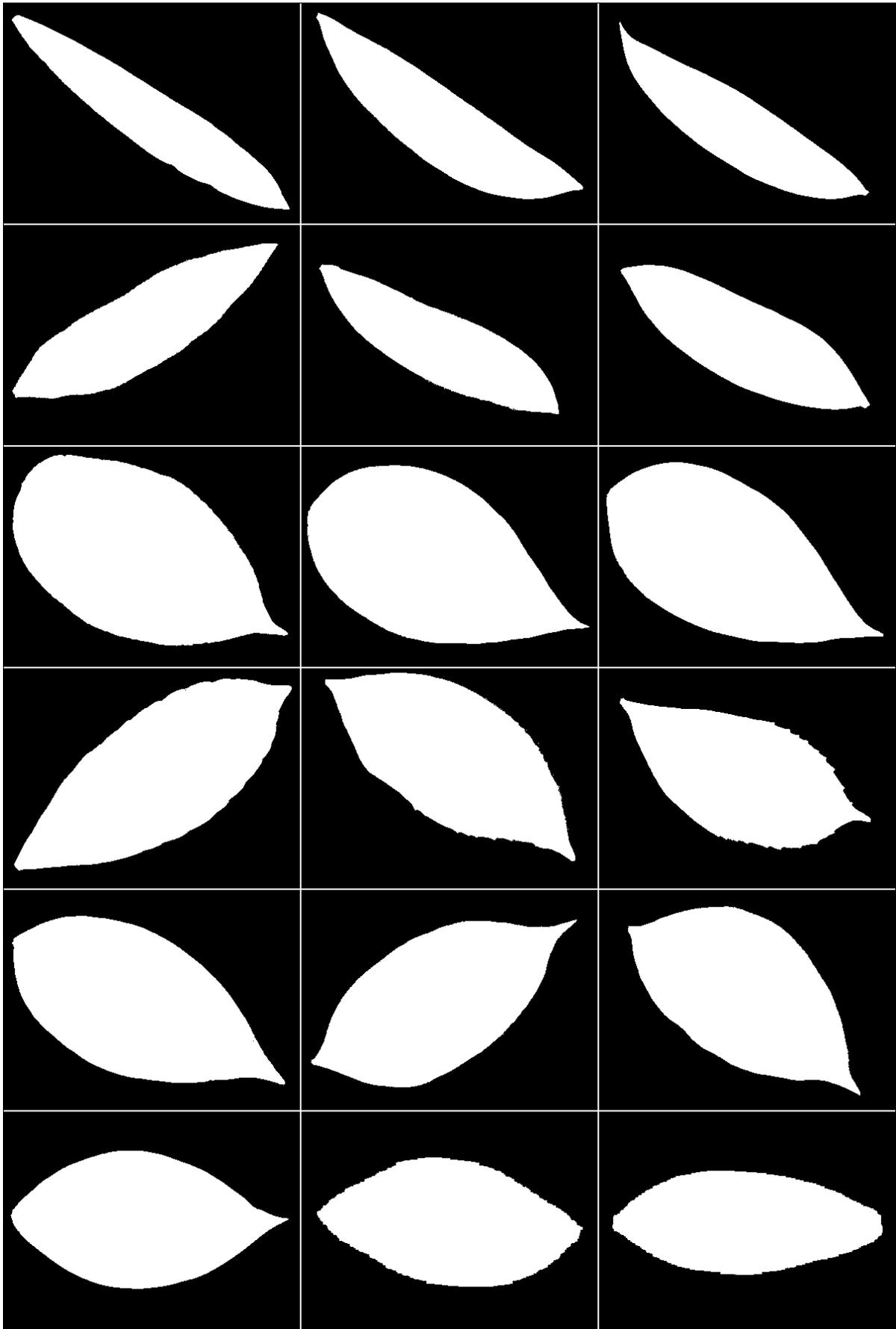


Abbildung 92: Binärdarstellung der fehlerhaft klassifizierten Instanzen aus dem Flavia-Datenset. Links: Nächster Nachbar, Mitte: zu klassifizierende Instanz, rechts: nächster Nachbar der korrekten Spezies.

Hier wird erneut der Schwierigkeitsgrad der Klassifikation deutlich. Die Blätter vieler Klassen unterscheiden sich nur bedingt und erscheinen vor allem in der Binärdarstellung für das menschliche Auge fast identisch.

Außerdem ist zu erwähnen, dass die hier abgebildeten, fehlerhaft klassifizierten Blätter keineswegs in irgendeiner Weise missgestaltet sind oder klassenuntypische Eigenarten aufweisen.

7.5.4 Evaluation auf dem Swedish Leaf Datenset

Das SLD besitzt mit 15 Spezies nur etwa halb so viele wie das Flavia-Datenset. Dennoch weist es eine leicht schwächere Klassifikationsrate auf. Dies lässt sich durch den höheren Schwierigkeitsgrad des SLD begründen, was in Abbildung 93 schnell deutlich wird.



Abbildung 93: Beispielblätter sich stark ähnelnder Spezies aus dem SLD. Von links nach rechts: *Corylus avellana*, *Ulmus carpinifolia*, *Ulmus minor*, *Tilia cordata*, *Tilia tomentosa*.

Die starke Ähnlichkeit der Blätter ist offensichtlich. Natürlich gibt es auch in den anderen Datensätzen Spezies, deren Vertreter sich stark ähneln. Allerdings ist dort das Verhältnis zu den einfacher zu klassifizierenden Spezies deutlich besser. Da das SLD nur über 15 Spezies verfügt, stellen diese fünf stark verwechslungsgefährdeten Spezies ein Drittel der Gesamtmenge an Blättern dar, was sich negativ auf das Ergebnis auswirkt.

Trotz allem konnten insgesamt auch auf dem SLD sehr gute Ergebnisse erzielt werden. So entspricht bei einer Instanzengesamtanzahl von 1.110 eine Erfolgsrate von 98,74% bei dem 1 x all-Verfahren einer Menge von nur 14 falsch klassifizierten Instanzen. Davon entfallen allein sechs Fehlklassifikationen auf die beiden Ulmen-Spezies *Ulmus carpinifolia* und *Ulmus minor*.

Diese fehlerhaft klassifizierten Instanzen sind in Abbildung 94 zu sehen.

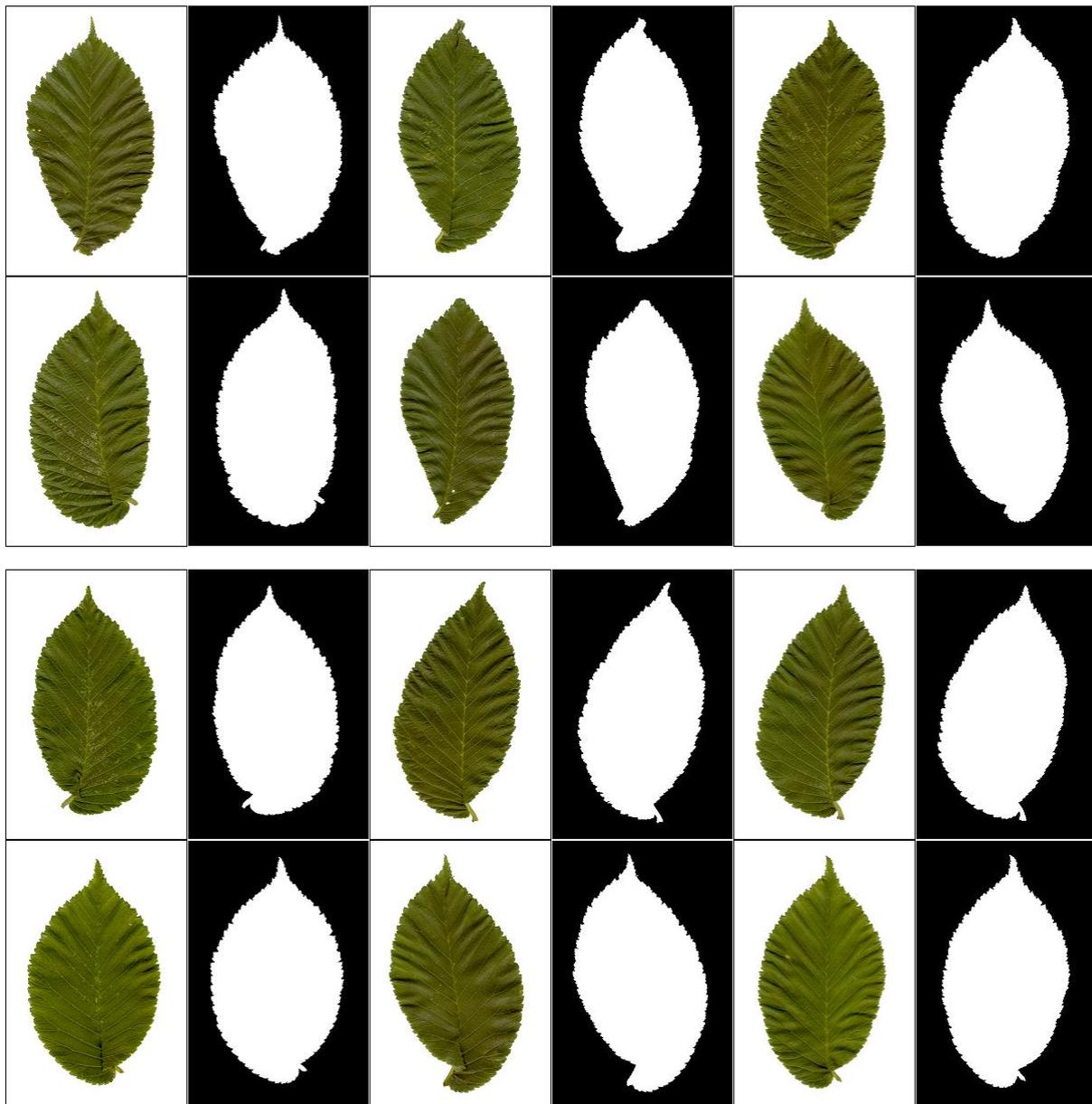


Abbildung 94: *Fehlklassifizierte Instanzen der beiden Ulmus-Spezies aus dem SLD. Zeilen 1 und 3: zu klassifizierende Instanzen, Zeilen 2 und 4: nächster Nachbar, jeweils mit zugehörigem Binärbild.*

Die Schwierigkeit bei der Unterscheidung dieser beiden Spezies ist offensichtlich. Die Blätter ähneln sich sehr stark und es lassen sich keine Merkmale ableiten, die eine sicherere Differenzierung ermöglichen.

7.5.5 Evaluation auf dem MEW-Datensatz

Trotz der weit höheren Anzahl an Spezies als bei den vorhergegangenen Datensätzen konnte ein sehr gutes Ergebnis von 95,66% erzielt werden. Die Fehldetektionen sind, anders als beim SLD, zum größten Teil gleichmäßig über den Datensatz verteilt, sodass für die meisten Spezies eine Klassifikationsrate von, zumeist sogar deutlich, mehr als 90% erzielt werden konnte.

Die einzigen Ausnahmen bilden zum einen die Spezies *Maclura pomifera*, bei der in diesem Datensatz zwischen der weiblichen und der männlichen Variante unterschieden wird. Diese erreichen nur eine Klassifikationsgenauigkeit von knapp 60% (weiblich) und 75% (männlich). Dabei ist die männliche Variante für 19 der insgesamt 21 Fehldetektionen des weiblichen Gegenstücks verantwortlich. Umgekehrt sind es sogar 17 von 17.

Zum anderen konnten bei der Spezies *Mespilus germanica* nur 64% der Instanzen richtig klassifiziert werden. Im Gegensatz zu den eben diskutierten Spezies sind hierbei die fehlerhaft zugewiesenen Klassen breit über das gesamte restliche Datenset verteilt. In Abbildung 95 sind je vier Beispielbilder der drei eben erwähnten Spezies zu sehen.

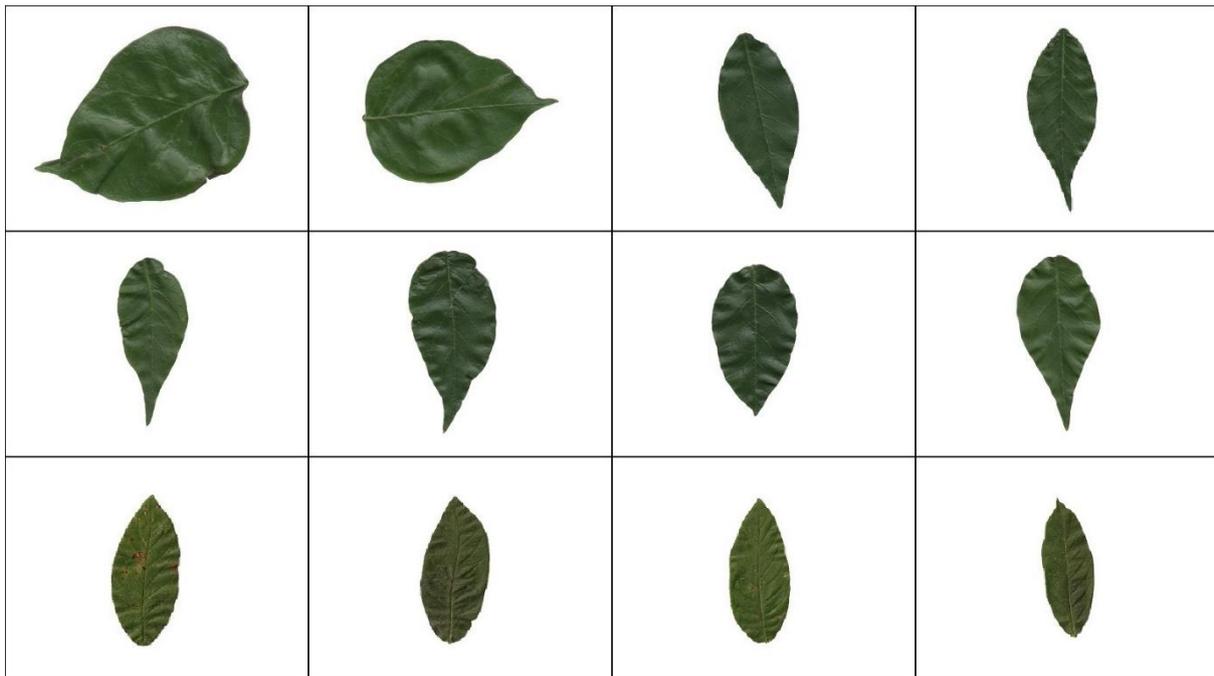


Abbildung 95: Beispielbilder der drei Spezies mit dem schlechtesten Klassifikationsergebnis im MEW-Datenset. Von oben nach unten: *Maclura pomifera* (weiblich), *Maclura pomifera* (männlich), *Mespilus germanica*.

In der ersten Zeile sind Instanzen der weiblichen Variante der *Maclura pomifera* zu sehen. Zunächst fällt auf, dass eine extreme Intraintravariabilität vorliegt. Die beiden linken Blätter unterscheiden sich deutlich von den beiden rechten. Des Weiteren ist zu sehen, dass die Ähnlichkeit zwischen den Blättern der zweiten und den beiden rechten der ersten Zeile sehr hoch ist. Die häufig auftretenden Verwechslungen sind demnach nicht überraschend.

Abschließend sind in Zeile 3 Beispielblätter der Spezies *Mespilus germanica* abgebildet. Auf den ersten Blick ist es nicht verwunderlich, dass diese Klasse das zweitschlechteste Klassifikationsergebnis aufweist und sich dabei die Fehldetektion auf eine Vielzahl von Spezies verteilt. Es sind keine herausragenden Merkmale zu erkennen, welche die Blätter klar von denen anderer Spezies abgrenzen. Je nach Wuchsform sind die Blätter länglich oder leicht eiförmig und der Rand ist glatt. Es gibt eine Vielzahl von Spezies, auf denen diese Beschreibung zutrifft. Dies ist kein Einzelfall. Gerade in den größeren Datensets kommt es häufiger vor, dass eine Klasse schlechte Klassifikationsergebnisse aufweist, nicht weil sie einer anderen Klasse extrem ähnelt, sondern weil ihr herausstechende Merkmale fehlen und sie demnach Gefahr läuft, mit vielen anderen Klassen verwechselt zu werden. An dieser Stelle sei allerdings auch erwähnt, dass es durchaus Spezies gibt,

deren Instanzen sehr zuverlässig klassifiziert werden können, obwohl sie für das menschliche Auge keine besonders auffälligen Merkmale aufweisen.

7.5.6 Ergebnisse unter Verwendung einer SVM

Dieselben Experimente wie in Abschnitt 7.5.2, mit Ausnahme der 1 x all-Evaluationsmethode, wurden auch mit einer SVM ($\gamma=0,335$; $C=20$) als Klassifikator durchgeführt. Die Ergebnisse sind in Tabelle 18 zu sehen.

Tabelle 18: Vergleich der Ergebnisse zwischen 1NN und SVM.

DS	Flavia		Foliage	SLD		MEW	ICL
EM	10 x 40	½ x ½	Te x Tr	50 x 25	½ x ½	½ x ½	½ x ½
1NN	99,37% ±0,08%	98,93% ±0,10%	95,83%	97,81% ±0,15%	98,24% ±0,24%	93,80% ±0,09%	90,19% ±0,07%
SVM	99,29% ±0,09%	98,92% ±0,08%	95,17%	98,35% ±0,12%	98,79% ±0,12%	96,54% ±0,10%	95,28% ±0,10%

Die Ergebnisse sind in den meisten Fällen sehr ähnlich. Der 1NN-Klassifikator erzielt beim Flavia- und Foliage-Datenset leicht bessere Ergebnisse, während die SVM vor allem bei den sehr großen Datensets MEW und ICL bessere Resultate liefert. Es lässt sich nicht endgültig festhalten, welcher Klassifikator für diese Klassifikationsaufgabe am besten geeignet ist.

Aufgrund der weit besseren Erklärbarkeit und der einfacheren Handhabung wird zur weiteren Evaluation der 1NN-Klassifikator verwendet.

7.5.7 Vergleich mit anderen Systemen

Im Folgenden werden die erzielten Klassifikationsergebnisse des in dieser Arbeit erstellten Systems mit denen anderer Systeme verglichen und diskutiert. Anschließend wird auf Verzerrungen und Ungenauigkeiten bezüglich der Vergleichbarkeit hingewiesen.

7.5.7.1 Ergebnisse und Diskussion

Tabelle 19 zeigt einen Vergleich der Ergebnisse mit anderen Systemen. Um maximale Vergleichbarkeit zu gewährleisten, werden dieselben Datensets (DS) und Evaluationsmethoden (EM) verwendet. Die Ergebnisse des in dieser Arbeit entwickelten Ansatzes sind in der letzten Zeile zu sehen. Die besten Klassifikationsleistungen anderer Systeme sind für jedes Datenset und für jede Klassifikationsmethode unterstrichen dargestellt. Alle Ergebnisse sind in Prozent angegeben.

Tabelle 19: Vergleich des in dieser Arbeit vorgestellten Systems mit anderen Ansätzen.

DS	Flavia		Foliage	SLD		MEW		ICL	
EM	10 x 40	1 x all	Te x Tr	50 x 25	1 x all	½ x ½	1 x all	½ x ½	1 x all
[1]	<u>97,19</u>								
[2]	95,00		<u>95,75</u>						
[3]	<u>97,19</u>		95,00						
[4]				<u>97,92</u>					
[5]						86,86			
[6]	91,53	<u>93,66</u>			<u>96,53</u>	84,92	<u>88,91</u>	<u>79,68</u>	<u>84,62</u>
1NN	99,37 ±0,08	99,69	95,83	97,81 ±0,15	98,74	93,80 ±0,09	95,66	90,19 ±0,07	93,48
SVM	99,29 ±0,09	-	95,17	98,35 ±0,12	-	96,54 ±0,10	-	95,28 ±0,10	-

[1]: [Lee 2013], [2]: [Kadir 2012], [3]: [Kadir 2014], [4]: [Wu 2007], [5]: [Suk 2013], [6]: [Novotny 2013]

Der vorgestellte Ansatz erreicht oder übertrifft bei jedem Datenset und für jeden Evaluationsmodus die Ergebnisse der Systeme, die im jeweiligen Bereich am besten abgeschnitten haben. Teilweise konnten sogar deutliche Steigerungen bei den Klassifikationsergebnissen erzielt werden. Betroffen sind vor allem die Datensets, die eine sehr große Anzahl an Spezies beinhalten, namentlich das MEW- und das ICL-Datenset. Dies spricht für die hohe Genauigkeit des in dieser Arbeit vorgestellten Ansatzes, mit der selbst eine solche Vielzahl von Klassen mit hoher Erfolgsrate differenziert werden kann.

7.5.7.2 Probleme bei der Vergleichbarkeit

Beim Vergleich der Systeme entstehen, trotz der Verwendung identischer Datensets und Evaluationsmethoden, einige Verzerrungen, die in diesem Abschnitt kurz erläutert werden sollen.

7.5.7.2.1 Unklare Herkunft mancher Werte

In der überwiegenden Mehrheit der Veröffentlichungen wird leider nicht erläutert, wie der jeweilige Klassifikationswert zustande gekommen ist. Bei varianzlosen Evaluationsmethoden wie 1 x all oder Testset x Trainingsset spielt das keine Rolle. Bei Methoden, bei denen das Set zufällig in kleinere Teile zerlegt wird, können sich jedoch signifikante Verzerrungen bei den Ergebnissen ergeben. So wurde z. B. in dieser Arbeit stets über zehn Läufe gemittelt und dabei immer unterschiedliche Seeds für den Zufallsgenerator verwendet, nämlich die ganzen Zahlen von 1 bis 10. Daraus lässt sich ein Mittelwert und ein Standardfehler berechnen, der das Ergebnis exakt beschreibt. Wird, wie bei vielen Veröffentlichungen üblich, lediglich ein einzelner Wert ohne Standardfehler angegeben, kann dieser auf vielen Wegen zustande gekommen sein. So ist es möglich, dass es sich um einen Mittelwert handelt, der nicht als solcher gekennzeichnet wurde, bei dem nicht bekannt ist, aus wie vielen Läufen er sich ergeben hat und wie varianzreich das zu untersuchende Verfahren ist. Eine andere Möglichkeit ist, dass es sich wirklich nur um einen einzigen Wert handelt. Dies könnte z. B. der erstbeste Wert sein, der berechnet wurde oder sogar der Beste aus einer Vielzahl von Werten. All das kann zu starken Verzerrungen führen. So betrug beispielsweise beim SLD der über zehn Läufe

ermittelte Durchschnittswert 97,81%, wobei der höchste Wert bei 98,38% und der niedrigste sogar nur bei 96,76% lag.

7.5.7.2.2 *Overfitting*

Ein weiterer Problempunkt ist, dass nicht bekannt ist, ob die Parameter der zu vergleichenden Systeme immer auf demselben Datenset optimiert wurden bzw. ob sogar vor jedem Test auf einem anderen Datensatz die Parameter erneut angepasst wurden. Dies kann wiederum zu starken Verzerrungen bei der Evaluation führen, da ein System in den allermeisten Fällen die besten Ergebnisse auf dem Datenset erzielt, für den auch die Parameter optimiert wurden. Dieses Overfitting, wie es z. B. in dieser Arbeit auf dem Flavia-Datensatz erfolgt ist, kann sich auf das Klassifikationsergebnis stark positiv auswirken. Deshalb wurde bei dem hier vorgestellten System auch größter Wert darauf gelegt, dass die Parameteroptimierung ausschließlich auf dem Flavia-Datensatz vorgenommen wurde und das resultierende System eins zu eins auf die anderen Datensätze übertragen wurde.

7.5.7.2.3 *Segmentierung*

Einige Datensets wie Flavia und Foliage sind bereits vollständig vorsegmentiert. Das bedeutet, dass alle Pixel, die nicht zum Blatt gehören, den Graustufenwert 255 besitzen. In diesem Fall ist eine Segmentierung trivial und wird stets dasselbe Ergebnis und somit die gleichen Voraussetzungen für die Klassifikation liefern. Bei den Datensets SLD, MEW und ICL verhält sich das allerdings etwas anders. Hier liegen die Blätter als Scans vor und die Segmentierung erfordert dementsprechend mehr Aufwand. In den meisten Fällen lassen sich mit Hilfe der in Abschnitt 4.1.1 vorgestellten Otsu Methode, die auch in dieser Arbeit zur Segmentierung verwendet wurde, sehr gute Ergebnisse erzielen. Allerdings können in bestimmten Fällen nicht zu vernachlässigende Fehler auftreten. Drei Beispiele aus dem MEW-Datensatz sind in Abbildung 96 zu sehen.



Abbildung 96: *Drei Beispielbilder aus dem MEW-Datensatz mit erheblichen Segmentierungsfehlern. Links und rechts: Schatten im unteren Bereich, Mitte: Schatten am Stiel und dadurch missglückte Stielentfernung.*

Die Segmentierungsfehler sind deutlich zu erkennen. Teilweise führen diese auch dazu, dass die Detektion und Entfernung des Stiels nicht mehr vorgenommen werden kann (s. mittleres Bild). Die Auswirkungen sind klar: Viele Features können verfälscht werden und zu Fehlklassifikationen führen.

Folglich kann eine ausgefeiltere Segmentierungsmethode zu genaueren Features und somit zu einer besseren Klassifikationsperformanz führen. Über die in den Vergleichssystemen verwendete Segmentierungsmethode ist in den meisten Fällen nichts bekannt. Da die simple Otsu Methode in den allermeisten Fällen zu sehr guten Ergebnissen führt, ist davon auszugehen, dass die Segmentierung keinen signifikanten

Einfluss auf die Klassifikationsergebnisse hat. Dies gilt auch für die Qualität der Stielentfernung.

7.5.8 Evaluation auf dem BLD

Wie in Abschnitt 3.6 bereits erläutert, unterscheiden sich das eigens für diese Arbeit erstellte Datenset in vielerlei Hinsicht von den online frei verfügbaren. Im ersten Teil wurden die Blätter auf Papier platziert und fotografiert, was eine automatische Segmentierung ermöglicht (vgl. 4.3). Diese ist zwar nicht so zuverlässig wie bei eingescannten Blättern, jedoch ist diese Methode deutlich benutzerfreundlicher. Im zweiten Teil erfolgten die Fotoaufnahmen direkt am Baum und somit vor natürlichem Hintergrund.

Im Folgenden sollen beide Teile, analog zu den online frei verfügbaren Datensets, einzeln evaluiert werden.

7.5.8.1 Ergebnisse

Tabelle 20 zeigt die erzielten Klassifikationsergebnisse. Es wurden wiederum die auf dem Flavia-Datenset optimierten Parameter verwendet.

Tabelle 20: Übersicht über die erzielten Ergebnisse auf den beiden BLD-Teilsets.

1NN-Klassifikator		Evaluationsmethode		
Datensets	# Spezies	1 x all	10FCV	½ x ½
Flavia	32	99,69%	99,61 ±0,02%	98,93 ±0,10%
Papier	10	99,75%	99,71 ±0,02%	99,49 ±0,13%
Baum	10	98,46%	98,20 ±0,04%	97,89 ±0,14%

Das auf Papier erstellte Datenset liefert eine ähnliche Klassifikationsgenauigkeit wie das Flavia. Bei der 1 x all-Evaluation ergaben sich nur drei Fehlklassifikationen. Diese sind in Abbildung 97 zu sehen.

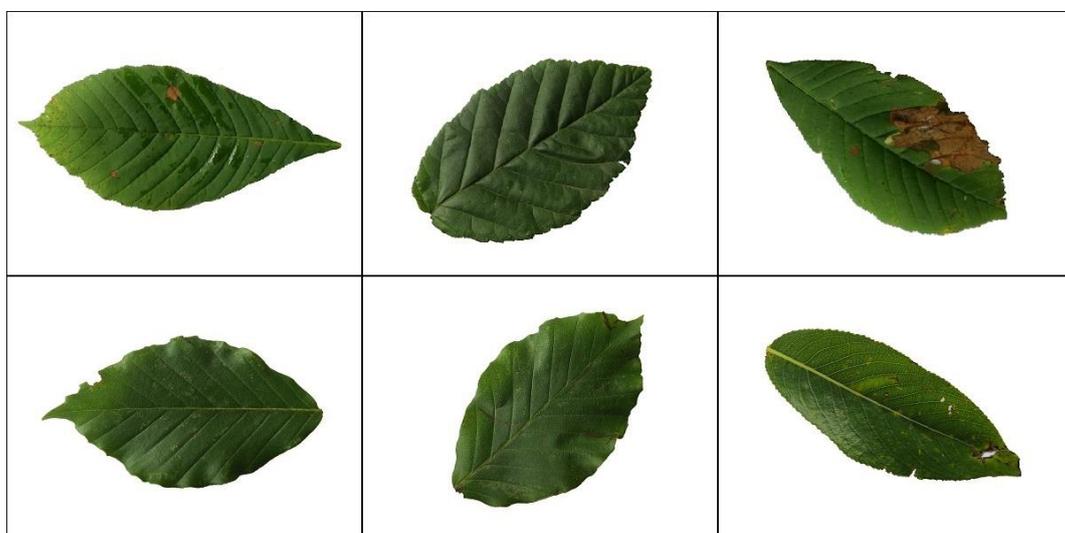


Abbildung 97: Fehlerhaft klassifizierte Instanzen im BLD (Papier).
Oben: zu klassifizierende Blätter, unten: nächste Nachbarn.

7.5.8.2 Problematik bei der Aufnahme am Baum

Das Datenset, dessen Blätter direkt am Baum aufgenommen wurden, liefert mit 98,46% ein um 1,28% schlechteres Ergebnis. Die Abweichung lässt sich dadurch erklären, dass die Blätter, anders als bei den Aufnahmen auf Papier, nicht stets in einer ähnlichen Pose flach auflagen, sondern am Baum hängen. Dies führte dazu, dass die Blätter aus unterschiedlichen Winkeln aufgenommen wurden und sich die Proportionen, z. B. durch durchhängende Blätter, veränderten. Trotzdem ist die Erkennungsrate auch am Baum noch als sehr hoch zu bezeichnen. Abbildung 98 verdeutlicht diese Problematik.

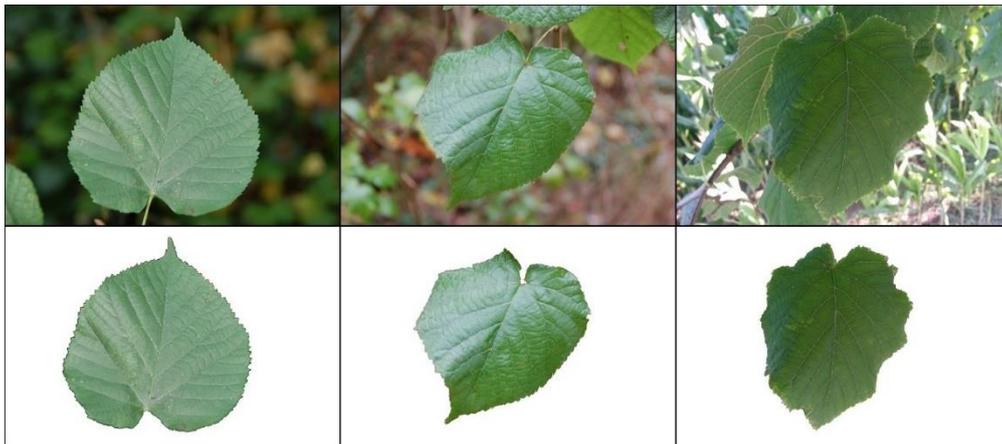


Abbildung 98: Drei Beispiele derselben Spezies (Stein-Linde: *Tilia cordata*) für die Aufnahmeproblematik am Baum. Oben: Originalaufnahme, unten: Ergebnis der Segmentierung.

Während links die charakteristische Herzform der Blätter der Linde noch klar zu erkennen ist, treten bei den mittleren und rechten Beispielen, bedingt durch Aufnahmewinkel und Schwerkraft, bereits deutliche Verzerrungen auf. Dies erschwert die Klassifikationsaufgabe erheblich.

Ebenfalls auffällig ist die vergleichsweise hohe Fehlerquote bei den beiden Eichen-Spezies Stiel-Eiche (*Quercus robur*) und Rot-Eiche (*Quercus rubra*). Diese sind allein für sieben der insgesamt 17 Fehlklassifikationen verantwortlich, während sie in allen Klassifikationsläufen auf dem Papierdatenset keine einzige Instanz falsch zuordneten. Drei dieser Fälle sind in Abbildung 99 zu sehen.

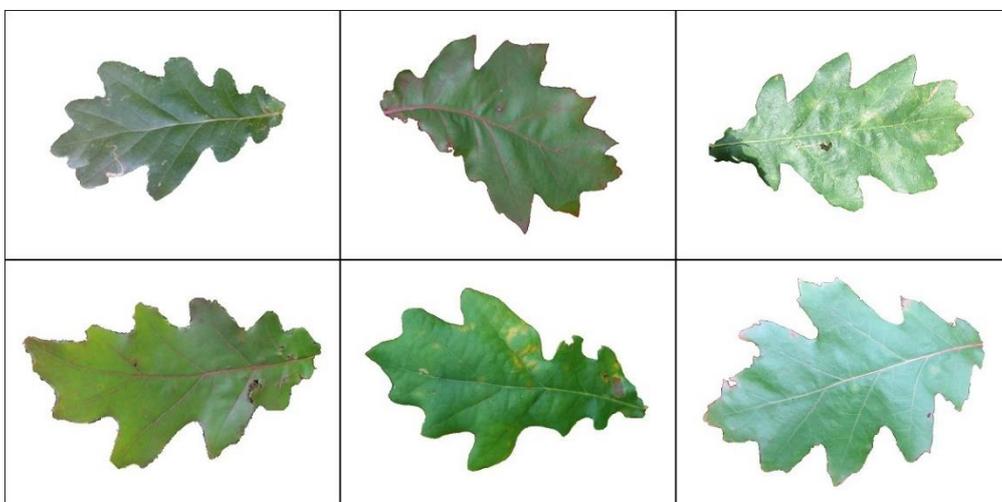


Abbildung 99: Fehlerhaft klassifizierte Instanzen der Stiel-Eiche und der Rot-Eiche im BLD (Baum). Oben: zu klassifizierende Blätter, unten: nächste Nachbarn.

Eine mögliche Erklärung für dieses Phänomen ist die im Schnitt schlechtere Segmentierung der Blätter. Da diese vor teils sehr komplexem Hintergrund fotografiert wurden, bleiben manchmal Hintergrundpixel am äußeren Rand des Blattes zurück. Dies kann dazu führen, dass die charakteristisch spitz zulaufenden Stellen der Blätter der Rot-Eiche deutlich runder erscheinen.

7.6 Farbproblematik

Wie bereits bei der Vorstellung der Farb-Features erwähnt, ist deren Verwendung zu Klassifikationszwecken zumindest fragwürdig. Grund dafür sind verschiedene Umwelteinflüsse, die die Färbung der Blätter stark beeinflussen können. Diese Problematik soll in diesem Abschnitt näher untersucht werden. Dazu wird zunächst eine Übersicht über den Einfluss der Farb-Features auf die Ergebnisse bei der Klassifikation innerhalb einzelner Datensets gegeben. Anschließend werden verschiedene Einflussfaktoren auf die Farb-Features und die daraus entstehende Problematik bei der Klassifikation vorgestellt. Zum Abschluss soll eine Möglichkeit diskutiert werden, mit der die Abhängigkeit der Farb-Features von äußeren Umständen überprüft werden kann.

7.6.1 Einfluss der Farb-Features auf das Klassifikationsergebnis

In Abschnitt 7.4 wurde die Nützlichkeit der Farb-Features bei der Klassifikation innerhalb des Flavia- und MEW-Datensets eindeutig nachgewiesen. Auch bei den anderen Datensets führt die Nutzung der Farb-Features zumeist zu einer signifikanten Erhöhung der Klassifikationsgenauigkeit. Dies ist in Tabelle 20 zu sehen.

Tabelle 21: Einfluss der Farb-Features auf das Klassifikationsergebnis.

1 x all; 1NN				
Datenset	# Spezies	Ergebnis des komplettes Systems		
		mit Farb-Features	ohne Farb-Features	Einfluss
SLD	15	98,74%	98,47%	-0,27%
Flavia	32	99,69%	97,33%	-2,36%
Foliage	60	96,45%	91,03%	-5,42%
MEW	153	95,66%	89,06%	-6,60%
ICL	220	93,48%	83,85%	-9,63%
BLD Papier	13	99,74%	99,57%	-0,17%
BLD Baum	12	98,46%	98,19%	-0,27%

Prinzipiell lässt sich sagen, dass die Farb-Features mit steigender Anzahl der Spezies an Einfluss gewinnen. Dies erscheint zunächst unlogisch, da es schwer vorstellbar ist, dass sich die Farbe von beispielsweise 153 oder sogar 220 Spezies signifikant unterscheidet. Allerdings ist das auch nicht notwendig, um eine deutliche Verbesserung des Ergebnisses zu erzielen. Es genügt, wenn sich einige wenige Spezies, deren Blattform sich stark ähnelt, mit Hilfe der Farb-Features unterscheiden lassen, um das Ergebnis deutlich zu erhöhen.

7.6.2 Auswirkungen von Umwelteinflüssen auf die Farb-Features

Blätter sind verschiedensten Umwelteinflüssen über lange Zeit ungeschützt ausgesetzt. Dies gilt sowohl für Niederschläge als auch für die vorherrschende Außentemperatur.

Beide Einflüsse sind stark ortsabhängig. So ist davon auszugehen, dass ein Blatt aus dem Flavia-Datenset, das an einem Baum im Yangtze-Delta in China gewachsen ist, deutlich anderen Umwelteinflüssen ausgesetzt war, als die Blätter aus dem BLD, das in Bayern zusammengestellt wurde.

Neben lokalen Differenzen der Umwelteinflüsse sind auch temporale zu betrachten. So wurde z. B. die Masterarbeit, für die das SLD zusammengestellt wurde (vgl. [SLD]), 2001 fertiggestellt. Dies legt die Annahme nahe, dass das Datenset im Zeitraum 2000/2001 entstanden ist. Im Vergleich dazu wurde das BLD in der zweiten Jahreshälfte 2014 kreiert. Es ist wahrscheinlich, dass sich die Umweltbedingungen zu beiden Zeitpunkten signifikant unterscheiden.

Des Weiteren muss beachtet werden, dass sich die Blattfärbung innerhalb eines Jahres stark verändert. Die im Frühjahr frisch gewachsenen Blätter erscheinen bei den meisten mitteleuropäischen Spezies in einem kräftigen Grün. Im Spätsommer beginnen die Blätter Flüssigkeit zu verlieren. Das Grün wird zunächst dunkler und geht in ein Gelb über. Im Herbst werden die Blätter braun, sterben letztendlich ab und fallen zu Boden.

7.6.3 Resultierende Problematik bei der Klassifikation

Die im vorletzten Abschnitt vorgestellten Ergebnisse lassen darauf schließen, dass die Farb-Features einen wertvollen Beitrag zur Klassifikation liefern. Dies ist soweit auch richtig. Allerdings muss beachtet werden, unter welchen Umständen die Klassifikation stattfindet. Es ist davon auszugehen, dass die Blätter der einzelnen Datensets alle innerhalb eines sehr begrenzten Zeitraums und Gebiets gesammelt wurden. Mit hoher Wahrscheinlichkeit wurde in vielen Fällen sogar eine Vielzahl von Blättern von ein und demselben Baum verwendet. All diese Faktoren führen dazu, dass sich die Blätter im Hinblick auf die Umwelteinflüsse, denen sie ausgesetzt waren, stark ähneln. Dies erklärt den großen Beitrag der Farb-Features zu den Klassifikationsergebnissen.

Mit der Realität haben diese Datensets aber nur begrenzt zu tun. Ein reales Anwendungsszenario wäre, dass mit Hilfe einer Datenbank von Blättern, die irgendwann einmal irgendwo gesammelt wurden, ein neues Blatt klassifiziert wird. Abbildung 100 macht das auftretende Problem schnell deutlich. Zu sehen sind je zwei zufällig ausgewählte Beispielbilder aus drei verschiedenen Datensets: SLD, BLD und MEW.

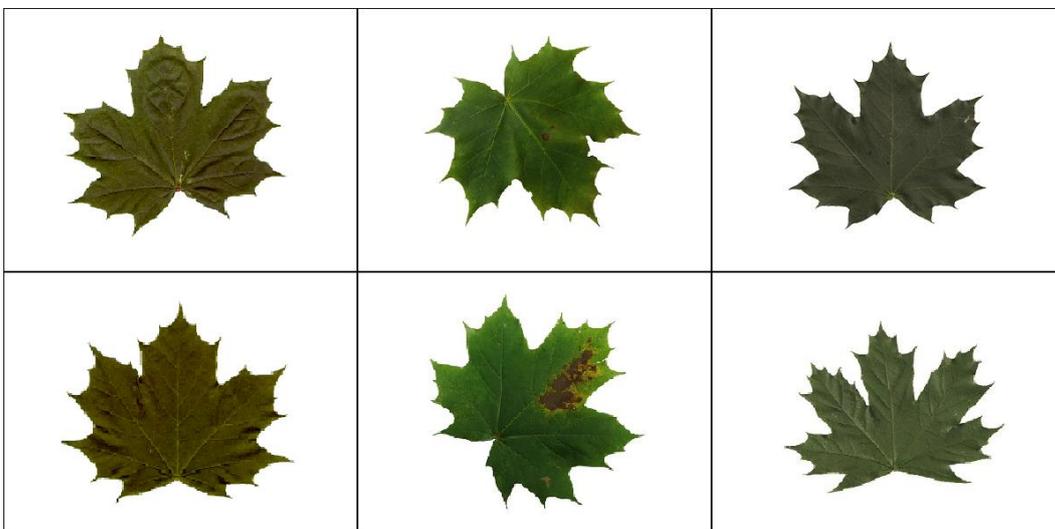


Abbildung 100: Je zwei Beispielblätter der Spezies Spitz-Ahorn (*Acer platanoides*).
Links: SLD, Mitte: BLD, rechts: MEW.

Es fällt sofort auf, dass die Blätter sich stark in ihrer Grundfärbung unterscheiden. Des Weiteren können Verunreinigungen oder großflächige Verfärbungen (s. Mitte unten), starken Einfluss auf die Farb-Features nehmen. Würde man versuchen, eines der hier gezeigten Blätter aus einem Datenset mithilfe der Blätter aus den anderen beiden Datensets zu klassifizieren, würden die Farb-Features wohl mehr schaden als sie nützen. Dieser Umstand soll im nächsten Abschnitt näher untersucht werden.

7.7 Evaluation der Datenset-übergreifenden Klassifikation

Im Folgenden wird das erstellte System weiter evaluiert, indem es nicht mehr wie zuvor innerhalb eines Datensets getestet wird, sondern ein Datenset als Trainingsdatensatz und ein anderes als Testdatensatz verwendet wird. Zunächst wird ein Überblick über die ausgewählten Datensets gegeben und die Vorgehensweise erläutert. Im Anschluss werden verschiedene Kombinationen von Datensets getestet und evaluiert. Abschließend sollen Möglichkeiten diskutiert werden, die Klassifikationsgenauigkeit zu verbessern.

7.7.1 Gewählte Datensets

Um eine möglichst aussagekräftige Evaluation zu gewährleisten, werden Datensets benötigt, in denen eine große Anzahl identischer Spezies vorkommt. In dieser Arbeit bieten sich dazu das MEW und das BLD an, da das MEW alle Spezies des BLD enthält. Ein weiterer Vorteil ist, dass das MEW mit insgesamt 153 Spezies sehr groß ist und somit genügend Raum für Fehlklassifikationen bietet. Dies kommt dem in Abschnitt 7.6.3 beschriebenen „realen Anwendungsbeispiel“ sehr nahe.

Zudem deckt BLD viele Klassen ab, die im MEW eine von vielen Unter-Spezies von Gattungen wie Ahorn (*Acer*), Eiche (*Quercus*) oder Linde (*Tilia*) darstellen. Die Blätter dieser Unter-Spezies ähneln sich häufig stark, so dass der Schwierigkeitsgrad der Klassifikation sehr hoch ist.

Ein weiterer Vorteil besteht darin, dass im MEW bei zusammengesetzten Blättern, genau wie beim BLD, stets nur einzelne Blättchen betrachtet werden.

Eine Liste der Spezies ist in Tabelle 22 zu sehen.

Tabelle 22: Zehn Spezies, die sowohl im MEW als auch im BLD vorkommen.

Spezies		# Blätter im jeweiligen Datenset			
		MEW	Papier	Baum	# MEW Spezies
Lateinischer Name	Deutscher Name				
<i>Acer platanoides</i>	Spitz-Ahorn	61	77	78	9
<i>Acer pseudoplatanus</i>	Berg-Ahorn	61	96	75	
<i>Alnus glutinosa</i>	Schwarz-Erle	72	85	95	3
<i>Betula pendula</i>	Hänge-Birke	62	70	66	4
<i>Carpinus betulus</i>	Hain-Buche	65	81	85	1
<i>Fagus sylvatica</i>	Rot-Buche	63	91	88	1
<i>Populus tremula</i>	Zitter-Pappel	64	94	90	2
<i>Quercus robur</i>	Stiel-Eiche	62	95	108	6
<i>Quercus rubra</i>	Rot-Eiche	73	83	89	
<i>Tilia cordata</i>	Stein-Linde	59	106	84	2

Zusätzlich ist jeweils die Anzahl der vorhandenen Blätter sowie der im MEW vorhandenen Spezies derselben Gattung gegeben. So existieren z. B. drei Spezies der Gattung Erle (*Alnus*): Grün- (*Alnus alnobetula*), Schwarz- (*Alnus glutinosa*) und Grau-Erle (*Alnus incana*).

Das Vergleichsdatenset MEW enthält somit knapp zehnmal mehr Spezies als jedes der beiden Testdatensets. Es ist anzumerken, dass die zu einer Gattung gehörigen Spezies lediglich nach dem ersten Teil ihres lateinischen Namens ausgewählt wurden. Dies muss nicht zwingend heißen, dass sich die Blätter sehr ähnlich sind. Meist ist dies aber der Fall.

7.7.2 Vorgehensweise

Als Trainingsdatenset wird stets das MEW verwendet. Die beiden Teile des BLD dienen sowohl als Testdatenset für das MEW als auch als Test- und Trainingsdatensätze untereinander. Das MEW zu testen, nachdem auf dem BLD trainiert wurde, ist sinnlos, da das MEW deutlich mehr Spezies besitzt. Für eine Evaluation müssten zunächst alle Spezies, die zwar im MEW, aber nicht in einem der beiden anderen Datensets vorkommen, entfernt werden. Dies würde den Schwierigkeitsgrad der Klassifikation enorm reduzieren und entspräche keinem realen Anwendungsszenario mehr.

Soweit nicht anders erwähnt, wird eine normale Trainings → Test-Klassifikation mit Hilfe des KNN-Klassifikators mit $K=1$ durchgeführt. Das bedeutet, dass jede Instanz des Testsets mit allen Instanzen des Trainingssets verglichen wird und die Klasse zugewiesen bekommt, die ihr nächster Nachbar innehat.

7.7.3 Ergebnisse mit dem Standardsystem

In Tabelle 23 sind die mit der eben beschriebenen Methodik erzielten Ergebnisse zu sehen.

Tabelle 23: Ergebnisse der Datenset-übergreifenden Klassifikation.

1NN; Komplettes System		
Training	Test	Ergebnis
Papier	Baum	87,30%
Baum	Papier	75,97%
MEW	Papier	10,93%
MEW	Baum	25,99%

Verglichen mit der Klassifikation innerhalb eines Datensets haben sich die Ergebnisse unter Verwendung der beiden Teile des BLD klar verschlechtert. Dies war teilweise zu erwarten, da sich die Blätter, die direkt am Baum fotografiert wurden, durch unterschiedliche Winkel und ein Schwerkraft-bedingtes Durchhängen stark von denen auf Papier unterscheiden können. Es fällt auf, dass die hohe Fehlerquote größtenteils durch Blätter bedingt ist, die durch das Durchhängen am Baum stark verformt sind. Von diesem Phänomen war am stärksten die Stein-Linde (*Tilia cordata*) betroffen, wie bereits in Abbildung 98 auf Seite 130 zu sehen war.

Die Ergebnisse, bei denen die Instanzen aus dem BLD mit denen aus dem MEW verglichen wurden, sind sehr schlecht. Speziell die erzielten Resultate, die mit den auf Papier aufgenommenen Blättern entstanden sind, sind mit weniger als 11% völlig unbrauchbar. Besonders auffällig ist, dass keine einzige Instanz der beiden Ahorn-Arten (Spitz- (*Acer*

platanoides) und Berg-Ahorn (*Acer pseudoplatanus*) korrekt klassifiziert wurde, da sie alle als Feld-Ahorn (*Acer campestre*) angesehen wurden. Abbildung 101 zeigt je einen Vertreter der zu klassifizierenden Blätter, als auch je zwei Beispiele der zugehörigen Klasse aus dem MEW, sowie zwei Instanzen der fälschlicherweise zugewiesenen Klasse.

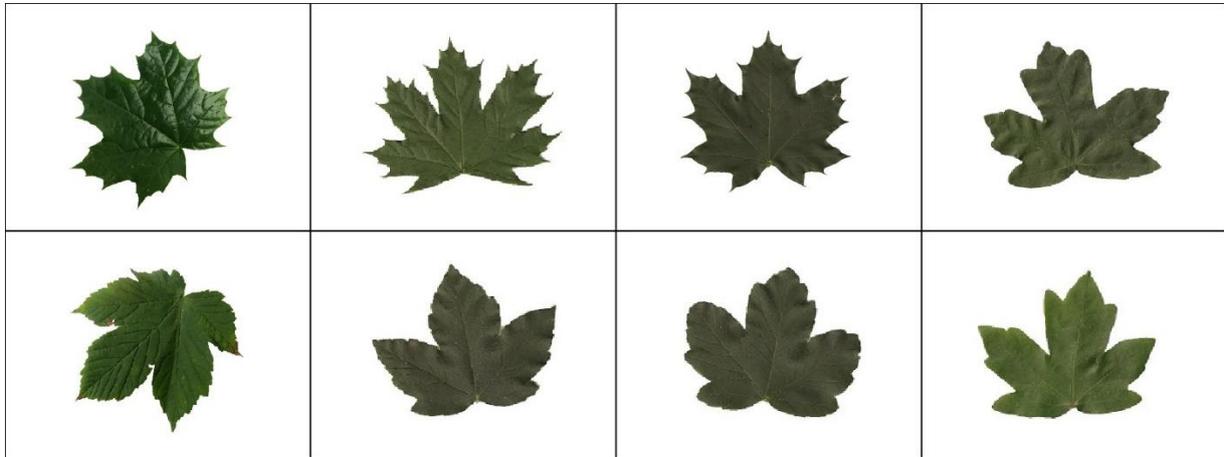


Abbildung 101: Verschiedene Ahorn-Arten aus dem BLD (Spalte 1) und dem MEW (Spalte 2-4).
Oben: Spitz-Ahorn (Spalte 1-3), Feld-Ahorn; unten: Berg-Ahorn (Spalte 1-3),
Feld-Ahorn.

Zwar ähneln sich die Spezies untereinander stark, dennoch ist das komplette Versagen bei der Klassifikation nur schwer erklärbar. Ein Ansatz ist, dass einige Features nicht für die Datenset-übergreifende Klassifikation geeignet sind. Speziell der Nutzen von Farb-Features ist fragwürdig, da sich die Färbung der einzelnen Blätter, wie in Abbildung 101 unten zu sehen ist, auch innerhalb einer Spezies stark unterscheiden kann. Auch die Binary Patterns erwiesen sich schon bei der Klassifikation innerhalb eines Datensets nicht als 100% zuverlässig. Diese Überlegungen sollen im Folgenden überprüft werden, indem die betreffenden Features entfernt werden und die Testläufe ansonsten auf dieselbe Art und Weise wiederholt werden.

7.7.4 Entfernung potentiell schädlicher Feature-Klassen

Im Folgenden soll das entwickelte System erneut überprüft werden, da sich die Datenset-übergreifende Klassifikation offensichtlich stark von der innerhalb eines Datensets unterscheidet. Dazu wird zunächst der Einfluss der Binary Patterns und der Farb-Features erneut überprüft. Anschließend werden die restlichen Feature-Klassen noch einmal getestet.

7.7.4.1 Überprüfung der Farb- und Binary Pattern-Features

Zunächst wird jeweils nur eine der zuvor als potentiell schädlich eingestuften Feature-Klassen entfernt und die Klassifikation durchgeführt. Anschließend erfolgt ein Durchlauf, bei dem beide Klassen ausgeschlossen werden. Die Ergebnisse sind in Tabelle 24 zu sehen.

Tabelle 24: Ergebnisse der Datenset-übergreifenden Klassifikation, nach Entfernung potentiell schädlicher Feature-Klassen.

1NN		Ergebnis - Komplettes System ohne			
Training	Test	-	Farb-F.	BP-F.	Farb-, BP-F.
Papier	Baum	87,30%	85,90%	91,72%	95,34%
Baum	Papier	75,97%	77,68%	95,10%	97,95%
MEW	Papier	10,93%	14,92%	46,70%	60,36%
MEW	Baum	25,99%	22,49%	33,57%	51,17%

Das bloße Entfernen der Farb-Features hat keinen signifikanten Effekt auf die erzielten Ergebnisse. Viel stärker wirkt sich der Ausschluss der Binary Pattern-Features aus. Alle Klassifikationsergebnisse steigen stark an. Bei der Verwendung des MEW als Trainings- und des BLD Papier als Testdatensatz verbessert sich das erzielte Ergebnis sogar um über 30%. Der starke Anstieg weist darauf hin, dass die Binary Patterns, zumindest in der in dieser Arbeit verwendeten Form, sich nicht zur Datenset-übergreifenden Klassifikation eignen. Der Hauptgrund dafür ist, dass die gewählten Radien mit nur acht Punkten zu spärlich besetzt sind. Ein weiteres Problem ist, dass im MEW die überwiegende Mehrheit aller Blätter einer Spezies jeweils gleich ausgerichtet ist. Das bedeutet, dass beispielsweise bei den Ahorn-Arten der Stiel jeweils senkrecht nach unten zeigt. Da auch die Binary Patterns für jeden einzelnen Radius stets dieselben Punkte betrachten, führt dies zu einer starken Verzerrung. Die Folge ist, dass die Wahrscheinlichkeit, mit der ein Blatt korrekt klassifiziert wird, stark von dessen Ausrichtung abhängt; ein Umstand, der in dieser Arbeit unbedingt vermieden werden sollte.

Entfernt man zusätzlich zu den Binary Pattern- noch die Farb-Features, so ist erneut ein deutlicher Anstieg der Ergebnisse zu beobachten. Dieser fällt erheblich höher aus als der, der bei der Entfernung der Farb-, aber Beibehaltung der Binary Pattern-Features auftrat, was auf die große Menge von Binary Pattern-Features zurückzuführen ist. Bei der Berechnung der Distanzen verlieren die wenigen Farb-Features an Einfluss. Nach einer deutlichen Reduzierung der Feature-Anzahl durch Entfernung der Binary Pattern-Features gewinnen sie wieder an Bedeutung.

Aufgrund dieser neuen Erkenntnisse muss davon ausgegangen werden, dass bei der Datenset-übergreifenden Klassifikation, auch unter Verwendung anderer Datensets, die Farb- und Binary Pattern-Features nicht nur keinerlei Mehrwert bieten, sondern dem

Klassifikationsergebnis sogar in höchstem Maße schaden. Aus diesem Grund werden sie im Folgenden keine Verwendung mehr finden.

7.7.4.2 Überprüfung der übrigen Feature-Klassen

Durch das Wegfallen der Farb- und Binary Pattern-Features hat sich das Gesamtsystem stark verändert. Außerdem wurde bereits gezeigt, dass sich das Datenset-übergreifende Klassifikationsszenario stark von dem innerhalb eines Datensets unterscheidet. Aus diesem Grunde wird im Folgenden das verbleibende Gesamtsystem erneut überprüft. Dazu wird jeweils eine Feature-Klasse entfernt und die Ergebnisse verglichen.

Tabelle 25: Ergebnisse der Datenset-übergreifenden Klassifikation, nach Entfernung der Farb- und Binary Pattern-Features und je einer weiteren Feature-Klasse.

1NN		Ergebnis des kompletten Systems ohne Farb-F., BP-F.,				
Training	Test	-	Kontur-F.	Kurvatur-F.	Hu-F.	HOCS-F.
Papier	Baum	95,34%	95,45%	95,10%	95,34%	74,24%
Baum	Papier	97,95%	97,95%	97,61%	97,72%	79,04%
MEW	Papier	60,36%	60,25%	59,68%	64,46%	15,26%
MEW	Baum	51,17%	51,17%	51,40%	55,01%	14,45%

Zunächst fällt auf, dass ein Entfernen der HOCS-Features die Klassifikationsergebnisse dramatisch verschlechtert. Sie bilden genau wie bei der Klassifikation innerhalb eines Datensets weiterhin das Herzstück des Systems.

Die Kontur- und Kurvatur-Features liefern ein gemischtes Bild. Bei der Klassifikation von am Baum fotografierten Blättern führt das Weglassen einer dieser Feature-Klassen teilweise sogar zu einer Steigerung des Klassifikationsergebnisses. Auf Papier dagegen sinken die Ergebnisse größtenteils. Da der positive Effekt der Kontur- und Kurvatur-Features gerade bei Verwendung des MEW-Datensets, was einer realistischen Anwendung sehr ähnlich ist, überwiegt, werden beide Feature-Klassen auch weiterhin verwendet.

Bei den Hu-Features ergibt sich ein wesentlich klareres Bild. Ihre Verwendung verbessert das Ergebnis in keinem der vier Testläufe und führt in den beiden realistischen Anwendungsszenarien sogar zu einer deutlichen Verschlechterung der Ergebnisse um fast 4%. Folglich finden sie im weiteren Verlauf dieser Arbeit keine Verwendung mehr. Das schlechte Abschneiden der Hu-Features bei der Datenset-übergreifenden Klassifikation ist etwas überraschend, da sie bei der Klassifikation innerhalb eines Datensets, gerade innerhalb des sehr umfangreichen MEW-Datensatzes, gute Ergebnisse liefern.

7.7.4.3 Übersicht über das finale System

Im weiteren Verlauf dieser Arbeit umfasst demnach das „komplette System“ lediglich die Kontur-, Kurvatur- und HOCS-Features. Die Gesamtanzahl der Features sinkt damit von 366 auf 172 und setzt sich wie folgt zusammen:

- 5 Kontur-Features.
- 5 Kurvatur-Features.
- 160 HOCS-Features.
- 1 ID-Feature.
- 1 Class-Feature.

7.7.4.4 Bewertung

Abschließend ist zu erwähnen, dass die am Baum fotografierten Blätter mit nur etwas über 55% eine signifikant schlechtere Klassifikationsrate aufweisen als die aus dem auf Papier erstellten Datenset. Dies kann wiederum durch den Einfluss von unterschiedlichen Winkeln und das Durchhängen der Blätter am Baum erklärt werden. Genannte Faktoren erschweren die sowieso bereits sehr anspruchsvolle Datenset-übergreifende Klassifikation noch einmal erheblich.

Da das finale System jetzt feststeht, sollen abschließend die einzelnen Feature-Klassen noch einmal evaluiert werden. Zuvor werden aber in den nächsten Abschnitten noch drei Erweiterungen eingeführt, die eine bessere Bewertung der Performanz des Systems zulassen: die Wertung mehrerer Ergebnisse, das Zusammenfassen verschiedener Spezies zu einer Gattung und ein größeres, künstlich zusammengestelltes Datenset.

7.7.5 Wertung mehrerer Ergebnisse

Trotz der Entfernung der schädlichen Feature-Klassen befinden sich die Ergebnisse, speziell bei Verwendung des MEW als Trainingsdatensatz, mit etwa 65% noch nicht in einem Bereich, den man sich für eine reale Anwendung erhofft. Daher soll im Folgenden ein Ansatz vorgestellt werden, der die Ergebnisse mit einem minimalen zusätzlichen Aufwand des Nutzers stark verbessert. Im Anschluss werden die Testläufe erneut durchgeführt und die Ergebnisse diskutiert.

7.7.5.1 Ansatz

Ein Nutzer, der ein von ihm fotografiertes Blatt identifizieren will, dürfte mit einer 60%igen Erkennungsrate nicht zufriedenzustellen sein. Allerdings spricht bei einer solchen Anwendung nichts dagegen, nicht nur das eine, beste Ergebnis zu präsentieren, sondern auch die Klassenzuweisungen, die etwas schlechter abgeschnitten haben. So ist es z. B. möglich, die fünf wahrscheinlichsten bzw. nächsten Klassen zu bestimmen, dem Nutzer jeweils einen Vertreter der entsprechenden Spezies zu zeigen und ihn selbst entscheiden zu lassen, um welche Spezies es sich handelt. Ein ähnlicher Ansatz kommt auch bei der bereits in 3.8.1 vorgestellten iOS-App „Leafsnap“ (vgl. [Kumar 2012]) zum Einsatz.

7.7.5.2 Ergebnisse und Diskussion

Die Evaluation erfolgt analog zu der im vorherigen Abschnitt vorgestellten. Es wird das komplette System verwendet, also alle Feature-Klassen, außer den Farb- und Binary Pattern-Features. Die Ergebnisse sind in Tabelle 26 zusammengefasst. Der TopX-Wert bezeichnet dabei die Anzahl der nächsten Spezies, die bei der Bestimmung der Klassifikationsergebnisse berücksichtigt werden. Beträgt dieser beispielsweise 5, so wird die Klassifikation als korrekt gewertet, wenn sich die zutreffende Klasse unter den fünf nächsten Spezies befinden. Es werden dabei nicht nur die fünf nächsten Nachbarn betrachtet, da dabei eine Spezies mehrfach vertreten sein kann. Stattdessen werden solange nächste Nachbarn hinzugenommen, bis die benötigte Anzahl unterschiedlicher Spezies, in diesem Beispiel fünf, erreicht wurde.

Tabelle 26: Ergebnisse der Datenset-übergreifenden Klassifikation, bei Variation der Anzahl der beachteten Resultate.

1NN		Ergebnisse unter Verwendung der TopX Spezies					
Training	Test	Top1	Top 2	Top 3	Top 5	Top 7	Top 10
Papier	Baum	95,34%	98,37%	99,19%	99,65%	99,77%	100%
Baum	Papier	97,72%	99,77%	99,88%	100%		
MEW	Papier	65,38%	81,09%	86,22%	91,69%	94,42%	96,13%
MEW	Baum	53,32%	72,76%	80,09%	86,50%	89,87%	93,71%

Wie erwartet verbessern sich die Werte mit steigendem TopX-Wert erheblich. Gerade die Berücksichtigung der zweit- und drittnächsten Spezies führt zu einer signifikant höheren Genauigkeit. Verantwortlich dafür ist hauptsächlich die große Anzahl von zwei bis drei sehr ähnlichen Spezies, bei denen die Blätter zwar dem Großteil der Blätter der übrigen Spezies relativ unähnlich sind, sich untereinander aber stark ähneln.

Bei der Klassifikation mit dem MEW als Trainingsdatensatz wird schon bei einem TopX-Wert von 5 eine Genauigkeit von 91,69% bzw. 86,50% erreicht. Nach Auffassung dieser Arbeit erlauben derartige Werte bereits eine sehr zufriedenstellende Nutzung in einem realen Anwendungsbereich.

In Abbildung 102 sind die Ergebnisse noch einmal als Diagramm dargestellt.

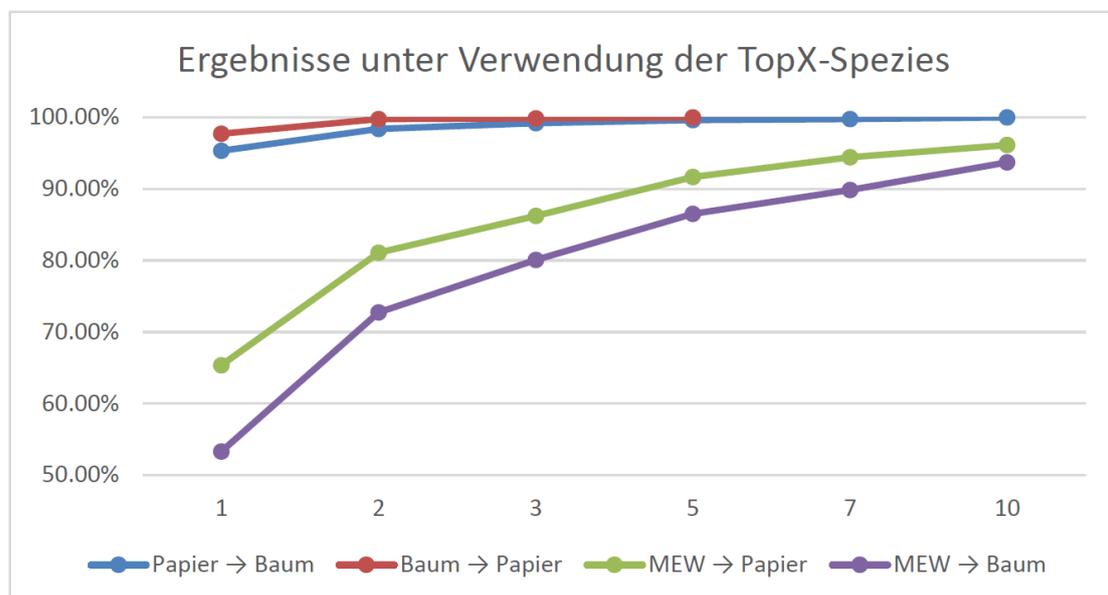


Abbildung 102: Entwicklung der Ergebnisse unter Verwendung der TopX-Spezies.

7.7.6 Zusammenfassung mehrerer Spezies

In manchen Fällen ist vielleicht eine Bestimmung der exakten Spezies gar nicht unbedingt nötig, und es genügt, die zugehörige Gattung zu bestimmen und somit die Anzahl in Frage kommender Arten erheblich einzuschränken. Dieser Ansatz soll im Folgenden näher erläutert werden. Anschließend werden erneut die Klassifikationsergebnisse bestimmt und diskutiert.

7.7.6.1 Begriffsklärung

Für den weiteren Verlauf ist es zunächst wichtig, zwei aus der Biologie stammende Begriffe zu klären: Eine *Spezies* bezeichnet eine bestimmte Baumart, beispielsweise einen Berg- oder Spitzahorn. Alle Ahorne zusammen bilden eine *Gattung*. Im Folgenden werden also jeweils mehrere Spezies zu einer Gattung zusammengefasst. [Wiki: Taxonomie]

7.7.6.2 Ansatz

Da diese Arbeit lediglich mit einem Grundwissen im Bereich der Botanik durchgeführt wird, ist die Zusammenfassung der zusammengehörigen Spezies nur bedingt exakt möglich. Sie kann zunächst lediglich basierend auf den lateinischen Bezeichnungen erfolgen. So bilden eine Gattung z. B. die Ahorne (*Acer*), deren Vertreter Namen wie *Acer platanoides* oder *Acer pseudoplatanus* tragen. Diese werden zu einer Gattung mit dem Namen *Acer* zusammengefasst und ab diesem Zeitpunkt nicht mehr unterschieden. Das führt dazu, dass beispielsweise die Klassifikation eines Blattes der Spezies *Acer platanoides* als *Acer pseudoplatanus* nicht als Fehler sondern als korrekte Klassifikation gewertet wird. Das dabei auftretende Problem wird in Abbildung 103 deutlich, in der jeweils ein Vertreter aller neun im MEW auftretenden Ahorn-Arten noch einmal zu sehen ist.

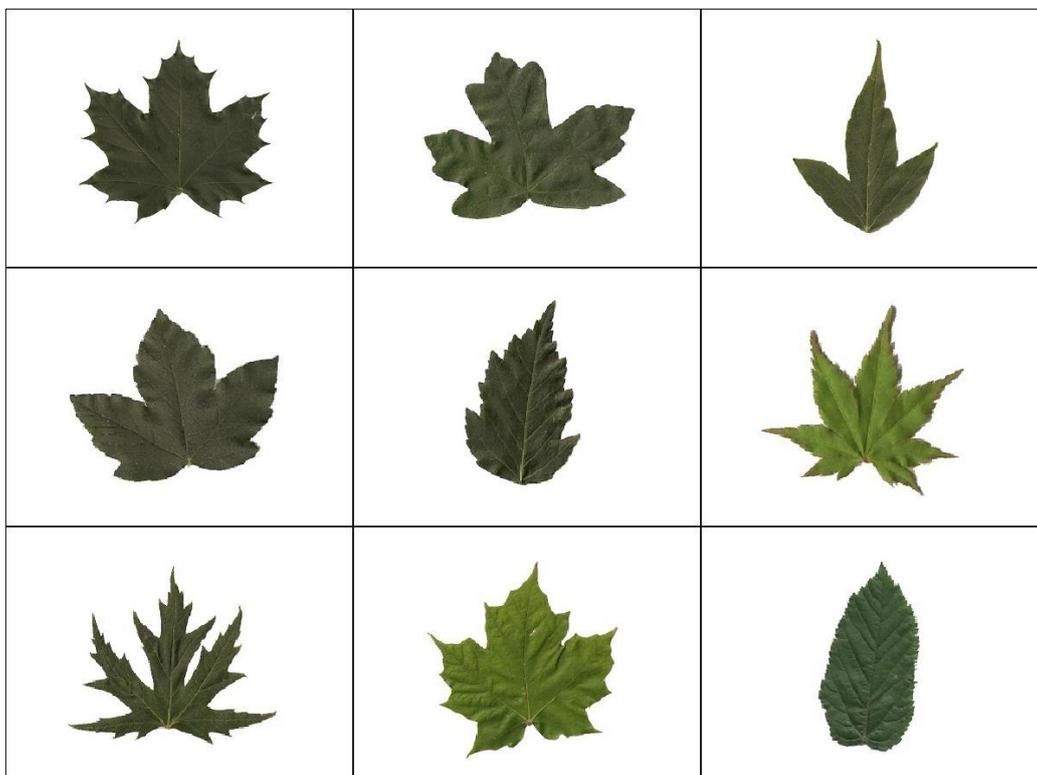


Abbildung 103: Je ein Vertreter aller neun Ahorn-Arten des MEW.

Es fällt sofort auf, dass die Zugehörigkeit zu einer Gattung keine Garantie für eine Ähnlichkeit der Blätter ist. Das bedeutet auch, dass sich nach dem Zusammenfassen einer Oberspezies die Blätter innerhalb dieser neu geschaffenen Klasse deutlich unterscheiden können. Da zur Klassifikation der K-Nearest-Neighbours-Klassifikator verwendet wird, spielt dies eine eher untergeordnete Rolle. Bei einem Klassifikator, der vor der Verwendung trainiert werden muss, könnten diese Abweichungen innerhalb der Klassen später zu einer schlechteren Klassifikationsrate führen.

Ein weiteres Problem an dieser Vorgehensweise ist, dass Spezies, trotz offensichtlicher Verbundenheit, nicht auf dem ersten Teil ihres lateinischen Namens übereinstimmen müssen. So lässt sich z. B. aus der wissenschaftlichen Bezeichnung der Hain-Buche (*Carpinus betulus*) und der Rot-Buche (*Fagus sylvatica*) keinerlei Zusammengehörigkeit ablesen. Demnach werden diese beiden Spezies auch nicht zusammengefasst.

Es ist allerdings davon auszugehen, dass das in dieser Arbeit angewandte Vorgehen insgesamt eine recht gute Zusammenfassung liefert.

Auf dem MEW-Datensatz führt der eben vorgestellte Ansatz zu einer Reduktion von 153 Spezies auf 95 Gattungen. Im BLD werden lediglich die beiden Ahorne und die beiden Eichen zusammengefasst, was zu einer Reduktion von zehn Spezies auf acht Gattungen führt.

7.7.6.3 Ergebnisse und Diskussion

Die Ergebnisse der Klassifikation mit zusammengefassten Datensets sind in Tabelle 27 zu sehen. Diese neu kreierten Datensets sind mit einem „+“ markiert. Zum Vergleich sind auch die Ergebnisse aus den vorherigen Testläufen, ohne die Zusammenfassung von Spezies, nochmals aufgeführt.

Tabelle 27: Ergebnisse der Datenset-übergreifenden Klassifikation, bei Variation der Anzahl der beachteten Spezies vor und nach (+) Zusammenfassung der Gattung.

1NN		Ergebnisse unter Verwendung der TopX Spezies			
Training	Test	Top1	Top2	Top5	Top10
Papier	Baum	95,34%	98,37%	99,65%	100%
Baum	Papier	97,72%	99,77%	100%	
Papier+	Baum+	96,15%	98,95%	99,88%	100%
Baum+	Papier+	97,95%	99,77%	100%	
MEW	Papier	73,50%	81,09%	91,69%	96,13%
MEW	Baum	53,32%	72,76%	86,50%	93,71%
MEW+	Papier+	75,74%	85,19%	93,51%	97,84%
MEW+	Baum+	68,88%	79,84%	89,16%	96,62%

Wie erwartet führt die Verwendung der neuen Datensets nochmals zu signifikant besseren Ergebnissen. Mit diesem Ansatz lassen sich einem Blatt, selbst wenn dessen Spezies nicht exakt bestimmt werden kann, zwei Klassen zuordnen, denen es mit einer Wahrscheinlichkeit von etwa 80% bis 85% angehört.

Etwas verwunderlich ist das relativ niedrige erreichte Ergebnis in dem realistischen Anwendungsszenario, also den letzten beiden Zeilen. Wird lediglich der nächste Nachbar betrachtet, führt dies in nur knapp 76% der Fälle dazu, dass die korrekte Gattung zugewiesen werden kann, wenn ein Blatt auf Papier fotografiert wurde. Am Baum sind es

sogar nur knapp 69%. Dies erscheint wenig, da ja lediglich die Gattung und nicht die exakte Spezies bestimmt werden soll. Dieser Umstand soll im Folgenden genauer evaluiert werden.

7.7.6.3.1 Betrachtung der einzelnen Gattungen

Werden die Ergebnisse für die einzelnen Gattungen getrennt voneinander betrachtet, ergibt sich das in Tabelle 28 zu sehende Bild.

Tabelle 28: Einzelergebnisse für die verschiedenen Gattungen bei der Datenset-übergreifenden Klassifikation.

1NN; Trainingset: MEW			
Lateinische Bezeichnung	Deutsche Bezeichnung	BLD Papier	BLD Baum
Acer	Ahorne	83,80%	79,10%
Alnus	Erlen	70,60%	75,80%
Betula	Birken	94,30%	72,70%
Carpinus	Hain-Buche	65,40%	21,20%
Fagus	Rot-Buche	20,90%	23,90%
Populus	Pappeln	100%	94,40%
Quercus	Eichen	93,80%	90,40%
Tilia	Linden	57,50%	57,10%

Es existieren klare Unterschiede zwischen den Klassifikationsraten der einzelnen Gattungen. Besonders herausstechend sind die extrem niedrigen Ergebnisse bei der Rot-Buche. Ebenfalls verwunderlich ist die Genauigkeit bei der Gattung der Ahorne. Diese ist zwar mit etwa 80% relativ hoch, allerdings wurden bei der Erstellung des Datensets auch neun Ahorn-Spezies zusammengefasst. Demnach war ein entsprechend höheres Ergebnis zu erwarten. Dem gegenüber stehen vor allem bei den Pappeln und Eichen sehr gute Ergebnisse von teilweise deutlich über 90%. Diese Diskrepanz lässt sich wie folgt erklären: Pappeln und Eichen unterscheiden sich als Gattung stark von den übrigen Spezies des MEW-Datensets. Zwar ist die Ähnlichkeit innerhalb der jeweiligen Gattung sehr hoch, jedoch tritt z. B. die typische Eichenform bei anderen Spezies so gut wie nicht auf. Dies führt zu sehr guten Ergebnissen bei der Klassifikation. Ganz anders verhält es sich bei der Rot-Buche, die bei der hier vorgenommenen Zusammenfassung ihre eigene Gattung definiert. Sie ist durch ihre unauffällige Form, d. h. eiförmig, etwas spitz zulaufend, glatter Rand, nach wie vor sehr anfällig für Fehldetektionen. Diese Problematik wird in Abbildung 104 deutlich. Die in der ersten Zeile zu sehenden Vertreter der Rot-Buche ähneln den Blättern anderer Gattungen in der zweiten Zeile sehr stark. Eine Zusammenfassung bringt in diesem Fall keinerlei Vorteile.

In Zeile 3 sind Vertreter der Ahorne zu sehen, je zwei Spitz- und Berg-Ahorne. Die Blätter in Zeile 4 zeigen deutlich, dass auch außerhalb der Gattung Ahorn andere Spezies bzw. Gattungen existieren, die ebenfalls die typische Ahorn-Form aufweisen. Dies erklärt die vergleichsweise niedrige Klassifikationsrate.

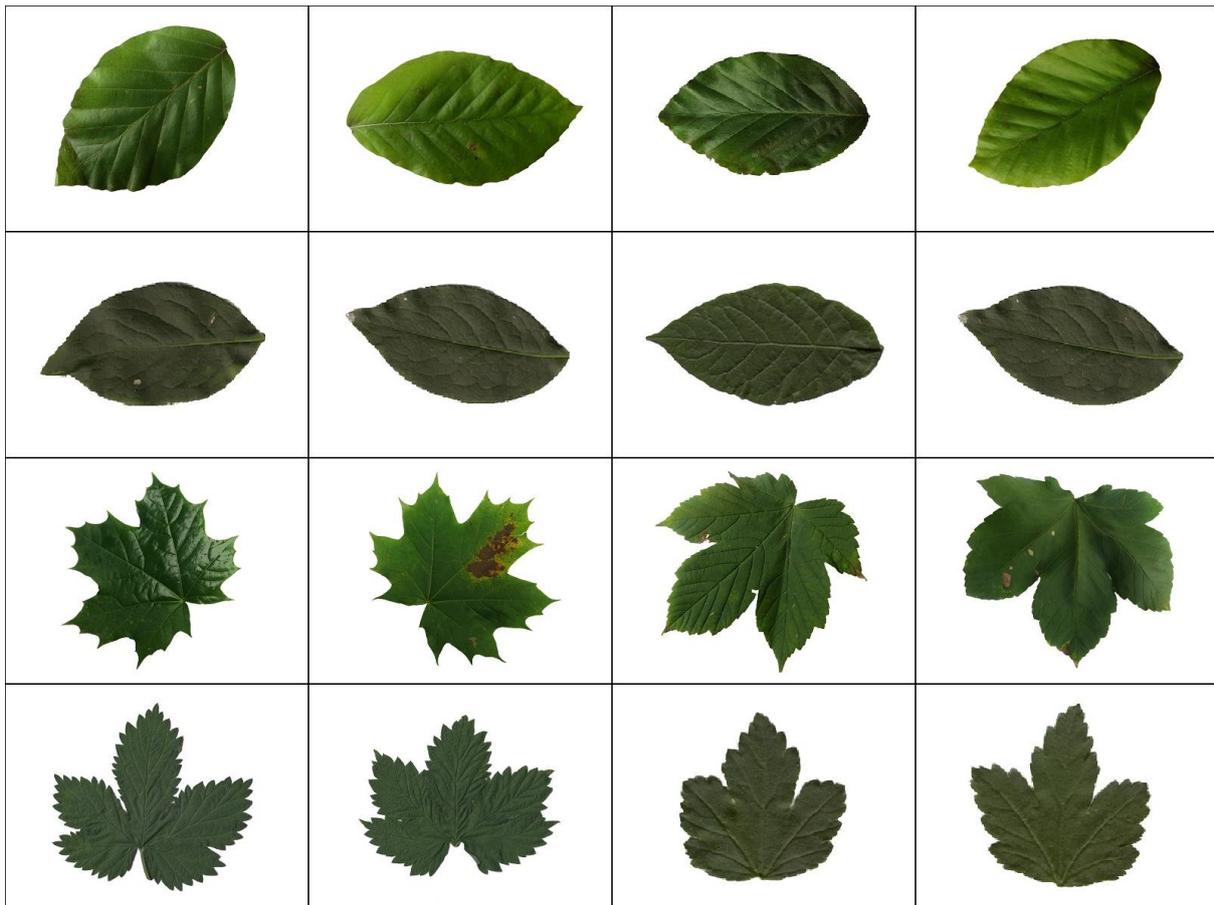


Abbildung 104: Problematik bei der Klassifikation von Gattungen. 1. Zeile: *Fagus sylvatica*; 2. Zeile: *Euonymus* (Bild 1 und 2), *Lonicera* (Bild 3 und 4); 3. Zeile: *Acer*; 4. Zeile: *Humulus lupulus* (Bild 1 und 2), *Ribes alpinum* (Bild 3 und 4).

Überdies fällt auf, dass die Hain-Buche bei der Klassifikation der Bilder, die direkt am Baum aufgenommen wurden, deutlich schlechter abschneidet, als bei den auf Papier fotografierten Blättern. Dies ist vermutlich auf die erschwerte Segmentierung am Baum zurückzuführen. Die Hain-Buche verfügt über einen fein gezackten Rand. Wird dieser nicht exakt segmentiert, was bei komplexem Hintergrund durchaus vorkommen kann, erscheint der Rand deutlich glatter und das Blatt ähnelt stark einer Rot-Buche, was wiederum zu den bereits erwähnten Problemen bei der Klassifikation führt. Es ist allerdings fraglich, ob sich der extrem starke Abfall um über 40% bei der Klassifikationsgenauigkeit einzig auf die Segmentierungsproblematik zurückführen lässt.

Bei den übrigen Gattungen unterscheiden sich die Ergebnisse zwischen den auf Papier und am Baum erstellten Datensets deutlich geringer.

7.7.6.3.2 Vergleich der Verwechslungsmatrix mit und ohne Zusammenfassung der Spezies

Zum Abschluss soll der Effekt der Zusammenfassung der Spezies verschiedener Gattungen anhand der resultierenden Verwechslungsmatrizen betrachtet werden. Eine Verwechslungsmatrix ist definiert als eine Matrix, deren Spalten und Zeilen in derselben Reihenfolge mit den auftretenden Speziesnamen benannt sind. Die Werte in den Zellen geben an, wie viele der tatsächlichen Spezies (Spalte) als die Spezies der zugehörigen Zeile klassifiziert wurden. Wird also eine Instanz der Spezies in Spalte 1 korrekt, also als

Spezies der Zeile 1 klassifiziert, so wird der Wert in Zelle (1,1) um 1 erhöht. Werden 100% der Instanzen korrekt zugeordnet, sind alle Werte abseits der Hauptdiagonalen 0.

In diesem Experiment wird das BLD Papier als Testdatensatz gewählt. Als Trainingsdatensatz dient eine Teilmenge des MEW. Bei der Verwendung des kompletten MEW hätte die resultierende Matrix eine Dimension von 153x153 und wäre demnach sehr unübersichtlich. Es werden lediglich die Spezies verwendet, die entweder direkt im BLD vorkommen oder derselben Gattung einer im BLD vorkommenden Spezies angehören. Insgesamt enthält das so neu kreierte Datenset 28 Spezies aus zehn Gattungen. Ohne die Zusammenfassung der Spezies zu Gattungen wird ein Klassifikationsergebnis von 79,04% erzielt. Die zugehörige Verwechslungsmatrix ist in Abbildung 105 zu sehen.

```

a b c d e f g h i j k l m n o p q r s t u v w x y z aa ab <-- classified as
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = Acer campestre
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | b = Acer ginnala
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = Acer negundo
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = Acer palmatum
0 0 0 0 72 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = Acer platanoides
0 0 0 0 0 96 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = Acer pseudoplatanus
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = Acer saccharinum
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = Acer saccharum
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = Acer tataricum
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = Alnus alnobetula
0 0 0 0 0 0 0 0 0 0 1 76 4 0 0 0 1 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 | k = Alnus glutinosa
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | l = Alnus incana
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | m = Betula nana
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | n = Betula papyrifera
0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 62 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | o = Betula pendula
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | p = Betula pubescens
0 0 0 0 0 0 0 0 0 9 6 0 2 0 1 0 0 62 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | q = Carpinus betulus
0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 1 0 82 0 0 0 0 0 0 0 0 0 0 0 0 | r = Fagus sylvatica
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | s = Populus canescens
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 83 0 0 0 0 0 0 0 0 0 0 0 | t = Populus tremula
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | u = Quercus cerris
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | v = Quercus frainetto
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | w = Quercus petraea
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | x = Quercus pubescens
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 19 52 23 0 0 0 | y = Quercus robur
0 0 1 0 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 1 0 0 76 0 0 0 | z = Quercus rubra
0 0 0 0 0 0 0 0 0 6 3 8 0 0 0 2 8 0 0 0 0 0 0 0 0 0 0 0 0 62 17 0 0 | aa = Tilia cordata
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | ab = Tilia tomentosa

```

Abbildung 105: Verwechslungsmatrix der Klassifikation ohne Zusammenfassung der Spezies einer Gattung.

Grün markiert sind einige Fehldetektionen, die durch alleinige Betrachtung der Gattungen ausgemerzt werden können. Besonders auffällig ist die große Menge an Klassifikationsfehlern zwischen verschiedenen Eichen- (Quercus) und Linden-Arten (Tilia). In Rot sind einige Fehler gekennzeichnet, bei denen es sich um Fehldetektionen zwischen Spezies handelt, die nicht derselben Gattung angehören. Diese treten gehäuft auf bei den bekannten Arten mit wenig charakteristischen Merkmalen, wie der Hain- (Carpinus betulus) oder der Rot-Buche (Fagus sylvatica) oder der mehrfach wegen ihrer teils stark deformierten Blätter erwähnten Stein-Linde (Tilia cordata).

Nach Zusammenfassung zu Gattungen verbessert sich das Klassifikationsergebnis auf 92,72%. Abbildung 106 zeigt die neue Verwechslungsmatrix.

	a	b	c	d	e	f	g	h	<-- classified as
173	0	0	0	0	0	0	0	0	a = Acer
0	81	1	0	0	3	0	0	0	b = Alnus
1	0	69	0	0	0	0	0	0	c = Betula
9	6	1	64	0	0	0	1	0	d = Carpinus
0	8	1	0	82	0	0	0	0	e = Fagus
0	0	0	0	0	94	0	0	0	f = Populus
5	0	0	0	0	2	171	0	0	g = Quercus
7	10	9	0	0	0	0	80	0	h = Tilia

Abbildung 106: Verwechslungsmatrix der Klassifikation nach Zusammenfassung der Spezies einer Gattung.

Es fällt sofort auf, dass die oben grün markierten Fehlklassifikationen wie erwartet verschwunden sind, da fehlerhafte Klassifikationen innerhalb einer Gattung nicht mehr länger gewertet werden. Die rot markierten Fehler bestehen allerdings nach wie vor.

7.7.7 Evaluation auf künstlich kreierten Datensets

Bisher wurde das System bei der Datenset-übergreifenden Klassifikation getestet, indem ein Teil des BLD als Testset und der andere Teil bzw. das MEW als Trainingsset verwendet wurde. Gerade das letztere Szenario kommt einem realistischen Anwendungsgebiet bereits recht nahe. Allerdings existieren weit mehr als 153 Spezies, und selbst bei einem z. B. in Deutschland fotografierten Blatt kann der Nutzer nicht sicher davon ausgehen, dass es sich um einen Baum handelt, der in Deutschland bzw. Mitteleuropa heimisch ist. Der entsprechende Baum könnte beispielsweise importiert und künstlich angelegt worden sein.

7.7.7.1 Aufbau des AC-Datensatzes

Um diese Problematik abzuschwächen und gleichzeitig die Menge an klassifizierbaren Spezies zu maximieren, wurde ein Großteil der bisher vorgestellten Datensets zu einem einzigen großen zusammengefasst. Dazu wurden alle Blätter der Datensets Flavia, Foliage, MEW, ICL und SLD kombiniert. Die Blätter des BLD wurden zunächst nicht integriert, da die dort vorkommenden Spezies vollständig in den anderen Datensets bereits enthalten sind. Aus diesem Grund bietet sich das BLD optimal zu Testzwecken an. Es spricht allerdings nichts gegen eine spätere Integration, um den Datensatz um weitere Instanzen zu erweitern.

Beim Zusammenlegen der fünf Datensets und bei der späteren Klassifikation werden Vertreter von Spezies, die mehrmals vorkommen, vollkommen gleichberechtigt behandelt. Das entstandene Datenset beinhaltet 430 Spezies mit insgesamt knapp 36.000 Instanzen. Im weiteren Verlauf dieser Arbeit wird das Datenset als „AC“ (*all combined*: englisch für „alle zusammengefasst“) bezeichnet.

7.7.7.2 Ergebnisse und Diskussion

Als Testdatensets dienen die beiden Teile des BLD. Zum Vergleich sind noch einmal die erhaltenen Ergebnisse unter Verwendung des MEW als Trainingsdatenset angegeben. Die Resultate sind in Tabelle 29 zusammengefasst.

Tabelle 29: Ergebnisse der Datenset-übergreifenden Klassifikation unter Verwendung des AC als Trainingsdatenset.

1NN		Ergebnisse unter Verwendung der TopX Spezies			
Training	Test	Top1	Top2	Top5	Top10
MEW	Papier	73,50%	81,09%	91,69%	96,13%
MEW	Baum.	53,32%	72,76%	86,50%	93,71%
AC	Papier	63,10%	76,31%	87,02%	92,81%
AC	Baum	49,48%	64,26%	79,86%	87,19%

Wie erwartet fallen die Klassifikationsergebnisse unter Verwendung des fast dreimal größeren AC schlechter aus als mit dem MEW. Durch die wesentlich höhere Anzahl an Spezies gibt es deutlich mehr Arten, die sich stark ähneln und dementsprechend häufiger zu Fehlklassifikationen führen. Unter Beachtung dieser Umstände entspricht eine Verschlechterung des Ergebnisses um ca. 10% bei der Klassifikation auf Papier bzw. um ca. 4% am Baum bei alleiniger Wertung des Topergebnisses durchaus den Erwartungen. Bei einem TopX-Wert von 5 wird auf Papier ein sehr ansprechendes Ergebnis von 87% erzielt. Am Baum werden immer noch knapp 80% erreicht.

Möchte man zuverlässige Ergebnisse erhalten, bietet es sich an, die Anzahl der Spezies im Vergleichsdatensatz möglichst gering zu halten. So ist es beispielsweise unnötig, ein Blatt, das an einem natürlich gewachsenen Baum in einem deutschen Mischwald fotografiert wurde, mit den Vertretern aller Spezies aus den im asiatischen Raum zusammengestellten Datensets zu vergleichen. Auf diese Problematik wird in Abschnitt 10.2.7.2 bei der Diskussion des finalen Programmes noch näher eingegangen.

7.7.8 Einfluss der Feature-Klassen des finalen Systems

Analog zu Abschnitt 7.4 werden zunächst die Ergebnisse evaluiert, die die einzelnen Feature-Klassen alleine erzielen. Anschließend wird der Einfluss der Wegnahme derselben auf das Klassifikationsergebnis untersucht. Als Testset dient jeweils das BLD Papier, als Vergleichsdatenset das MEW und AC.

7.7.8.1 Evaluation einzelner Feature-Klassen

Die Ergebnisse sind in Tabelle 30 zu sehen. Zum Vergleich wird zusätzlich das erzielte Ergebnis des gesamten Systems angegeben und der jeweils höchste Klassifikationswert grün markiert.

Tabelle 30: Ergebnisse der Datenset-übergreifenden Klassifikation unter Verwendung einzelner Feature-Klassen.

1NN	Ergebnis unter Verwendung der TopX Spezies und des					
	MEW			AC		
Features	Top1	Top2	Top5	Top1	Top2	Top5
Kontur	18,45%	33,71%	58,54%	10,59%	19,93%	39,64%
Kurvatur	12,30%	20,73%	36,90%	8,66%	15,03%	30,41%
HOCS	63,78%	80,18%	90,66%	64,81%	77,56%	89,64%
Alle	65,38%	81,09%	91,69%	63,1%	76,31%	87,02%

Zunächst ist wiederum festzuhalten, dass die HOCS-Features klar das beste Einzelergebnis liefern. Mit großem Abstand folgen die Kontur- und schließlich die Kurvatur-Features. Außerdem wird deutlich, dass die Performanz der HOCS-Features deutlich weniger von der Größe des Vergleichsdatensets abhängt. Anders als die beiden anderen Feature-Klassen, die jeweils signifikant bessere Ergebnisse bei Verwendung des kleineren MEW liefern, erzielen die HOCS-Features fast gleichwertige Resultate im MEW und AC. Wird lediglich das Topergebnis betrachtet, erreichen sie mit dem MEW als Vergleichsdatenset sogar einen leicht höheren als mit dem AC. Dies ist auf zusätzliche Instanzen der korrekten Spezies im AC zurückzuführen, die unsicheren Kandidaten im Testdatenset stärker ähneln als die Vertreter des MEW.

Überdies zeigt sich, dass bei der Verwendung des kleineren MEW als Vergleichsdatenset das Gesamtsystem stets das beste Ergebnis liefert. Beim AC hingegen erzielen allein die HOCS-Features die Topwerte. Dies lässt darauf schließen, dass die Kontur- und Kurvatur-Features zwar relativ gut eine kleine bis mittelgroße Anzahl, in diesem Fall 153, an Spezies unterscheiden können, aber bei einer sehr großen Anzahl ihre Aussagekraft nachlässt. Die HOCS-Features dagegen erlauben auch bei einer großen Vielfalt an Spezies eine exakte Beschreibung.

7.7.8.2 Einfluss der Wegnahme einzelner Feature-Klassen

Tabelle 31 zeigt die Ergebnisse unter Wegnahme einzelner Feature-Klassen. Wiederum ist als Vergleich der erzielte Wert des Gesamtsystems sowie der tatsächliche Einfluss, also die Differenz, angegeben.

Tabelle 31: Einfluss der Wegnahme einzelner Feature-Klassen auf die Ergebnisse der Datenset-übergreifenden Klassifikation.

1NN	Ergebnis unter Verwendung der TopX Spezies und des					
	MEW			AC		
Features	Top1	Top2	Top5	Top1	Top2	Top5
Alle	65,38%	81,09%	91,69%	63,1%	76,31%	87,02%
ohne Kontur	64,24%	81,09%	91,69%	63,4%%	76,65%	87,70%
	-1,14%	0	0	+0,34%	+0,34%	+0,68%
ohne Kurvatur	63,9%	80,07%	90,77%	64,69%	77,90%	87,24%
	-1,48%	-1,02%	-0,92%	+1,59%	+1,59%	+0,22%
ohne HOCS	22,67%	35,08%	55,58%	16,06%	28,82%	47,04%
	-42,71%	-46,01%	-36,11%	-47,04%	-47,49%	-39,98%

Das im vorherigen Abschnitt gewonnene Bild wird hier bestätigt. Bei Verwendung des MEW liefert das Gesamtsystem die besten Ergebnisse. Außerdem spielen die Krümmungs-Features eine etwas größere Rolle als die Kontur-Features.

Dient das AC als Vergleichsdatenset, können beide Beobachtungen nicht mehr bestätigt werden. Die Wegnahme der Kontur- und Krümmungs-Features verbessert das Klassifikationsergebnis. Dabei fallen letztere zumindest bei Verwendung kleiner TopX-Werte stärker ins Gewicht.

7.7.8.3 Fazit

Aufgrund der erzielten Ergebnisse muss das System neu überdacht werden. Die HOCS-Features bilden das Herzstück und können keinesfalls entfernt werden. Allerdings gibt die Performanz der Kontur- und Krümmungs-Features zu denken, da beide dem Klassifikationsergebnis schaden, wenn die Anzahl der zu unterscheidenden Spezies sehr groß ist. Andererseits tragen sie bei einer kleinen und mittleren Anzahl von Spezies klar zur Gesamtklassifikation bei.

Im Folgenden wird weiterhin die Kombination aus Kontur-, Krümmungs- und HOCS-Features als Gesamtsystem verwendet. Allerdings ist zu überlegen, ob bei einer weiteren Optimierung des Systems eine Reduktion auf die HOCS-Features, zusammen mit einer erneuten Überarbeitung der verwendeten Radien, nicht die aussichtsreichere Variante wäre. Somit könnten evtl. die Ergebnisse bei der Klassifikation mit einer kleinen oder mittleren Menge an Spezies auch ohne Kontur- und Krümmungs-Features verbessert werden. Im Moment stellt allerdings diese Art der Klassifikation einen zu wichtigen Teil des Systems dar. Aus diesem Grund wird an allen Feature-Klassen festgehalten.

7.7.9 Test auf zwei unabhängig erstellten Evaluationsdatensets

Nachdem in diesem Kapitel das finale System nach und nach fertig entwickelt wurde, wird abschließend ein letztes Experiment unter realistischen Anwendungsbedingungen durchgeführt. Zu diesem Zweck wurden zwei neue Evaluationsdatensets zusammengestellt. Diese werden im Folgenden zunächst genauer vorgestellt und anschließend evaluiert.

7.7.9.1 Eckdaten der Evaluationsdatensets

Die Blätter wurden einmal auf Papier fotografiert und einmal direkt am Baum. Die Datensets beinhalten jeweils zehn Spezies, von denen wiederum je 20 Blätter fotografiert wurden. Die Spezies entsprechen denen des ursprünglichen BLD. Um maximale Vergleichbarkeit zu gewährleisten, wurden diesmal exakt dieselben Blätter für beide Sets verwendet. Es wurde also immer ein Blatt am Baum fotografiert, anschließend eingesammelt und erneut auf Papier fotografiert. Tabelle 32 gibt eine Übersicht über die beiden Datensets.

Tabelle 32: Übersicht über die verwendeten Spezies in den Evaluationsdatensets.

Spezies		# Blätter im jeweiligen Datenset	
Lateinische Bezeichnung	Deutsche Bezeichnung	Eval Papier	Eval Baum
Acer platanoides	Spitz-Ahorn	20	20
Acer pseudoplatanus	Berg-Ahorn	20	20
Alnus glutinosa	Schwarz-Erle	20	20
Betula pendula	Hänge-Birke	20	20
Carpinus betulus	Hain-Buche	20	20
Fagus sylvatica	Rot-Buche	20	20
Populus tremula	Zitter-Pappel	20	20
Quercus robur	Stiel-Eiche	20	20
Quercus rubra	Rot-Eiche	20	20
Tilia cordata	Stein-Linde	20	20

7.7.9.2 Durchgeführte Experimente

Im Folgenden werden die durchgeführten Experimente erläutert und deren Ergebnisse vorgestellt und diskutiert. Zunächst wird eine simple Datenset-übergreifende Klassifikation evaluiert. Anschließend wird der Einfluss des ursprünglichen BLD auf die Ergebnisse näher untersucht.

7.7.9.2.1 Datenset-übergreifende Klassifikation

Analog zu den in 7.7.7.2 durchgeführten Experimenten werden die selbst erstellten Evaluationsdatensätze als Testdatensets verwendet. Als Vergleichsdatsensets dienen das MEW und das AC. Das auf Papier erstellte Datenset wird in zwei Ausführungen untersucht: einmal automatisch und einmal manuell segmentiert. Die Ergebnisse sind in Tabelle 33 zu sehen.

Tabelle 33: Ergebnisse der Datenset-übergreifenden Klassifikation unter Verwendung der Evaluationsdatensätze.

1NN		Ergebnisse unter Verwendung der TopX Spezies			
Training	Test	Top1	Top2	Top5	Top10
MEW	EvalPapier	73,5%	87,5%	98,0%	100%
MEW	EvalAuto	66,5%	82,0%	94,0%	99,0%
MEW	EvalBaum	60,5%	78,5%	93,5%	99,0%
AC	EvalPapier	68,5%	86,5%	94,0%	97,0%
AC	EvalAuto	63,0%	78,0%	90,5%	96,5%
AC	EvalBaum	58,0%	75,5%	86,5%	92,5%

Die Ergebnisse liegen fast ausnahmslos höher als bei der Verwendung des ursprünglichen BLD, teilweise sogar deutlich. So steigen die Klassifikationsraten unter Beachtung der besten fünf Spezies jeweils im Schnitt um etwa 7% an. Selbst bei der schwierigsten Aufgabenstellung, also mit dem AC als Vergleichsdatsenset und den am Baum fotografierten Blättern als Testset, werden letztere mit einer Genauigkeit von über 86% korrekt klassifiziert. Dies stellt ein sehr gutes Ergebnis dar.

Die verbesserten Resultate lassen sich durch die Art der Aufnahmen erklären. Während bei der Zusammenstellung des BLD auch stark deformierte Blätter fotografiert und verwendet wurden, wurde bei den Evaluationsdatensets darauf geachtet, realistische Bedingungen zu schaffen. Ein Nutzer, der daran interessiert ist zu erfahren, welcher Spezies ein Baum angehört, wird zur Klassifikation sicherlich kein Blatt verwenden, das beispielsweise nur noch zu 60% vorhanden ist. Es ist allerdings klar zu erwähnen, dass abgesehen davon kein Blatt aussortiert wurde. Kleinere Schäden oder weniger ausgeprägte, charakteristische Merkmale wurden ignoriert und die betreffenden Blätter dennoch verwendet.

7.7.9.2.2 Einfluss der Wachstumsregion auf die Klassifikation

Die Evaluationsdatensets wurden in derselben Region zusammengestellt wie das ursprüngliche BLD, allerdings knapp ein Jahr später. Zur Überprüfung des Einflusses der Region auf die Ähnlichkeit der Blätter wurde folgendes Experiment durchgeführt:

Dem AC werden die Instanzen aus dem BLD Papier hinzugefügt und die Testläufe aus dem vorherigen Abschnitt wiederholt. Tabelle 34 zeigt die erhaltenen Ergebnisse.

Tabelle 34: Einfluss der zusätzlichen Verwendung der Instanzen aus dem BLD Papier im Vergleichsdatenset.

1NN		Ergebnisse unter Verwendung der TopX Spezies			
Training	Test	1	2	5	10
AC	EvalPapier	68,5%	86,5%	94,0%	97,0%
	EvalAuto	63,0%	78,0%	90,5%	96,5%
	EvalBaum	58,0%	75,5%	86,5%	92,5%
AC + Papier	EvalPapier	88,0%	95,0%	99,0%	100%
	EvalAuto	81,0%	89,0%	100%	
	EvalBaum	82,5%	90,5%	94,5%	97%

Es lässt sich sofort feststellen, dass das Hinzufügen der Instanzen aus dem BLD die Ergebnisse gravierend verbessert. Bei alleiniger Betrachtung des Topergebnisses steigen die Klassifikationsraten auf Papier um knapp 20% an, am Baum sogar um fast 25%. Erreicht wurden derartige Steigerungen lediglich durch das Erweitern des AC um 878 weitere Instanzen. Bei der Größe des ursprünglichen AC von 35.981 Blättern entspricht das nicht einmal einer Zunahme von 2,5%. Dies zeigt den erheblichen Einfluss, den die Region auf die Wuchsform der Blätter hat, selbst wenn ein volles Jahr zwischen den Aufnahmen liegt.

Auffällig ist außerdem, dass die automatische Segmentierung in diesem Experiment früher die 100%-Marke erreicht als die manuelle. Verantwortlich dafür sind kleinere Segmentierungsfehler, die zufälligerweise eher zu korrekten Klassifikationen führen.

7.8 Evaluation der Klassifikation von automatisch segmentierten Blättern

In Abschnitt 4.7.2 wurde gezeigt, dass die automatische Segmentierung auf Papier sehr gute Ergebnisse liefert. Beim BLD betrug der durchschnittliche Fehler lediglich 4,10%. Im Folgenden soll die Auswirkung dieser leichten Fehlsegmentierung untersucht werden. Zuerst werden die erzielten Ergebnisse diskutiert, anschließend die auftretenden zusätzlichen Fehler bei der automatischen Segmentierung anschaulich erläutert.

7.8.1 Vergleich der Ergebnisse

Zunächst werden die zuvor erhaltenen Segmentierungsergebnisse mit Hilfe des EvalPapier überprüft. Als Trainingsdatenset dient das MEW. Zum Vergleich sind die erzielten Werte unter Verwendung des BLD mit angegeben. Es wird erneut angenommen, dass bei der manuellen Segmentierung mittels GrabCut die Fehlerrate 0% beträgt. Die Ergebnisse sind in Tabelle 35 zu sehen.

Tabelle 35: Klassifikationsergebnisse automatisch segmentierter Blätter.

Training	Test	Segmentierungs-		Ergebnis unter Verwendung der TopX-Spezies		
		Methode	Fehler (Ø)	Top1	Top2	Top5
MEW	BLD Papier	Manuell	0,00%	65,38%	81,09%	91,69%
		Automatisch	4,10%	49,11%	64,54%	80,65%
MEW	Eval Papier	Manuell	0,00%	73,50%	87,50%	98,00%
		Automatisch	3,97%	66,50%	82,00%	94,50%

Zunächst ist festzuhalten, dass die Fehlerquote der automatischen Segmentierung des EvalPapier mit 3,97% erneut sehr niedrig ist und sich im selben Bereich befindet wie beim BLD. Die geringe Abweichung ist normale Varianz.

Insgesamt erzielt das EvalPapier eine höhere Klassifikationsgenauigkeit. Dies wurde bereits durch die strengere Zusammenstellung des Datensets begründet. Die manuelle Segmentierung schneidet in beiden Fällen besser ab als die automatische, was ebenfalls zu erwarten war. Auffällig ist das abfallende Ergebnis der automatischen Segmentierung des BLD. Selbst bei Verwendung der Top5-Spezies wird lediglich ein Klassifikationswert von etwas über 80% erreicht. Die manuelle Segmentierung liegt etwa 11% besser, während die automatische Segmentierung des EvalPapier sogar einen Wert von 94,5% erzielt. Die hohe Abweichung liegt vermutlich darin begründet, dass kleine Fehler bei der Segmentierung von ohnehin bereits deformierten Blättern deren korrekte Klassifikation noch einmal erheblich erschweren. Selbst eine minimal schlechtere Segmentierung scheint in der Lage zu sein, den negativen Einfluss einer Deformation eines Blattes deutlich zu verstärken.

7.8.2 Veranschaulichung anhand von Beispielen

Abbildung 107 zeigt drei Blätter aus dem EvalPapier, die bei der eben diskutierten Datenset-übergreifenden Klassifikation korrekt zugeordnet wurden, wenn die manuelle Segmentierung verwendet wurde, bei denen die automatische Segmentierung aber zu einer Fehlklassifikation führte.

Diese Beispiele vermitteln erneut einen Eindruck über den Schwierigkeitsgrad der Klassifikationsaufgabe. Die Segmentierungsfehler sind minimal und mit dem bloßen Auge kaum zu erkennen. Lediglich bei der Stein-Linde (*Tilia cordata*) in der dritten Zeile lässt sich bei dem Ergebnis der manuellen Segmentierung (rechts) an der Spitze des Blattes ein mittelschwerer Fehler ausmachen. Bei den anderen beiden Beispielen führen offensichtlich kleinste Ungenauigkeiten beim Freischneiden des Blattrandes bereits zu einer Fehldetektion. Dabei ist zu erwähnen, dass beide Blätter Spezies angehören, die auch bei allen anderen Klassifikationsaufgaben stets für eine der höchsten Fehlerquoten sorgen: die Hain-Buche (*Carpinus betulus*) in Zeile 1 und die Rot-Buche (*Fagus sylvatica*) in Zeile 2. Diese beiden Spezies sind im Test unter Verwendung der Top5-Spezies für acht der insgesamt elf Fehldetektionen verantwortlich. Die restlichen drei entfallen auf die Stein-Linde. Daraus lässt sich folgern, dass die Auswirkungen von Fehlern bei der Segmentierung stark von der Spezies abhängen, bei der sie auftreten. So ist es beispielsweise durchaus vorstellbar, dass sich ein Fehler von 10% bei einem Vertreter der Ahorne oder Eichen deutlich weniger negativ auf die Klassifikation auswirkt als bei einer Hain-Buche. Dies wiederum liegt daran, dass sich grobe, charakteristische Formen, wie z. B. die typische Ahornform, deutlich schwerer durch Segmentierungsfehler verfälschen lassen als beispielsweise ein fein gezackter Rand.

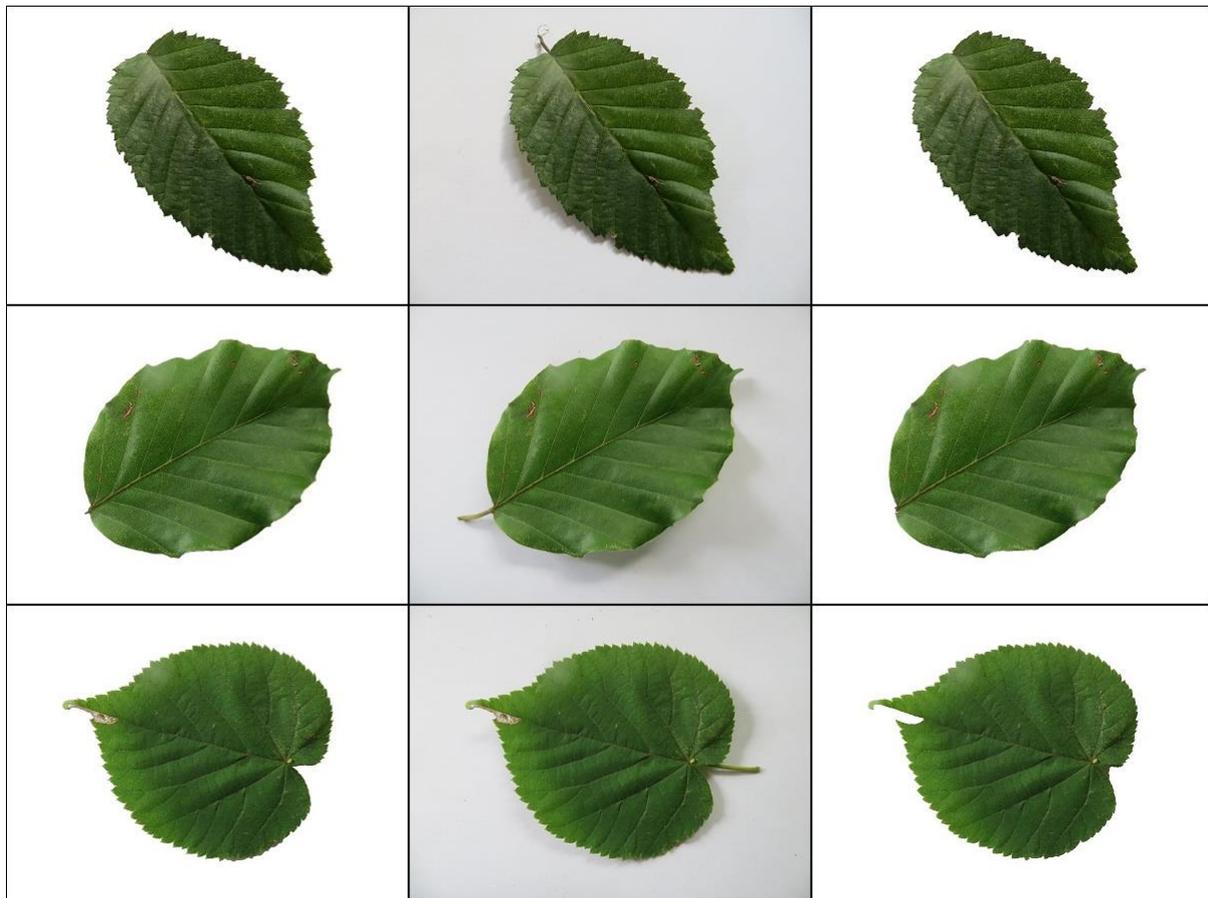


Abbildung 107: Auswirkungen der Verwendung der manuellen und automatischen Segmentierung auf das Klassifikationsergebnis. Links: Ergebnis der manuellen Segmentierung (korrekt klassifiziert), Mitte: Originalbild, rechts: Ergebnis der automatischen Segmentierung (falsch klassifiziert).

Abschließend sollen kurz die konkreten Ergebnisse für das Beispiel aus Zeile 3 in Abbildung 107 dargestellt werden. Dazu werden zunächst für die Klassifikation des

manuell segmentierten Blattes die fünf passendsten Vertreter aus verschiedenen Spezies aufgezeigt (vgl. Abbildung 108).

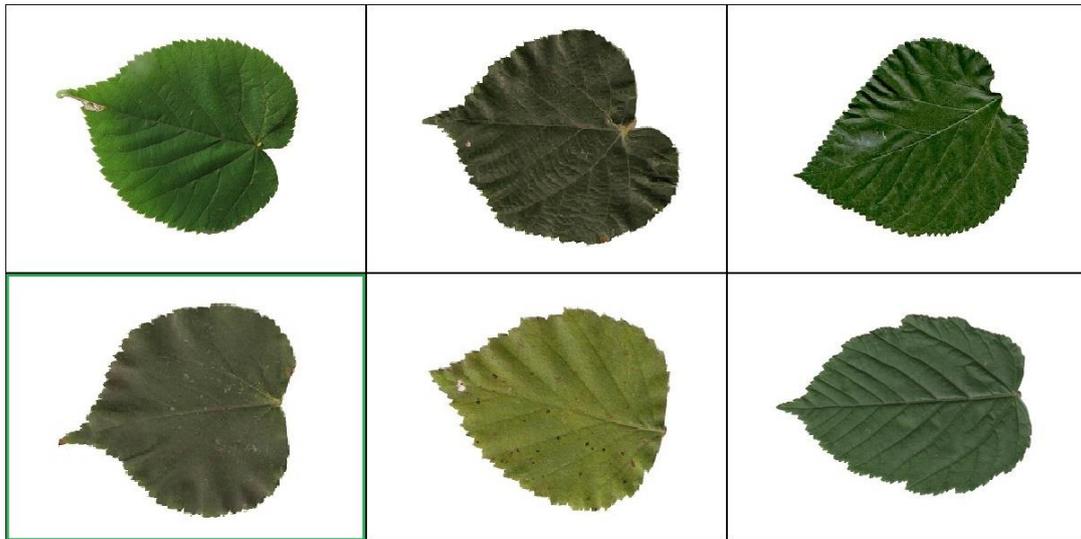


Abbildung 108: Top5-Ergebnisse bei der Klassifikation des manuell segmentierten Stein-Linden-Blattes. Oben links: Ergebnis der manuellen Segmentierung, unten links: korrekte Spezies.

Analog sind in Abbildung 109 die Ergebnisse zu sehen, die bei einer Klassifikation des automatisch segmentierten Blattes die größte Ähnlichkeit aufweisen.

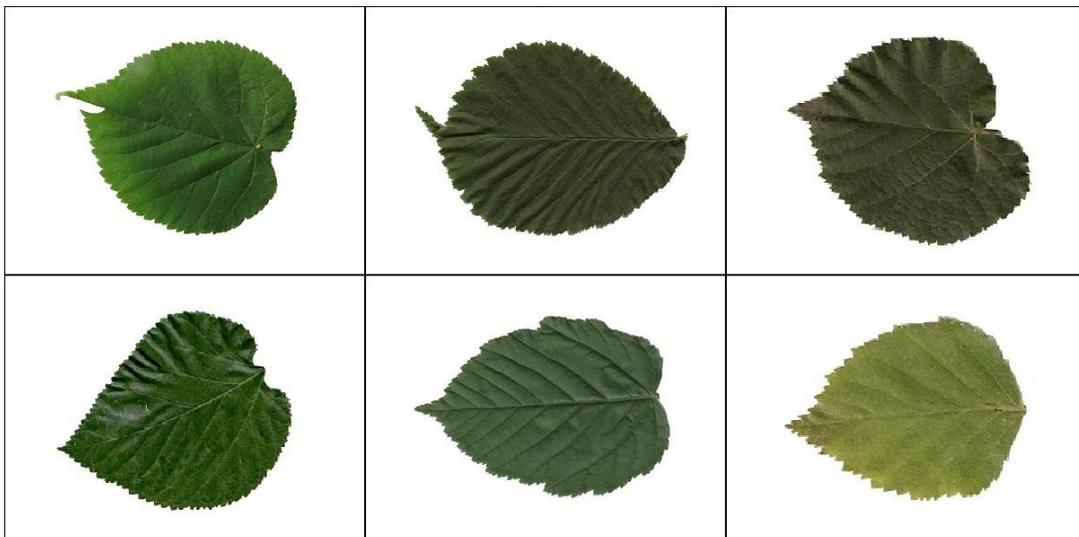


Abbildung 109: Top5-Ergebnisse bei der Klassifikation des automatisch segmentierten Stein-Linden-Blattes. Oben links: Ergebnis der manuellen Segmentierung.

Trotz großer Ähnlichkeit ist die korrekte Spezies nicht unter den besten fünf Ergebnissen zu finden. Die Klassifikation schlägt aufgrund eines geringfügigen Segmentierungsfehlers fehl.

7.9 Evaluation der Laufzeiten

Zum Abschluss dieses Kapitels soll die zeitliche Komponente des Systems evaluiert werden. Dies beinhaltet zum einen den Prozess der Feature-Extraktion, zum anderen die Dauer der Klassifikation.

Analog zu Abschnitt 4.7.2 wird folgende Hardware verwendet:

- Prozessor: Intel(R) Core(TM) i7-4770 CPU @3.40 GHz
- Arbeitsspeicher: 8,00 GB
- Betriebssystem: Windows 7 (64 Bit)

7.9.1 Zeitaufwand der Feature-Extraktion

Die Laufzeitmessung erfolgt mittels des vorsegmentierten EvalPapier-Datensets. Die Zeitmessung umfasst sämtliche Schritte vom Einlesen und der Vorverarbeitung des Bildes, über die eigentliche Feature-Extraktion, bis hin zur Ablage der gewonnenen Information in einer ARFF-Datei. Abschließend wird die verstrichene Zeit durch die Anzahl der Blätter im EvalPapier-Datenset, also durch 200 dividiert.

Untersucht wurde zunächst das entwickelte Gesamtsystem. Anschließend werden einzelne Feature-Klassen entfernt, um deren Einfluss auf die Gesamtzeit zu überprüfen. Die Ergebnisse sind in Tabelle 36 zu sehen.

Tabelle 36: Zeitaufwand der Feature-Extraktion.

Extrahierte Features	Zeitaufwand	Einfluss
Alle	0,79s	-
ohne Kontur-Features	0,76s	0,03s
ohne Kurvatur-Features	0,74s	0,05s
ohne HOCS-Features	0,26s	0,53s

Der insgesamt benötigte Zeitaufwand ist mit 0,79s sehr überschaubar. Die komplette Verarbeitung eines kleinen Datensets wie dem EvalPapier mit 200 Blättern dauert nur etwas über 2,5 Minuten. Ein sehr großes Datenset, beispielsweise das AC mit knapp 36.000 Blättern, benötigt etwas weniger als 8 Stunden.

Wie erwartet sind die HOCS-Features für den größten Teil der benötigten Zeit verantwortlich. Trotz der in 5.5.3 vorgestellten Speed-Ups nimmt ihre Berechnung beinahe 0,53s in Anspruch. Dagegen sind die Kontur- und Kurvatur-Features mit 0,03s bzw. 0,05s fast vernachlässigbar. Die restliche Zeit von etwa 0,18s entfällt zum größten Teil auf die Vorverarbeitung: Laden des Originalbildes, Anpassung der Größe, Berechnung der Graustufen- und der Binärdarstellung und Extraktion der Kontur.

7.9.2 Zeitaufwand der Klassifikation

Zur Überprüfung der Laufzeit der Klassifikation wird das EvalPapier-Datensatz als Testdatensatz verwendet. Das MEW und AC dienen als Vergleichsdatensätze. Tabelle 37 zeigt die erzielten Ergebnisse.

Tabelle 37: Zeitaufwand der Klassifikation.

# zu klassifizierende Instanzen	Vergleichsdatensatz	# Instanzen im Vergleichsdatensatz	Zeitaufwand pro Instanz
200	MEW	9.717	0,055s
200	AC	35.981	0,210s

Wie zu erwarten, hängt der Zeitaufwand zur Klassifikation einer Instanz fast linear von der Anzahl der Instanzen im Vergleichsdatensatz ab. Das AC enthält etwa 3,7mal so viele Instanzen wie das MEW, was sich in einem ca. 3,8mal längeren Zeitaufwand pro Klassifikation widerspiegelt.

Es wurden jeweils vier Läufe mit verschiedenen TopX-Werten, nämlich 1, 2, 5 und 10, durchgeführt. Dabei konnte kein nennenswerter Unterschied beim Zeitaufwand festgestellt werden. Das liegt daran, dass bei jeder Klassifikation ohnehin alle Distanzen zu den Instanzen im Vergleichsdatensatz berechnet werden müssen. Der Aufwand, anschließend die nächsten TopX Nachbarn verschiedener Spezies zu identifizieren, ist demgegenüber zu vernachlässigen.

8 Finales Programm

Nachdem das System umfassend evaluiert wurde, wird ein Java-Programm inklusive GUI entwickelt, das einem Nutzer die direkte Klassifikation eines Blattes anhand eines Fotos ermöglicht. Dieses soll im Folgenden näher vorgestellt werden. Es lässt sich grob in zwei Hauptbestandteile zerlegen: das Verarbeitungs- und das Ergebnisfenster.

Vorab ist zu erwähnen, dass es nicht das Ziel dieser Arbeit war, ein Programm zu erstellen, das eine möglichst große Anzahl Nutzer generiert und bis ins kleinste Detail optimiert ist. Vielmehr soll das Programm auf einfache Art und Weise die Leistungsfähigkeit der implementierten Segmentierung und der Feature-Extraktion, sowie der daraus resultierenden Klassifikation demonstrieren.

8.1 Das Verarbeitungsfenster

Zu Beginn des Programms wird das Startfenster geladen, wie es in Abbildung 110 zu sehen ist. Es besteht aus einem großen Label (links), auf dem das zu verarbeitende Blatt angezeigt wird. Rechts befinden sich Buttons und Auswahlelemente wie Radiobuttons und eine SpinBox.



Abbildung 110: Startfenster der GUI.

Zu Beginn spezifiziert der Nutzer, ob ein Bild geladen werden soll, das ein Blatt auf Papier oder am Baum zeigt. Erst danach ist der Load-Button verfügbar. Diese Auswahl hat nur begrenzte Auswirkungen auf den weiteren Ablauf des Programms. Vielmehr soll durch die Abfrage sichergestellt werden, dass der Nutzer sich über den Unterschied der beiden Aufnahmemöglichkeiten bewusst ist.

8.1.1 Laden eines Bildes

Zum besseren Verständnis werden im weiteren Verlauf dieses Kapitels zwei Blätter Schritt für Schritt verarbeitet. Dabei handelt es sich zum einen um ein auf Papier fotografiertes Spitz-Ahorn (*Acer platanoides*)-Blatt und zum anderen um ein direkt am Baum aufgenommenes Bild einer Zitter-Pappel (*Populus tremula*). Abbildung 111 zeigt die GUI nach dem Laden des jeweiligen Blattes.

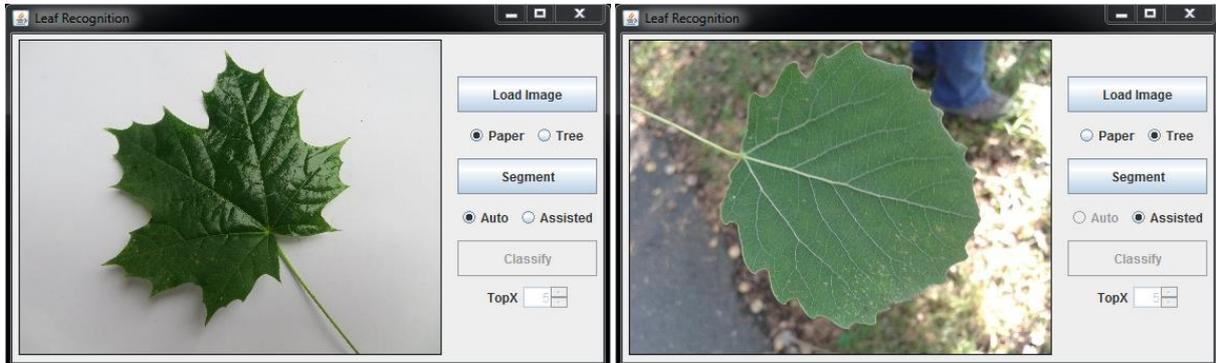


Abbildung 111: Die GUI nach dem Laden der zu klassifizierenden Blätter.
Links: Spitz-Ahorn auf Papier; rechts: Zitter-Pappel am Baum.

8.1.2 Segmentierung des Blattes

Nach dem Laden der Blätter muss die Art der Segmentierung festgelegt werden. Dabei kann ein auf Papier aufgenommenes Blatt sowohl automatisch als auch manuell durch den Nutzer mit Unterstützung des GrabCut-Algorithmus vorgenommen werden. Bei am Baum fotografierten Blättern steht die automatische Segmentierung nicht zur Auswahl.

8.1.2.1 Automatisch

Ist die automatische Methode ausgewählt, erfolgt beim Klicken des „Segment“-Buttons direkt die Segmentierung. Diese entspricht exakt dem Ablauf, der in Abschnitt 4.3 vorgestellt wurde. Die anschließende Stielentfernung erfolgt analog zu dem Algorithmus aus 4.6. Nach der Segmentierung wird das Ergebnis so angezeigt, wie es in Abbildung 112 zu sehen ist.

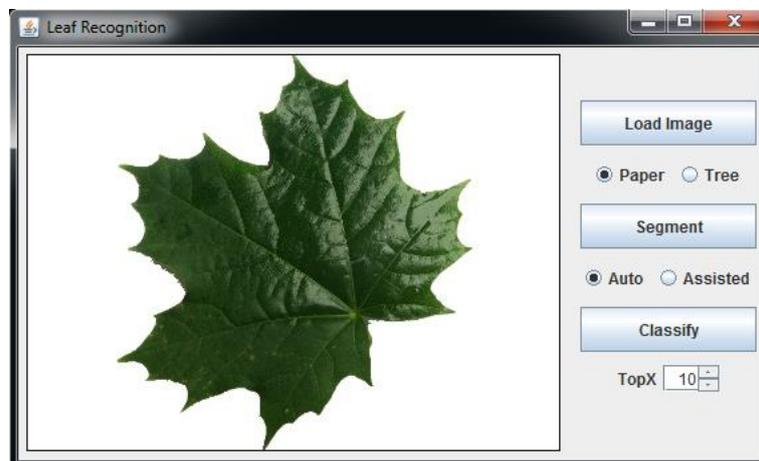


Abbildung 112: Ergebnis der automatischen Segmentierung.

Zur besseren Übersicht wird das segmentierte Blatt so vergrößert und neu positioniert, dass es den zur Verfügung stehenden Platz optimal nutzt. Dazu wird die Blattkontur extrahiert, das *StraightBoundingRect* berechnet und anschließend an die Maße des Anzeigelabels angepasst.

8.1.2.2 Manuell

Bei Auswahl der manuellen Segmentierung öffnet sich das GrabCut-Fenster. Ein Beispiel des bereits bekannten Ablaufs der Segmentierung zeigt Abbildung 113.

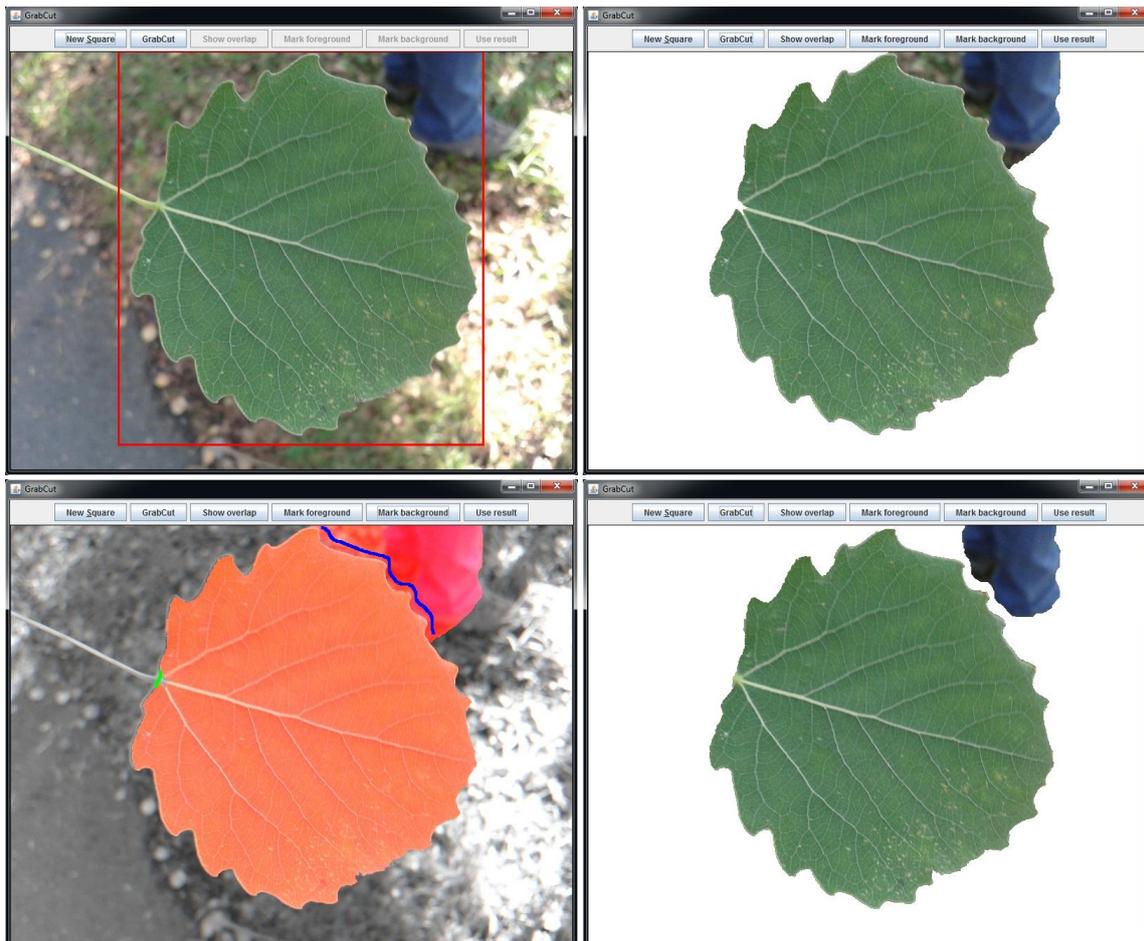


Abbildung 113: Ein möglicher Ablauf der manuellen Segmentierung mit Hilfe des GrabCut-Algorithmus. Oben: Initialisierendes Rechteck (links), erstes Segmentierungsergebnis (rechts); unten: Overlap-Anzeige und manuell markierter Vorder- und Hintergrund (links), zweites Segmentierungsergebnis (rechts).

Im hier gewählten Beispiel ist ein häufig auftretendes Phänomen zu sehen: Ein weiteres, sich stark vom restlichen Hintergrund unterscheidendes Objekt, in diesem Fall ein Teil einer Jeans rechts oben im Bild, wird neben dem Blatt ebenfalls als Vordergrund klassifiziert. Tritt dieser Fall ein, ist es nicht nötig, das Störobjekt komplett zu entfernen. Es genügt, das Blatt vollständig freizulegen, wie es rechts unten zu sehen ist. Beim Klicken auf den „Use-Result“-Button wird automatisch die größte äußere Kontur extrahiert, also die des Blattes. Alles außerhalb dieser Kontur wird im Ergebnisbild mit weißen Hintergrundpixeln überschrieben. An dieser Stelle ist zu erwähnen, dass selbstverständlich die Jeans bereits im links unten zu sehenden Schritt hätte großzügiger als Hintergrund markiert werden können. Zu Demonstrationszwecken wurde dies hier bewusst unterlassen.

Genügt dem Nutzer das Ergebnis der Segmentierung, wird es analog zu dem der automatischen Methode an das Label angepasst und angezeigt (vgl. Abbildung 114).

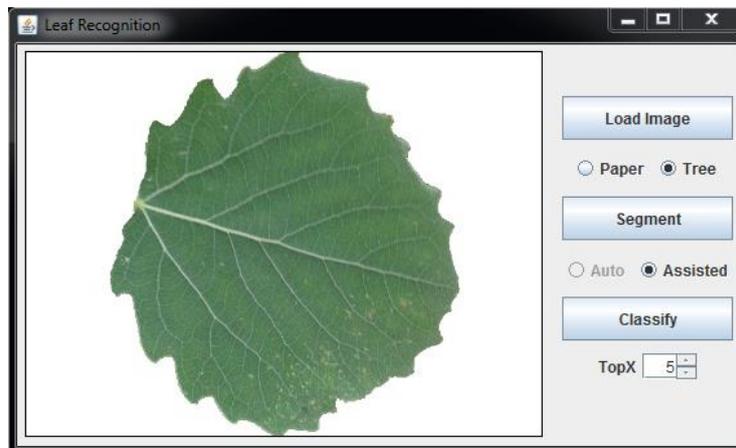


Abbildung 114: Ergebnis der automatischen Segmentierung.

Nach der erfolgreichen Segmentierung kann die Klassifikation erfolgen. Dazu muss der gewünschte TopX-Wert angegeben werden, der die Anzahl der angegebenen Resultate bestimmt. In diesem Beispiel wurde für das auf Papier fotografierte Blatt ein Wert von 10 und für das am Baum aufgenommene ein Wert von 5 gewählt. Nach Klicken des „Classify“-Buttons werden die Ergebnisse mittels des 1NN-Ansatzes kalkuliert. Als Vergleichsdatenset dient das AC mit 430 Spezies. Das Verarbeitungsfenster wird geschlossen und das Ergebnisfenster geöffnet.

8.2 Das Ergebnisfenster

Die Ergebnisse werden dem Nutzer, wie in Abbildung 115 zu sehen, präsentiert.

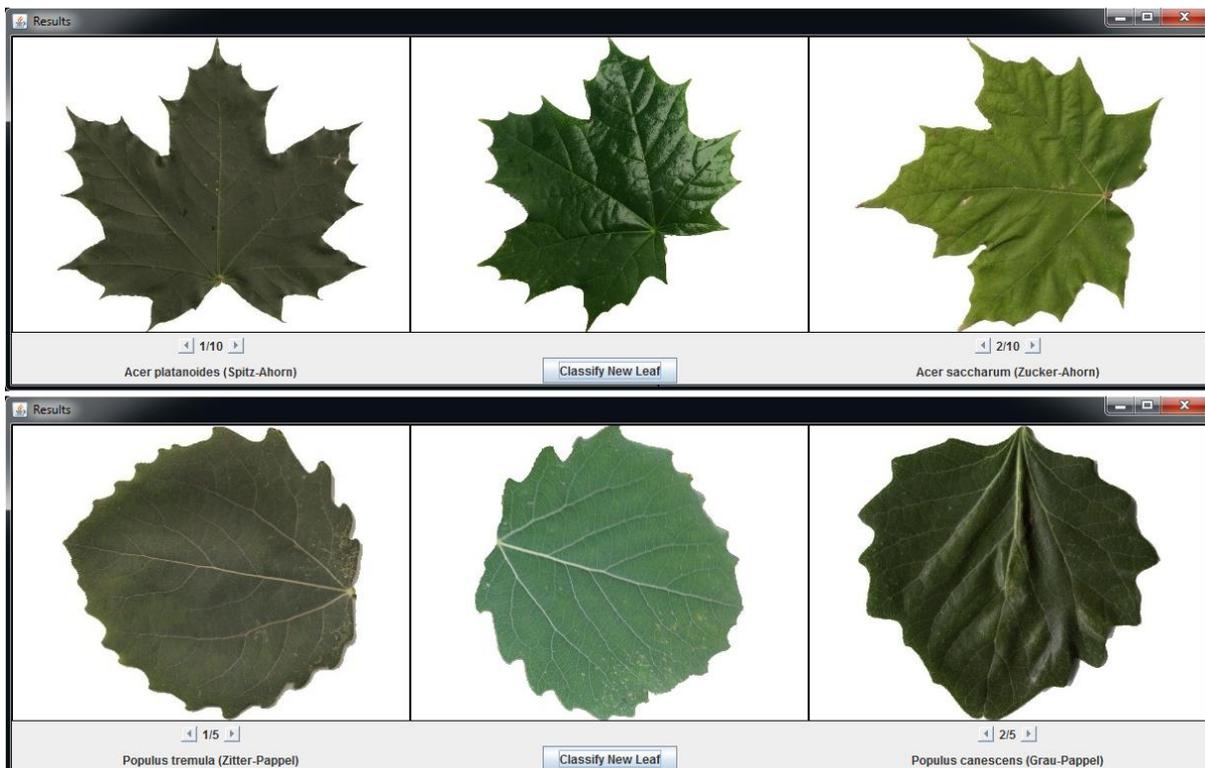


Abbildung 115: Ergebnisfenster der beiden Klassifikationen. Oben: auf Papier, unten: am Baum.

Das Fenster ist in drei Anzeigelabel unterteilt. Das mittlere zeigt das zu klassifizierende Blatt. Links und rechts wird jeweils ein Vertreter einer Ergebnis-Spezies angezeigt. Die vorderen Zahlen unterhalb der Blätter geben an, welches Ergebnis gerade angezeigt wird. 1 ist das wahrscheinlichste Ergebnis, 2 das zweitwahrscheinlichste usw.. Die hinteren Zahlen entsprechen der Gesamtzahl der Ergebnisse, also dem zuvor gewählten TopX-Wert. Die Bezeichnung „2/10“ sagt demnach aus, dass im Moment der Vertreter der zweitwahrscheinlichsten Spezies angezeigt wird und insgesamt zehn Ergebnisse berechnet wurden. Zu jeder Spezies wird der wissenschaftliche Name und, falls vorhanden, die deutsche Bezeichnung angegeben.

An dieser Stelle sei ausdrücklich erwähnt, dass die angezeigten Blätter nicht die wahren nächsten, zweitnächsten, usw. Nachbarn des zu klassifizierenden Blattes sind. Zu jeder der 430 Spezies ist ein Bild eines Repräsentanten der Spezies abgespeichert und wird bei Bedarf in die GUI geladen. Dies kann selbstverständlich dazu führen, dass der Repräsentant dem zu klassifizierenden Blatt deutlich weniger ähnelt, als das Blatt, das für das Klassifikationsergebnis verantwortlich ist. Alle im Vergleichsdatenset verfügbaren Instanzen jeweils mit Bild abzuspeichern, ist allerdings auch keine erstrebenswerte Lösung.

Der Nutzer kann sich durch die verschiedenen Ergebnisse klicken und dabei durch direktes Vergleichen entscheiden, welcher Spezies das zu klassifizierende Blatt am wahrscheinlichsten angehört. Dabei stellt das Programm automatisch sicher, dass das linke und das rechte Label niemals gleichzeitig dieselbe Spezies anzeigen. Würde dieser Fall durch das Wechseln der Spezies in einem Label auftreten, wird die nun doppelte Spezies übersprungen und die nächste angezeigt.

Durch einen Klick auf den „Classify New Leaf“-Button gelangt der Nutzer zurück zum Startfenster der GUI und kann ein neues Bild einlesen.

9 Vergleich mit anderen Domänen

Die Klassifikation anhand optischer Merkmale spielt in einer Vielzahl von Bereichen eine wichtige Rolle. In diesem Kapitel sollen drei ausgewählte Anwendungen behandelt werden: die Gesichtserkennung, die Klassifikation handschriftlicher Ziffern und die Detektion von potentiellen Tumoren in Mammographien. Dabei werden die Anwendungen jeweils nur kurz vorgestellt. Das Hauptaugenmerk soll auf dem Vergleich der zum Einsatz kommenden mit den in dieser Arbeit vorgestellten Methoden liegen. Dabei werden jeweils nur ausgewählte Herangehensweisen betrachtet, da eine ausführliche Übersicht über alle Ansätze im Rahmen dieser Arbeit nicht möglich ist. Speziell werden Methoden betrachtet, die im hier erstellten System zur Blatterkennung ebenfalls verwendet wurden oder diesen zumindest ähneln.

9.1 Gesichtserkennung

Die Identifikation von Gesichtern in Fotos und Videos ist eine der am häufigsten genutzten Klassifikationsanwendungen anhand optischer Merkmale. Dabei muss zwischen zwei Anwendungsfällen unterschieden werden:

Detektion von Gesichtern: Ziel ist es, in einem Bild alle sichtbaren Gesichter zu detektieren und zu markieren. Diese Anwendung ist heutzutage weit verbreitet und beispielsweise bei der Kamera eines jeden Smartphones zuschaltbar.

Identifikation von Personen: Anders als bei der Detektion von Gesichtern, reicht es bei dieser Anwendung nicht aus zu unterscheiden, ob ein oder mehrere Gesichter zu erkennen sind oder nicht. Vielmehr soll die zu sehende Person möglichst eindeutig identifiziert werden. Diese Technik kommt vor allem in Bereichen zum Einsatz, in denen eine Vielzahl von Personen verkehrt, z. B. an Flughäfen oder bei Massenveranstaltungen wie Sportereignissen oder Demonstrationen. Die in der Menge detektierten Gesichter werden mit einer Datenbank verglichen. So ist es möglich, vollkommen automatisch als gefährlich eingestufte Individuen zu identifizieren.

Im Folgenden soll sich auf den Bereich der Identifikation von Personen beschränkt werden. Es wird zunächst eine Auswahl von Ansätzen kurz vorgestellt und anschließend die Übertragbarkeit der Methoden diskutiert.

9.1.1 Methoden

Die Vorverarbeitung verläuft bei den meisten Herangehensweisen identisch: Jedes Bild wird hinsichtlich Rotation, Skalierung und ggf. Helligkeit normiert.

Zur Feature-Extraktion wurden in der Vergangenheit zahlreiche mögliche Herangehensweisen entwickelt, die gute Ergebnisse erzielten: Neben den im Folgenden näher erläuterten geometrischen Features und Binary Patterns, kommen häufig Verfahren wie die Hauptkomponentenanalyse, bei der eine Vielzahl statistischer Variablen durch eine (deutlich) geringere Anzahl ihrer Linearkombinationen abgebildet wird (vgl. [Wiki: PCA]), zum Einsatz. Weitere bekannte Ansätze nutzen einen Graph-Matching-Algorithmus, der markante Punkte als Knoten und deren Verbindungen als Kanten modelliert. Beim Vergleich von Gesichtern wird versucht, den Graphen so zu positionieren, dass der Wert einer Ähnlichkeitsfunktion des neuen und des ursprünglichen Graphen maximiert wird (vgl. [Lu 2003]).

9.1.1.1 Aus markanten Punkten extrahierte geometrische Features

Der klassische Ansatz der Gesichtserkennung definiert zunächst auffällige Punkte, wie beispielsweise den Mittelpunkt der Augen oder des Mundes. Eine mögliche Auswahl ist in Abbildung 116 zu sehen.

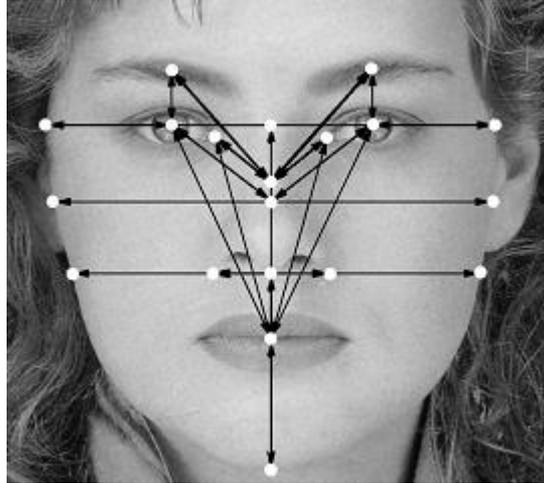


Abbildung 116: Eine Auswahl von markanten Punkten und Distanzen. [Starovoitov 2002]

Anschließend werden die Features extrahiert. Diese beschreiben hauptsächlich Abstände zwischen den zuvor definierten Punkten, aber auch von den Verbindungslinien eingeschlossene Flächen.

Die Nachteile dieser Herangehensweise werden schnell deutlich: Wird ein Punkt falsch lokalisiert, kann das zu schwerwiegenden Verfälschungen einer Vielzahl von Features führen. Überdies können schlechte Bildqualität und Haare, sowohl ins Gesicht hängende Kopfbehaarung als auch Bartwuchs, das Ergebnis bedeutend verschlechtern.

[Starovoitov 2002]

9.1.1.2 Binary Patterns

In Abschnitt 5.6 wurden die Binary Pattern-Features vorgestellt, die eine wichtige Rolle in einer Vielzahl von Anwendungsbereichen spielen, so auch in der Gesichtserkennung. Die in dieser Arbeit beschriebene Herangehensweise wurde dazu wie folgt erweitert:

Zum einen werden die Patterns, anders als bei der Beschreibung eines Blattes, nicht aus einem Binär-, sondern aus einem Graustufenbild extrahiert. Dazu wird bei jeder neuen Betrachtung der Nachbarschaft eines Pixels dessen Helligkeitswert als Grenzwert verwendet. Der Vorgang ist in Abbildung 117 zu sehen.

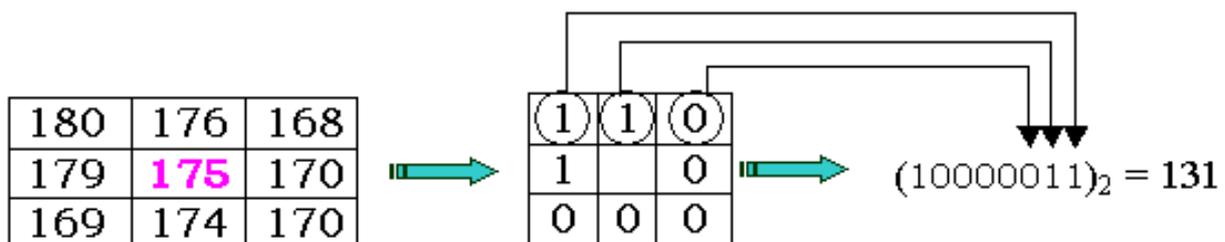


Abbildung 117: Extraktion von Binary Patterns aus einem Graustufenbild. Links: Beispiel einer Nachbarschaft von Pixeln, Mitte: Binärdarstellung, rechts: entstandener Pattern. [Pietikäinen 2005]

In diesem Fall hat der mittlere Pixel einen Wert von 175. Die Pixel mit einem Wert von 175 oder mehr werden demnach mit 1, alle anderen mit 0 kodiert. Außerdem wird aufgrund der oben erwähnten Vorverarbeitungsschritte davon ausgegangen, dass alle Gesichter, aus denen Features extrahiert werden sollen, sich in einer einheitlichen Position und Ausrichtung befinden. Demnach ist es nicht mehr nötig, durch Zusammenfassung Rotationsinvarianz bei den Binary Patterns zu erreichen.

Zum anderen wurde festgestellt, dass bestimmte Bereiche eines Gesichtes deutlich deskriptiver sind als andere. So sind beispielsweise die Augen und der Mund besonders aussagekräftig, während der äußere Bereich der Wangen fast vernachlässigbar ist. Die Information wird sich zunutze gemacht, indem jedes Bild in gleichviele Sektionen derselben Größe unterteilt wird und diesen, je nach Aussagekraft, bestimmte Gewichte zugewiesen werden. So werden z. B. die Werte, die aus dem Bereich der Augen gewonnen wurden, stärker gewichtet als die der Wangen. Eine beispielhafte Gewichtung ist in Abbildung 118 zu sehen.

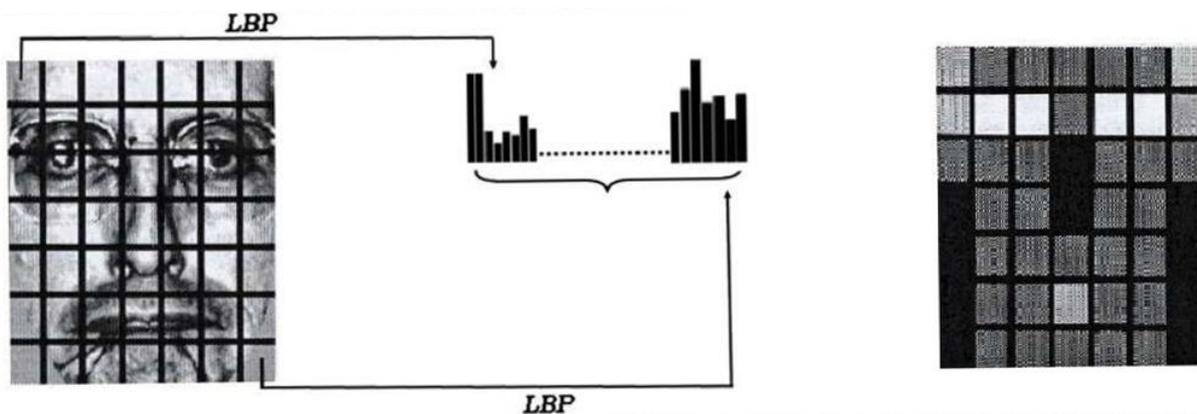


Abbildung 118: Gewichtung einzelner Sektionen. Links: Beispielhafte Unterteilung in 49 Sektionen, rechts: manuell zugewiesene Gewichte (heller=höher gewichtet). [Pietikäinen 2005]

Auffällig ist, dass dem Bereich der Nase das niedrigste Gewicht zugewiesen wurde. Dies ist verwunderlich, da nach menschlichem Empfinden die Form der Nase eine erhebliche Rolle bei der Identifikation eines Gesichtes spielt. Leider wird in der Originalarbeit keine Begründung genannt. Eine mögliche Erklärung ist, dass die Nase besonders anfällig gegenüber Lichteinfall und Aufnahmewinkel des Bildes ist und dies in der Vorverarbeitung nicht ausreichend neutralisiert werden kann.

9.1.2 Übertragbarkeit

Die Verwendung von markanten Punkten ist ein Standardansatz bei der Klassifikation unter Verwendung von optischen Merkmalen und kam demnach bereits bei der Blatterkennung zum Einsatz (vgl. [Mouine 2013]). Die erreichten Resultate waren vielversprechend. Allerdings lohnt es sich nach Ansicht dieser Arbeit nicht, die dort verwendeten Features in das entwickelte System zu integrieren, da es sehr fraglich ist, ob sie einen Mehrwert zu den bereits vorhandenen Features, speziell den HOCS-Features, liefern können.

Die vorgestellte Abwandlung der Binary Patterns würde für die Baumklassifikation nach Ansicht dieser Arbeit keinerlei Mehrwert darstellen, da bereits festgestellt wurde, dass die Färbung und damit auch die Helligkeitswerte der Blätter stark variieren können.

Neben verschiedenen Umwelteinflüssen hätten auch Verunreinigungen oder eventuelle Krankheiten starken Einfluss auf die Klassifikation.

Interessanter ist die Frage, ob die HOCS-Features einen Beitrag zur Gesichtserkennung leisten können. Prinzipiell eignen sie sich zur Beschreibung nahezu beliebiger Formen, und es ist auch durchaus vorstellbar, dass eine sehr exakte Beschreibung der äußeren Kontur von Augen, Nase oder Mund möglich ist. Jedoch sprechen zwei Punkte erheblich gegen die Verwendung der HOCS-Features zur Gesichtserkennung: Bisher sind sie ausnahmslos für die Verwendung auf Binärbildern konzipiert. Eine Anpassung analog zu der oben vorgestellten Abänderung der Berechnung der Binary Patterns ist zwar vorstellbar, muss aber keineswegs zum Erfolg führen. Das wichtigste Gegenargument ist allerdings der hohe Berechnungsaufwand. In fast allen Bereichen der Gesichtserkennung ist es von größter Wichtigkeit, dass die Feature-Extraktion und anschließende Klassifikation innerhalb kürzester Zeit durchgeführt werden kann. So ist es beispielsweise bei moderner Gesichtserkennungssoftware, wie sie an Flughäfen oder bei Großveranstaltungen zum Einsatz kommt, üblich, dass jede Sekunde 10, 15 oder sogar mehr Bilder aufgenommen und verarbeitet werden können. Dabei ist es möglich, dass in jedem Bild eine Vielzahl von Personen zu sehen ist, deren Gesicht analysiert werden muss. Dies ist mit den HOCS-Features in ihrer derzeitigen Form nicht realisierbar. Eine Anwendung in weniger zeitkritischen Bereichen, bei denen evtl. auch eine einzelne, kontrollierte Aufnahme des Gesichts möglich ist, wie z. B. bei der Zugangskontrolle zu sicherheitssensiblen Arealen, ist allerdings durchaus vorstellbar.

9.2 Klassifikation handschriftlicher Ziffern

Ähnlich wie die Gesichtserkennung ist die automatische Klassifikation von Schriftzeichen ein Gebiet, dem in der Forschung viel Aufmerksamkeit gewidmet wird und das in zahlreichen praktischen Anwendungsbereichen bereits Verwendung findet. So werden beispielsweise bei der Post Adressen, speziell Postleitzahlen, automatisch erfasst und die Briefe und Pakete dementsprechend verteilt. Ein weiteres Anwendungsgebiet ist die Verkehrsüberwachung. Durch die automatische Kennzeichenerkennung von Kraftfahrzeugen können z. B. Straßennutzungsgebühren ohne menschliches Zutun direkt erfasst werden. [Grafmüller 2012]

Im Folgenden soll sich auf Methoden beschränkt werden, die zur automatischen Erkennung von handgeschriebenen Ziffern, also der Zahlen von 0 bis 9, dienen. Für diese Anwendung existiert ein anerkannter Datensatz, der eine optimale Vergleichbarkeit verschiedener Ansätze und Systeme ermöglicht und in fast allen Publikationen verwendet wird: *The MNIST Database* (vgl. [MNIST]). Dieser besteht aus einem Trainingsset mit 60.000 und einem Testset mit 10.000 Beispielen von handgeschriebenen Ziffern. Der Schwierigkeitsgrad der zu klassifizierenden Ziffern variiert dabei stark.

9.2.1 Methoden

Genau wie bei der Gesichtserkennung müssen zunächst einige Vorverarbeitungsschritte erfolgen, die die Vergleichbarkeit der anschließend extrahierten Features signifikant erhöhen: Störendes Rauschen wird entfernt und es erfolgt eine Normalisierung von Rotation (soweit möglich) und Skalierung. [Goswami 2013]

Bei den verwendeten Methoden liefern, analog zur Gesichtserkennung, viele verschiedene Ansätze gute Ergebnisse. Dies umfasst beispielsweise geometrische

Features, ähnlich den in 5.1 verwendeten Kontur-Features, oder Bildmomente, wie die aus 5.4 bekannten Hu- oder Zernike-Momente (vgl. [Gao 2015]). Überdies finden häufig statistische Methoden Verwendung, die z. B. die Häufigkeit und Distanzen erfassen, mit denen die Kontur der zu identifizierenden Zahl horizontal oder vertikal verlaufende Hilfslinien überschreitet (vgl. [Goswami 2013]). Auch Ansätze, die die Konvexe Hülle oder die in 5.7.3 vorgestellten HOG-Features nutzen, führen zu ansprechenden Ergebnissen. Zwei Vertreter dieser beiden Gruppen sollen im Folgenden kurz näher vorgestellt werden, bevor mit dem sogenannten Deep Learning der vielversprechendste Ansatz erläutert wird.

9.2.1.1 Konvexe Hülle

Die Konvexe Hülle bzw. die daraus ableitbaren Verhältnisse stellen eine simple Möglichkeit dar, die grobe Form eines Objektes zu beschreiben (vgl. 5.1). Speziell Eigenschaften wie die Kompaktheit lassen sich direkt ablesen. In Abbildung 119 sind einige Beispiele von Zahlen und ihrer jeweiligen Konvexen Hülle zu sehen.



Abbildung 119: Konvexe Hülle von Ziffern. Oben: Verschiedene Beispiele, unten: Definition von Buchten und Seen. [Das 2015]

Das et al. (vgl. [Das 2015]) definierten die Bereiche, die komplett von der Kontur eingeschlossen sind, als Seen (englisch: *lakes*) und die Regionen, die zwischen der Konvexen Hülle und der Kontur liegen, als Buchten (englisch: *bays*).

Aus der Konvexen Hülle und den daraus resultierenden Seen und Buchten lässt sich eine Vielzahl von Features ableiten. Zum größten Teil handelt es sich dabei um simpel abzählbare Werte, wie den Konturumfang in Pixeln, die Anzahl der auftretenden Buchten und Seen und deren durchschnittliche und maximale Höhe und Breite.

Mit dieser Herangehensweise konnte auf dem oben erwähnten MNIST-Datensatz ein Ergebnis von 97,44% erreicht werden, was einen guten Wert darstellt, wenn man die

Einfachheit der Features bedenkt. Es sei allerdings erwähnt, dass bei der Ziffernerkennung auf dem MNIST-Datensatz bereits einige Male Ergebnisse von über 99% oder sogar 99,5% erzielt wurden. Einer dieser erfolgreichen Ansätze wird in Abschnitt 9.2.1.3 kurz erläutert werden.

9.2.1.2 HOG-Features

Wie in Abschnitt 5.7.3 erwähnt, bieten die HOG-Features eine hohe Flexibilität und sind vielseitig einsetzbar, so auch bei der Klassifikation handgeschriebener Ziffern.

Nach den Vorverarbeitungsschritten, also nachdem die Ziffern hinsichtlich Skalierung, Rotation und Helligkeit normiert wurden, sind optimale Voraussetzungen für die Anwendung der HOG-Features gegeben. Die Feature-Extraktion erfolgt genau analog zu dem bereits vorgestellten Ablauf: Das vorliegende Bild wird in kleine Zellen aufgeteilt. Ebrahimzadeh und Jampour wählten eine Unterteilung in insgesamt $9 \times 9 = 81$ Zellen (vgl. [Ebrahimzadeh 2014]). Es folgt die Berechnung der Histogramme der Gradientenrichtungen. Diese werden wiederum konkateniert und mittels einer SVM klassifiziert.

Das erzielte Ergebnis liegt mit 97,25% etwa auf dem Niveau des im letzten Abschnitt vorgestellten Systems.

9.2.1.3 Deep Learning

In den letzten Jahren hat ein neuer Ansatz hervorragende Ergebnisse auf dem Gebiet der Computer Vision und des Machine Learnings erzielt: das sogenannte Deep Learning.

Die Idee dahinter ist, aus einem Eingabebild immer komplexere Feature-Hierarchien zu extrahieren. Man spricht dabei von unbeaufsichtigtem Lernen, da keine Features vorgegeben werden, wie es beispielsweise in dieser Arbeit der Fall war. Dabei werden die Resultate aus niedrigeren Schichten in höheren Schichten wiederverwendet. Der gesamte Prozess ist in Abbildung 120 schematisch dargestellt.

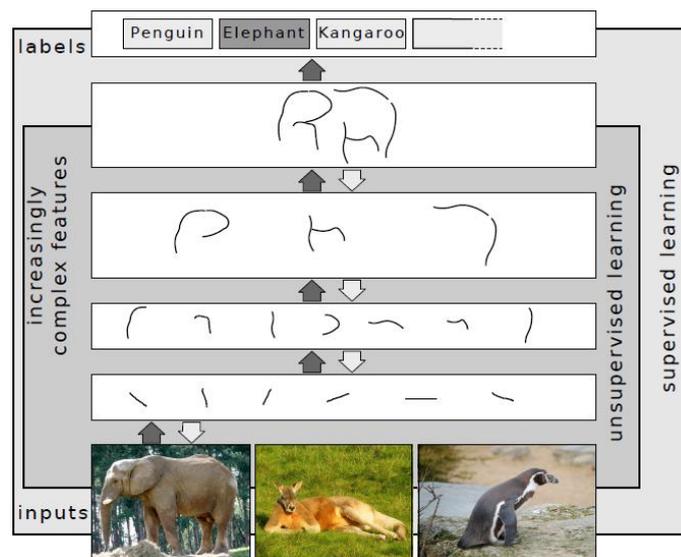


Abbildung 120: Schematische Übersicht des Schichtenweisen Lernens von Feature-Hierarchien. [Schulz 2012]

Dieser Ansatz ähnelt dem des menschlichen Sehzentrums, in dem in einer hierarchischen Struktur Sequenzen von Darstellungen des Sichtfelds berechnet werden. Dabei nimmt die

räumliche Auflösung immer weiter ab, die Feature-Komplexität dagegen erhöht sich. Eine solche Netzwerkstruktur entspricht der typischen hierarchischen Struktur von Bildern, die in Abbildung 120 zu sehen ist: Mehrere Kanten können zu Gruppen zusammengefasst werden, die z. B. ein Körperteil darstellen. Werden diese Gruppen weiter kombiniert, entsteht ein komplettes Objekt. [Schulz 2012]

Die verwendeten Neuronalen Netze können sehr große Ausmaße annehmen, was zu aufwendigen Berechnungen führt. Auf normalen CPUs würde das Training eines solchen Netzes Wochen, Monate oder sogar Jahre dauern. Daher werden GPUs verwendet, die es ermöglichen, unter Nutzung eines hohen Grades an Parallelisierung einen Speed-Up-Faktor von 50-100 zu erreichen. Somit können Prozesse, deren Berechnung vor wenigen Jahren mehrere Monate in Anspruch genommen hätte, innerhalb einiger Tage durchgeführt werden. [Ciresan 2012]

Bei der Klassifikation von handgeschriebenen Ziffern konnten Ciresan et al. (vgl. [Ciresan 2012]) mit ihrem Deep Learning-System ein Ergebnis erreichen, das der Leistungsfähigkeit des Menschen sehr nahe kommt. Auf dem MNIST Datensatz wurde eine Fehlerquote von lediglich 0,23% erzielt. Ein Mensch kann eine nur unwesentlich niedrigere Quote von etwa 0,20% erreichen. Um dieses Ergebnis besser einschätzen zu können, sind alle 23 aufgetretenen Fehler in Abbildung 121 zu sehen.

 ⁸ 3 2	 ⁵ 3 5	 ⁵ 3 5	 ⁸ 3 8	 ⁹ 4 9	 ⁵ 6 5
 ⁶ 0 6	 ⁶ 8 6	 ² 7 2	 ³ 5 3	 ⁷ 2 7	 ⁴ 7 4
 ⁶ 1 6	 ⁶ 1 6	 ⁵ 6 5	 ⁸ 2 7	 ² 7 2	 ⁴ 7 4
 ⁴ 9 4	 ² 0 8	 ⁵ 3 5	 ⁴ 9 4	 ⁷ 1 7	

Abbildung 121: Aufgetretene Fehler beim MNIST Datensatz. Jeweils rechts oben: korrekte Zahl, links unten: beste Vorhersage, rechts unten: zweitbeste Vorhersage. [Ciresan 2012]

Es fällt sofort der enorm hohe Schwierigkeitsgrad der Aufgabe ins Auge. Viele Ziffern weisen Lücken auf und sind insgesamt sehr undeutlich. Außerdem ist zu sehen, dass in 20 von 23 Fällen die zweitbeste Vorhersage zutraf.

9.2.2 Übertragbarkeit

Die Ausdrucksmächtigkeit der HOG-Features ist unbestritten. Jedoch schränkt ihre fehlende Invarianz gegenüber Rotation ihren Einsatzbereich erheblich ein. So eignen sie sich, nach Auffassung dieser Arbeit, auch nur bedingt zur Klassifikation von Blättern. Könnte sichergestellt werden, dass die zu klassifizierenden Blätter stets dieselbe Ausrichtung haben, sei es durch Anweisungen an den Nutzer oder einen nahezu 100% zuverlässigen Algorithmus, wären die HOG-Features eine interessante Ergänzung zu dem entwickelten System.

Die hier vorgestellten, aus der Konvexen Hülle extrahierten Features eignen sich nur zum Teil für eine Blattklassifikation, da bei Blättern im Normalfall keine Seen auftreten. Demgegenüber bieten die Buchten einen interessanten Ansatz. So könnte mit deren Hilfe beispielsweise bei einem gefingerten oder gelappten Blatt direkt festgestellt werden, wie viele Finger bzw. Lappen es besitzt, also ob es sich z. B. um eine drei- oder fünfflappige Ahorn-Art handelt. Das entspricht einer Information, die aus den in dieser Arbeit verwendeten Features nicht unbedingt sofort ablesbar ist.

Deep Learning-Strategien lassen sich prinzipiell auf jede Art von Computer Vision- oder Machine Learning-Problemen anwenden und erzielen stets sehr gute Resultate. Es ist zu erwarten, dass dies auch bei einer Übertragung auf die Blattklassifikation der Fall wäre.

Eine Anwendung der in dieser Arbeit vorgestellten Kontur- bzw. Krümmungs-Features auf die Ziffernerkennung erscheint, abgesehen von der Konvexen Hülle, nur bedingt sinnvoll. Die meisten Kontur-Features wie z. B. die *AspectRatio*, also das Länge-zu-Breite-Verhältnis der Kontur, haben fast keinerlei Aussagekraft, da die Zahlen sehr unterschiedlich geschrieben sein können. Gleiches gilt für die Krümmungs-Features, deren Aussagekraft, gerade im Vergleich zu den HOG-Features, vermutlich nur sehr begrenzt ist.

Interessant könnte die Integration einer Abwandlung der auf einem Kreisbogen basierenden HOCS-Features sein. Da Ziffern oft Aussparungen aufweisen und über variierende Liniendicken verfügen, dürfte allerdings nicht der Anteil der Vordergrundpixel berechnet werden, der auf einem um einen Konturpunkt aufgespannten Kreis liegt. Vielmehr müsste zum einen gezählt werden, wie häufig ein solcher Kreis von der Kontur der Ziffer geschnitten wird. Zum anderen wäre die Größe der Stücke im Verhältnis zum Umfang zu bestimmen, die bei einer Teilung durch die schneidende Kontur entstehen. Dieser Vorgang müsste für alle Konturpunkte und verschiedene Radien durchgeführt werden. Ohne ausgiebige Tests lässt sich allerdings schwer sagen, ob dieser Ansatz Erfolg verspricht.

9.3 Medizinische Anwendung: Mammographie

Mammographie ist die am häufigsten zum Einsatz kommende Methode zur Früherkennung von Brustkrebs. Es handelt sich um ein bildgebendes Verfahren, das Röntgenstrahlung nutzt, um sogenannte Mammakarzinome, also bösartige Tumore in der weiblichen, in Ausnahmefällen aber auch in der männlichen Brust zu detektieren. Mit einem Anteil von 32% ist Brustkrebs die häufigste Krebserkrankung bei Frauen in Deutschland. Im Schnitt erkrankt jede achte Frau im Laufe ihres Lebens an Brustkrebs. [Wiki: Brustkrebs] [Wiki: Mammographie]

Die korrekte Identifikation eines Mammakarzinoms hängt stark von der Dichte des umgebenden Brustgewebes ab, wie in Abbildung 122 zu sehen ist.



Abbildung 122: Zwei Beispielergebnisse einer Mammographie. Links: gesunde Brust, rechts: Mammakarzinom (rot). [Wiki: Mammographie]

Rechts ist der Tumor aufgrund des weniger dichten Brustgewebes klar zu erkennen. Befände sich links ein ähnliches Mammakarzinom, fiel die Detektion, vor allem im unteren Bereich der Brust, deutlich schwerer.

Durch eine bessere Früherkennung könnte die Sterblichkeitsrate deutlich reduziert werden. Daher beschäftigen sich zahlreiche Publikationen mit der Entwicklung von Systemen, die anhand des Ergebnisbildes einer Mammographie entweder eigenständig entscheiden können, ob ein Mammakarzinom vorliegt oder nicht (vgl. [Cardona 2014]), oder einen Experten in einer interaktiven Sitzung bei dessen Entscheidung unterstützen können (vgl. [Samulski 2010]).

Beide Ansätze sollen im Folgenden kurz vorgestellt werden.

9.3.1 Automatische Detektion

Auch bei der Tumorsuche in Mammographie-Aufnahmen erweisen sich die Binary Pattern-Features wiederum als nützlich. Cardona et al. (vgl. [Cardona 2014]) erreichten auf ihrem selbst erstellten Datensatz unter Verwendung von Binary Patterns eine Klassifikationsgenauigkeit von etwa 70%. Ein deutlich besseres Ergebnis von ca. 85% konnten sie unter Verwendung einer Segmentation-Based Fractal Texture Analysis (SFTA) erzielen, die im Folgenden kurz erläutert werden soll:

Zunächst wird die Brustkontur detektiert und ein Unschärfefilter angewendet, der mögliche Tumorkandidaten hervorheben soll. Anschließend werden, basierend auf der Graustufenwerteverteilung des Eingangsbildes, verschiedene optimale Grenzwerte berechnet. Dazu kommt eine Anpassung von Otsus Methode zum Einsatz, die analog zur Originalmethode das Bild nicht in zwei, sondern in mehrere Abschnitte gleicher Wertedistribution unterteilt. Im nächsten Schritt wird eine Menge von Binärbildern

berechnet. Dazu werden paarweise zwei der zuvor berechneten Grenzwerte ausgewählt. Pixel, deren Graustufenwert dazwischen liegt, erhalten den Wert 1, sonst 0. Durch diese Vorgehensweise ist es möglich, Informationen aus den Bereichen mit mittleren Intensitätswerten zu generieren, die unter Verwendung eines einzelnen Grenzwertes verloren gehen würden. [Costa 2012]

Für jedes der so entstandenen Binärbilder wird ein Feature-Vektor extrahiert, der die Menge an Pixeln mit dem Wert 1, den durchschnittlichen Graustufenwert dieser Pixel im Originalbild und Konturinformationen enthält. Letztere sind als Kantenbild gespeichert, in dem alle Pixel, die im zugehörigen Binärbild den Wert 1 besitzen und in deren 8-Nachbarschaft sich ein Pixel mit dem Wert 0 befindet, mit 1 markiert sind.

Zur finalen Klassifikation wird eine SVM eingesetzt.

9.3.2 Entscheidungsunterstützung

In realen Anwendungsszenarien sind selbst Methoden, die eine sehr hohe Genauigkeit garantieren, nicht autonom einsetzbar. Zu groß ist das Risiko, dass ein kritischer Fall übersehen wird. Jedoch sind natürlich auch menschliche Experten nicht unfehlbar. Daher bietet es sich an, durch eine Kombination beider Herangehensweisen die höchste Genauigkeit anzustreben und grobe Fehler weitestgehend auszuschließen. Dies wird als *Computer Assisted Diagnosis* (englisch für: Computer-gestützte Diagnostik) oder *CAD* bezeichnet.

Dabei ist die Unterstützung des menschlichen Experten auf viele Arten möglich. Durch Manipulation des Graustufenhistogramms einer Mammographie-Aufnahme kann beispielsweise der Kontrast deutlich erhöht werden, was das Auffinden eines Mammakarzinoms erheblich erleichtern kann. Die besten Ergebnisse lassen sich erreichen, wenn nicht das gesamte Bild bearbeitet wird, sondern nur ein kleiner Ausschnitt. Dieser Bereich kann wiederum automatisch detektiert oder vom Experten manuell markiert werden. [Mina 2015]

Ein solches Vorgehen kann sich als sehr hilfreich erweisen, da verdächtiges Gewebe bei weitem nicht immer so einfach zu detektieren ist wie in obigem Beispiel (vgl. Abbildung 122). Dies wird in Abbildung 123 schnell deutlich.

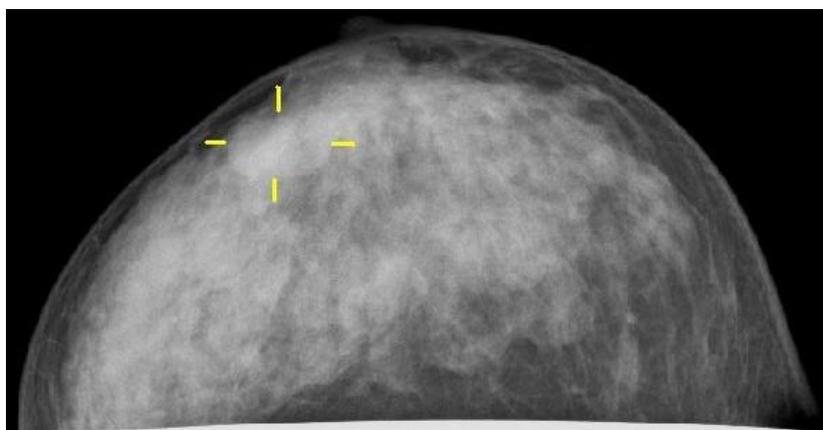


Abbildung 123: Schwer zu detektierendes Mammakarzinom. [Wiki: Mammographie]

Selbst mit den gelben Markierungen ist das Mammakarzinom nur äußerst schwierig zu erkennen, da das umliegende Brustgewebe ebenfalls sehr dicht ist.

9.3.3 Übertragbarkeit

Es ist nur schwer vorstellbar, dass die bei der Automatischen Detektion erwähnte SFTA-Methode einen Mehrwert für die Blattklassifikation liefert. Wie in dieser Arbeit gezeigt, hängen die Farbe und damit auch die Intensität der Blattpixel stark von Umwelteinflüssen ab, weshalb die inneren Pixel des Blattes im Vergleich zur äußeren Kontur nur eine sehr untergeordnete Rolle spielen.

Ebenso ist es fraglich, ob die in dieser Arbeit verwendeten Features einen Mehrwert für die Identifikation von Tumoren in Mammographie-Aufnahmen liefern können. Die Form eines möglichen Mammakarzinoms spielt oft lediglich eine untergeordnete Rolle, während andere Parameter, wie beispielsweise die Dichte, im Vordergrund stehen, gerade wenn es darum geht zu unterscheiden, ob es sich um einen Tumor handelt oder nicht. Die Form spielt erst bei der späteren, exakteren Klassifikation eine wichtigere Rolle. Es ist jedoch davon auszugehen, dass die HOCS-Features ein mindestens ebenso gutes Ergebnis liefern können wie die Binary Patterns. Da die Analyse von Mammographie-Aufnahmen keine zeitkritische Aufgabe ist, könnten die HOCS-Features durchaus eine sinnvolle Ergänzung zu bestehenden Systemen darstellen.

Die Segmentierung stellt bei der Analyse von Mammographie-Aufnahmen und bei medizinischen Anwendungen allgemein eine große Herausforderung dar. Die in dieser Arbeit untersuchten Methoden, vor allem der Watershed-Algorithmus oder eine Abwandlung der GrabCut-unterstützten Methode, könnten sich dabei als wichtiges Werkzeug erweisen.

10 Diskussion und Ausblick

Im Folgenden sollen zum einen noch einmal die erzielten Ergebnisse und erhaltenen Erkenntnisse dieser Arbeit zusammengefasst und diskutiert werden. Zum anderen wird ein Ausblick auf mögliche Verbesserungen und Erweiterungen des aktuellen Systems gegeben.

10.1 Zusammenfassung der Ergebnisse

Bei der Evaluation des Systems wurde schnell deutlich, dass ein erheblicher Unterschied zwischen der Klassifikation innerhalb eines Datensatzes und der Datenset-übergreifenden Klassifikation besteht. Dies kann als die wichtigste Erkenntnis dieser Arbeit angesehen werden. Es ist zwar aus anderen Anwendungsbereichen bekannt, dass es bei Klassifikationsaufgaben je nach verwendeter Trainings- und Testmenge zu starken Verzerrungen kommen kann, allerdings war eine dermaßen große Diskrepanz zwischen den Resultaten nicht zu erwarten. So konnten innerhalb der Datensätze teils Ergebnisse von bis zu über 99% erreicht werden, während bei der Datenset-übergreifenden Klassifikation zunächst nur sehr niedrige Bestimmungsraten von teilweise deutlich unter 30% auftraten. Dies weist darauf hin, dass Systeme, die dazu entwickelt wurden, möglichst gute Ergebnisse innerhalb einzelner Datensätze zu erzielen, sich nicht zwingend dazu eignen, in einem realistischen Anwendungsgebiet zum Einsatz zu kommen.

Bei der Evaluation der einzelnen Feature-Klassen wurde schnell klar, dass vor allem die Farb-Features bei der Datenset-übergreifenden Klassifikation keinerlei Mehrwert bieten und dem Ergebnis sogar schaden, da sie stark von den jeweiligen Umwelteinflüssen abhängen, denen ein Blatt ausgesetzt war. Ebenfalls negativ wirkten sich die Binary Pattern- und die Hu-Features aus. Während das System, das bei der Klassifikation innerhalb eines Datensatzes die besten Ergebnisse erzielte, noch aus sechs Feature-Klassen bestand, blieben nach der Entfernung der Farb-, Hu- und Binary Pattern-Features für das finale System lediglich noch die Kontur-, Kurvatur- und HOCS-Features übrig. Allein die HOCS-Features konnten zu 100% überzeugen. Sie bilden das Herzstück des Systems und erlauben eine exakte Beschreibung der äußeren Kontur des Blattes. Dies führt auch bei einer sehr großen Anzahl von bis zu 430 verschiedenen Spezies zu einer hohen Zuverlässigkeit der Klassifikation. Anders verhält es sich bei den Kontur- und Kurvatur-Features: Bei Verwendung von sehr großen Datensets fiel deren Klassifikationsgenauigkeit stark ab.

Eine deutliche Verbesserung lässt sich erreichen, indem nicht nur das beste Ergebnis, also die wahrscheinlichste Spezies, betrachtet wird, sondern beispielsweise die fünf passendsten. Mit dieser Erweiterung stiegen die Klassifikationsraten unter Verwendung des Evaluationsdatensets als Test- und des AC als Trainingsmenge auf Papier von 68,5% auf 94% und am Baum von 58% auf 86%. Weitere Steigerungen konnten durch die Zusammenfassung einzelner Spezies zu Gattungen erzielt werden.

Wie erwartet, werden die am Baum fotografierten Blätter insgesamt etwas schlechter klassifiziert als die gescannten oder auf Papier aufgenommenen. Dies liegt in Schwierigkeiten bei der Aufnahme der Fotos begründet: Die Blätter hängen durch oder werden z. B. aufgrund von Wind etwas unschärfer fotografiert.

Zur Segmentierung der Blätter wurden mehrere Methoden getestet. Als besonders effektiv erwies sich dabei der GrabCut-Algorithmus, mit dessen Hilfe sich die Blätter mit geringer Unterstützung des Nutzers sauber segmentieren lassen. Dies gilt sowohl für Blätter, die auf Papier fotografiert wurden, als auch für am Baum aufgenommene. Die manuell unterstützte Methode ist simpel und liefert hervorragende Ergebnisse. Eine Modifikation des GrabCut-Algorithmus wurde auch zur automatischen Segmentierung auf Papier eingesetzt. Der dabei auftretende durchschnittliche Fehler belief sich auf lediglich 4,10% und wirkte sich auf das Klassifikationsergebnis nur minimal aus.

10.2 Ausblick

Es sind sowohl konzeptionelle Änderungen als auch Anpassungen bei der Umsetzung des Systems denkbar. Diese sollen im Folgenden näher vorgestellt werden. Einige davon wurden im Verlauf dieser Arbeit bereits erwähnt und werden an dieser Stelle lediglich noch einmal zusammengefasst.

10.2.1 Gewichtung der Features

Bisher wurden bei der Berechnung der Distanz zwischen zwei Instanzen genau die Attributwerte verwendet, die bei der Feature-Extraktion berechnet wurden. Das hat zur Folge, dass Attribute mit einem kleineren Wertebereich im Schnitt einen geringeren Einfluss auf die Distanzberechnung und demnach auf das Ergebnis der Klassifikation haben als Attribute mit einem großen Wertebereich. Der minimale und maximale Wert eines Attributs wird dabei durch die Art und Weise der Berechnung festgelegt.

Durch Manipulation der Wertebereiche der Attribute könnte eine Verbesserung der Performanz der Klassifikation erreicht werden. So ist es beispielsweise möglich, Attribute, die als besonders wichtig und aussagekräftig angesehen werden, stärker zu gewichten als vermeintlich weniger ausdrucksstarke. Die Manipulation kann mittels einer Gewichtung durchgeführt werden. Wird z. B. einem Attribut das Gewicht 2 zugewiesen, bedeutet das, dass die bestimmte Differenz bei der Distanzberechnung verdoppelt wird.

Dabei können sowohl die Gewichte einzelner Attribute individuell festgelegt werden, als auch Attribute zu Gruppen zusammengefasst und gemeinsam gewichtet werden. Wird beispielsweise angenommen, dass die HOCS-Features, die sich auf die Blattfläche beziehen, eine höhere Aussagekraft besitzen als diejenigen, die zur Berechnung die Blattkontur verwenden, können die Gewichte der Attribute entsprechend angepasst werden: Die Flächen-Features werden stärker gewichtet, die Bogen-Features weniger stark. Dieses Vorgehen lässt sich beliebig erweitern.

Es ist zu erwarten, dass durch Optimierung der Gewichte eine merkliche Steigerung der Klassifikationsergebnisse erreicht werden kann. Allerdings ist die Gefahr eines Overfittings sehr groß, sodass die Resultate auf anderen Datensets wahrscheinlich nicht im selben Maße bestätigt werden könnten.

10.2.2 Zurückweisen von Instanzen

In vielen Bereichen der Klassifikation, wie beispielsweise der automatischen Erkennung handgeschriebener Ziffern, ist es üblich, dem Klassifikator die Möglichkeit zu geben, bestimmte Instanzen keiner Klasse zuzuordnen und damit abzuweisen [Russel & Norvig 2010]. Diese werden bei der Berechnung der Klassifikationsgenauigkeit ignoriert.

Zurückgewiesen werden Instanzen, die der Klassifikator nicht mit ausreichender Sicherheit einer bestimmten Klasse zuzuordnen vermag. Das könnten im in dieser Arbeit beschriebenen Anwendungsfall z. B. Blätter sein, die einer Spezies angehören, die nicht im Vergleichsdatenset vorkommt. Die Herausforderung ist es, die Voraussetzungen für eine Abweisung so zu definieren, dass ein möglichst gutes Klassifikationsergebnis erreicht wird, ohne dass zu viele Instanzen abgewiesen werden. Im Folgenden sollen zwei Ansätze kurz erläutert werden.

10.2.2.1 Einführen einer Grenzdistanz

Eine Möglichkeit ist es, beim 1NN-Klassifikator die Distanz zwischen der zu klassifizierenden Instanz und dem nächsten Nachbarn näher zu betrachten. Diese kann als Maß für die Konfidenz der Klassifikation dienen. Je größer die Distanz, desto unähnlicher sind sich die beiden Instanzen und desto unwahrscheinlicher ist es, dass eine korrekte Klassifikation vorliegt. Die Entscheidung, ob die zu bestimmende Instanz abgewiesen wird, könnte beispielsweise mit Hilfe eines Grenzwertes gefällt werden. Ist die Distanz größer als dieser, erfolgt die Ablehnung, andernfalls wird das Klassifikationsergebnis akzeptiert. Selbstverständlich hat der gewählte Grenzwert einen großen Einfluss auf die Klassifikationsresultate und muss dementsprechend sorgfältig ausgewählt werden. Ein Ansatz ist, einen Testlauf durchzuführen und den Mittelwert aus allen Distanzen zu bilden, die zu korrekten Klassifikationen führten. Anschließend wird der erhaltene Wert mit einem Sicherheitsfaktor multipliziert und das Ergebnis als Grenzwert verwendet. Die Bestimmung des optimalen Sicherheitsfaktors muss wiederum durch Testläufe erfolgen. Ein zu niedriger Wert führt dazu, dass zu viele Klassifikationen abgelehnt werden, während ein zu groß gewählter Grenzwert die Anzahl der akzeptierten Fehlklassifikationen erhöht.

Ähnlich wie bei der Gewichtung der Features (vgl. 10.2.1) besteht auch hier die Gefahr des Overfittings.

10.2.2.2 Beachtung der Ergebnisdichte

Ein alternativer Ansatz ist es, nicht nur den Nearest Neighbour zu betrachten, sondern auch die späteren Ergebnisse. Die Auftrittsrates der zugeordneten Spezies unter den Topergebnissen ist ein Maß für das Vertrauen in die Klassifikation. Es ist zu erwarten, dass bei einem Blatt, das mit hoher Konfidenz einer bestimmten Spezies zugeordnet werden konnte, sich unter den weiteren Ergebnissen mehr andere Vertreter dieser Spezies befinden, als bei einem Blatt, bei dem sich der Klassifikator entsprechend unsicher ist.

Analog zum vorherigen Abschnitt müssen Grenzwerte definiert werden. Beispielsweise könnten die zehn nächsten Nachbarn betrachtet werden: Befinden sich darunter mindestens vier Vertreter der Spezies, die den Nearest Neighbour stellt, wird das Ergebnis akzeptiert, ansonsten abgewiesen. Eine allgemeingültige Optimierung dieser Parameter erscheint problematisch, da die Ergebnisdichte offensichtlich stark von der Anzahl sehr ähnlicher Spezies und damit indirekt von der Gesamtmenge von Spezies im Vergleichsdatenset abhängt. Auch die Anzahl vorhandener Instanzen der einzelnen Spezies spielt eine nicht zu vernachlässigende Rolle.

Ein Overfitting an den zur Optimierung verwendeten Datensatz ist sehr wahrscheinlich. Nach Auffassung dieser Arbeit ist der im vorherigen Abschnitt vorgestellte Ansatz der

Einführung einer Grenzdistanz der deutlich vielversprechendere. Allerdings könnte es sich als sinnvoll erweisen, die beiden Herangehensweisen zu kombinieren und beispielsweise den zur Berechnung des Distanzgrenzwertes verwendeten Sicherheitsfaktor in Abhängigkeit von der Ergebnisdichte festzulegen.

10.2.3 Einführen von Merkmalsklassen und Lösungsklassen

Im jetzigen System erfolgt die Klassifikation ausschließlich mit Hilfe vieler abstrakter Merkmale, die, für den Menschen schwer nachvollziehbar, direkt eine finale Lösung bestimmen. Um die Nachvollziehbarkeit zu erhöhen, wäre es möglich, Merkmals- und Lösungsklassen einzuführen. Die Lösungen würden dann nicht mehr direkt aus den Merkmalen (Features) hervorgehen, sondern aus Lösungsklassen, die wiederum zuvor aus den Merkmalsklassen generiert wurden.

Dabei könnten Lösungen basierend auf unterschiedlichen Kriterien zu Lösungsklassen zusammengefasst werden: Zum einen wäre es möglich, Blattspezies nach ihrer groben Form zu klassifizieren, also beispielsweise Ahorn- oder Eichen-förmige Blätter zu gruppieren. Dazu würden bestimmte Merkmalsklassen dienen, wie z. B. der Grad der Kompaktheit des Blattes, der wiederum aus einzelnen Merkmalen wie bestimmten Kontur-Features oder den HOCS-Features mit großem Radiusbereich hervorgeht. Dieser Schritt entspricht im Prinzip bereits einer groben Klassifikation.

Ist die Auswahl von in Frage kommenden Spezies auf eine oder wenige Merkmalsklassen begrenzt, kann die Feinklassifikation erfolgen. Dazu werden, basierend auf den Ergebnissen der Grobklassifikation, Merkmalsklassen verwendet, die eine Unterscheidung der in Frage kommenden Spezies erlauben. Wurde z. B. bei der Grobklassifikation festgestellt, dass es sich um ein Ahorn-förmiges Blatt handelt, könnten im Folgenden Merkmalsklassen verwendet werden, die speziell die feine Beschaffenheit des Randes beschreiben und die grobe Form der Blattkontur ignorieren: die HOCS-Features mit kleinem Radiusbereich. Durch die vorherige, erhebliche Reduktion der in Frage kommenden Lösungen ist eine korrekte Klassifikation nun deutlich wahrscheinlicher.

Zur Umsetzung dieser Herangehensweise wäre ein großer Aufwand vonnöten, da eine Vielzahl von möglichen Merkmals- und Lösungsklassen definiert und getestet werden müssten. Allerdings bietet der vorgestellte Ansatz nach Auffassung dieser Arbeit sehr großes Potential und viel Platz für weitere Verfeinerungen. So könnten beispielsweise bei der Feinklassifikation, je nach Ergebnis der Grobklassifikation, die Features der zum Einsatz kommenden Merkmalsklassen entsprechend gewichtet werden.

10.2.4 Zusammenfassung ähnlicher Blätter durch Hierarchisches Clustering

In Abschnitt 7.7.6 wurden mehrere Spezies zu ihren jeweiligen Gattungen zusammengefasst. Dies zog offensichtlich eine Verbesserung der Klassifikationsergebnisse nach sich. Allerdings ist diese Herangehensweise nicht optimal: Einerseits garantiert, wie bereits in Abbildung 103 auf Seite 140 zu sehen war, die Zugehörigkeit zur selben Gattung nicht zwingend eine hohe Ähnlichkeit der Blätter der einzelnen Spezies. Andererseits können sich auch Blätter von Spezies, die nicht derselben Gattung angehören, sehr stark ähneln.

Um eine aussagekräftigere Bewertung des Systems vornehmen zu können, muss demnach eine andere Methode entwickelt werden, die es erlaubt „zu ähnliche“ Spezies zusammenzufassen. Die Idee dahinter ist, dass Fehlklassifikationen von Blättern, die sich so stark ähneln, dass eine Unterscheidung praktisch nicht mehr möglich ist, nicht gewertet werden.

Als Mittel der Umsetzung bietet sich ein Hierarchisches Clustering an, dessen Grundprinzip im Folgenden kurz vorgestellt werden soll. Anschließend wird erläutert, wie sich dieses Konzept auf die Blattklassifikation übertragen lässt.

10.2.4.1 Grundprinzip des Hierarchischen Clustering

Der Grundgedanke dieser Variante des Clusterings ist es, die sich am nächsten liegenden Datenpunkte zu Clustern zusammenzufassen. Dies wird solange wiederholt, bis nur noch ein Cluster übrig ist oder eine vorgegebene Anzahl von Clustern erreicht wird. Ein Beispiel für ein Hierarchisches Clustering ist in Abbildung 124 zu sehen.

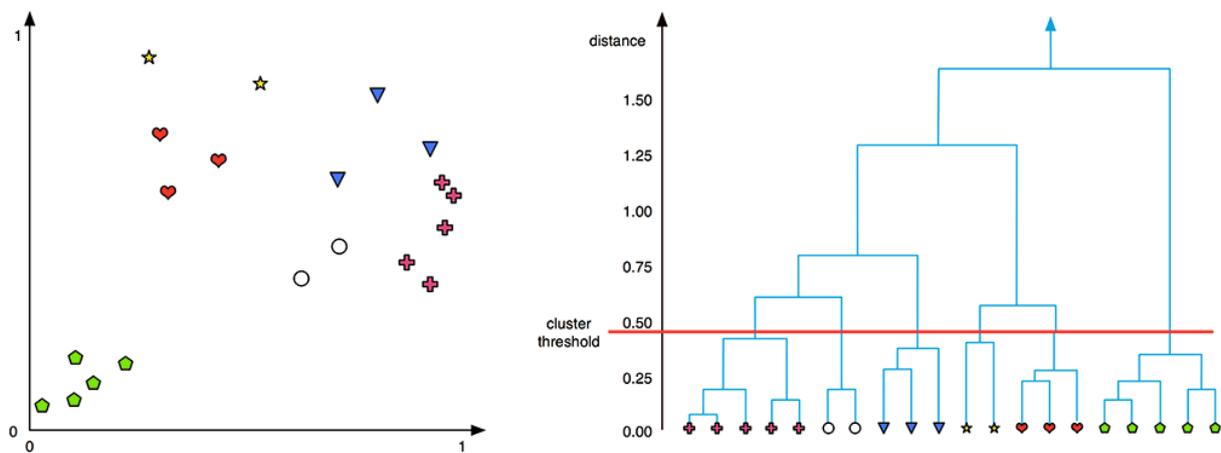


Abbildung 124: Beispiel für ein Hierarchisches Clustering.

Links: Lage der Datenpunkte, rechts: Ergebnis des Clustering. [TAI]

10.2.4.2 Übertragung auf die Blattklassifikation

In Abbildung 124 ist im unteren Teil des rechten Bildes gut der potentielle Nutzen für die Blattklassifikation zu sehen. Wird das Clustering bei einer bestimmten Distanz abgeschnitten, bleiben Gruppen von Instanzen zurück, die sich untereinander stärker ähneln als den Vertretern der anderen Cluster. Je niedriger die Grenzdistanz angesetzt wird, desto mehr Gruppen verbleiben.

Die Distanz zwischen einzelnen Blättern lässt sich mit Hilfe der Features leicht berechnen. Problematisch ist allerdings der extrem hohe Berechnungsaufwand und Speicherbedarf bei der Clusterung einer großen Anzahl von Instanzen.

Abhilfe könnte die Verwendung eines Repräsentanten für jede Spezies schaffen, der als Attributwerte die Mittelwerte aller zugehörigen Instanzen seiner Spezies erhält. Beim anschließenden Hierarchischen Clustering werden nur noch die jeweiligen Instanzen betrachtet, was den Berechnungsaufwand und Speicherbedarf drastisch reduziert.

Überdies könnten bei dieser Herangehensweise Ausreißer ausgeschlossen werden, indem jede Spezies durch einen Repräsentanten und einen Bereich definiert ist, in dem

sich Vertreter derselben Spezies befinden müssen, um beachtet zu werden. Zur Definition dieses Bereichs könnte z. B. die empirische Varianz dienen.

10.2.5 Automatische Segmentierung am Baum

Eine weitere interessante Erweiterung des Systems wäre die Option, am Baum fotografierte Blätter automatisch segmentieren zu lassen. Im bisherigen Verlauf dieser Arbeit wurde stets betont, dass dies nicht möglich ist bzw. eine automatische Segmentierung, aufgrund der verschiedenen komplexen Hintergründe, nicht zuverlässig genug funktionieren könne. Davon wird nach wie vor ausgegangen. Allerdings ist es durchaus denkbar, dass unter Einhaltung gewisser Vorgaben eine brauchbare Segmentierung am Baum nicht völlig utopisch ist. So könnte beispielsweise vom Nutzer verlangt werden, das Blatt vor dem Stamm des Baumes oder gegen den Boden zu fotografieren und sicherzustellen, dass keine anderen Blätter im Bild sind. Überdies muss sich das Blatt in der Mitte des Bildes befinden und vollständig zu erkennen sein. So wäre es z. B. möglich, die starken Kanten zwischen Blattkontur und Hintergrund zu detektieren und als Anhaltspunkt für die Initialisierung der GrabCut-Maske zu verwenden. Dies könnte mit Hilfe eines um die Kontur aufgespannten Rechtecks erfolgen. Gerade bei Aufnahmen mit relativ hochwertigen Kameras und somit gut geblurrtem Hintergrund sollte dieser Ansatz zu sehr brauchbaren Ergebnissen führen.

10.2.6 Behandlung zusammengesetzter Blätter

Wie bereits in Abschnitt 6.3.2 erwähnt, gibt es noch einige weitere Möglichkeiten, mit zusammengesetzten Blättern umzugehen. Nach Auffassung dieser Arbeit ist der vielversprechendste Ansatz der, zusammengesetzte Blätter zunächst als Ganzes einer Spezies zuzuordnen. Anschließend wird ein Blättchen ausgewählt, extrahiert und klassifiziert. Dabei werden lediglich die Spezies behandelt, die als zusammengesetzt markiert wurden. Dazu wäre eine Aufteilung oder zumindest eine Modifikation des Vergleichsdatensatzes nötig.

Diese Methode führt zu einem erheblichen Mehraufwand, könnte allerdings die Ergebnisse bei der Klassifikation zusammengesetzter Blätter stark verbessern. Allein die Information, dass es sich nicht um ein einfaches Blatt handelt, schränkt die in Frage kommenden Spezies bei der Klassifikation des Blättchens so stark ein, dass eine korrekte Bestimmung deutlich wahrscheinlicher wird.

10.2.7 Mögliche Erweiterungen des finalen Programms

Wie zu Beginn von Kapitel 8 bereits erwähnt, dient die GUI in ihrer jetzigen Form lediglich dazu, die Leistungsfähigkeit des Systems anschaulich darzustellen. Um das Programm einer breiten Masse von Nutzern zugänglich zu machen, wären einige Änderungen sinnvoll, die im Folgenden kurz vorgestellt werden sollen.

10.2.7.1 Verwendung im Browser

Im Moment liegt das Programm als Desktopanwendung vor. Ein potentieller Nutzer müsste es demnach zunächst herunterladen und entpacken, bevor er ein fotografiertes Blatt klassifizieren lassen kann. Wesentlich praktischer wäre eine Verwendung des Programms direkt im Browser: Der Nutzer ruft eine URL auf, lädt sein Foto hoch und kann

direkt mit der Segmentierung und Klassifikation beginnen, ohne einen vorherigen Download oder Installationsprozess durchführen zu müssen.

Eine Einbindung dieser Art ist bereits angedacht.

10.2.7.2 Auswahl der Vergleichsdatensets durch den Nutzer

Das Programm erlaubt aktuell die Unterscheidung von 430 Spezies. Diese große Anzahl wirkt sich zum einen negativ auf die Laufzeit des Klassifikationsprozesses aus, zum anderen verschlechtert sie die Klassifikationsrate, da viele, teils sehr ähnliche Spezies zur Auswahl stehen. Eine Einschränkung der Menge der Vergleichsspezies wäre daher wünschenswert. Dies kann durch das Hinzufügen von Checkboxen erreicht werden, bei denen der Nutzer auswählen kann, welche Datensets zur Klassifikation verwendet werden und welche nicht. Eine mögliche Aufteilung wäre beispielsweise: Mitteleuropa (Verwendung von BLD, MEW und SLD) und Asien (Flavia, Foliage und ICL). Überdies könnten unter Verwendung des Leafsnap-Datensets in Nordamerika heimische Spezies abgedeckt werden.

Auch die Integration eines neu zusammengestellten Datensets, das lediglich die in Deutschland heimischen Bäume beinhaltet, ist denkbar.

10.2.7.3 Mehrere Beispielbilder pro Spezies

In Abschnitt 8.2 wurde bereits die Problematik diskutiert, dass dem Nutzer nicht die Blätter angezeigt werden können, die dem zu klassifizierenden Blatt am stärksten ähneln, sondern lediglich ein Vertreter der entsprechenden Spezies. Gerade bei Blättern mit hoher Intra-Klassen-Varianz kann dies zur Verwirrung beim Nutzer führen, wenn das angegebene Topergebnis sich stark von dem zu klassifizierenden Blatt unterscheidet.

Eine Möglichkeit diese Problematik zu beheben wäre, zumindest bei Spezies mit hoher Intra-Klassen-Varianz, mehrere Repräsentanten abzuspeichern und jeweils das Bild dessen anzuzeigen, der dem zu klassifizierenden Blatt am stärksten ähnelt.

10.2.8 Fazit

Auch wenn das erarbeitete System bereits sehr gute Ergebnisse erzielt, gibt es noch eine Reihe vielversprechender Ansätze, die die Klassifikationsgenauigkeit weiter erhöhen und eine bessere Evaluation ermöglichen können. Besonders die Gewichtung der Features und die Einführung von Merkmals- und Lösungsklassen stellen interessante Erweiterungen dar.

Durch einige kleine Ergänzungen und eine mögliche Verwendung innerhalb eines Browsers kann das erstellte Programm eine ansprechende Möglichkeit bieten, eine Vielzahl unbekannter Spezies schnell und einfach zu identifizieren.

11 Quellenverzeichnis

Literatur

- [Ahonen 2006] Ahonen, T. et al.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 28, Issue 12, pp 2037-2041. 2006.
- [Alpaydin 2010] Alpaydin, E.: Introduction to Machine Learning. The MIT Press. 2010.
- [Beucher 1991] Beucher, S.: The Watershed Transformation Applied to Image Segmentation. 10th Pfefferkorn Conference on Signal and Image Processing in Microscopy and Microanalysis; pp 299-314. 1991.
- [Boykov 2001] Boykov, Y., Jolly, M.-P.: Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. Proceedings IEEE International Conference on Computer Vision; Vol. 1, pp 105-112. 2001.
- [Bradski 2008] Bradski, G., Kaehler, A.: Learning OpenCV. O'Reilly. 2008.
- [Brahmbhatt 2013] Brahmbhatt, S.: Practical OpenCV. Springer. 2013.
- [Burger 2009a] Burger, W., Burge, M.: Principles of Digital Image Processing – Fundamental Techniques. Springer. 2009.
- [Canny 1986] Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. PAMI-8, Issue 6. 1986.
- [Cardona 2014] Cardona, H. et al.: Automatic Recognition of Microcalcifications in Mammography Images through Fractal Texture Analysis. Advances in Visual Computing; Vol. 8888 of the series Lecture Notes in Computer Science, pp 841-850. 2014.
- [Chang 2013] Chang, C., Lin, C.: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST); Vol. 2, Issue 3. 2011.
- [Chih-Wei 2003] Chih-Wei, H.: A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science, National Taiwan University. 2003.
- [Ciresan 2012] Ciresan, D. et al.: Multi-column Deep Neural Networks for Image Classification. Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition. 2012.
- [Costa 2012] Costa, A. et al.: An Efficient Algorithm for Fractal Analysis of Textures. XXV SIBGRAPI Conference on Graphics, Pattern and Images; pp 39-46. 2012.
- [Das 2009] Das, N. et al.: Design of a Novel Convex Hull Based Feature Set for Recognition of Isolated Handwritten Roman Numerals. Proceedings of UB NE ASEE 2009. 2009.
- [Davies 2012] Davies, E.R.: Computer & Machine Vision – Theory, Algorithms, Practicalities. Academic Press. 2012.
- [Ebner 2007] Ebner, M.: Color Constancy. Wiley. 2007.
- [Ebrahimzadeh 2014] Ebrahimzadeh, R., Jampour, M.: Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM. International Journal of Computer Applications; Vol. 104, No. 9. 2014.

- [Efford 2000] Efford, N.: Digital Image Processing – A Practical Introduction Using Java. Pearson Education Limited. 2000.
- [Gao 2015] Gao, X. et al.: Handwritten Digit Recognition Based on Support Vector Machine. International Conference on Information Sciences, Machinery, Materials and Energy. 2015.
- [Goswami 2013] Goswami, R, Sharma, O.P.: A Review on Character Recognition Techniques. International Journal of Computer Applications; Vol 83, No. 7. 2013.
- [Grafmüller 2012] Grafmüller, M.: Verfahrensfortschritte in der robusten Echtzeiterkennung von Schriftzeichen. Schriftenreihe Automatische Sichtprüfung und Bildverarbeitung; Band 5. KIT Scientific Publishing. 2012.
- [Hecker 2012] Hecker, U.: Bäume und Sträucher – Treffsicher bestimmen in drei Schritten. BLV Buchverlag GmbH & Co. KG. 2012.
- [Hermes 2005] Hermes, T.: Digitale Bildverarbeitung. Carl Hanser Verlag. 2005.
- [Hotho 2014] Hotho, A. et al.: Data Mining. Vorlesung an der Julius-Maximilians-Universität Würzburg im Sommersemester 2014. Kapitel, Folien 67-86. 5 2014.
- [Hu 1962] Hu, M.: Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory; Vol. 8, pp 179-187. 1962.
- [Kadir 2011] Kadir, A.: Foliage Plant Retrieval Using Polar Fourier Transform, Color Moments and Vein Features. Signal & Image Processing: An International Journal; Vol. 2, No. 3. 2011.
- [Kadir 2012] Kadir, A.: Performance Improvement of Leaf Identification System Using Principal Component Analysis. International Journal of Advanced Science and Technology; Vol. 44. 2012.
- [Kadir 2014] Kadir, A.: A Model of Plant Identification System Using GLCM, Lacunarity and Shen Features. Research Journal of Pharmaceutical, Biological and Chemical Sciences; Vol. 5, No. 2. 2014.
- [Kremer 1989] Kremer, B.: Welches Blatt ist das? Bäume und Sträucher Mitteleuropas. Franck'sche Verlagshandlung, W. Keller & Co. 1989.
- [Kumar 2012] Kumar, N. et al.: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. Computer Vision – ECCV 2012, pp 502-516. 2012.
- [Lee 2013] Lee, K., Hong, K.: An Implementation of Leaf Recognition System using Leaf Vein and Shape. International Journey of Bio-Science and Bio-Technology; Vol. 5, No. 2. 2013.
- [Liu 2012] Liu, D. et al.: A Review of Computer Vision Segmentation Algorithms. Course on Computer Vision Spring 2012. University of Washington. 2012.
- [Lu 2003] Lu, X.: Image Analysis for Face Recognition. Proceedings of Personal Notes, Department of Computer Science and Engineering, USA, pp 1-37. 2003.
- [Mahesh 2008] Mahesh, P.: Multi-class Approaches for Support Vector Machine Based Land Cover Classification. National Institute of Technology, Haryana, India. 2008.
- [Mayer 2009] Mayer, J.: Welcher Baum ist das? Franckh-Kosmos Verlags-GmbH & Co.KG, Stuttgart. 2009.

- [Mina 2015] Mina, L., Isa, N.: A Review of Computer-Aided Detection and Diagnosis of Breast Cancer in Digital Mammography. *Journal of Medical Sciences*; Vol. 15, pp 110-121. [2015]
- [Mouine 2013] Mouine, S. et al.: Combining Leaf Salient Points and Leaf Contour Descriptions for Plant Species Recognition. *Image Analysis and Recognition*; Vol. 7950 of the series *Lecture Notes in Computer Science*, pp 205-214. 2013.
- [Mzoughi 2012] Mzoughi, O. et al.: Extraction of Leaf Parts. 9th International Conference, ICIAR 2012, Aveiro, Portugal, June 25-27; pp 348-358. 2012.
- [Novotny 2013] Novotný, P., Suk, T.: Leaf Recognition of Woody Species in Central Europe, *Biosystems Engineering*; Vol. 115, No. 4, pp 444–452. 2013.
- [Otsu 1979] Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*; Vol 9, pp 62-66. 1979.
- [Pietikäinen 2005] Pietikäinen, M., Hadid, A.: Texture Features in Facial Image Analysis. *Advances in Biometric Person Authentication*; Vol. 3781 of the series *Lecture Notes in Computer Science*, pp 1-8. 2005.
- [Pietikäinen 2011] Pietikäinen, M. et al.: Local Binary Patterns for Still Images. *Computational Imaging and Vision*; Vol. 40, pp 13-47. 2011.
- [Reul 2013] Reul, C.: Implementierung und Evaluierung einer Objekterkennung für einen Quadrocopter, Bachelorarbeit. Universität Würzburg. 2013.
- [Rother 2004] Rother, C., Kolmogorov, V., Blake, A.: „GrabCut“ – Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (TOG)* 23 (3), pp 309-314. 2004.
- [Russel & Norvig 2010] Russel, S., Norvig, P.: *Artificial Intelligence – A Modern Approach*. Pearson. 2010.
- [Sabhara 2013] Sabhara, R.: Comparative Study of Hu Moments and Zernike Moments in Object Recognition. *Smart Computing Review*; Vol. 3, No. 3, pp 166-173. 2013.
- [Samulski 2010] Samulski, M. et al.: Using computer-aided detection in mammography as a decision support. *European Radiology*; Vol. 20, Issue 10, pp 2323-2330. 2010.
- [Schulz 2012] Schulz, H., Behnke, S.: Deep Learning – Layer-wise Learning of Feature Hierarchies. *Künstliche Intelligenz*; Vol. 26, Issue 4, pp 357-363. 2012.
- [Starovoitov 2002] Starovoitov, V. et al.: Three Approaches for Face Recognition. 6th International Conference on Pattern Recognition and Image Analysis, Velikiy Nogorod; pp 707-711. 2002.
- [Suk 2013] Suk, T. et al.: Comparison of Leaf Recognition by Moments and Fourier Descriptors. *Lecture Notes in Computer Science*; Vol. 8047, pp 221-228. 2013.
- [Suzuki 1985] Suzuki, S., Abe, K.: Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics and Image Processing*; Vol. 30, pp 32-46. 1985.
- [Theofilis 2013] Theofilis, G.: *Weka Classifiers Summary*. Athens University of Economics and Business. 2013.

[Tsolakidis 2014] Tsolakidis, D. et al.: Plant Leaf Recognition Using Zernike Moments and Histogram of Orientated Gradients. 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17; pp 406-417. 2014.

[Witten 2011] Witten, I. et al.: Data Mining – Practical Machine Learning Tools and Techniques. Elsevier. 2011.

[Wu 2007] Gang Wu, S. et al.: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. IEEE 7th International Symposium on Signal Processing and Information Technology; pp 11-16. 2007.

Onlinequellen

Falls nicht explizit anders erwähnt, ist das letzte Abrufdatum jeweils der 28. September 2015.

[Baumkunde] Online-Datenbank für Bäume und Sträucher: Homepage.

URL: <http://www.baumkunde.de/>

[Gil] Gil's Computer Vision blog: Homepage.

URL: <https://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/>

[Heise] Heise online: Homepage.

URL: <http://www.heise.de/foto/artikel/Immer-noch-exotisch-L-a-b-228542.html>

[Leafsnap] iOS-App zur Blattidentifikation: Homepage.

URL: <http://www.leafsnap.com>

[LibSVM] Projektseite der LibSVM-Bibliothek: Homepage.

URL: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

[MNIST] LeCun, Y. et al.: The MNIST Database of Handwritten Digits. Homepage.

URL: <http://yann.lecun.com/exdb/mnist/> - Letztes Abrufdatum: 01.10.2015

[OpenCV] Open Source Computer Vision Library: Homepage. URLs:

<http://opencv.org/about.html>

[OpenCV: Morph]

http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html

[OpenCV: Watershed]

http://docs.opencv.org/master/d3/db4/tutorial_py_watershed.html

[TAI] Tangible Auditory Interfaces: Homepage.

URL: <http://tangibleauditoryinterfaces.de/index.php/tai-applications/durcheinander/>

[University of Alberta] University of Alberta: Homepage. URL:

<http://www.ualberta.ca/~ccwj/teaching/image/morph/Figs/PNG/connectivity.png>

[Weka] Data-Mining und Machine Learning Tool. Homepage.

URL: <http://www.cs.waikato.ac.nz/ml/weka/>

[Wikipedia] Wikipedia: *Homepage*. Wikimedia Foundation. URLs:
 [Wiki: Blatt] [https://de.wikipedia.org/wiki/Blatt_\(Pflanze\)](https://de.wikipedia.org/wiki/Blatt_(Pflanze))
[https://de.wikipedia.org/wiki/Blatt_\(Pflanze\)#/media/File:Leaf_Morphology.png](https://de.wikipedia.org/wiki/Blatt_(Pflanze)#/media/File:Leaf_Morphology.png)
 [Wiki: Brustkrebs] <https://de.wikipedia.org/wiki/Brustkrebs>
 [Wiki: Chebyshev] https://en.wikipedia.org/wiki/Chebyshev_distance
 [Wiki: DT] https://en.wikipedia.org/wiki/Distance_transform
 [Wiki: Euklid] https://en.wikipedia.org/wiki/Euclidean_distance
 [Wiki: GLCM] <https://de.wikipedia.org/wiki/Grauwertematrix>
 [Wiki: HSV] <https://de.wikipedia.org/wiki/HSV-Farbraum>
 [Wiki: Mammographie] <https://de.wikipedia.org/wiki/Mammographie>
 [Wiki: Manhattan] https://en.wikipedia.org/wiki/Taxicab_geometry
 [Wiki: Minkowski] https://en.wikipedia.org/wiki/Minkowski_distance
 [Wiki: Otsu] http://en.wikipedia.org/wiki/Otsu%27s_method
 [Wiki: PCA] https://en.wikipedia.org/wiki/Principal_component_analysis -
 Letztes Abrufdatum: 01.10.2015
 [Wiki: RapidMiner] <https://de.wikipedia.org/wiki/RapidMiner>
 [Wiki: Schiefe] [http://de.wikipedia.org/wiki/Schiefe_\(Statistik\)](http://de.wikipedia.org/wiki/Schiefe_(Statistik))
 [Wiki: StdDev] <http://de.wikipedia.org/wiki/Standardabweichung>
 [Wiki: Taxonomie] <https://de.wikipedia.org/wiki/Taxonomie>
 [Wiki: Wölbung] [http://de.wikipedia.org/wiki/W%C3%B6lbung_\(Statistik\)](http://de.wikipedia.org/wiki/W%C3%B6lbung_(Statistik))

Verwendete Datensets

Letztes Abrufdatum jeweils der 28. September 2015.

[Flavia] Wu, S. et al.: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. 2007 IEEE International Symposium on Signal Processing and Information Technology; pp 11-16. 2007.

URL: <http://flavia.sourceforge.net/>

[Foliage] Kadir, A. et al.: Experiments of Zernike Moments for Leaf Identification. Journal of Theoretical and Applied Information Technology (JATIT); Vol 41, No. 1, pp 82-93. 2012.

URL: https://www.researchgate.net/profile/Abdul_Kadir4/

[ICL] Intelligent Computing Laboratory, Chinese Academy of Sciences.

URL: <http://www.intelengine.cn/English/dataset>

[Leafsnap] Kumar, N. et al.: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. Computer Vision – ECCV 2012, pp 502-516. 2012.

URL: <http://www.leafsnap.com/dataset/>

[LifeCLEF 2015] LifeCLEF 2015 Plant task.

URL: <http://www.imageclef.org/lifeclef/2015/plant>

[MEW] Novotný, P., Suk, T.: Leaf recognition of woody species in Central Europe. Biosystems Engineering; Vol. 115, No. 4, pp 444–452. 2013.

URL: <http://zoi.utia.cas.cz/node/662>

[SLD] Söderkvist, O.: Computer Vision Classification of Leaves from Swedish Trees. Masterarbeit, Linköping Universität. 2001.

URL: <http://www.isy.liu.se/cvl/ImageDB/public/blad/index.html>

12 Anhang

12.1 Übersicht über die verwendeten Spezies

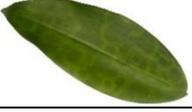
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Acalypha australis				x				Acer tataricum			x				
Acalypha wilkesiana		x						Achyranthes bidentata				x			
Acer buergerianum	x			x				Actinidia arguta			x				
Acer campestre			x					Adina rubella				x			
Acer cinna-monifolium				x				Aesculus chinensis	x			x			
Acer ginnala			x	x				Aesculus glabra			x				
Acer mono				x				Aesculus hippocastanum			x				
Acer negundo			x					Aesculus octandra			x				
Acer palmatum	x		x	x				Aesculus pavia			x				
Acer platanoides			x		x	x		Aglaonema commutatum (Maria)		x					
Acer pseudo-platanus			x			x		Aglaonema commutatum (Malay Beauty)		x					
Acer rubrum				x				Ailanthus altissima			x				
Acer saccharinum			x					Alchornea davidii				x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Alisma plantago-aquatica				x				Aster subulatus				x			
Alnus alnobetula			x					Basella rubra				x			
Alnus glutinosa			x					Bauhinia acuminata		x					
Alnus incana			x		x	x		Begonia glabra		x					
Alpinia sanderae		x						Begonia heraclifolia		x					
Alternanthera philoxeroides				x				Begonia rajah		x					
Amaranthus tricolor				x				Begonia Rosy Morn		x					
Amorpha fruticosa			x	x				Begonia semperflorens (green)				x			
Amygdalus persica				x				Begonia semperflorens (red)				x			
Anemone flaccida				x				Berberis anhwaiensis	x						
Artemisia annua				x				Berberis thunbergii			x	x			
Artemisia argyi				x				Berberis vulgaris			x				

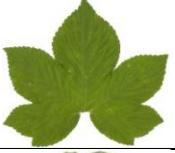
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Betula pubescens			x					Buxus microphylla				x			
Bidens frondosa				x				Buxus sempervirens			x				
Bischofia polycarpa				x				Calibrachoa				x			
Betula nana			x					Calissia elegans		x					
Betula papyrifera			x					Calystegia hederacea				x			
Betula pendula			x			x		Camellia oleifera				x			
Betula pubescens			x					Campsis radicus				x			
Bidens frondosa				x				Camptotheca acuminata				x			
Bischofia olycarpa				x				Capsicum annum				x			
Broussonetia papyrifera				x				Caragana arborescens			x				
Broussonetia papyrifera				x				Carpesium abrotanoides				x			
Buddleja davidii			x					Carpinus betulus			x			x	
Buxus bodinieri				x				Carya cathayensis				x			

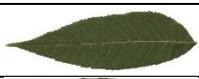
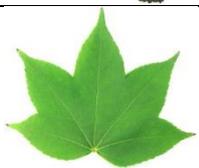
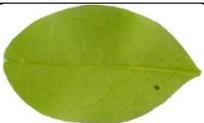
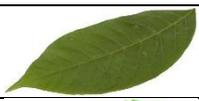
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
<i>Carya ovata</i>			x					<i>Cercidiphyllum japonicum</i>			x	x			
<i>Cassia tora</i>				x				<i>Cercis chinensis</i>	x			x			
<i>Castanea mollissima</i>				x				<i>Chaenomeles japonica</i>			x				
<i>Castanea sativa</i>			x					<i>Chaenomeles sinensis</i>				x			
<i>Catalpa bignonioides</i>			x					<i>Chaenomeles speciosa</i>				x			
<i>Catalpa bungei</i>				x				<i>Chenopodium ambrosioides</i>				x			
<i>Cayratia japonica</i>				x				<i>Chenopodium serotinum</i>				x			
<i>Cedrus deodara</i>	x							<i>Chimonanthus praecox</i>	x						
<i>Celosia cristata</i> (1)				x				<i>Christia verpetilionis</i>		x					
<i>Celosia cristata</i> (2)				x				<i>Cinnamomum camphora</i>	x			x			
<i>Celosia spicata</i>		x						<i>Cinnamomum japonicum</i>	x						
<i>Celosiae cristata</i>				x				<i>Cirsium setosum</i>				x			
<i>Celtis occidentalis</i>			x					<i>Cissus discolor</i>		x					
<i>Cerasus yedoensis</i>				x				<i>Citrus reticulata</i>	x			x			

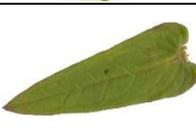
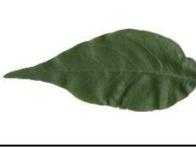
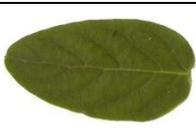
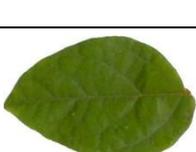
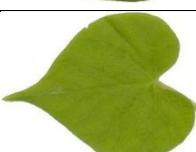
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Clematis vitalba				x				Corylus avellana			x		x		
Codiaeum variegatum (Captain Jack)		x						Corylus colurna			x				
Codiaeum variegatum (Whorthon)		x						Corylus maxima			x				
Coleus blumei				x				Cotinus coggygria			x				
Coleus tricolor		x						Cotoneaster horizontalis			x				
Colocasia		x						Cotoneaster integerrimus			x				
Colutea arborescens			x					Crataegus monogyna			x				
Commelina benghalensis				x				Cudrania tricuspidata				x			
Comniza canadensis				x				Cyclobalanopsis glauca				x			
Corchoropsis tomentosa				x				Dalbergia hupeana				x			
Cornus alba			x					Daphne mezereum			x				
Cornus mas			x					Dendrobenthamia japonica				x			
Cornus sanguinea			x					Deutzia scabra			x				

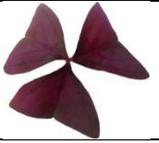
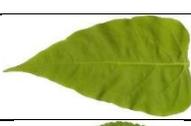
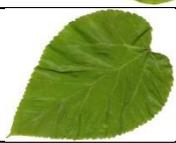
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Dichondra repens				x				Episcia cupreata		x					
Diospyros kaki (1)				x				Epri premnum amplissimum		x					
Diospyros kaki (2)				x				Epri premnum pinnatum (Aureum)		x					
Dolichos ladlad				x				Epri premnum pinnatum (Marble Queen)		x					
Dorstenia contrajerva		x						Erigeron annuus (1)				x			
Dracaena sanderiana		x						Erigeron annuus (2)				x			
Dracaena surculosa		x						Eriobotrya japonica				x			
Dracaena surculosa (Variegata)		x						Erythrina corallo-dendron				x			
Duchesnea indica				x				Eucommia ulmoides				x			
Echinodorus cordifolius		x						Euonymus europaea			x				
Eclipta prostrata				x				Euonymus japonica (1)				x			
Elaeagnus angustifolia			x					Euonymus japonica (2)				x			
Elaeocarpus decipens				x				Euonymus verrucosa			x				

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Fagaceae				x				Fructus pinnatifida				x			
Fagus sylvatica			x		x	x		Gardenia jasminoides (1)				x			
Fallopia aubertii			x					Gardenia jasminoides (2)				x			
Fatsia japonica				x				Ginkgo biloba	x		x	x			
Ficus religiosa		x						Gleditsia sinensis				x			
Ficus sarmentosa				x				Gleditsia triacanthos			x				
Ficus triangularis		x						Glycine soja				x			
Forsythia intermedia			x					Gomphrena globosa				x			
Forsythia suspensa				x				Gossypium hirsutum				x			
Frangula alnus			x					Graptho-phyllum (1)		x					
Fraxinus chinensis				x				Graptho-phyllum (2)		x					
Fraxinus excelsior			x			x		Graptho-phyllum (3)		x					
Fraxinus ornus			x					Gymnocladus dioicus			x				

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Hamamelis virginiana			x					Hovenia acerba				x			
Hedera helix (fertile)			x					Humulus lupulus			x				
Hedera helix (sterile)			x					Humulus scandens				x			
Hedra nepalensis				x				Hydrocotyle sibthorpioides				x			
Hemigraphis reptans		x						Hypericum monogynum				x			
Hibiscus mutabilis				x				Ilex aquifolium			x				
Hibiscus rosa-sinensis		x						Ilex macrocarpa	x						
Hibiscus rosa-sinensis (2)		x						Ilex purpurea				x			
Hibiscus syriacus			x	x				Impatiens balsamina				x			
Hibiscus syriacus (2)				x				Indigofera tinctoria	x						
Hibiscus tiliaceus		x						Ipomoea aquatica				x			
Hippophae rhamnoides			x					Ipomoea batatas		x		x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Ipomoea Batatas (Yellow)		x						Kolkwitzia amabilis			x				
Ipomoea lacunosa		x						Kummerowia stipulacea				x			
Ipomoea lacunosa (2)		x						Kummerowia striata				x			
Isotoma longiflora		x						Laburnum anagyroides			x				
Jassminum nudiflorum				x				Lager- stroemia indica	x			x			
Juglans cinerea			x					Lager- stroemia indica (2)				x			
Juglans nigra			x					Laurus nobilis				x			
Juglans regia			x					Lavandula angustifolia			x				
Kalancho daigre- montiana		x						Ledum palustre			x				
Kalopanax septemlobus	x							Ligustrum lucidum	x						
Kerria japonica			x					Ligustrum obtusifolium				x			
Koelreuteria bipinnata				x				Ligustrum ovalifolium			x				
Koelreuteria paniculata	x		x					Ligustrum quihoui				x			

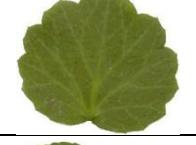
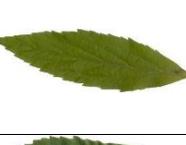
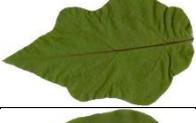
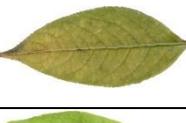
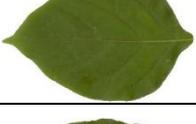
Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Ligustrum vulgare			x					Lycium barbarum			x				
Lindera glauca				x				Lycium chinense				x			
Liquidambar formosana				x				Lythrum salicaria				x			
Liquidambar styraciflua			x					Maclura pomifera (female)			x				
Liriodendron chinense	x			x				Maclura pomifera (male)			x				
Liriodendron tulipifera			x					Magnolia denudata				x			
Lonicera involucrata			x					Magnolia grandiflora	x			x			
Lonicera japonica				x				Magnolia soulangeana				x			
Lonicera tatarica			x					Mahonia aquifolium			x				
Lonicera xylosteum			x					Mahonia bealei	x						
Loranthus europaeus			x					Mahonia fortunei				x			
Lorpetalum chinense				x				Mallotus japonicus				x			
Lpomoea purpurea				x				Manglietia fordiana	x			x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Manihot esculenta		x						Nerium indicum				x			
Maranthamians		x						Nerium oleander	x		x				
Mazus gracilis				x				Oplismenus compositus				x			
Melia azedaeach				x				Orychophragmus violaceus				x			
Mespilus germanica			x					Osmanthus fragrans	x			x			
Metaplexis japonica				x				Oxalis corniculata				x			
Michelia figo				x				Oxalis corymbosa				x			
Michelia maudiae				x				Oxalis regnelli		x					
Mirabilis jalapa				x				Paederia scandens				x			
Morus alba			x	x				Paeonia Lactiflora				x			
Morus australis				x				Parthenocissus henryana				x			
Myrica rubra				x				Parthenocissus inserta			x				
Nandina domestica				x				Parthenocissus quinquefolia				x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Parthenocissus tricuspidata			x					Photinia serrulata				x			
Passiflora Quadrangularis		x						Phyllanthus urinaria				x			
Paulownia tomentosa			x					Phyllostachys edulis	x						
Peperomia pellucida				x				Physocarpus opulifolius			x				
Pharbitis nil				x				Phytolacca americana				x			
Phello-dendron amurense			x					Pilea cadieri		x					
Philadelphus coronarius			x					Pilea Nummulariifolia		x					
Philodendron barosoanum		x						Pilea repens		x					
Philodendron esculenta		x						Piper crocatum		x					
Philodendron hederaceum		x						Pistacia chinensis				x			
Phoebe nanmu	x							Pittosporum tobira	x			x			
Photinia davidsoniae				x				Plantago asiatica				x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Platanus hispanica			x	x				Prunella vulgaris				x			
Podocarpus macrophyllus	x							Prunus armeniaca			x				
Polyathia longifolia		x						Prunus cerasifera				x			
Polygonum longisetum				x				Prunus laurocerasus			x				
Polygonum orientale				x				Prunus mahaleb			x				
Polygonum perfoliatum				x				Prunus padus			x		x		
Pontederia cordata				x				Prunus persica	x						
Populus adenopoda				x				Prunus serrulata	x						
Populus canadensis	x							Prunus spinosa			x				
Populus canescens			x					Pterocarya pterocarpa			x				
Populus tremula			x		x	x		Pterocarya stenoptera				x			
Portulaca oleracea				x				Pterostyrax psilophyllus				x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Pueraria lobata				x				Ravenia spectabilis		x					
Punica granatum				x				Rhamnus cathartica			x				
Pyracantha coccinea			x					Rhododendron pulchrum				x			
Pyrus betulifolia				x				Rhus hirta			x				
Quamoclit pennata				x				Ribes alpinum			x				
Quercus acutissima				x				Robinia pseudacacia			x				
Quercus cerris			x					Robinia viscosa			x				
Quercus frainetto			x					Rostellularia procumbens				x			
Quercus glandulifera				x				Salix babylonica (1)				x			
Quercus petraea			x					Salix babylonica (2)				x			
Quercus pubescens			x					Salix caprea			x				
Quercus robur			x		x	x		Salix fragilis					x	x	
Quercus rubra			x			x		Salix matsudana				x			

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Salvia plebeia				x				Sorbus aria			x				
Salvia splendens				x				Sorbus aucuparia			x		x	x	
Sambucus nigra			x					Sorbus hybrida					x		
Sambucus racemosa			x					Sorbus intermedia			x				
Sapium sebiferum				x				Sorbus torminalis			x				
Saxifraga stolonifera				x				Spiraea japonica			x	x			
Sedum aizoon				x				Spiraea salicifolia			x				
Sedum sarmentosum				x				Spiraea vanhouttei			x				
Solanum melongena				x				Staphylea pinnata			x				
Solanum nigrum				x				Stellaria media				x			
Solanum pseudo-capsicum				x				Strobilanthes dyerianus		x					
Sophora japonica			x	x				Symphoricarpos albus			x				
Sorbaria sorbifolia			x					Symphoricarpos orbiculatus			x				

Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel	Spezies	Flavia	Foliage	MEW	ICL	SLD	BLD	Beispiel
Syringa vulgaris			x					Tradescantia zebrina		x					
Syzygia oblata				x				Trifolium repens				x			
Talinum paniculatum				x				Ulmus carpinifolia					x		
Thunbergia aurifolia		x						Ulmus chenmoui				x			
Tilia cordata			x		x	x		Ulmus glabra			x				
Tilia tomentosa			x		x			Ulmus laevis			x				
Tonna sinensis	x							Ulmus minor			x		x		
Torenia fournieri				x				Ulmus parvifolia				x			
Toxicodendron vernicifluum				x				Vaccinium myrtillus			x				
Trachelospermum jasminoides				x				Vaccinium uliginosum			x				
Tradescantia rdflexa				x				Vaccinium vitisidaea			x				
Tradescantia spathacea		x						Var intermedius				x			

12.2 Deutsche Bezeichnungen der verwendeten Spezies

Folgende Tabelle zeigt die deutschen Bezeichnungen der verwendeten Spezies. Diese wurden mittels des Online-Baum-Lexikons Baumkunde.de (vgl. [Baumkunde]) und des deutschsprachigen Wikipedia-Portals (vgl. [Wiki]) zusammengestellt.

Dort nicht aufgeführte deutsche Bezeichnungen werden als nicht bekannt angenommen. Ließ sich die Übersetzung des Namens einer Spezies nicht bestimmen, wurde, soweit möglich, zumindest der Name der übergeordneten Gattung angegeben und mit „+“ markiert.

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
<i>Acalypha australis</i>		<i>Alnus alnobetula</i>	Grün-Erle
<i>Acalypha wilkesiana</i>		<i>Alnus glutinosa</i>	Schwarz-Erle
<i>Acer buergerianum</i>	Dreizähniger Ahorn	<i>Alnus incana</i>	Grau-Erle
<i>Acer campestre</i>	Feld-Ahorn	<i>Alpinia sanderae</i>	Ingwer
<i>Acer cinnamomifolium</i>	Ahorn+	<i>Alternanthera philoxeroides</i>	Alligatorkraut
<i>Acer ginnala</i>	Ahorn+	<i>Amaranthus tricolor</i>	Gemüse-Amaranth
<i>Acer mono</i>	Ahorn+	<i>Amorpha fruticosa</i>	Falscher Indigo
<i>Acer negundo</i>	Eschen-Ahorn	<i>Amygdalus persica</i>	
<i>Acer palmatum</i>	Fächer-Ahorn	<i>Anemone flaccida</i>	Windröschen+
<i>Acer platanoides</i>	Spitz-Ahorn	<i>Artenisia annua</i>	Einjähriger Beifuß
<i>Acer pseudoplatanus</i>	Berg-Ahorn	<i>Artenisia argyi</i>	
<i>Acer rubrum</i>	Rot-Ahorn	<i>Aster subulatus</i>	Aster+
<i>Acer saccharinum</i>	Silber-Ahorn	<i>Basella rubra</i>	Malabar-Spinat
<i>Acer saccharum</i>	Zucker-Ahorn	<i>Bauhinia acuminata</i>	Orchidee
<i>Acer tataricum</i>	Tataren-Ahorn	<i>Begonia glabra</i>	Begonie+
<i>Achyranthes bidentata</i>		<i>Begonia heraclifolia</i>	Blatt-Begonie
<i>Actinidia arguta</i>	Scharfzähniger Strahlengriffel	<i>Begonia rajah</i>	Begonie+
<i>Adina rubella</i>		<i>Begonia Rosy Morn</i>	Begonie+
<i>Aesculus chinensis</i>	Chinesische Rosskastanie	<i>Begonia semperflorens (green)</i>	Eis-Begonie
<i>Aesculus glabra</i>	Ohio-Rosskastanie	<i>Begonia semperflorens (red)</i>	Eis-Begonie
<i>Aesculus hippocastanum</i>	Gemeine Rosskastanie	<i>Berberis anhweiensis</i>	Berberspitze+
<i>Aesculus octandra</i>	Gelbe Rosskastanie	<i>Berberis thunbergii</i>	Thunbergs Berberspitze
<i>Aesculus pavia</i>	Echte Pavie	<i>Berberis vulgaris</i>	Gewöhnliche Berberspitze
<i>Aglaonema commutatum (Maria)</i>	Kolbenfaden	<i>Betula nana</i>	Zwerg-Birke
<i>Aglaonema commutatum (Malay Beauty)</i>	Kolbenfaden	<i>Betula papyrifera</i>	Papier-Birke
<i>Ailanthus altissima</i>	Götterbaum	<i>Betula pendula</i>	Hänge-Birke
<i>Alchornea davidii</i>		<i>Betula pubescens</i>	Moor-Birke
<i>Alisma plantagoaquatica</i>	Gewöhnlicher Froschlöffel	<i>Bidens frondosa</i>	Schwarzfrüchtiger Zweizahn

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
<i>Bischofia polycarpa</i>		<i>Celosiae cristata</i>	Celosie+
<i>Broussonetia papyrifera</i>	Maulbeerbaum+	<i>Celtis occidentalis</i>	Amerikanischer Zürgelbaum
<i>Buddleja davidii</i>	Schmetterlings-Flieder	<i>Cerasus yedoensis</i>	
<i>Buxus bodinieri</i>	Buchsbaum+	<i>Cercidiphyllum japonicum</i>	Japanischer Kuchenbaum
<i>Buxus microphylla</i>	Buchsbaum+	<i>Cercis chinensis</i>	Chinesischer Judasbaum
<i>Buxus sempervirens</i>	Gewöhnlicher Buchsbaum	<i>Chaenomeles japonica</i>	Japanische Zier-Quitte
<i>Calibrachoa</i>	Calibrachoa+	<i>Chaenomeles sinensis</i>	Zier-Quitte+
<i>Calissia elegans</i>	Calissia+	<i>Chaenomeles speciosa</i>	Chinesische Zier-Quitte
<i>Calystegia hederacea</i>	Zaunwinde+	<i>Chenopodium ambrosioides</i>	Gänsefuß+
<i>Camellia oleifera</i>	Kamelie+	<i>Chenopodium serotinum</i>	Gänsefuß+
<i>Campsis radicus</i>	Amerikanische Klettertrompete	<i>Chimonanthus praecox</i>	Chinesische Winterblüte
<i>Camptotheca acuminata</i>	Glücksbaum	<i>Christia verpetilionis</i>	
<i>Capsicum annuum</i>	Spanischer Pfeffer	<i>Cinnamomum camphora</i>	Kampferbaum
<i>Caragana arborescens</i>	Gemeiner Erbsenstrauch	<i>Cinnamomum japonicum</i>	
<i>Carpesium abrotanoides</i>	Kragenblume+	<i>Cirsium setosum</i>	
<i>Carpinus betulus</i>	Hain-Buche	<i>Cissus discolor</i>	Bunte Klimme
<i>Carya cathayensis</i>	Hickory+	<i>Citrus reticulata</i>	Mandarine
<i>Carya ovata</i>	Schuppenrinden-Hickorynuss	<i>Clematis vitalba</i>	Gewöhnliche Waldrebe
<i>Cassia tora</i>	Kassie+	<i>Codiaeum variegatum (Captain Jack)</i>	Kroton
<i>Castanea mollissima</i>	Chinesische Ess-Kastanie	<i>Codiaeum variegatum (Whornton)</i>	Kroton
<i>Castanea sativa</i>	Ess-Kastanie	<i>Coleus blumei</i>	Bunt-Nessel
<i>Catalpa bignonioides</i>	Gewöhnlicher Trompetenbaum	<i>Coleus tricolor</i>	Nessel+
<i>Catalpa bungei</i>	Kugel-Trompetenbaum	<i>Colocasia</i>	Zehrwurz
<i>Cayratia japonica</i>		<i>Colutea arborescens</i>	Gelber Blasenstrauch
<i>Cedrus deodara</i>	Himalaya-Zeder	<i>Commelina benghalensis</i>	Tagblume+
<i>Celosia cristata (1)</i>	Hahnenkamm	<i>Conyza canadensis</i>	Kanadisches Berufkraut
<i>Celosia cristata (2)</i>	Hahnenkamm	<i>Corchoropsis tomentosa</i>	
<i>Celosia spicata</i>	Celosie+	<i>Cornus alba</i>	Tatarischer Hartriegel

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
Cornus mas	Kornelkirsche	Erigeron annuus	Feinstrahl
Cornus sanguinea	Roter Hartriegel		
Corylus avellana	Gemeine Hasel	Eriobotrya japonica	Japanische Wollmistel
Corylus colurna	Baum-Hasel	Erythrina coraliodendron	
Corylus maxima	Lamberts-Hasel	Eucommia ulmoides	Gummi-Ulme
Cotinus coggygria	Perückenstrauch	Euonymus europaea	Gewöhnlicher Spindelstrauch
Cotoneaster horizontalis	Fächer-Stein-Mispel	Euonymus japonica	Japanischer Spindelstrauch
Cotoneaster integerrimus	Gewöhnliche Zwerg-Mispel	Euonymus verrucosa	Weizen-Spindelstrauch
Crataegus monogyna	Eingriffeliger Weißdorn	Fagaceae	Buchengewächs+
Cudrania tricuspidata		Fagus silvatica	Rot-Buche
Cyclobalanopsis glauca		Fallopia aubertii	Schling-Knöterich
Dalbergia hupeana		Fatsia japonica	Zimmer-Aralie
Daphne mezereum	Echter Seidelbast	Ficus religiosa	Pappel-Feige
Dendrobenthamia japonica		Ficus sarmentosa	Feige+
Deutzia scabra	Raue Deutzie	Ficus triangularis	Feige+
Dichondra repens	Kriechende Dichondra	Forsythia intermedia	Garten-Forsythie
Diospyros kaki	Kakibaum	Forsythia suspensa	Hänge-Forsythie
		Frangula alnus	Faulbaum
Dolichos laddad		Fraxinus chinensis	Chinesische Esche
Dorstenia contrajerva	Dorstenie+	Fraxinus excelsior	Gewöhnliche Esche
Dracaena sanderiana	Glücks-Bambus	Fraxinus ornus	Manna-Esche
Dracaena surculosa	Drachenbaum	Fructus pinnatifida	
Dracaena surculosa (Variegata)	Drachenbaum	Gardenia jasminoides	Gardenie+
Duchesnea indica	Indische Scheinerdbeere	Ginkgo biloba	Ginko+
Echinodorus cordifolius	Herzblättrige Schwertpflanze	Gleditsia sinensis	Chinesische Gleditschie
Eclipta prostrata		Gleditsia triacanthos	Amerikanische Gleditschie
Elaeagnus angustifolia	Schmalblättrige Ölweide	Glycine soja	
Elaeocarpus decipens	Japanische Blaubeere	Gomphrena globosa	Echter Kugelmarant
Episcia cupreata	Gesnerie+	Gossypium hirsutum	
Epri premnum amplissimum		Graphophyllum	
Epri premnum pinnatum (Aureum)		Gymnocladus dioicus	Geweißbaum
Epri premnum pinnatum (Marble Queen)		Hamamelis virginiana	Virginische Zaubernuss

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
Hedera helix (FERTILE)	Gewöhnlicher Efeu	Kalopanax septemlobus	Baum-Aralie
Hedera helix (STERILE)	Gewöhnlicher Efeu	Kerria japonica	Japanische Kerrie
Hedera nepalensis	Nepal-Efeu	Koelreuteria bipinnata	
Hemigraphis reptans		Koelreuteria paniculata	Blasen-Esche
Hibiscus mutabilis	Hibiskus+	Kolkwitzia amabilis	Kolkwitzie
Hibiscus rosa-sinensis	Zimmer-Hibiskus	Kummerowia stipulacea	
Hibiscus syriacus	Strauch-Hibiskus	Kummerowia striata	
Hibiscus tiliaceus	Lindenblättriger Hibiskus	Laburnum anagyroides	Gemeiner Goldregen
Hippophae rhamnoides	Sanddorn	Lagerstroemia indica	Chinesische Lagerströmie
Hovenia acerba	Japanischer Rosinenbaum	Laurus nobilis	Echter Lorbeer
Humulus lupulus	Echter Hopfen	Lavandula angustifolia	Echter Lavendel
Humulus scandens	Japanischer Hopfen	Ledum palustre	Sumpfporst
Hydrocotyle sibthorpioides	Kleinblättriger Wassernabel	Ligustrum lucidum	Glänzender Liguster
Hypericum monogynum	Johanniskraut+	Ligustrum obtusifolium	Stumpfbältriger Liguster
Ilex aquifolium	Europäische Stechpalme	Ligustrum ovalifolium	Ovalblättriger Liguster
Ilex macrocarpa	Stechpalme+	Ligustrum quihoui	Quihois Liguster
Ilex purpurea	Stechpalme+	Ligustrum vulgare	Gewöhnlicher Liguster
Impatiens balsamina	Garten-Balsamine	Lindera glauca	
Indigofera tinctoria	Indigopflanze	Liquidambar formosana	Chinesischer Amberbaum
Ipomoea aquatica	Wasser-Spinat	Liquidambar styraciflua	Amerikanischer Amberbaum
Ipomoea batatas	Süß-Kartoffel	Liriodendron chinense	Chinesischer Tulpenbaum
Ipomoea batatas (Yellow)	Süß-Kartoffel	Liriodendron tulipifera	Tulpenbaum
Ipomoea lacunosa	Kartoffel+	Lonicera involucrata	Heckenkirsche+
Isotoma longiflora	Glockenblumengewächse+	Lonicera japonica	Japanisches Geißblatt
Jasminum nudiflorum	Winter-Jasmin	Lonicera tatarica	Tataren-Heckenkirsche
Juglans cinerea	Butter-Nuss	Lonicera xylosteum	Rote Heckenkirsche
Juglans nigra	Schwarz-Nuss	Loranthus europaeus	Eichen-Mistel
Juglans regia	Echte Walnuss	Lorpetalum chinense	Riemenblüte
Kalancho daigremontiana	Goethe-Pflanze	Ipomoea purpurea	Purpur-Prunkwinde

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
Lycium barbarum	Gemeiner Bocksdorn	Osmanthus fragrans	Süße Duftblüte
Lycium chinense	Chinesischer Bocksdorn	Oxalis corniculata	Horn-Sauerklee
Lythrum salicaria	Gewöhnlicher Blutweiderich	Oxalis corymbosa	Sauerklee+
Maclura pomifera (FEMALE)	Milchorangenbaum	Oxalis regnelli	Sauerklee+
Maclura pomifera (MALE)	Milchorangenbaum	Paederia scandens	Stink-Wein
Magnolia denudata	Yulan-Magnolie	Paeonia Lactiflora	Milchweiße Pfingstrose
Magnolia grandiflora	Immergrüne Magnolie	Parthenocissus henryana	Jungfernrebe+
Magnolia soulangeana	Tulpen-Magnolie	Parthenocissus inserta	Gewöhnliche Jungfernrebe
Mahonia aquifolium	Gewöhnliche Mahonie	Parthenocissus quinquefolia	Selbstkletternde Jungfernrebe
Mahonia bealei	Beals Mahonie	Parthenocissus tricuspidata	Dreispitziige Jungfernrebe
Mahonia fortunei	Mahonie+	Passiflora quadrangularis	Riesen-Granadilla
Mallotus japonicus		Paulownia tomentosa	Blauglockenbaum
Manglietia fordiana		Peperomia pellucida	
Manihot esculenta	Maniok	Pharbitis nil	Blaue Prunkwinde
Maranthamians		Phellodendron amurense	Amur-Korkbaum
Mazus gracilis		Philadelphus coronarius	Europäischer Pfeifenstrauch
Melia azedarach	Zedrachbaum	Philodendron barosolanum	
Mespilus germanica	Echte Mispel	Philodendron esculenta	Philodendron+
Metaplexis japonica		Philodendron hederaceum	Kletter-Philodendron
Michelia figo		Phoebe nanmu	Phoebe+
Michelia maudiae		Photinia davidsoniae	
Mirabilis jalapa	Wunderblume	Photinia serrulata	Kahle Glanzmispel
Morus alba	Weißer Maulbeere	Phyllanthus urinaria	
Morus australis	Maulbeere+	Phyllostachys edulis	Moso-Bambus
Myrica rubra		Physocarpus opulifolius	Schneebältrige Blasenpiere
Nandina domestica	Himmels-Bambus	Phytolacca americana	Amerikanische Kermesbeere
Nerium indicum		Pilea cadierei	Kanonierblume
Nerium oleander	Oleander	Pilea nummulariifolia	
Oplismenus compositus		Pilea repens	Pellionie
Orychophragmus violaceus	Japanische Blüten-Kresse	Piper crocatum	Zierblatt-Pfeffer

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
<i>Pistacia chinensis</i>		<i>Quercus frainetto</i>	Ungarische Eiche
<i>Pittosporum tobira</i>	Chinesischer Klebsame	<i>Quercus glandulifera</i>	Eiche+
<i>Plantago asiatica</i>	Wegerich+	<i>Quercus petraea</i>	Trauben-Eiche
<i>Platanus hispanica</i>	Ahornblättrige Platane	<i>Quercus pubescens</i>	Flaum-Eiche
<i>Podocarpus macrophyllus</i>	Großblättrige Steineibe	<i>Quercus robur</i>	Stiel-Eiche
<i>Polyathia longifolia</i>		<i>Quercus rubra</i>	Rot-Eiche
<i>Polygonum longisetum</i>		<i>Ravenia spectabilis</i>	
<i>Polygonum orientale</i>	Chinesischer Knöterich	<i>Rhamnus cathartica</i>	Purgier-Kreuzdorn
<i>Polygonum perfoliatum</i>		<i>Rhododendron pulchrum</i>	Rhododendron+
<i>Pontederia cordata</i>	Herzblättriges Heckkraut	<i>Rhus hirta</i>	Essigbaum
		<i>Ribes alpinum</i>	Alpen-Johannisbeere
<i>Populus adenopoda</i>	Pappel+	<i>Robinia pseudacacia</i>	Gewöhnliche Robinie
<i>Populus canadensis</i>	Bastard-Schwarz-Pappel	<i>Robinia viscosa</i>	Klebrige Robinie
<i>Populus canescens</i>	Grau-Pappel	<i>Rostellularia procumbens</i>	
<i>Populus tremula</i>	Zitter-Pappel	<i>Salix babylonica</i> (1)	Echte Trauer-Weide
<i>Portulaca oleracea</i>	Gemüse-Portulak	<i>Salix babylonica</i> (2)	Echte Trauer-Weide
<i>Prunella vulgaris</i>	Gewöhnliche Braunelle	<i>Salix caprea</i>	Sal-Weide
<i>Prunus armeniaca</i>	Aprikose	<i>Salix fragilis</i>	Bruch-Weide
<i>Prunus cerasifera</i>	Kirschpflaume	<i>Salix matsudana</i>	Korkenzieher-Weide
<i>Prunus laurocerasus</i>	Lorbeer-Kirsche	<i>Salvia plebeia</i>	Salbei+
<i>Prunus mahaleb</i>	Felsen-Kirsche	<i>Salvia splendens</i>	Feuer-Salbei
<i>Prunus padus</i>	Gewöhnliche Trauben-Kirsche	<i>Sambucus nigra</i>	Schwarzer Holunder
<i>Prunus persica</i>	Zwerg-Pfirsich	<i>Sambucus racemosa</i>	Roter Holunder
<i>Prunus serrulata</i>	Japanische Blüten-Kirsche	<i>Sapium sebiferum</i>	Chinesischer Talgbaum
<i>Prunus spinosa</i>	Schlehdorn	<i>Saxifraga stolonifera</i>	Judenbart
<i>Pterocarya pterocarpa</i>	Flügelnuss+	<i>Sedum aizoon</i>	Große Gold-Fetthenne
<i>Pterocarya stenoptera</i>	Flügelnuss+	<i>Sedum sarmentosum</i>	Ausläufer-Fetthenne
<i>Pterostyrax psilophyllus</i>		<i>Solanum melongena</i>	Aubergine
<i>Pueraria lobata</i>	Kudzu	<i>Solanum nigrum</i>	Schwarzer Nachtschatten
<i>Punica granatum</i>	Granatapfel	<i>Solanum pseudocapsicum</i>	Korallenstrauch
<i>Pyracantha coccinea</i>	Mittelmeer-Feuerdorn	<i>Sophora japonica</i>	Japanischer Schnurbaum
<i>Pyrus betulifolia</i>	Birkenblättrige Birne	<i>Sorbaria sorbifolia</i>	Sibirische Fiederspiere
<i>Quamoclit pennata</i>		<i>Sorbus aria</i>	Echte Mehlbeere
<i>Quercus acutissima</i>	Japanische Kastanien-Eiche	<i>Sorbus aucuparia</i>	Vogelbeere
<i>Quercus cerris</i>	Zerr-Eiche	<i>Sorbus hybrida</i>	Bastard-Vogelbeere

Lateinische Bezeichnung	Deutsche Bezeichnung	Lateinische Bezeichnung	Deutsche Bezeichnung
<i>Sorbus intermedia</i>	Schwedische Mehlbeere	<i>Ulmus minor</i>	Feld-Ulme
<i>Sorbus torminalis</i>	Elsbeere	<i>Ulmus parvifolia</i>	Chinesische Ulme
<i>Spiraea japonica</i>	Sommer-Spiere	<i>Vaccinium myrtillus</i>	Heidelbeere
<i>Spiraea salicifolia</i>	Weidenblättrige Spiere	<i>Vaccinium uliginosum</i>	Rauschbeere
<i>Spiraea vanhouttei</i>	Pracht-Spiere	<i>Vaccinium vitisidaea</i>	Preiselbeere
<i>Staphylea pinnata</i>	Gemeine Pimpernuss	Var <i>intermedius</i>	
<i>Stellaria media</i>	Gewöhnliche Vogelmiere	<i>Viburnum awabuki</i>	Immergrüner Schneeball
<i>Strobilanthes dyerianus</i>		<i>Viburnum farreri</i>	Duftender Schneeball
<i>Symphoricarpos albus</i>	Gewöhnliche Schneebeere	<i>Viburnum lantana</i>	Wolliger Schneeball
<i>Symphoricarpos orbiculatus</i>	Korallenbeere	<i>Viburnum odoratissimum</i>	Schneeball+
<i>Syringa vulgaris</i>	Gemeiner Flieder	<i>Viburnum opulus</i>	Gewöhnlicher Schneeball
<i>Syzygia oblata</i>		<i>Viburnum pragense</i>	Prager Schneeball
<i>Talinum paniculatum</i>	Erd-Ginseng	<i>Viburnum rhytidophyllum</i>	Runzelblättriger Schneeball
<i>Thunbergia aurifolia</i>		<i>Vinca minor</i>	Kleines Immergrün
<i>Tilia cordata</i>	Winter-Linde	<i>Viola betonicifolia</i>	Betonieblättriges Veilchen
<i>Tilia tomentosa</i>	Silber-Linde	<i>Viola concordifolia</i>	Veilchen+
<i>Tonna sinensis</i>	Chinesischer Gemüsebaum	<i>Vitis adstricta</i>	Rebe+
<i>Torenia fourneri</i>	Fournier-Torenie	<i>Vitis vinifera</i>	Wein-Rebe
<i>Toxicodendron vernicifluum</i>	Lackbaum	<i>Weigela florida</i>	Liebliche Weigelia
<i>Trachelospermum jasminoides</i>	Stern-Jasmin	<i>Wisteria sinensis</i>	Chinesische Wisteria
<i>Tradescantia rdflexa</i>		<i>Xanthium sibiricum</i>	Gewöhnliche Spitzklette
<i>Tradescantia spathacea</i>	Purpurblättrige Dreimasterblume	<i>Zamia pumila</i>	
<i>Tradescantia zebrina</i>	Zebra-Ampelkraut	<i>Zamioculcas</i>	Glücksfeder
<i>Trifolium repens</i>	Weiß-Klee	<i>Zanthoxylum bungeanum</i>	
<i>Ulmus carpinifolia</i>	Feld-Ulme	<i>Zelkova schneideriana</i>	Zelkove+
<i>Ulmus chenmoui</i>	Ulme+	<i>Zelkova serrata</i>	Japanische Zelkove
<i>Ulmus glabra</i>	Berg-Ulme	<i>Zizyphus jujuba</i>	Chinesische Jujube
<i>Ulmus laevis</i>	Flatter-Ulme		

12.3 Inhalt der DVD

Folgende Materialien sind auf der beigelegten DVD zu finden:

- **Baumklassifikation_Reul_Masterarbeit:** Diese Arbeit im .pdf-Format.
- **GUI:** Das in Kapitel 8 vorgestellte finale Programm.
- **Code:** Die übrigen in dieser Arbeit verwendeten Programme.
- **Anleitung:** Hinweise zur Nutzung der Programme.
- **Datensets:** Alle in dieser Arbeit verwendeten Datensets in vorsegmentierter Form. Die gesamte Auswahl an Datensets wurde aufgrund ihrer Größe auf einem USB-Stick beigelegt.
- **ARFFs:** Die Datensets im ARFF-Format.