
OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

FAKULTÄT FÜR INFORMATIK

Institut für Technische und Betriebliche Informationssysteme

Arbeitsgruppe AMSL - Projekt DigiDak+



Bachelorarbeit

Zur Erlangung des akademischen Grades

„Bachelor of Science“

im Fach Informatik

Thema:

**Untersuchungszielspezifische Merkmalsselektion
für die textile Faserspurenforensik**

Vorgelegt von: Jens Dieskau
Matrikelnummer: 187970
Betreuer: Prof. Dr.-Ing. Jana Dittmann
Abgabetermin: 13.06.2014

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
Formelverzeichnis	v
Abkürzungsverzeichnis	vi
1 Einleitung	1
1.1 Ausgangssituation	2
1.2 Zielsetzung	3
1.3 Abgrenzung	4
1.4 Aufbau der Arbeit	5
2 Grundlagen	6
2.1 Klassifikation	6
2.2 Merkmalsgüte	7
2.2.1 Selektion	8
2.2.2 Kombination	8
2.3 Statistische Kenngrößen	9
2.4 Textur	10
3 Konzept	11
3.1 Datenerfassung und Digitalisierung	11
3.2 Vorverarbeitung	12
3.3 Merkmalsextraktion	13
3.4 Klassifizierung	14
3.5 Auswertung	15
3.6 Arbeitsablauf	15
4 Methodologie	17
4.1 Anforderungen	17
4.2 Einordnung	18
4.3 Schwerpunkt	18
4.4 Statistische Analyse	18
4.4.1 Tamuras Texturmerkmale	19
4.4.2 Co-occurrence Matrix	21
4.4.3 Gray Level Difference Method	23
4.4.4 Gray Level Run Length	23

4.4.5	Census-Transformation	25
4.4.6	Local Binary Pattern	26
4.4.7	Webers Lokaler Deskriptor	30
4.4.8	Autokorrelation	30
4.4.9	Energiefilter nach Laws	32
4.4.10	Zusammenfassung	34
4.5	Signalverarbeitungsanalyse	36
4.5.1	Fourier-Transformation	36
4.5.2	Gaborfilter	37
4.5.3	Waveletfilter	37
4.6	Modellbasierende Analyse	38
4.6.1	Markov Random Fields Modell	38
4.6.2	Fraktale Modelle	40
4.7	Geometrische Analyse	40
4.7.1	Voronoi-Tessellation	40
4.7.2	Syntactic Pattern Recognition	41
5	Umsetzung	42
5.1	Verwendete Technologien	42
5.2	Technischer Aufbau	43
5.2.1	Modularität	43
5.2.2	Parallelisierung	44
5.3	Benutzeroberfläche	44
5.4	Experimenteller Aufbau	48
6	Evaluierung	50
6.1	Anforderungen	50
6.2	Ergebnisse	51
6.2.1	Tamura	52
6.2.2	Haralick	54
6.2.3	Galloway	58
6.2.4	Local Binary Pattern	60
6.2.5	Autokorrelation	61
6.2.6	Laws Energiefilter	62
7	Zusammenfassung und Ausblick	63
7.1	Ausblick	64
7.2	Fazit	65
	Literaturverzeichnis	66
8	Anhang	77

Abbildungsverzeichnis

1.1	Klassifikation der Fasertypen	3
3.1	Konzeptionelle Verarbeitungspipeline	11
3.2	Vergleich des originalen Faserscans mit algorithmisch generierter Textur	13
4.1	Beispiel einer Gray Level Co-occurrence Matrix	21
4.2	Beispiel einer Gray Level Run Length Matrix	24
4.3	Beispiel einer 3x3 Census-Transformation	26
4.4	Beispielergebnis einer LBP-Berechnung	27
4.5	Veranschaulichung der Auswirkung einer Bildrotation auf die Nachbarschaft	28
4.6	Uniforme LBP Muster	29
5.1	Screenshot von einem geöffneten Projekt	45
5.2	Screenshot einer Verarbeitungspipeline	46
5.3	Fasertypen und deren Anzahl im Testset	48
5.4	Konfiguration der verwendeten Computersysteme	48
6.1	Tamura Klassifizierungsergebnisse	53
6.2	Haralick - Vergleich mit unterschiedlichen Fenstergrößen	54
6.3	Haralick - Klassifizierung mit unterschiedlichen Graustufen	55
6.4	Haralick - Klassifizierung mit unterschiedlichen Abstandsvektoren	57
6.5	Galloway - Vergleich mit unterschiedlichen Fenstergrößen	59
6.6	LBP - Vergleich mit unterschiedlichen Fenstergrößen	60

Tabellenverzeichnis

2.1	Kontingenztafel mit grundlegenden statistischen Kenngrößen	9
4.1	Haralick-Texturmerkmale	22
4.2	Mögliche 5×5 Laws Masken	33
4.3	Laws finale Merkmale	34
4.4	Zusammenfassung der statistischen Merkmalsextraktionsmethoden	35
4.5	Beispiel eines Zustandes x_t einer Markov-Kette.	39
6.1	Verwendete Klassifikatoren und deren Parameter in <i>Weka</i>	51
6.2	Galloway Klassifizierungsergebnisse	58
6.3	Autokorrelation Klassifizierungsergebnisse	62
8.1	Tamura Klassifizierungsergebnisse (Histogrammgröße: 32)	77
8.2	Tamura Klassifizierungsergebnisse (Histogrammgröße: 48)	78
8.3	Tamura Klassifizierungsergebnisse (Histogrammgröße: 64)	78
8.4	Tamura Klassifizierungsergebnisse (Histogrammgröße: 96)	79
8.5	Tamura Klassifizierungsergebnisse (Histogrammgröße: 128)	79
8.6	Haralick Klassifizierungsergebnisse (Fenstergröße: 48x48 Pixel)	80
8.7	Haralick Klassifizierungsergebnisse (Fenstergröße: 64x64 Pixel)	80
8.8	Haralick Klassifizierungsergebnisse (Fenstergröße: 96x96 Pixel)	81
8.9	Haralick Klassifizierungsergebnisse (Fenstergröße: 128x128 Pixel)	81
8.10	LBP Klassifikationsergebnisse	82
8.11	Laws Energiefilter Klassifizierungsergebnisse	82

Formelverzeichnis

2.2	Merkmalsgüte	8
2.3	F-Maß	10
4.1	Tamura - Coarsness	20
4.2	Tamura - Contrast	20
4.3	Tamura - Directionality	20
4.4	Co-occurrence Matrix	21
4.5	Galloway - Short Runs Emphasis	24
4.6	Galloway - Long Runs Emphasis	24
4.7	Galloway - Long Runs Emphasis	24
4.8	Galloway - Run Length Non-uniformity	25
4.9	Galloway - Run Percentage	25
4.10	Galloway - Low Gray Level Run Emphasis	25
4.11	Galloway - High Gray Level Run Emphasis	25
4.12	Census-Transformation	26
4.13	Local Binary Pattern	27
4.14	Rotationsinvariantes Local Binary Pattern	28
4.15	Uniformitätsmaß	28
4.16	Uniformes Local Binary Pattern	29
4.17	Webers Gesetz	30
4.18	Autokorrelation	31
4.19	Vereinfachte Formel der Autokorrelation	31
4.21	Schiefe	31
4.22	Kurtosis	32
4.23	Autokorrelation höherer Ordnung	32
4.24	Fourier-Transformation	36
4.25	Power-Spektrum	36
4.27	Heisenbergsche Unschärferelation	37
4.28	Anforderung an ein Mutter-Wavelet	38
4.29	Fraktale Dimension	40

Abkürzungsverzeichnis

CBIR	Content Based Image Retrieval	21
CT	Census-Transformation	25
DFT	Diskrete Fourier-Transformation	36
FFT	Fast Fourier-Transformation	37
FT	Fourier-Transformation	36
GC	Garbage-Collector	44
GLCM	Gray Level Co-occurrence Matrix	21
GLDM	Gray Level Difference Method	23
GLRL	Gray Level Run Length	23
GLRLM	Gray Level Run Length Matrix	23
GUI	grafische Benutzeroberfläche	43
HLAC	Higher Order Local Autocorrelation	32
LBP	Local Binary Pattern	26
MRF	Markov Random Fields	38
PCA	Principle Component Analysis	8
ROC	Receiver Operating Characteristic	10
Weka	Waikato Environment for Knowledge Analysis	14
WFT	Kurzzeit-Fourier-Transformation	37
WLD	Webers Lokaler Deskriptor	30

1. Einleitung

Textile Fasern sind aus unserer heutigen Welt nicht mehr wegzudenken. Allein 2005 wurden mehr als 67 Millionen Tonnen Textilfaser weltweit produziert [Bur06]. Einrichtungsgegenstände zu Hause oder auf der Arbeit, Sitzbezüge im Auto, Kleidung und noch viele weitere Alltagsgegenstände enthalten die unterschiedlichsten Arten von Fasern. Wir kommen mit ihnen in Kontakt, tragen sie weiter, verlieren sie. Das Locard'sche Prinzip sagt aus, dass bei jedem Kontakt zwischen zwei Dingen ein Austausch von Spuren stattfindet [CT00] - beispielsweise zwischen zwei Personen oder aber zwischen einer Person und einem Tatort. Genau diese wichtige forensische Regel gilt besonders bei Fasern. Das macht sie zu einer der wichtigsten und weitverbreitetsten Tatortspuren überhaupt.

Textile Fasern treten in vielen verschiedenen Formen auf, alle mit recht individuellen Merkmalen. Da jede Person einen einzigartigen Geschmack bei beispielsweise der Auswahl von Kleidungsstücken, Autos oder Möbeln hat, wäre es ungewöhnlich, wenn zwei verschiedene Personen an einem Tag mit exakt den gleichen Textilien in Kontakt kommen würden. Mit dieser Annahme und Faserspureninformationen von Tatorten oder Betroffenen lässt sich oft schon der Kreis der Verdächtigen eingrenzen.

Die Herausforderung besteht nun also in der Zuordnung der verschiedenen Faserspuren zu seinem Ursprung. Dabei sind viele Teile dieses Auswertungsprozesses im Bereich der Tatortforensik auch in der modernen Kriminologie weiterhin komplett manuell. Die Ergebnisse kommen durch einen oder mehrere Sachverständige zusammen [RG99] [SWG04] [Bal08]. Dabei spielt nicht nur eine gute Ausbildung, sondern auch viel Erfahrung eine wichtige Rolle. Die Qualität der Ergebnisse kann dadurch stark variieren. Damit ist das Resultat der Untersuchung höchst subjektiv. Hinzu kommt, dass die ohnehin schon viel Zeit in Anspruch nehmende Analyse bei Anfechtung der Ergebnisse von weiteren unabhängigen Experten nachgestellt und bestätigt werden muss. Der Befund ist dadurch jedoch weiterhin subjektiv und immernoch nur schwer nachvollziehbar.

Um Personal zu entlasten, den Untersuchungsprozess zu beschleunigen, um nachvollziehbare und vor allem reproduzierbare Ergebnisse zu erzielen, werden zur Unterstützung immer öfter computergestützte mathematische Modelle zum Untersuchungsprozess hinzugezogen.

Im Rahmen dieser Arbeit werden diverse Möglichkeiten der automatischen Auswertung und Zuordnung der verschiedenen Fasern aufgezeigt. Dazu werden bekannte Methoden aus der

Literatur recherchiert und gegebenenfalls demonstrativ umgesetzt. Zudem werden Techniken aus anderen Fachgebieten vorgestellt und deren Nützlichkeit und Anwendbarkeit im Bereich der Tatortforensik beurteilt.

1.1. Ausgangssituation

Eine Faser ist der kleinste noch sichtbare Baustein, aus dem Textilstoff besteht. Es gibt unzählige verschiedene Fasertypen mit ihren eigenen Vor- und Nachteilen. Man unterteilt sie grob in natürlich vorkommende und künstliche, vom Menschen gemachte Fasern. Natürliche Fasern kommen in der Natur in der Form vor, dass man aus ihnen direkt Stoff herstellen kann. Sie stammen entweder von Tieren (auf Proteinbasis), von Pflanzen (aus Cellulose) oder aus Mineralien. Künstliche Fasern bestehen dagegen entweder aus natürlich vorkommenden Materialien, die erst zu einer Faser weiterverarbeitet werden oder ganz aus synthetischen, rein chemisch gewonnenem Material. Einen Überblick über die Einteilung erhält man mit Blick auf Abbildung 1.1. Die rechts stehenden kursiven Einträge sind jeweils eine Beispielfaser aus der entsprechenden Kategorie.

Es gibt viele Verfahren, um diese Vielzahl an Typen zu unterscheiden. Von einfacher Begutachtung mit bloßem Auge oder Fühlen der Oberfläche [WoCC84] bis hin zu mikroskopischen Untersuchungen [Hou09], Verbrennungstests [Lew10] oder chemischen Analysen [Fan05]. Eigenschaften von Fasern über Verbrennung oder mit Hilfe von Chemie festzustellen, zerstört jedoch oft wichtige Merkmale, im schlimmsten Fall sogar die gesamte Faser selbst. Die Reproduzierbarkeit der Ergebnisse wird dadurch stark eingeschränkt.

Das Erkennen von Merkmalen über sichtbare und spürbare Differenzen ist dagegen subjektiv. Jedoch können, zumindest zum Teil, genau diese Merkmale auch algorithmisch bestimmt werden. Dadurch wird zum einen die Arbeit für Forensiker erleichtert und zum anderen das Ergebnis ein Stück weit objektiver.

Andere Anwendungsdomänen stehen vor ähnlichen Herausforderungen und müssen vergleichbare Merkmale untersuchen. In der Bildverarbeitung [Jen86] [Ban08], -wiederherstellung [Yar] und -komprimierung [NC10] [CP00] oder der Signalverarbeitung findet man viele Merkmale, die sich auch im Bereich der Tatortforensik nutzen lassen. Auch existiert ein großes Forschungsinteresse zur Detektion von Produktionsfehlern wie zum Beispiel von Nahtfehlern in Kleidung [CP00] [KS01] [ZPY01]. Viele der dortigen Forschungsergebnisse können auch im Bereich der Faserforensik nützlich sein.

Im Forschungskolleg DigiDak+¹ entstanden bereits einige Merkmalsräume, die hier berücksich-

¹<https://omen.cs.uni-magdeburg.de/digi-dak-plus/cms>

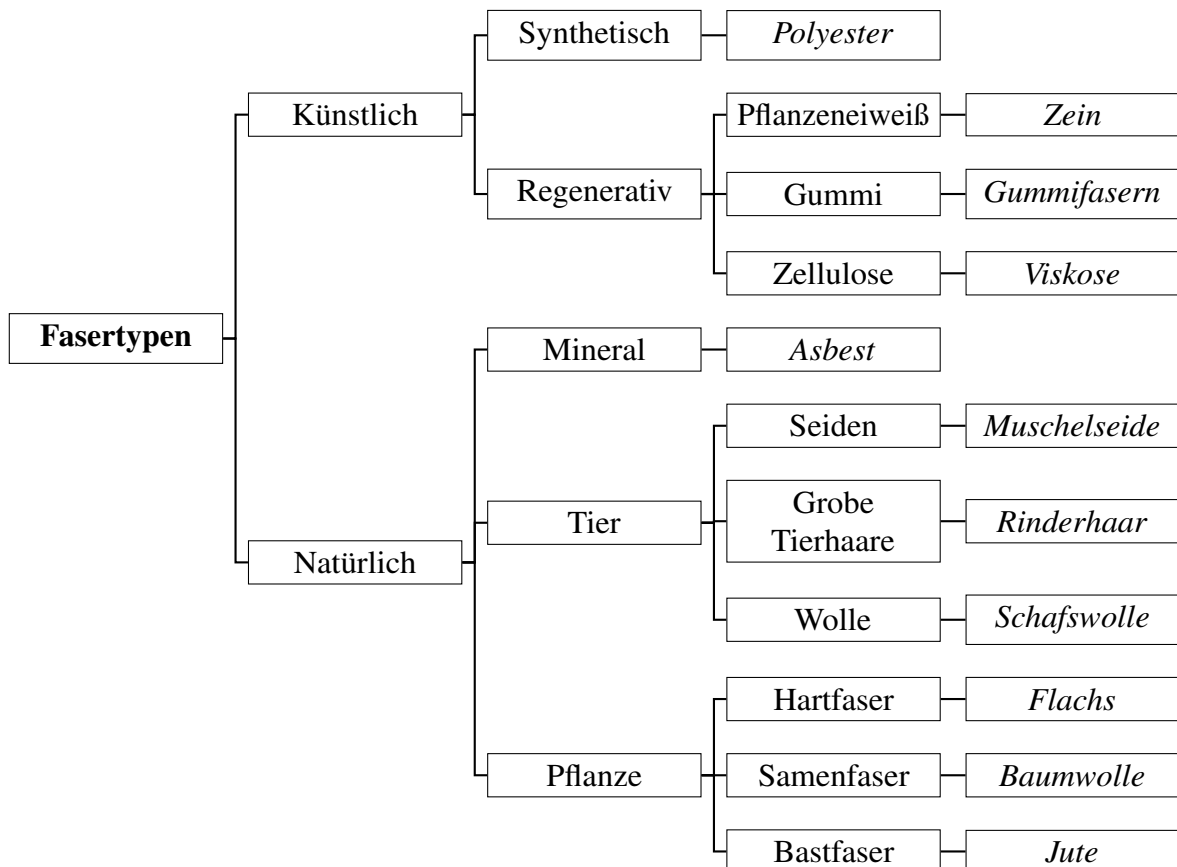


Abbildung 1.1.: Übliche Klassifikation der verschiedenen Fasertypen mit jeweils einem Beispiel als letztes Element in kursiver Schrift (angelehnt an [Nee86])

tigt werden. Diese dienen als Einstiegspunkt für die Literaturrecherche [HAMD12] [AKV12]. Dazu gehören unter anderem Abmessungsmerkmale, Bildmomente [Jan02] [Flu05] [FS05] [BJx02] [LP96] [SF02], Texturmerkmale [HDS73] [GV11] [Oul03] [CP00] [ZPY01] [TH03] [PA97] sowie biometrische Merkmale [Vie05].

1.2. Zielsetzung

Für unterschiedliche Untersuchungsziele werden entsprechende Merkmale aufgezeigt, für jedes dieser Ziele Alternativen genannt und miteinander verglichen. Ihre Eignung für das entsprechende Ziel soll gut begründet sein und wird schlussendlich mit teilweiser Umsetzung evaluiert.

Zur Vergleichbarkeit und zum Zwecke der Demonstration ihrer Anwendbarkeit werden die Merkmalsuntermengen mit gegebenen Testdaten beurteilt.

Ziel dieser Arbeit ist es, eine umfassende, aber dennoch kompakte Sammlung von relevanten

Merkmale aufzuzeigen. Dazu werden schon jetzt verwendete und untersuchte Merkmale sowie Merkmale ganz anderer Domänen herangezogen. Im Vordergrund steht durchgehend ihre Zweckmäßigkeit und Effizienz.

1.3. Abgrenzung

Die Merkmale werden nach dem Untersuchungsziel unterschieden. In der Literatur findet man einige Möglichkeiten der Klassifizierung anhand der Zielsetzung. Als Basis dienen die folgenden fünf Untersuchungsziele (engl.: „forensic use cases“), wie sie auch so zum Teil in anderen Veröffentlichungen [HAMD12] [AKV12] verwendet wurden.

1. Detektion/Lokalisierung
2. Faserkategorisierung
3. Fasertypisierung
4. Individualisierung
5. Spurenkombination

Die Detektion und Lokalisierung beschäftigt sich mit der Erkennung, ob überhaupt und wo Faserspuren auf einer gegebenen Oberfläche vorhanden sind. Hier geht es also in erster Linie um die Segmentierung von Faser und Hintergrund.

Bei der Kategorisierung geht es um die Bestimmung der Herkunft der Faser. Die Einteilung erfolgt dabei wie in den ersten ein bis zwei Ebenen aus Abbildung 1.1.

Genauere Informationen über die Faser erhält man im Bereich Typisierung. Hier soll der konkrete Fasertyp bestimmt werden, beispielsweise Baumwolle oder Schafswolle. Die Einteilung dazu ist ebenfalls in Abbildung 1.1 dargestellt.

Im Bereich der Individualisierung geht man noch einen Schritt weiter. Hier sollen verschiedene Fasern genau zugeordnet werden können, zum Beispiel eine Faser vom Tatort mit einer von einem Kleidungsstück. Die Zuordnung kann über Faserfarben, Verschmutzungen oder angeheftete Partikel geschehen.

Das letzte Untersuchungsziel kann noch einmal in Intraspurkombination und Interspurkombination unterteilt werden. Bei Ersterem geht es um die Informationsgewinnung durch Kombination

mit anderen Mikrospuren wie Staub- oder Schmutzanhäufungen, bei Letzterem um Informationsgewinnung im Zusammenhang mit verschiedenen Spurenarten und dem dazugehörigen a-priori Wissen.

Da man sich sowohl in der Literatur als auch im Forschungskolleg DigiDak+ schon intensiv mit der Detektion und Lokalisierung von textilen Faserspuren beschäftigt hat, wurde dieser Bereich wenig beachtet [Ots79] [KSW85] [Zha04] [SL99] [BS10] [Alg10] [SS04] [RYS95] [BdKGM06]. Die nächste Zielsetzung ist dementsprechend die Faserkategorisierung. Auf genau diese und das nächste eng verbundene Ziel der Fasertypisierung wird in dieser Arbeit der größte Fokus gelegt. Da die Zielsetzungen aufeinander aufbauen, wird den Bereichen Individualisierung und Spurenkombination weniger Beachtung geschenkt.

1.4. Aufbau der Arbeit

Nachdem in diesem ersten Kapitel die Ausgangslage, Motivation und das Ziel der Arbeit dargelegt wurden, werden im zweiten Teil Grundlagen vermittelt, die für das weitere Verständnis zwingend notwendig sind.

Im darauffolgenden Kapitel wird der Ablauf erläutert, mit dem das Ziel erreicht werden soll. Die Arbeit wird dort in kleinere Aufgabenpakete unterteilt. Zu jedem Schritt werden mögliche Einschränkungen sowie getroffene Annahmen aufgezeigt.

Das vierte Kapitel stellt die recherchierten Methoden dar. Diese werden detailliert erläutert und dabei auf ihre Eignung hin untersucht.

Der nächste Abschnitt stellt die Umsetzung einiger dieser Methoden vor. Dargelegt wird hier alles vom technischen Unterbau mit seinen spezifischen Implementierungsdetails bis hin zur entwickelten Benutzeroberfläche.

Das vorletzte Kapitel dreht sich um die Auswertung der gewonnenen Daten. Auch hier werden Anforderungen definiert und getroffene Annahmen motiviert. Die Ergebnisse der gesamten Arbeit werden an dieser Stelle vorgestellt.

Zum Schluss wird im letzten Abschnitt ein Fazit gezogen. Dazu werden auch kurz Vorschläge für mögliche Erweiterungen des Themas diskutiert.

2. Grundlagen

Gewisse Grundlagen aus dem Bereich der Statistik und Bildverarbeitung sind für das weitere Verständnis der nachfolgenden Abschnitte vonnöten. Deshalb werden in diesem Kapitel notwendige Definitionen erklärt, die aufgrund ihres Umfangs nicht im Fließtext an den verwendeten Stellen erörtert werden können. Aber auch hier können sie dank ihrer Komplexität nur kurz umrissen werden. Daher sind für ein tieferes Verständnis an den entsprechenden Stellen Verweise auf dazugehörige Fachliteratur hinterlegt.

2.1. Klassifikation

Unterschiedliche Fasern werden durch unterschiedliche Merkmale beschrieben. Auch die verschiedenen Untersuchungsziele bedingen meist andere Merkmale. Diese Menge an Merkmalen mit ihren dazugehörigen Ausprägungen werden in unterschiedliche Klassen einsortiert. Bei der Detektion beispielsweise gibt es nur die zwei Klassen „Faser“ und „Keine Faser“, bei der Typisierung hingegen gibt es für jeden Fasertyp eine Klasse.

Klassifikation hat nun das Ziel, neue Objekte anhand ihrer Merkmale in die entsprechend vorher definierten Klassen einzuordnen. Es gibt eine große Anzahl von sogenannten Klassifikatoren, die diese Aufgabe erfüllen. In den folgenden Abschnitten wird kurz auf formale Definitionen eingegangen und welche Anforderungen an einen guten Klassifikator gestellt werden.

Formal ausgedrückt bekommt ein Klassifikator f als Eingabe N Samples $X = \{x_1, x_2, \dots, x_N\}$, wobei $x_i = (x_{i,1}; x_{i,2}; \dots; x_{i,d})$ einem Merkmalsvektor entspricht. Die Ausgabe besteht dann aus der Menge $Y \subseteq X$.

Herausforderungen

Eine zu große Anzahl an Merkmalen d und eine zu kleine Zahl an Testdatensätzen N kann zu verschiedenen Problemen führen. Zum einen kann es so leicht zu einer sehr großen Varianz in den Ergebnissen kommen und zum anderen läuft man Gefahr, sein Ergebnis zu stark von den gegebenen Testsamples abhängig zu machen und damit keine allgemeinen Aussagen mehr treffen zu können („Overfitting“ [TLL95]).

Ein zu großer Merkmalsvektor kann ebenfalls zu Problemen führen. Erstmals untersucht und geprägt wurde dazu der Begriff *Fluch der Dimensionalität* von Richard Bellman [Bel61]. Das

Problem kann veranschaulicht werden, indem man sich zuerst einen eindimensionalen Merkmalsraum vorstellt. Weiterhin wird angenommen, dass die Werte in dieser einen Dimension nur natürliche Zahlen von eins bis einschließlich acht annehmen dürfen. Um diesen Merkmalsraum komplett abdecken zu können, benötigen man nun mindestens acht verschiedene Stichproben. Es werden dementsprechend nur wenige Beobachtungen benötigt, um den Raum zumindest gut abdecken zu können. Wird eine Dimension hinzugefügt, so braucht man für diesen zweidimensionalen Raum schon $8^2 = 64$ Stichproben, bei drei Dimensionen schon ganze 512. Bei einem Vektor mit 100 Einträgen benötigt man $8^{100} \approx 2 \cdot 10^{30}$ Stichproben, um eine genauso gute Abdeckung zu erreichen. Das Erstellen einer solchen Menge an Stichproben ist sehr schwierig, wenn nicht gar unmöglich.

Der eigentliche Fluch jedoch besteht vielmehr darin, dass die Ergebnisse von Distanzfunktionen, die zur Klassifikation benutzt werden, nicht mehr zu gebrauchen sind, denn der Unterschied zwischen der kleinsten und größten Distanz im Vergleich zur kleinsten wird umso kleiner, je mehr Dimensionen im Spiel sind.

$$\lim_{Dimension \rightarrow \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \rightarrow 0 \quad (2.1)$$

Eine weitere Problematik ist, dass in vielen Fällen das Ergebnis Y_i nicht durch den gesamten Vektor X_i bestimmt wird, sondern nur durch eine Untermenge $\{x_{(1)}, x_{(2)}, \dots, x_{(n)}\}$ mit $n < N$. Im schlimmsten Fall können einige Merkmale die Klassifikationspräzision sogar stark senken [BL97].

Auswahl

Es existieren sehr viele Klassifikationsverfahren [Aly05]. Angemessen ist es, aus zumindest einigen der verschiedenen Kategorien, in die Klassifikatoren eingeteilt werden, mindestens einen Vertreter zu verwenden.

Weit verbreitet und in Hinblick auf die Nutzung von Feature-Ranking-Algorithmen können hier beispielsweise k-Nearest-Neighbor (k-NN) und C4.5 [Qui92] verwendet werden. Entscheidungsbäume wie beim C4.5 haben das Potenzial, automatisch relevante Features zu bevorzugen und irrelevante zu ignorieren. k-NN hingegen besitzt keine eingebaute Selektion.

2.2. Merkmalsgüte

Zur Beurteilung der Güte eines Merkmalsvektors x wird eine Gütefunktion $G(x_i) = G(x^1, x^2, \dots, x^d)$ benötigt. Bei statistisch unabhängigen Merkmalen ist die Gesamtgüte des Vektors die Summe

der einzelnen Merkmalsgütwerte:

$$G(x_1, x_2, \dots, x_d) = \sum_{k=0}^d G(x_k) \quad (2.2)$$

Nach der absteigenden Sortierung der Güte G aller einzelnen Merkmale x_i , sind die ersten $N' < N$ Merkmale die beste Merkmalskombination. Die Herangehensweise nennt sich „Feature Ranking“.

Um dafür zu sorgen, dass die Merkmale unabhängig sind, wird zuvor oft eine Merkmalskombination durchgeführt. Bei der häufig auch Extraktion genannten Methode werden verschiedene Merkmale zu einem einzigen neuen Merkmal zusammengeführt. Detailliert wird darauf in Abschnitt 2.2.2 eingegangen.

Bei der Merkmalsselektion werden ganze Features noch vor der Klassifikation aus dem Feature-Vektor herausgenommen.

2.2.1. Selektion

Die Idee hinter der Merkmalsselektion ist es, redundante oder irrelevante Merkmale auszusortieren [JKP94]. Dazu wird eine neue Untermenge X' erzeugt, mit $|X'| < |X|$.

2.2.2. Kombination

Die Merkmalskombination kann als Spezialfall der Dimensionsreduktion angesehen werden. Das Ziel ist es, die Anzahl an Attributen zu senken, ohne dabei viel an Informationen aus dem originalen Attributraum zu verlieren. So lassen sich Ergebnisse auch teilweise übersichtlicher und besser präsentieren [TSK05], was sehr gut für die spätere Visualisierung ist. Ein großer Nachteil ist jedoch, dass die Linearkombinationen der Originalmerkmale dann meist schwerer zu interpretieren sind. Zudem geht die Information, welches Attribut wieviel zum Ergebnis beigetragen hat, im Allgemeinen ebenfalls verloren [RS00] [Fod02]. Weitverbreitet ist hierfür die Hauptkomponentenanalyse (engl.: Principle Component Analysis (PCA)) [Pea01] [Jol02].

Um zwei oder mehr Merkmale sinnvoll zu kombinieren, müssen diese sich ähneln oder zumindest zu fast identischen Ergebnissen führen. Je nach Verfahren kann dies abhängig vom jeweiligen Problem oder dem verwendeten Klassifikator sein. Diese Information kann man verwenden, um zumindest eine relative Güte der Merkmale bzw. der gebildeten Merkmalsunterräume abzulesen.

Jedoch verhindern die schon genannten Nachteile einen direkten Rückschluss auf einzelne

Merkmale. Damit sind diese Verfahren weniger sinnvoll zur Bewertung einzelner Features, helfen aber, die Gesamtbewertung des Merkmalsraumes zu erhöhen.

Da es für die Selektionsmethoden wichtig ist, dass die Merkmale statistisch unabhängig sind, muss die Kombination noch vor der Selektion durchgeführt werden. Gegebenenfalls dürfen sich die korrelierenden Merkmale zur Bestimmung der Güte nicht in einem Merkmalsvektor befinden.

2.3. Statistische Kenngrößen

Da der gesamte Auswertungsprozess auf Statistiken beruht, sollen hier noch einmal wichtige grundlegende Kenn- und Messgrößen erklärt werden.

Jede zu klassifizierende Faser gehört genau einer Kategorie an. Der Klassifizierungsalgorithmus weist dem Scan ebenfalls eine Kategorie zu. Diese Zuordnung kann entweder falsch oder richtig sein.

		Gegeben		
		Positiv	Negativ	
Ergebnis	Positiv	richtig positiv (R_p)	falsch positiv (F_p)	Genauigkeit $= \frac{\sum R_p}{\sum R_p + F_p}$
	Negativ	falsch negativ (F_n)	richtig negativ (R_n)	Trennfähigkeit $= \frac{\sum R_n}{\sum F_n + R_n}$
		Sensitivität $= \frac{\sum R_p}{\sum R_p + F_n}$	Spezifität $= \frac{\sum R_n}{\sum F_p + R_n}$	Präzision $= \frac{\sum R_p + \sum R_n}{\sum \text{Gesamtinstanzen}}$

Tabelle 2.1.: Kontingenztabelle mit den wichtigsten ablesbaren statistischen Kenngrößen

Zur Vereinfachung wird erst einmal angenommen, es gäbe nur zwei Klassen A und B, also es handele sich um ein sogenanntes 2-Klassen-Problem. Dann kann der Ausgang der Klassifizierung insgesamt vier Zustände annehmen. Entweder eine Faser aus der Klasse A wurde richtig als A erkannt oder fälschlicherweise der Klasse B zugeordnet oder aber entweder eine Faser der Klasse B wurde richtig als B erkannt oder falsch als A. Eine Verallgemeinerung mit Positiv und Negativ als Zuweisung kann man gut in einer Kontingenztabelle darstellen, wie sie 1904 von K. Pearson [Pea04] erstmals genannt wurde. Eine Darstellung dieser ist nach [AD81] in

Tabelle 2.1 zu sehen. Dort sind ebenfalls alle vier Möglichkeiten übersichtlich dargestellt.

Zusätzlich abgebildet sind dort gängige Kenngrößen, die als Basis für die spätere Auswertung herangezogen werden. Mit ihnen lassen sich weitere Gütemaße errechnen, so beispielsweise das F-Maß mit:

$$F = 2 \cdot \frac{\text{Genauigkeit} \cdot \text{Sensitivität}}{\text{Genauigkeit} + \text{Sensitivität}} \quad (2.3)$$

Es ist nicht möglich, nach allen Kriterien zu optimieren. Beispielsweise haben Sensitivität und Spezifität einen negativen Korrelationskoeffizienten. Man muss dementsprechend immer einen Kompromiss finden zwischen zu viel fälschlich als positiv bzw. als negativ klassifiziert.

Um einen solchen Schwellwert zu finden, wird oft eine Receiver Operating Characteristic (ROC)-Kurve benutzt. In ein Diagramm werden Sensitivität und Spezifität von Klassifikationsergebnissen mit unterschiedlichen Parametern eingetragen. Daraus ergibt sich die ROC-Kurve. Mit dieser lässt sich zum einen leicht der beste Parameter ablesen und zum anderen dient sie als wichtiges Qualitätsmaß.

Die Fläche unter der ROC-Kurve kann Werte von 0,0 bis 1,0 annehmen, wobei 0,5 das Ergebnis eines Zufallsprozesses ist. Niedrigere Werte sind im Allgemeinen ebenfalls negativ zu interpretieren. Ein Wert größer als 0,5 bedeutet hingegen, dass zumindest bei geeigneter Parameterwahl das Ergebnis gut ist.

2.4. Textur

In der Computer Vision Literatur existieren sehr viele unterschiedliche Definitionen, was eine Textur ist. In einer schon etwas älteren Definition des IEEE heißt es zum Beispiel:

“Textur ist ein Attribut, das die räumliche Anordnung der Grauwerte von Pixeln in einer Bildregion darstellt.” [IEE90]

Andere Beschreibungen gehen bis in das Jahr 1950 zurück [Gib50]. Angelehnt an [Sk178] wird hier eine Textur als eine Aneinanderreihung von sich wiederholenden kleineren Mustern definiert. Nur mit Hilfe des Musters, auch Primitive genannt [Har79], und einer Platzierungsregel lassen sich diese deterministischen Texturen beschreiben. Manche Muster folgen statistischen, nicht deterministischen Regeln. Diese Texturen werden auch *Stochastic Textures* genannt.

In der Praxis existieren keine rein deterministischen Primitive. Genauso ist auch die Platzierung immer etwas verschoben oder folgt nicht genau einer festen mathematisch abbildbaren Regel. Man kann auch feststellen, dass einige Primitive noch weiter in kleinere stochastische Muster unterteilt werden können, sogenannte Mikrotexturen.

3. Konzept

In diesem Kapitel wird ein grober Plan über das weitere Vorgehen skizziert. Die Aufteilung der Verarbeitung ist an die computergestützte forensische Faseranalyse angelehnt, wie sie in Abbildung 3.1 zu sehen ist [HAMD12].

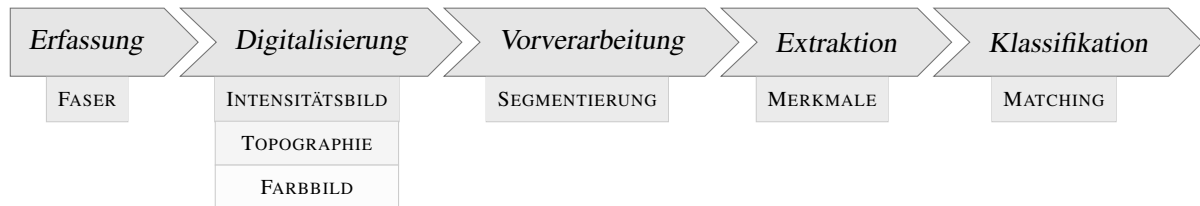


Abbildung 3.1.: Verarbeitungspipeline, bestehend aus fünf Phasen mit ihren dazugehörigen Ausgaben.

Zuerst wird geklärt, woher die Daten zur Analyse stammen und welche Annahmen man über sie treffen kann. Dabei wird auch auf den Digitalisierungsvorgang und mögliche Vorverarbeitungsschritte eingegangen. Ebenfalls erörtert wird, welche der enthaltenen Informationen speziell für uns wichtig sind und welche nicht näher betrachtet werden.

Danach wird kurz darauf eingegangen, in welcher Form die Analyse (Merkmalsextraktion) stattfinden soll und worauf hier der Fokus liegt.

Im letzten Teil der Verarbeitungspipeline geht es dann um die Klassifizierung. Hier geht es um die Frage, welche Techniken Anwendung finden sollen und wie deren Ergebnisse auszuwerten sind.

Abschließend wird kurz dargelegt, wie der Arbeitsablauf vonstatten ging, um diese Schritte durchführen zu können.

3.1. Datenerfassung und Digitalisierung

Die Fasern sind im Rahmen der Forschungstätigkeiten des DigiDak+ Projekts gesammelt und erfasst worden. Diese wurden sowohl an der FH Brandenburg mit dem Konfokalmikroskop vom Typ *Keyence VK-X 105* [Keya] sowie mit dem *Keyence VK-X 110* [Keyb] der Otto-von-Guericke-Universität Magdeburg digitalisiert. Entstanden sind daraus digitale Abbilder mit Topographie-, Intensitäts- und Farbinformationen in unterschiedlichen Vergrößerungsstufen von mindestens 5- bis maximal 100-facher Vergrößerung. Je nach Fasertyp sind dennoch nicht alle Stufen geeignet. So können Kunstfasern beispielsweise sehr schmal sein und nur wenig Oberflächenstruktur aufweisen. Diese würde man mit 5-facher Vergrößerung kaum erkennen

können.

Da die Farbinformationen starken Schwankungen bei unterschiedlichen Scanvorgängen unterworfen waren und diese zudem von vielen Algorithmen nicht ohne weitere Anpassung benutzt werden können, werden diese hier nicht weiter betrachtet.

Topographie- und Intensitätsbilder stammen zwar vom gleichen Sensor, doch werden beide während der Digitalisierung durch die Keyence-Soft- und Firmware unterschiedlich (vor-)verarbeitet. Prinzipiell sollten sie dennoch beide ein ähnliches Maß an Merkmalen bieten und werden deshalb gleichberechtigt mit in die Analyse einbezogen. Die Auswertung wird zeigen, ob und inwieweit Unterschiede in ihrem Informationsgehalt vorhanden sind.

Um die Analyse im späteren Verlauf zu vereinfachen, sind alle Fasern unter gleichen Bedingungen aufgenommen worden. Sämtliche Fasern wurden auf dem gleichen Objektträger, nämlich Glas, befestigt. Glas bietet eine homogene Oberfläche, ist einfach von Schmutzpartikeln zu befreien und bietet günstige Reflektionseigenschaften.

Da nicht nur der Untergrund, sondern auch das zu untersuchende Objekt durch das konfokale Messprinzip recht flach sein muss, wurden alle Fasern begradigt und flach auf den Objektträger an den Enden festgeklebt. Dadurch gehen leider spezifische Informationen wie die Kräuselung verloren. Da die Tiefe des Objekts bei einem Konfokalmikroskop über die ganze Fläche im Mittel gleich sein muss, ist eine andere Herangehensweise nur sehr schwer möglich..

Daraus ergibt sich, dass in diesem Fall nur die Oberfläche der Faser untersucht werden kann. Die Größe oder Form des Querschnitts sind zwar nicht zu unterschätzende Merkmale von Fasern, können aber hier aufgrund der Datenerfassungstechnik nicht betrachtet werden. Durch das Abkleben der Enden werden zudem weitere gängige Merkmale vernachlässigt.

Aus diesem Grund liegt der Fokus in dieser Arbeit komplett auf der Oberflächenstruktur der Faser, in unserem Kontext also der Textur, wie sie in Abschnitt 2.4 beschrieben wurde.

Damit die automatische Klassifizierung später validiert werden kann, ist es wichtig, nicht zu vergessen, jeden Scan gut zu beschriften. Fasertyp, Scantyp und Vergrößerung müssen mindestens von jedem Sample bekannt sein.

3.2. Vorverarbeitung

Um die Messergebnisse nicht zu verfälschen, wird keine Vorverarbeitung durchgeführt. Die Analyse findet nur auf den Rohmessdaten statt. Einzig die Segmentierung von Hintergrund und eigentlicher Faser wird noch vor der Merkmalsextraktion durchgeführt.

3.3. Merkmalsextraktion

Die zu extrahierenden Merkmale sind allesamt von der Textur der Faser abgeleitet. Durch die Art der Datenbeschaffung können andere typische Eigenschaften hier einfach nicht untersucht werden.

In Abbildung 3.2 ist gut zu erkennen, dass es sich dabei auch wirklich um Texturen handelt. Die obere Darstellung ist ein Intensitätsbild einer Schafswollfaser, aufgenommen mit einer 50-fachen Vergrößerung. Im unteren Bild hingegen ist zur Demonstration der Musterwiederholungseigenschaft von Texturen eine künstlich computergenerierte Oberfläche zu sehen. Dabei wurde als Grundmuster der rot umrandete Ausschnitt des originalen Scans verwendet. Dieser wurde dann wiederholt nacheinander abgebildet. Da natürliche Texturen sich niemals perfekt wiederholen, wurden auch hier die Grundprimitiven mit einem zufälligen Versatz platziert. Auch wenn die generierte Oberfläche nicht aussieht wie das obige Original, so könnte man auf den ersten Blick doch vermuten, es handele sich hier um einen legitimen Scan.

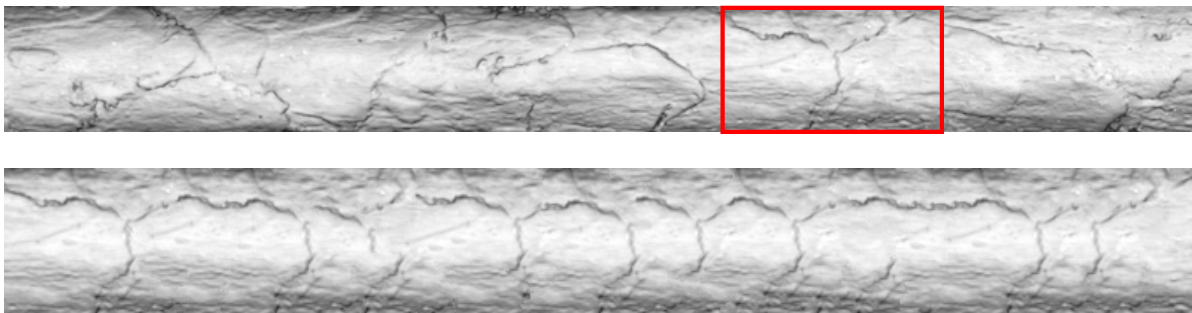


Abbildung 3.2.: Oben: Intensitätsscan einer Schafswollfaser mit 50-facher Vergrößerung;
Unten: Algorithmisch generierte Textur, bestehend aus der gekennzeichneten Primitive der oberen Faser

Texturen werden auch in vielen anderen Anwendungsgebieten untersucht und klassifiziert. Genau diese Algorithmen werden im nächsten Kapitel aufgezeigt und näher begutachtet. Jedes Verfahren soll dabei verschiedene Anforderungen zumindest zum Teil erfüllen können. Nachdem einige mögliche Kandidaten ausgewählt wurden, werden ausgesuchte davon zur näheren Untersuchung implementiert.

Jeder Algorithmus hat zum Ziel, aus einem Eingabescan ein oder mehrere Merkmale zu extrahieren. Dazu soll eine minimale Testumgebung geschaffen werden. In dieser können die Scans eingelesen und gegebenenfalls manuell vorverarbeitet werden. Die unterschiedlichen Algorithmen werden dort zur Auswahl gestellt und auf die entsprechenden Scandaten angewandt. Als Ausgabe erhält man einen Merkmalsvektor pro Faser, Vergrößerung und Scantyp.

Auch soll die Implementierung typische forensische Anforderungen erfüllen. Die einzelnen Schritte der Verarbeitung müssen offensichtlich und leicht nachvollziehbar sein. Ebenfalls müssen die Ergebnisse jederzeit wiederholt werden können. Details darüber sind in Kapitel 5 zu finden.

3.4. Klassifizierung

Nachdem alle Merkmalsvektoren errechnet wurden, werden diese im nächsten Schritt klassifiziert. Wie in Abschnitt 1.3 besprochen, soll hier eine automatische Einteilung nach der genauen Faserart stattfinden. Pro Scantyp und pro Vergrößerungsfaktor ergibt sich eine zu klassifizierende Menge. Auch wenn intuitiv nicht sofort ersichtlich, ist es dennoch angebracht, dieselbe Faser mit unterschiedlichem Vergrößerungsfaktor als einzelne Klasse anzusehen. Nicht alle Algorithmen sind invariant gegenüber Skalierung. Zudem entfallen bei nur geringer Vergrößerung einige Details der Oberfläche, die die Algorithmen dann nicht erfassen können. Für den forensischen Vorgang entsteht hier kein Nachteil. Auch in realen Anwendungsszenarien müssen die Fasern mit verschiedenen, fixen Vergrößerungsstufen eingescannt werden.

Im Gegensatz zu den Analysealgorithmen werden die Klassifizierungsalgorithmen nicht selbst implementiert. Hierzu wird das Open Source Softwaretool Waikato Environment for Knowledge Analysis (Weka) ¹ verwendet. Dieses bietet eine sehr große Auswahl von Klassifikatoren sowie auch Selektoren und Verfahren zur Merkmalskombination wie sie im Grundlagenkapitel 2.1 beschrieben sind.

Wenn möglich, und wenn nicht weiter angegeben, werden nur jeweils die Standardeinstellungen des jeweiligen Klassifikators benutzt. Die nur schwer überschaubare Menge an Einstellmöglichkeiten macht es ohnehin sehr schwierig, wenn nicht sogar unmöglich, die optimalen Parameter zu finden.

Es werden Klassifikatoren aus unterschiedlichen Kategorien verwendet. Wie gut welche Art von Klassifikator abschließt, ist im vornherein nur schwer abzuschätzen. Die Auswahl findet deshalb eher zufällig statt. Immerhin sollen die Ergebnisse experimentell zustande kommen und eine gewisse Allgemeingültigkeit besitzen.

Alternative Klassifikationsmethoden werden nicht ausgewertet. Diese werden nur kurz im Kapitel 7 (Ausblick) beschrieben.

¹<http://www.cs.waikato.ac.nz/ml/weka/>

3.5. Auswertung

Der letzte Schritt ist die Auswertung und Interpretation der gesammelten Erkenntnisse. Auch wenn dieser nicht mehr ins forensische Vorgehensmodell gehört, ist er hier einer der wichtigsten. Durch gängige statistische Messverfahren soll geklärt werden, welche der Merkmalsextraktionsalgorithmen sich für die Klassifizierung von Fasern am besten eignen und welche Einschränkungen sie mit sich bringen.

Weiterhin sollen hier Aussagen über die verschiedenen Klassifizierer getroffen werden. Wie schon im Grundlagenkapitel 2.1 erklärt, eignen sich nicht alle gleichermaßen und sind ebenfalls abhängig von verschiedenen Parametern und ihrer Eingabe.

3.6. Arbeitsablauf

Die Arbeit wurde in folgende vier Phasen aufgeteilt:

1. Recherche und Zuordnung bekannter Merkmale aus der digitalen Tatortforensik
2. Nicht berücksichtigte Merkmale identifizieren
3. Umsetzung
4. Evaluierung anhand von Testdaten

Der erste Teil bestand aus einer ausführlichen Literaturrecherche. Ausgehend von den gegebenen Merkmalen wurden weitere relevante Features recherchiert. Relevant sind unter anderem all diejenigen Merkmale, die in anderen Fachdomänen schon länger Anwendung finden, in der Auswertung von Faserspuren jedoch noch nicht auf ihre Tauglichkeit getestet wurden. Die gefundenen Merkmale wurden in einer systematischen Herangehensweise auf ihre Plausibilität geprüft. Die Ergebnisse dieser Recherche sind in Kapitel 4 zu finden.

In der zweiten Phase wurden weitere Merkmale aus anderen Anwendungsbereichen ermittelt. Für sämtliche Merkmale wird die Primärquelle und die spezifische Anwendungsdomäne genannt. Bei Merkmalen, die in der reinen Bildverarbeitung noch nicht bekannt sind, wird die Art der Neuheit genau spezifiziert. Ein besonderer Fokus wurde hier auf die Frage gelegt, ob diese sich in der forensischen Faserklassifizierung überhaupt anwenden lassen. Näheres zu der Merkmalsextraktion ist in den entsprechenden Unterpunkten in Kapitel 4 zu finden.

In Phase drei wurden einige der Merkmale demonstrativ umgesetzt. Da nicht alle evaluierten Merkmale in der Praxis verwirklicht werden konnten, lag hier das Hauptaugenmerk auf denjenigen, zu denen es in der digitalen Tatortforensik nur wenige bis gar keine Auswertungen gibt. Auf die für die Umsetzung verwendete Technik wird in Abschnitt 5 noch näher eingegangen.

Im finalen Schritt wurden die umgesetzten Merkmalsräume evaluiert. Hierzu werden getrennt für jedes Untersuchungsziel die Merkmale an gegebenen experimentellen Testdaten untersucht. Näheres zu den Testdaten findet sich im Abschnitt 3.1. Um den Aufwand der Berechnungen zu verringern und um die Güte einzelner Merkmale festzustellen, werden die Merkmale gegebenenfalls zuvor selektiert (siehe Abschnitt 2.2.1).

Anschließend wurden geeignete Testgrößen aufgestellt bzw. aus der Literatur ermittelt, um die Ergebnisse untereinander und mit Ergebnissen anderer Veröffentlichungen vergleichen zu können. Mithilfe der daraus gewonnenen Daten wurde dann ein abschließendes Fazit in Kapitel 7 für die verschiedenen Merkmale aufgestellt.

4. Methodologie

Jetzt, da alle notwendigen Grundlagen vermittelt worden sind, beschreibt dieses Kapitel nun die im Laufe der Arbeit recherchierten Algorithmen. Dazu werden zu Beginn erst einmal notwendige Anforderungen definiert und diese begründet. Die Algorithmen werden dabei zur Übersichtlichkeit in verschiedene Kategorien eingeordnet. Am Ende einer jeden Kategorie folgt eine Zusammenfassung, um alle wichtigen Fakten im Überblick zu behalten.

4.1. Anforderungen

Um die schiere Menge an möglichen Algorithmen einzugrenzen, werden hier einige Anforderungen definiert. In vielen Fällen ist eine konkrete Einordnung, ob der Algorithmus der jeweiligen Anforderung gerecht wird oder nicht, nur sehr schwer möglich. Aus dem Grund sind auch nicht alle Anforderungen immer zwingend notwendig.

1. Robustheit gegenüber Rauschen

Wie gut kann der Algorithmus mit verrauschten Eingangsdaten umgehen? Inwieweit beeinflussen Bildstörungen das Ergebnis?

2. Robustheit gegenüber Parameteranpassung

Der Algorithmus hat gegebenenfalls unterschiedliche Parameter, die das Ergebnis beeinflussen. Es ist nicht einfach, immer die optimalen Parameterwerte zu finden. Sind diese dann nur auf einem kleinen Bereich von Testdaten gut angepasst oder können die Werte auch für eine sehr große und differenzierte Testmenge verwendet werden?

3. Lokalität

Features sollen möglichst lokal sein. Damit wird die Gefahr reduziert, dass kleinere Verdeckungen des Scans oder Fehler an der Faser das Ergebnis zu stark beeinflussen.

4. Notwendige Stichprobengröße

Wie viele Trainingsdaten sind notwendig, um eine gute Beschreibung und Abgrenzung der unterschiedlichen Merkmalsausprägungen zu gewinnen?

5. Invarianz

Wie gut verkraftet der Algorithmus leichte geometrische Veränderungen der Eingangs-Informationen wie zum Beispiel Skalierung, Rotation oder aber eine unterschiedliche Projektion?

6. Geringer Rechenaufwand

Die Verarbeitungsdauer des Algorithmus muss auf angebrachter Hardware vertretbar sein. Da hierbei eine Menge Faktoren eine Rolle spielen, wird an dieser Stelle wenn möglich die theoretische Komplexität der Algorithmen angegeben.

Wie gut die einzelnen Verfahren untereinander im jeweiligen Bereich abschneiden, wird in der Auswertung diskutiert.

4.2. Einordnung

Angelehnt an Haralicks Texture Survey [Har79] werden Feature-Extraction-Algorithmen oft in zwei Kategorien unterteilt, zum einen die strukturbedingten (*structural*) und zum anderen die statistischen (*statistical*). Diese grobe Einteilung hat lange Zeit gereicht und wurde vielfach in der Literatur verwendet [Wec80] [VDO85]. Da sich jedoch in den letzten 30 Jahren viel auf dem Gebiet getan hat, wird hier eine erweiterte Kategorisierung angewandt. Angelehnt an [TJ93] werden die verschiedenen Algorithmen in insgesamt vier Kategorien eingeordnet: statistische, geometrische, modellbasierende und Methoden der Signalverarbeitung.

Diese Einteilung wird für die in diesem Kapitel genannten Algorithmen zur Übersichtlichkeit und besseren Recherchierbarkeit benutzt.

4.3. Schwerpunkt

In den letzten Jahrzehnten sind außerordentlich viele Texturextraktionsmethoden entwickelt worden. Bei weitem nicht alle können hier wiedergegeben werden, es müssen Abstriche gemacht werden. Aus diesem Grund liegt der Schwerpunkt in dieser Arbeit auf der statistischen Analyse. Algorithmen aus den anderen drei Kategorien werden nur kurz vorgestellt und mit Literatur für eine tiefgehendere Recherche versehen.

Zudem bestehen sehr viele Methoden aus Kombinationen von Grundalgorithmen, die sich experimentell für einen speziellen Anwendungsbereich als sinnvoll herausgestellt haben. Diese werden hier ebenfalls nicht betrachtet. Nur die zugrundeliegenden Vorgehen sind hier von Interesse.

4.4. Statistische Analyse

Bei der statistischen Analyse wird die räumliche Verteilung von meist Grauwerten der Scans analysiert. Dabei werden lokale Merkmale in jedem Punkt eines Bildes berechnet. Zu jedem

dieser Merkmale werden Statistiken zur gesamten Verteilung abgeleitet. Man unterscheidet hier zwischen Statistik erster Ordnung, bei der die Merkmale abhängig von einem einzigen Pixel sind, und zweiter oder höherer Ordnung, bei der die Merkmale durch Beschreibung der Abhängigkeit von zwei oder mehreren Pixeln entstehen.

Für die nachfolgenden Algorithmen gilt: $I(x, y)$ mit $x \in [0, N)$ und $y \in [0, M)$ ist ein $N \times M$ Intensitätsbild mit G verschiedenen möglichen Grauwerten und $n = N \cdot M$ Pixeln. Einfachheitshalber werden die meisten Verfahren eine $N \times N$ Matrix voraussetzen.

4.4.1. Tamuras Texturmerkmale

Tamura et al. definieren sechs verschiedene Texturmerkmale - Coarsness, Contrast, Directionality, Linelikeness, Regularity und Roughness [TMY78]. Diese orientieren sich an der optischen Wahrnehmung des Menschen.

Dabei werden meist nur die ersten drei Eigenschaften verwendet, da die anderen drei eine Linearkombination der ersteren sind.

Coarsness

Beschreibt, wie rau beziehungsweise granular eine Textur ist. Sie wird von Tamura als die wichtigste Eigenschaft von allen bezeichnet.

Als erstes wird zu jedem Pixel (x, y) der Durchschnitt seiner Umgebung bei unterschiedlichen Skalierungen berechnet:

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{f(i, j)}{2^{2k}}$$

mit $k = 1, 2, \dots, 5$.

Im nächsten Schritt wird die Differenz von allen Paaren aller nicht überlappender Nachbarschaften von sich gegenüberliegenden Punkten in horizontaler ($E_{k,h}$) wie auch in vertikaler ($E_{k,v}$) Richtung berechnet.

$$E_{k,h} = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|$$

$$E_{k,v} = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})|$$

Für jedes Pixel wird das k herausgesucht, das die größte Veränderung bewirkt (in beide Richtungen).

$$E_k = E_{max} = \max(E_1, E_2, \dots, E_L)$$

Die Coarsness F_{crs} ist dann der Durchschnitt eines Bildes oder Bildbereiches mit $S_{opt}(x, y) = 2^{k_{opt}}$.

$$F_{crs} = \frac{1}{M \cdot N} \sum_x^n \sum_y^m S_{opt}(x, y) \quad (4.1)$$

Contrast

Beschreibt, wie Grauwerte in einem Bild variieren und inwieweit sich die Grauwertverteilung eher in Richtung Schwarz oder Weiß verzerrt. Berechnet wird das mit Hilfe der Standardabweichung σ und der Kurtosis α_4 .

$$F_{con} = \frac{\sigma}{(\alpha_4)^n} \quad (4.2)$$

Wobei $\alpha_4 = \mu_4 / \sigma^4$ und μ_4 das vierte Bildmoment ist. Tamura hat experimentell herausgefunden, dass $n = \frac{1}{4}$ ungefähr der menschlichen Wahrnehmung entspricht.

Directionality

Für jedes Pixel wird der Gradient in horizontaler (Δ_H) und vertikaler (Δ_V) Richtung berechnet. Zu jedem Gradienten wird ein Winkel

$$\Theta = \arctan\left(\frac{\Delta_V}{\Delta_H}\right) + \frac{\pi}{2}$$

berechnet. Daraus ergibt sich ein Richtungshistogramm

$$H_D(k) = \frac{N_\Theta(k)}{\sum_{i=0}^{n-1} N_\Theta(i)}$$

mit $k = 0, 1, \dots, n-1$. Das Maß der Direktionalität kann nun wie folgt berechnet werden:

$$F_{dir} = 1 - r \cdot n_p \sum_{p=1}^{n_p} \sum_{a \in w_p} (a - a_p)^2 H_D(a) \quad (4.3)$$

wobei n_p die Anzahl der Höhepunkte, a_p die Position des p -ten Höhepunkts, w_p der Bereich der Winkel, die zu a_p dazugehören (sprich, der Bereich des Tals um den Höhepunkt herum)

und r der normalisierte Faktor der Anzahl der quantisierten Bereiche des Winkels a ist.

Sehr häufig werden die Tamura-Merkmale im Bereich Content Based Image Retrieval (CBIR) eingesetzt, also dem Finden von Bildern in einer Datenbank, die einem Suchbild ähnlich sind.

4.4.2. Co-occurrence Matrix

Die Co-occurrence Matrix beschreibt Eigenschaften eines Bildes mit Hilfe von Statistiken zweiter Ordnung. Wir beziehen uns hier speziell auf die von Haralick [HDS73] untersuchte Gray Level Co-occurrence Matrix (GLCM). Diese ist sehr bekannt und wird häufig zur Beschreibung von Texturen verwendet.

Die $G \times G$ Co-occurrence Matrix $P_d(i, j)$ mit einem Versatzvektor $d = (dx, dy)$ ist wie folgt definiert: Für jeden Eintrag (i, j) wird die Anzahl der Vorkommen (engl.: occurrences) von Paaren von Grauwerten i und j mit dem Abstand d gespeichert. Formal ausgedrückt bedeutet das:

$$P_d(i, j) = |\{(r, s), (t, v) : I(r, s) = i, I(t, v) = j\}| \quad (4.4)$$

wobei $(r, s), (t, v) \in N \times N$ und $(t, v) = (r + dx, s + dy)$ gilt.

Ein Beispiel zur Veranschaulichung ist in Abbildung 4.1 zu sehen.

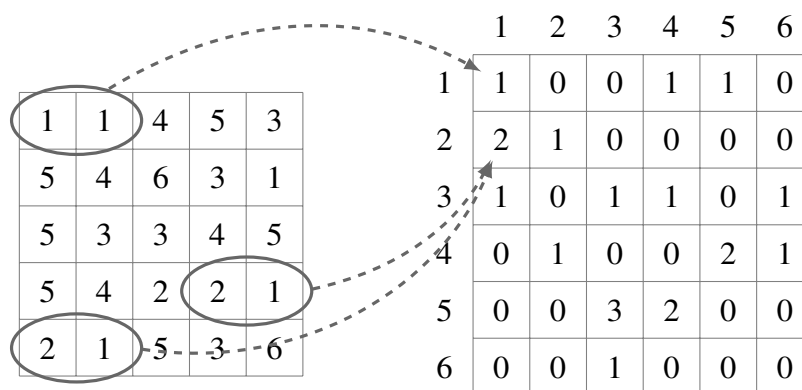


Abbildung 4.1.: Links: 5x5 Bild mit 6 Graustufen; Rechts: GLCM mit $d = (1, 0)$

Die GLCM gibt Auskunft über die örtliche Verteilung der Grauwerte in einem Bild. Mit deren Hilfe kann man verschiedenste Eigenschaften eines Bildes extrahieren. Die von Haralick veröffentlichten Merkmale [HDS73] sind in Tabelle 4.1 aufgelistet.

Texturmerkmal	Formel
Trägheit	$\sum_{i,j=0}^{N-1} P_d(i,j)^2 = ASM$
Kontrast	$\sum_{n=0}^{N-1} n^2 \cdot \sum_{i,j=0}^{N-1} P_d(i,j), i-j = n$
Verschiedenheit	$\sum_{i,j=0}^{N-1} P_d(i,j) \cdot i-j $
Homogenität	$\sum_{i,j=0}^{N-1} \frac{P_d(i,j)}{1+(i-j)^2}$
Entropie	$\sum_{i,j=0}^{N-1} P_d(i,j) \cdot (-\log P_d(i,j))$
Korrelation	$\sum_{i,j=0}^{N-1} (i - \mu_x) \cdot (j - \mu_y) \cdot P_d(i,j) \cdot \frac{1}{\sigma_x \sigma_y}$
Energie	\sqrt{ASM}

Table 4.1.: Einige Texturmerkmale, die mit Hilfe einer GLCM gewonnen werden können

μ_x, μ_y sind die Mittelwerte und σ_x, σ_y die Standardabweichungen von $P_d(x)$ beziehungsweise von $P_d(y)$. Dabei ist $P_d(x) = \sum_j^{N-1} P_d(x, j)$ und $P_d(y) = \sum_i^{N-1} P_d(i, y)$.

Hier muss beachtet werden, dass das Eingangsbild in kleinere Unterbereiche, sogenannte Fenster, aufgeteilt werden muss. Dabei darf der Ausschnitt nicht größer sein als zusammenhängende wichtige Bereiche im Bild, aber auch nicht zu klein, um wichtige Charakteristiken außen vor zu lassen. Die Fenstergröße ist unter Umständen ein ausschlaggebender Parameter für den Erfolg des Algorithmus. Ebenfalls ist der Vektor d wichtig. Mehrere Matrizen mit verschiedenen Vektoren zu berechnen ist möglich, erhöht den Rechenaufwand jedoch enorm. Aber besonders die Anzahl der Graustufen der GLCM ist für den Rechenaufwand entscheidend. Je mehr Graustufen, desto größer ist die GLCM und desto aufwändiger sind die Berechnungen. Tatsächlich steigt die Verarbeitungsdauer sogar exponentiell mit der Anzahl der Graustufen.

All das macht es schwierig, wiederholbare und allgemeingültige Ergebnisse mit Hilfe der GLCM zu erhalten. Dennoch wurden in anderen Bereichen, wie zum Beispiel der Klassifizierung von Landschaften in Luftaufnahmen, sehr gute Ergebnisse mit diesem Algorithmus erzielt. Deshalb werden auch hier im Bereich der Faserkategorisierung gute Ergebnisse erwartet.

Da die Berechnung der GLCM so aufwändig ist, gibt es ein großes Forschungsinteresse in der Optimierung des Algorithmus. Nennenswert ist die Gray level Co-occurrence Link List (GLCLL) [CZ01]. Hier werden nur die nicht-leeren Häufigkeiten mit Hilfe einer verketteten

Liste als Datenstruktur abgespeichert. Darauf aufbauend gibt es noch die Gray Level Co-occurrence Hybrid Structure (GLCHS) [CZ02], welche die GLCLL um eine Hash-Tabelle zur Speicherung der Wahrscheinlichkeiten erweitert.

Weiterhin gibt es Ansätze, die Co-occurrence Matrix auf nicht-CPU's berechnen zu lassen. Durch die hohe Parallelisierbarkeit bieten sich hier Grafikprozessoren [GMH⁺12] oder aber auch FPGAs [RR09] an.

4.4.3. Gray Level Difference Method

Sehr ähnlich zur GLCM ist die Gray Level Difference Method (GLDM) [WR75]. Hier beschreibt die Matrix $D_d(i, j)$ die Häufigkeit, dass zwei Pixel mit einem Abstand d eine bestimmte Differenz zueinander haben.

Auf diesen Deskriptor können dann die gleichen Merkmalsextraktionensmethoden wie für die GLCM aus Tabelle 4.1 benutzt werden.

4.4.4. Gray Level Run Length

Erstmals 1975 erwähnt von Galloway [Gal75], ist die Gray Level Run Length (GLRL) ebenfalls eine Methode, um Informationen einer Textur mit Hilfe Statistik höherer Ordnung zu extrahieren.

Eine Menge von zusammenhängenden Pixeln mit gleicher Intensität, die auf einer Geraden liegen, also kollinear sind, nennt man hier Grauwertlauf (engl.: gray level run). Die Anzahl der zusammenhängenden Pixel ist die Lauflänge (engl.: run length) und die Anzahl, wie oft die jeweilige Länge auftritt, ist der Lauflängenwert (engl.: run length value).

Darstellen lässt sich das äquivalent zur GLCM mit einer Gray Level Run Length Matrix (GLRLM) $R_d(i, j)$ und dem Richtungsvektor $d = (dx, dy)$. Ein Beispiel für ein 5x5 Pixel großes Bild mit vier Graustufen ist in Abbildung 4.2 zu sehen.

Ein Eintrag der Matrix an der Stelle (i, j) gibt die Gesamtanzahl von Vorkommen der Länge j mit dem Grauwert i wieder und wird mathematisch als $r_d(i, j)$ dargestellt.

Die Anzahl der Graustufen kann zur Vereinfachung der Berechnung reduziert werden. Weniger Graustufen bedeutet weniger Details. Andererseits bedeutet das ebenfalls eine geringere Anfälligkeit gegenüber Störungen und Bildrauschen.

Mit Hilfe der Matrix können eine Reihe von statistischen Skalaren berechnet werden. Fünf

1	1	4	2	3
2	4	3	3	1
2	3	3	4	3
2	4	2	2	1
2	1	3	3	3

	1	2	3	4	5	
1	3	1	0	0	0	Grauwert
2	5	1	0	0	0	
3	2	2	1	0	0	
4	4	0	0	0	0	
	Lauflänge					

Abbildung 4.2.: Links: 5x5 Bild mit 4 Graustufen; Rechts: GLRLM mit $d = (1, 0)$

Merkmale hat Galloway in seinem Paper genannt [Gal75]. Zwei weitere wurden einige Jahre später von Chu et al. aufgestellt [CSG90].

Short Runs Emphasis Betont kürzere Läufe, indem jeder Lauflängenwert durch das Quadrat der Länge geteilt wird.

$$SRE = \frac{1}{R_G} \sum_{i=0}^G \sum_{j=0}^{R_l} \frac{r_d(i, j)}{j^2} \quad (4.5)$$

Wobei R_l der längste Lauf ist. Die Gesamtanzahl aller Läufe wird mit R_G dargestellt und dient hier als Normalisierung.

Long Runs Emphasis Entspricht genau dem Gegenteil von SRE . Hier wird durch die Multiplikation jedes Laufs mit dem Quadrat der Länge der Schwerpunkt auf längere Läufe gelegt.

$$LRE = \frac{1}{R_G} \sum_{i=0}^G \sum_{j=0}^{R_l} j^2 r_d(i, j) \quad (4.6)$$

Gray Level Non-uniformity Größere Lauflängenwerte wirken sich hier mehr aus als kleinere Werte.

$$GLN = \frac{1}{R_G} \sum_{i=1}^G \left(\sum_{j=1}^{R_l} r_d(i, j) \right)^2 \quad (4.7)$$

Dieses Merkmal hat einen kleinen Wert, wenn die Lauflängen gleichmäßig über alle Grauwerte verteilt sind.

Run Length Non-uniformity Ähnlich wie *GLN*, nur dass hier der niedrigste Wert genau dann erreicht wird, wenn die Anzahl und nicht die Länge der Läufe gleichmäßig über alle Grauwerte verteilt sind.

$$RLN = \frac{1}{R_G} \sum_{i=1}^{R_l} \left(\sum_{j=1}^G r_d(i, j) \right)^2 \quad (4.8)$$

Run Percentage Gibt das Verhältnis zwischen allen Läufen und allen möglichen Läufen, wenn sie genau eine Länge von eins hätten, wieder.

$$RP = \frac{1}{n} \sum_{i=1}^G \sum_{j=1}^{R_l} r_d(i, j) \quad (4.9)$$

Low Gray Level Run Emphasis Eines der beiden von Chu et al. [CSG90] vorgeschlagenen Merkmale.

$$LGRE = \frac{1}{R_G} \sum_{i=0}^G \sum_{j=0}^{R_l} \frac{r_d(i, j)}{i^2} \quad (4.10)$$

High Gray Level Run Emphasis

$$HGRE = \frac{1}{R_G} \sum_{i=0}^G \sum_{j=0}^{R_l} i^2 r_d(i, j) \quad (4.11)$$

Anwendung findet GLRL unter anderem in der Medizin [Nai12] [CBLC04]. Dort allerdings wird diese Methode bis auf wenige Ausnahmen [AR11] nicht zur Klassifikation verwendet. Da es zudem im Bereich der Faseranalyse bislang offenbar wenig verwendet wurde, wird die GLRL hier implementiert und mit zur Auswertung herangezogen.

4.4.5. Census-Transformation

Eine sehr einfache Methode zur Gewinnung eines Texturmerkmals ist die Census-Transformation (CT) wie sie von Zabih et al. [ZW94] beschrieben wird. Im Gegensatz zu vielen anderen Methoden ist der Rechenaufwand hier besonders gering. Nichtsdestotrotz ist sie überraschend effektiv.

Der Grauwert eines zentralen Pixels g_c wird mit seinen P Nachbarn $g_p | p \in [0, P)$ verglichen. g_c fungiert hier als Schwellwert. Alle Nachbarn, die den Schwellwert erreichen, werden binär

als Eins dargestellt, die restlichen als Null. Formal zeigt dies die Gleichung 4.12.

$$C(x,y) = \sum_{p=0}^{P-1} \xi(g_p, g_c) \cdot 2^p \quad (4.12)$$

Wobei

$$\xi(p, c) = \begin{cases} 1 & , p \geq c \\ 0 & , \text{sonst} \end{cases}$$

ist. Abbildung 4.3 veranschaulicht dies an einem typischen 3x3 großen Ausschnitt.

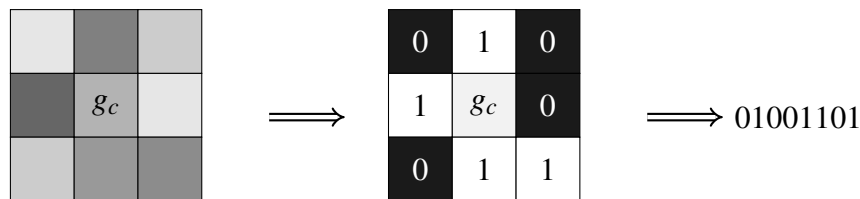


Abbildung 4.3.: Beispiel einer 3x3 Census-Transformation

Die Reihenfolge des Ergebnisbitstrings ist unerheblich, jedoch fest.

Der Bitstring beschreibt hier die Mikrotextur und ist invariant gegenüber monotonen Intensitätsänderungen wie unterschiedliche Belichtungen oder verschiedene Aufnahmewinkel. Auch leichtes Rauschen ändert am Ergebnis wenig.

Mithilfe von zum Beispiel der Hamming-Distanz lassen sich zwei ähnliche Eingaben finden. Denn sind beide gleich, so ergibt sich eine Distanz von Null. Zur Beschreibung von Texturen ist diese direkte Methode jedoch eher ungünstig. Schon leichte Rotation oder Translation ergeben völlig unterschiedliche Ergebnisse.

Um diese Probleme zu umgehen, kann man zum Beispiel ein Histogramm aus den verschiedenen Bitstrings erstellen. Dadurch wird zumindest das Ergebnis unabhängig gegenüber der Translation.

4.4.6. Local Binary Pattern

Eine verallgemeinerte Form der CT ist das Local Binary Pattern (LBP) [PONH94]. Auch diese Methode ist sehr einfach zu berechnen und liefert in verschiedenen Tests ausgezeichnete Ergebnisse [Oul03].

Beim CT wurden immer nur sehr kleine Bereiche einer Textur analysiert, beispielsweise eine 3×3 große Fläche. Größere Operationen würden die Berechnung stark erschweren.

Deshalb benutzt LBP hier keine komplette rechteckige Fläche, sondern sieht sich nur die Nachbarn, die auf einer Kreisbahn um den zu untersuchenden Mittelpunkt liegen, an. Durch lineare Interpolation der Pixelwerte ist hier jeder Radius R und jede Anzahl an Nachbarn P möglich. Dadurch lassen sich wesentlich größere Strukturen abbilden.

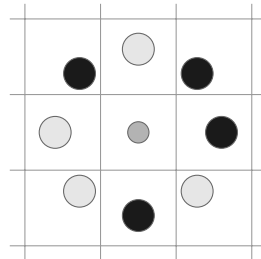


Abbildung 4.4.: Beispielergebnis eines LBP mit Radius 1 und 8 Punkten

Ein Beispiel einer LBP-Operation ist in Abbildung 4.4 zu sehen. Dabei entsprechen schwarze Punkte einer 0 und die anderen einer 1.

Die Berechnung ist analog zu der vom CT und sieht wie folgt aus:

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} \xi(g_p, g_c) 2^p \quad (4.13)$$

Ein möglicher Merkmalsdeskriptor ist die Analyse eines Histogramms. Dazu wird das Eingangsbild in kleinere Fenster aufgeteilt. Zu jedem Pixel (x,y) in dem Ausschnitt wird der $LBP_{P,R}(x,y)$ berechnet. Damit wird dann ein Histogramm, also die Häufigkeiten jeder vorkommenden Zahl, aufgestellt. Die einzelnen Histogramme werden zum Schluss aneinandergelassen und ergeben einen Merkmalsvektor.

Es gibt einige Optimierungsmöglichkeiten. So kann beispielsweise das Histogramm vorher normalisiert werden.

i. Invariant Local Binary Pattern

Eine weitere einfache Verbesserung ist es, das Ergebnis rotationsinvariant zu machen. Denn wird das Eingangsbild gedreht, so ändert sich das Ergebnis vom LBP.

Dreht man das Eingangsbild um α -Grad, so dreht sich das durch den LBP Algorithmus entstandene Muster ebenfalls um die entsprechenden α -Grad. Gut erkennen kann man dies in Abbildung 4.5.

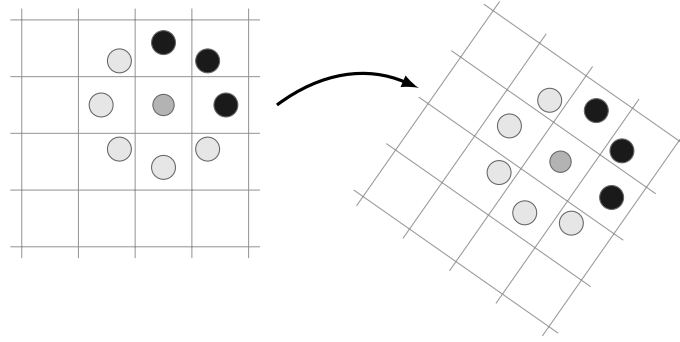


Abbildung 4.5.: Veranschaulichung der Auswirkung einer Bildrotation auf die Nachbarschaft

Wie in [OPM00] vorgeschlagen, kann man hier durch einen zirkulären bitweisen Rechtsshift-Operator, ROR , eine rotationsinvariante LBP-Version $LBP_{P,R}^{ri}(x,y)$ aufstellen.

$$LBP_{P,R}^{ri}(x,y) = \min\{ROR(LBP_{P,R}(x,y),i) : i \in [0, P-1]\} \quad (4.14)$$

Dazu wird die ROR -Operation P -mal auf den originalen Algorithmus angewandt und das minimale Ergebnis wird verwendet. Dies entspricht einem zusätzlichen Normalisierungsschritt. Die gesamte Funktion ist in Gleichung 4.14 zu sehen.

ii. Uniform Local Binary Pattern

Ein Bitstring wird genau dann als uniform bezeichnet, wenn es maximal zwei Übergänge, nämlich von 0 nach 1 beziehungsweise von 1 nach 0, gibt [MOPS00]. Das in Formel 4.15 angegebene Uniformitätsmaß $U(x)$ berechnet die Anzahl dieser Übergänge. Uniform ist demnach das Eingangsmuster, für das $U(x) \leq 2$ gilt.

$$U(LBP_{P,R}) = |\xi(g_{P-1}, g_c) - \xi(g_0, g_c)| + \sum_{p=1}^P |\xi(g_p, g_c) - \xi(g_{p-1}, g_c)| \quad (4.15)$$

Diese Muster können verschieden interpretiert werden. In Abbildung 4.6 sind beispielhaft sechs mögliche Muster dargestellt. Sogenannte Ebenen wie die ersten beiden sind eher uninteressant, da sie keinerlei spezifische Merkmale aufweisen. Andere hingegen beschreiben Kanten, Ecken oder mögliche Linienenden.

Insgesamt gibt es $(P-1) \cdot P + 2$ verschiedene dieser Muster. Zwei davon sind die nicht rotierbaren Ebenen, wie sie in den ersten beiden Beispielen in Abbildung 4.6 dargestellt sind. Die anderen sind die rotierten Varianten von Ecken, Kanten und Linienenden. Bei beispielsweise acht Nachbarn ergeben sich dadurch 58 verschiedene uniforme Muster.

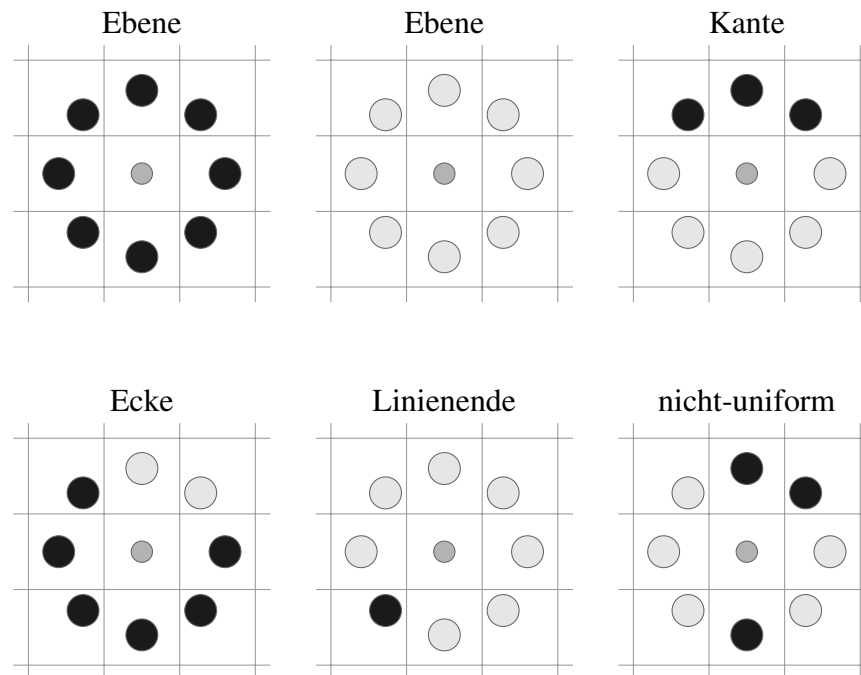


Abbildung 4.6.: Verschiedene lokale Muster und ihre Bedeutung im Bild

Daraus kann man nun folgende Formel ableiten:

$$LBP_{P,R}^{u2}(x,y) = \begin{cases} I(LBP_{P,R}(x,y)) & , U(LBP_{P,R}) \leq 2 \\ (P-1)P+2 & , \text{sonst} \end{cases} \quad (4.16)$$

Das hochgestellte $u2$ steht hier für den Maximalwert von 2 für das Uniformitätsmaß. Alle uniformen Muster werden mit Hilfe der Funktion $I(z) \in [0, (P-1)P+2)$ indexiert. Den restlichen nicht-uniformen wird jeweils der Indexwert $(P-1)P+2$ zugewiesen. Daraus lässt sich nun ein Histogramm mit allen wichtigen uniformen Mustern konstruieren.

Leider lässt sich hier das in Unterpunkt i) beschriebene Verfahren nicht anwenden. Die Rotationsinformationen spielen hier eine wichtige Rolle und können nicht einfach ignoriert werden. Errechnet man jedoch das Powerspektrum mit Hilfe der Diskreten Fourier-Transformation (DFT), so erhält man einen rotationsinvarianten Deskriptor. Denn werden Einträge eines Eingabevektors zyklisch verschoben, so ändert sich nur die Phase der DFT-Koeffizienten.

Ein großer Vorteil der LBP ist seine einfache Berechnung. Die Komplexität liegt bei $O(n)$ und es sind nicht einmal teure Rechenoperationen wie Multiplikation oder Division nötig.

4.4.7. Webers Lokaler Deskriptor

Ähnlich wie LBP ist auch Webers Lokaler Deskriptor (WLD) [JSG⁺08] eine Möglichkeit, lokale Texturmerkmale zu beschreiben. Die zugrundeliegende Idee stammt aus dem 19. Jahrhundert, vom Physiologen Ernst Heinrich Weber. Er erkannte, dass Änderungen von verschiedenen Sinnesorganen erst ab einem bestimmten Änderungsbetrag spürbar werden, der sogenannten differentiellen Wahrnehmbarkeitsschwelle.

So bemerkt man einen Gewichtsunterschied zum Beispiel erst ab ca. 2%. Intuitiv ausgedrückt kann man das gut am Beispiel des Hörsinns verstehen. In einer lauten Umgebung muss man sehr laut reden, um gehört zu werden, wohingegen bei Stille schon ein Flüstern ausreicht.

Daraus leitete Weber folgendes Gesetz ab:

$$k = \frac{\Delta R}{R} \quad (4.17)$$

Wobei ΔR hier der Unterschied zur vorherigen Reizstärke R ist und k immer konstant bleibt. Ist das Ergebnis des Bruchs kleiner als die Konstante k , so wird die Änderung nicht wahrgenommen.

WLD besteht aus zwei Komponenten. Zum einen die differenzielle Anregung (δ) und zum anderen die Orientierung (θ) der Gradienten. Ersteres ist äquivalent zu Webers Gesetz. Unser Reiz R ist die Intensität eines Pixels p_c in unseren Texturbild und ΔR die relative Intensitätsdifferenz von p_c zu seinen Nachbarn. Beides wird für jedes Pixel berechnet. Daraus ergibt sich dann ein Histogramm, das WLD-Histogramm.

Im Bereich der Texturklassifikation und Gesichtserkennung hat diese Methode sich bewährt und führte in anderen Arbeiten zu sehr guten Ergebnissen [JSC⁺10].

4.4.8. Autokorrelation

Eine der wichtigsten Eigenschaften von Texturen ist, dass sie aus sich immer wiederholenden Grundmustern bestehen. Mit Autokorrelation ist es möglich, genau diese Regel- beziehungsweise auch Unregelmäßigkeit dieser Muster auf skalare Werte abzubilden [Pri81].

Die Formel für die Autokorrelation eines Grauwertbildes ist:

$$p(dx, dy) = \frac{\sum_{u,v=0}^N I(u, v)I(u + dx, v + dy)}{\sum_{u,v=0}^N I^2(u, v)} \quad (4.18)$$

Wobei die Verschiebungsskalare dx und dy jeweils im Intervall von $[0, \eta_x)$ bzw. $[0, \eta_y)$ liegen und η frei wählbar ist. Einfacher ausgedrückt wird hier das Skalarprodukt (in dem Fall auch die Energie genannt) des Eingangsbildes mit verschiedenen verschobenen Varianten des Bildes berechnet.

$$p(dx, dy) = \frac{I(u, v) \circ I(u + dx, v + dy)}{I(u, v) \circ I(u, v)} \quad (4.19)$$

Durch Analyse des sich daraus ergebenden Korrelogramms können verschiedene Merkmale berechnet werden, jedoch werden diese meist direkt miteinander verglichen. Aus diesem Grund lassen sich in der Literatur nur wenige Merkmalsextraktionsmöglichkeiten finden. Da die Auswertung ausschließlich über numerische Klassifikatoren stattfinden soll, werden hier eigens entwickelte Extraktionsmethoden vorgestellt.

Regelmäßigkeit

Je mehr Spitzen und Täler ein Korrelogramm hat, desto regulärer ist eine Textur. Daher wird das Merkmal *Regelmäßigkeit* wie folgt definiert:

$$autOC_{reg} = \frac{\text{Anzahl lokaler Höhepunkte}}{0.5(\eta_x + \eta_y)} \quad (4.20)$$

Komplett zufällige Texturen haben nur einen einzigen Höhepunkt an der Stelle $p(0, 0)$.

Schiefe

Ebenfalls ein Maß, um die Regelmäßigkeit einer Textur zu messen, ist die Schiefe der Verteilung. Unregelmäßige bis hin zu komplett zufälligen Texturen besitzen eine rechtsschiefe Verteilung, regelmäßige Texturen sind eher symmetrisch.

Definiert ist die Schiefe folgendermaßen:

$$autOC_{skew} = \frac{\sum_{i=0}^{\eta_x} \sum_{j=0}^{\eta_y} (p(i, j) - \bar{p})^3}{(\eta - 1)s^3} \quad (4.21)$$

Wobei \bar{p} das arithmetische Mittel aller Werte, η die Summe von η_x und η_y und s die Standardabweichung ist.

Kurtosis

Bei groben Texturen flachen die Spitzen nur langsam ab. Im Gegensatz dazu fallen die Werte nach jeder Spitze bei rauen Texturen sehr schnell. Oftmals reicht hier schon die Betrachtung des ersten Höhepunkts.

Aus diesem Grund wird hier die Steilheit mit der Kurtosis (Wölbung) von der ersten Spitze bis in das dazugehörige Tal gemessen.

$$autoc_{kurt} = \frac{\sum_{i=0}^{\eta_x} \sum_{j=0}^{\eta_y} (p(i, j) - \bar{p})^4}{(\eta - 1)s^4} \quad (4.22)$$

Viele weitere Merkmale sind möglich. In dieser Arbeit beschränkt sich die Evaluation jedoch auf genau diese drei.

Lokale Autokorrelation höherer Ordnung Die normale Autokorrelation lässt sich nicht nur als Statistik zweiter Ordnung realisieren. Eine Verallgemeinerung auf höhere Ordnungen, die sogenannte Higher Order Local Autocorrelation (HLAC) [MR68], ist durch folgende Funktion gegeben:

$$x(a_1, a_2, \dots, a_K) = \int I(r)I(r + a_1)I(r + a_2) \dots I(r + a_K)dr \quad (4.23)$$

Wobei $r = (x, y)$ ein Pixel des Eingangsbildes ist und a_1, a_2, \dots, a_K sind K Abstandsvektoren.

4.4.9. Energiefilter nach Laws

Die Grundlage für Laws Energiefilter [Law80] ist hier, dass das Bild mit mehreren Masken gefiltert wird. Aus dem gefiltertem Bild werden dann statistische Kenngrößen wie zum Beispiel Bildmomente gemessen und als Merkmale verwendet. Damit können bestimmte Grundeigenschaften einer Textur aufgezeigt werden. Die von Laws vorgeschlagenen Basisvektoren sind:

$$\begin{aligned} L5(Level) &= [1 \quad 4 \quad 6 \quad 4 \quad 1] \\ E5(Edge) &= [-1 \quad -2 \quad 0 \quad 2 \quad 1] \\ S5(Spot) &= [-1 \quad 0 \quad 2 \quad 0 \quad -1] \\ W5(Wave) &= [-1 \quad 2 \quad 0 \quad -2 \quad 1] \\ R5(Ripple) &= [1 \quad -4 \quad 6 \quad -4 \quad 1] \end{aligned}$$

1. Schritt - Vorverarbeitung

Das Eingangsbild wird in kleinere Fenster aufgeteilt, zum Beispiel mit einer Größe von 15×15 Pixeln. Innerhalb dieses Bereiches wird der Durchschnitt berechnet und von allen Pixeln subtrahiert. Damit wird der Effekt der Illuminierung vermindert und die durchschnittliche Intensität zu jedem Nachbarn ist damit nahe null.

2. Schritt - Faltungsmatrizen anwenden

Aus dem Kreuzprodukt zweier Vektorpaare entsteht eine 2D Maske. Diese wird auch Faltungsmatrix oder Kernel genannt. Es existieren insgesamt 25 solcher Kernel, wie sie in Tabelle 4.2 zu sehen sind. Dabei gilt $X5Y5 = X5^T \times Y5$.

L5L5	E5L5	S5L5	W5L5	R5L5
L5E5	E5E5	S5E5	W5E5	R5E5
L5S5	E5S5	S5S5	W5S5	R5S5
L5W5	E5W5	S5W5	W5W5	R5W5
L5R5	E5R5	S5R5	W5R5	R5R5

Tabelle 4.2.: Mögliche 5×5 Laws Masken

Zum Beispiel ist hier E5L5 ein Maß für die horizontale Kantenstärke und sieht wie folgt aus:

$$E5L5 = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & - & 6 & 4 & 1 \end{bmatrix}$$

L5E5 ist das Äquivalent für die vertikale Kantenstärke.

Für jedes Pixel des vorverarbeiteten Bildes wird nun eine Faltung mit allen 25 Kernen durchgeführt. Daraus ergeben sich 25 neue gefilterte Bilder F_k mit $k \in \{0, 24\}$. In einigen Anwendungsszenarien ist es sinnvoll, nur eine Untermenge aller Kernel zu benutzen. Als Resultat erhält man ein Graustufen-Bild der Größe $N - 6 \times M - 6$. Dieses Bild dient als Basis für unsere Texturextraktion.

3. Schritt - Deskriptoren berechnen

Innerhalb der Fensterausschnitte werden nun unterschiedliche statistische Kenngrößen berechnet. Oft verwendet werden die folgenden drei:

Texturdeskriptor	Formel
Durchschnitt	$\mu_{law} = \frac{\sum_D \text{Nachbarpixel}}{D}$
Abs. Durchschnitt	$\mu_{law}^{abs} = \frac{\sum_D \text{Nachbarpixel} }{D}$
Standardabweichung	$\sigma_{law} = \sqrt{\frac{\sum_D (\text{Nachbarpixel} - \mu_{law})^2}{D}}$

Alle drei Formeln werden auf jedes Pixel aller 25 gefilterten Bilder angewandt. Daraus ergeben sich wiederum $3 \times 15 = 75$ neue Bilder, unsere Deskriptoren.

L5E5/E5L5	L5S5/S5L5	L5R5/R5L5
E5S5/S5E5	E5R5/R5E5	E5E5
S5R5/R5S5	S5S5	R5R5

Tabelle 4.3.: Laws finale Merkmale

4. Schritt - Merkmale berechnen

Nachdem alle Deskriptoren berechnet sind, werden nun bestimmte symmetrische Paare kombiniert. Die ursprünglichen, von Laws vorgeschlagenen Kombinationen, sind in Abbildung 4.3 zu sehen.

Daraus ergeben sich neun Merkmale pro Pixel des ursprünglichen Bildes. Man kann nun wiederum daraus den Durchschnitt von gleichgroßen Ausschnitten berechnen und erhält dann einen Merkmalsvektor, mit dessen Hilfe man den Bereich klassifizieren kann.

4.4.10. Zusammenfassung

In Tabelle 4.4 sind noch einmal alle wichtigen Fakten für die in diesem Abschnitt beschriebenen Algorithmen zusammengefasst.

In der ersten Zeile sind alle vorgestellten Deskriptoren aufgelistet, mit den entsprechenden aus ihnen gewinnbaren Merkmalen in der zweiten Zeile. Rechts davon werden noch einmal die wichtigsten und häufigsten Anwendungsgebiete dargestellt.

Die nächsten beiden Spalten geben an, ob der Deskriptor robust gegenüber Rauschen bzw. gegenüber Parameteranpassungen ist, wie sie in den Anforderungen in Abschnitt 4.1 beschrieben sind.

Deskriptor	Merkmal	Anwendung	Robust		Invariant		
			RA	PA	TL	RO	SK
Tamura	Coarsness, Contrast, Directionality	CBIR	●	✓	✓	●	●
GLCM / GLDM	Trägheit, Kontrast, Verschiedenheit, Homogenität, Entropie, Korrelation, Energie	Landschafts-klassifizierung	×	×	✓	×	●
GLRLM	SRE, LRE, GLN, RLN, RP, LGRE, HGRE	Medizinische Bildauswertung	×	×	✓	×	×
Census	Histogramm	Gesichtserkennung	✓	✓	●	×	×
LBP	LBP Histogramm	Gesichtserkennung	✓	✓	✓	✓	●
Weber		Gesichtserkennung, Textur-klassifikation	✓	✓	✓	?	×
Autokorrelation	Regelmäßigkeit, Schiefe, Kurtosis	Landschafts-klassifizierung, Medizin, Musik-analyse	●	✓	✓	●	✓
Laws Energiefilter	Level, Edge, Spot, Wave, Ripple	Medizin	✓	✓	✓	●	●

Tabelle 4.4.: Zusammenfassung der wichtigsten Informationen aller statistischen Merkmalsextraktionsmethoden.
 (✓) Anforderung erfüllt; (×) Anforderung nicht erfüllt; (●) teilweise erfüllt;
 (?) unbekannt

Über die Invarianz gegenüber verschiedenen Transformationen geben die letzten drei Spalten Auskunft - Translation (TL), Rotation (RO) und Skalierung (SK).

4.5. Signalverarbeitungsanalyse

Methoden aus dem Bereich der Signalverarbeitung versuchen, den Frequenzanteil der räumlichen Komponente eines Bildes zu analysieren. Diese Art der Feature Extraction ist besonders interessant, da es Indizien gibt, dass auch das menschliche Gehirn eine Frequenzanalyse vornimmt [Geo79] [Dau80].

4.5.1. Fourier-Transformation

Hier soll nicht auf die Einzelheiten der Fourier-Transformation (FT) eingegangen werden, Details sind gegebenenfalls entsprechender Literatur zu entnehmen [Tit37] [Bra00]. Wichtig bei der FT ist nur, zu wissen, dass man jedes Signal als Kombinationen von Sinus- bzw. Cosinuskurven darstellen kann. Im Bereich der Bildverarbeitung wird außerdem nur die einfachere Diskrete Fourier-Transformation (DFT) benutzt, da Bildpunkte auch nur diskret abgespeichert werden können.

Ein zweidimensionales Bild $f(x, y)$ kann über die folgende Funktion in den Frequenzraum transformiert werden.

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-\frac{2\pi j}{N}(xu+yv)} \quad (4.24)$$

Wobei $j = \sqrt{-1}$.

Daraus lässt sich das Power-Spektrum wie folgt berechnen [PFTV92]

$$P(u, v) = \text{Real}(F(u, v))^2 + \text{Imaginär}(F(u, v))^2 \quad (4.25)$$

Dieses gibt an, wieviel Energie die jeweiligen Frequenzen in einem Signal besitzen. Konzentriert sich die Energie auf einen Punkt, so ist die Textur sehr regelmäßig. Ist sie hingegen gleichmäßiger verteilt, besonders in den höheren Frequenzen, dann hat die Textur feine Strukturen. Weist die Energieverteilung eine bestimmte Richtung auf, so ist die Textur auch gerichtet.

Die Phaseninformationen enthalten generell wenig Informationen über die Textur. Deswegen wird meist nur das Amplitudenspektrum betrachtet, die Quadratwurzel des Power-Spektrums. Das arithmetische Mittel und die Standardabweichung jeder einzelnen Spalte des Amplitudenspektrums können hier als Merkmale verwendet werden.

Mithilfe der Fast Fourier-Transformation (FFT) kann die DFT mit einer Komplexität von $O(N \log N)$ effizient berechnet werden.

4.5.2. Gaborfilter

Die Gabor-Transformation ist eine spezielle Fourier-Transformation, bei der die Eingabe gefenstert und mit einer Gaußfunktion multipliziert wird. Eine gefensterte FT wird auch als Kurzzeit-Fourier-Transformation (WFT) bezeichnet. Im Gegensatz zur FT wird hier der lokale Ortsbereich analysiert. Der Frequenzbereich ist bei beiden Verfahren lokal.

Die allgemeine Formel für die gefensterte FT ist:

$$F_{\text{Fen}}(\omega, \tau) = \int_{-\infty}^{+\infty} f(t)w(t - \tau)e^{-i\omega t} dt \quad (4.26)$$

Hier ist ω die Frequenz, τ das Zentrum der Lokalisierung und die Funktion $w(\cdot)$ die Fensterfunktion. Ist $w(\cdot)$ eine Gauß-Funktion, so handelt es sich um eine Gabor-Transformation.

Die maximale Auflösung des Zeit- und Frequenzbereichs ist durch die Heisenbergsche Unschärferelation begrenzt [Dau85].

$$\Delta t \Delta u \geq \frac{1}{4\pi} \quad (4.27)$$

Daraus ergibt sich ein reziproker Zusammenhang zwischen der Ortsunschärfe und der Frequenzunschärfe. Um beispielsweise die Auflösung im Frequenzbereich zu verdoppeln, muss die Auflösung im Ortsbereich halbiert werden.

Gaborfilter mit unterschiedlichen Frequenzen und Orientierungen können nun zur Merkmalsextraktion benutzt werden. Einige Methoden dazu sind unter [Tur86] zu finden.

4.5.3. Waveletfilter

Die Wavelet-Transformation kann als eine verallgemeinerte WFT angesehen werden [Dau92]. Ähnlich der Gabor-Transformation und der WFT wird hier der lokale Orts- und auch Frequenzbereich betrachtet. Auch ist die Wavelet-Transformation mit $O(N)$ weniger aufwändig als die FT bzw. die WFT.

Wavelet, was soviel wie *kleine Welle* bedeutet, ist eine Basisfunktion, mit der ein zu analysierendes Signal verglichen werden soll. Dabei geht man von einem sogenannten Mutter-Wavelet

aus, dieses ist selbst eine Funktion mit folgender Anforderung:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (4.28)$$

Ein Signal wird bei der Transformation in gestauchte, gestreckte und verschobene Varianten des Mutter-Wavelets zerlegt. Ähnlich der FT, bei der das Signal in Sinus- bzw. Cosinuswellen zerlegt wird. Ein weiterer großer Vorteil gegenüber der FT ist, dass man auch unstetige Funktionen oder Funktionen mit starken Spitzen problemlos darstellen kann.

Für die diskrete Wavelet-Transformation, wie man sie in der Bildverarbeitung benutzt, stehen einem verschiedene Verfahren zur Auswahl. Darunter fallen Haar-Wavelets [Haa10] und Daubechies Wavelets [Dau92].

Angewandt werden Wavelets bei der Bildkompression [DJL92], Rauschentfernung [CDLL98], Segmentierung [Uns95] und seltener auch bei der Bildklassifikation [DL94].

4.6. Modellbasierende Analyse

Bei den modellbasierenden Analysen wird versucht, ein zugrunde liegendes analytisches Modell zur Beschreibung der Textur zu finden [AS81]. Das Modell ist dabei eine Verallgemeinerung für eine ganze Klasse von Texturen. Mit Hilfe unterschiedlicher Parameter können damit nicht nur verschiedene Texturen der gleichen Klasse zugeordnet, sondern auch neue Texturen erzeugt werden.

4.6.1. Markov Random Fields Modell

Markov-Ketten sind der eindimensionale Spezialfall von Markov Random Fields (MRF). Eine Markov-Kette besteht aus einer Sequenz von Zufallsvariablen x_1, x_2, \dots, x_n . Dabei beschreibt x_t den Zustand des Modells zum Zeitpunkt t . Der jeweilige Zustand ist allein von seinem Vorgängerzustand abhängig, dementsprechend gilt $p(x_t | x_{t-1})$. Bei Markov-Ketten höherer Ordnung ist der Zustand von mehreren Vorgängern abhängig. Für Markov-Ketten N ter-Ordnung gilt $p(x_t | x_{t-1}, \dots, x_{t-N})$.

Im folgenden wird ein Beispiel für ein einfaches Modell, aufgebaut aus Wörtern und Sätzen, aufgestellt.

Markov Ketten sind Markov Random Fields. Mit Markov Ketten lassen sich Modelle aufstellen.

Daraus lässt sich eine Wahrscheinlichkeitstabelle für den Zustand x_t aufstellen wie sie in Tabelle 4.5 zu sehen ist.

Markov		2/3		1/3							
Ketten			1/2					1/2			
sind	1										
Random					1						
Fields						1					
.							1				
Mit	1										
lassen									1		
sich										1	
Modelle										1	
aufstellen						1					
	Markov	Ketten	sind	Random	Fields	.	Mit	lassen	sich	Modelle	aufstellen

Tabelle 4.5.: Beispiel eines Zustandes x_t einer Markov-Kette.

Solche Tabellen können zur Merkmalsextraktion benutzt werden, da die Verteilung benachbarter Einträge selbst ein lokales Merkmal ist. Dazu wird oft ein Wahrscheinlichkeitshistogramm gebildet und dieses untersucht oder direkt mit anderen verglichen.

MRF ist nun die Erweiterung auf den mehrdimensionalen Fall. Die Grundannahme hier ist, dass die Intensität eines Pixels allein nur von seinen Nachbarn abhängig ist. Im Fall der Bildanalyse ist jedes Pixel eine Zufallsvariable. Nachbarn erster Ordnung sind die vier direkt nebeneinanderliegenden, bei zweiter Ordnung kommen die vier weiteren diagonalen Pixel hinzu. Es gilt dementsprechend für ein Pixel x , dass seine Ausprägung, beispielsweise seine Intensität in einem Schwarz-Weiß-Bild, alleinig abhängig von seinen Nachbarn ist. Damit gilt: $p(x|\text{Nachbarn von } x)$.

Somit werden hier nur lokale räumliche Informationen in einem Bild erfasst.

Anstatt einzelne Pixel zu betrachten, kann man zur Beschleunigung des Verfahrens auch ganze Gruppen von Pixeln benutzen.

Angewendet wird MRF unter anderem in der Texturgenerierung [CJ83], Segmentierung [KP06] [The83], Kompression [CCB85] und der Klassifizierung [CC85] [Blu04].

4.6.2. Fraktale Modelle

B. Mandelbrot war einer der ersten, der bemerkte, dass sich die Umwelt aus Fraktalen zusammensetzt [Man82]. Fraktale beschreiben hierbei besondere geometrische Muster. Diese Muster besitzen eine hohe Selbstähnlichkeit und sind zudem invariant gegenüber Skalierung.

Durch die Wiederholungseigenschaften sind diese gute Kandidaten, um auch Texturen zu beschreiben.

Eine beschränkte Menge A in einem euklidischen n -dimensionalen Raum ist genau dann selbstähnlich, wenn A wieder aus der Vereinigung von N unterschiedlichen, nicht überlappenden, um den Faktor r verkleinerten Kopien von sich selbst besteht. Fraktale sind außerdem dadurch gekennzeichnet, dass sie keine ganzzahligen Dimensionen haben, daher auch ihr Name. Die fraktale Dimension D ist wie folgt definiert [Hut79]:

$$D = \frac{\log N}{\log(1/r)} = \frac{\text{Anzahl selbstähnlicher Teile}}{\text{Verkleinerungsfaktor}} \quad (4.29)$$

Die Dimensionszahl D ist hier ein Merkmal, das zur Klassifikation benutzt werden kann. Die Schwierigkeit liegt nun darin, D zu berechnen. In der Natur sind diese nur selten eindeutig zu finden. Eine mögliche Methode ist die von Keller et al. [KCC89].

In [OD92] ist ein Performance Benchmark über, unter anderem, fraktale Modelle zu finden. Dort schneiden Merkmale aus fraktalen Modellen sehr gut ab, nur die Merkmale der Co-occurrence Matrix sind wenige Prozentpunkte präziser.

4.7. Geometrische Analyse

Bei der geometrischen Analyse geht es speziell um die geometrischen Eigenschaften der Texturprimitiven. Auch hier wird die Platzierung der Primitiven untersucht.

4.7.1. Voronoi-Tessellation

Voronoi ist eine Möglichkeit, einen gegebenen Raum zu partitionieren. Zu jedem gegebenen Punkt teilt der Algorithmus den ursprünglichen Raum in eine Zelle auf. Dabei ist jede Position innerhalb einer Zelle am nächsten zu seinem entsprechendem Ausgangspunkt.

Texturmerkmale mit Hilfe der Voronoi-Tessellation zu extrahieren, wurde das erste Mal von Tuceryan and Jain vorgeschlagen [TJ90]. Zur Erinnerung, Texturen bestehen aus Primitiven,

die unterschiedlich räumlich verteilt sind. Diese räumliche Verteilung soll sich durch Voronoi-Diagramme darstellen lassen. Die Primitiven können in diesem Fall sehr unterschiedlich gewonnen werden. Es gibt sehr viele Verfahren, um interessante Punkte in einem Bild zu finden, möglich ist es beispielsweise, Ecken, Kanten oder ganze Regionen durch verschiedene Filter zu extrahieren. Diese dienen dann als Ausgangspunkte für die Tessellation.

Nachdem die Voronoi-Regionen erstellt wurden, können aus diesen wiederum unterschiedliche Merkmale extrahiert werden. Die Form der Region allein gibt schon Auskunft über den Aufbau der Textur. Hier bieten sich also die Flächenmomente als Merkmale an.

Aussagen über die Robustheit lassen sich hier nur schwer treffen. Diese sind abhängig von den verwendeten Algorithmen zur Bestimmung der Primitiven.

Das gleiche gilt für die Invarianz. Translation, Rotation und Skalierung ändern am Ergebnis der eigentlichen Voronoi-Tessellation nichts.

4.7.2. Syntactic Pattern Recognition

Eine weitere geometrische Methode wurde 1982 von Fu [Fu82] vorgestellt. Texturen werden hierbei mit Hilfe einer Sprache beschrieben. Diese Sprache hat eine bestimmte Grammatik. Die Platzierung der Primitiven wird mit dieser Grammatik bestimmt.

Primitive selbst können sehr simpel sein, wie ein einzelnes Pixel, oder aber komplexer und aus einer Gruppe von Pixeln bestehen.

Eine komplette Textur ist dementsprechend ein Satz aus der Sprache. Sätze lassen sich mit Template Matching schnell und einfach miteinander vergleichen. Wesentlich aufwändiger, aber mit der Möglichkeit, auch Fehlertoleranzen zuzulassen, ist das Parsen. Da es sich hierbei um eine kontextfreie Grammatik handelt, reichen schon einfache Top-Down- bzw. Bottom-Up-Parser aus. Kommt es auf die Verarbeitungsgeschwindigkeit an, kann auch ein CYK-Parser benutzt werden [Coc69].

Oft wird diese Methode zur Generierung von Texturen verwendet. Dennoch ist eine Klassifizierung mit ihr ebenso möglich, nur sind hier dann speziell angepasste Klassifikationsmethoden nötig.

5. Umsetzung

In diesem Kapitel geht es um die Umsetzung der zuvor beschriebenen Algorithmen und des Konzepts. Das übergeordnete Ziel war nicht nur die Implementierung zu Test- und Auswertungszwecken, sondern auch, ein System zu konstruieren, das die Wiederverwendbarkeit und Modifikation der einzelnen Komponenten garantiert. Auch die Erweiterbarkeit im Hinblick auf neue Verfahren sowie alternative Datenformate und -darstellungen waren wichtige Kriterien beim Entwurf.

Zum Schluss wird zudem der experimentelle Aufbau beschrieben, mit dem im nächsten Kapitel die Auswertung erfolgen soll.

5.1. Verwendete Technologien

Aufgrund der Wiederverwendbarkeit auch über diese Arbeit hinaus wurde die schon innerhalb der Forschungsgruppe DigiDak+ verwendete Programmiersprache C# benutzt. Diese bietet einen guten Kompromiss zwischen Entwicklungs- und Ausführungsgeschwindigkeit. Das objektorientierte Sprachparadigma erleichterte den modularen Aufbau des Programmes. Plattformunabhängigkeit wurde durch die durchgängige Benutzung der C# Implementierung des Mono-Projekts erreicht. Dennoch sind auch mit der originalen .Net Implementierung bei der Ausführung keine Einschränkungen zu erwarten, so dass dank der weiten Verbreitung von .Net unter Windows auf den meisten Rechnern keine weiteren Abhängigkeiten zu installieren sind.

Folgende externe Bibliotheken wurden verwendet:

- Das Framework *Xwt*, um verschiedene GUI Bibliotheken zu abstrahieren. So wird unter Windows das native WinForms, unter MacOSX Cocoa und unter Linux Gtk zum Rendern der Benutzeroberfläche benutzt.
Ausführlich getestet ist jedoch nur das Gtk-Backend. Da Gtk jedoch auch auf allen drei Plattformen als Rückfalloption lauffähig ist, schränkt dies die Benutzung keinesfalls ein.
Lizenz: MIT
Webseite: <https://github.com/mono/xwt>
- *SharpZipLib* zum Lesen und Schreiben von zip-Dateien. Zip wird als Speicherformat von Projekten und Ergebnissen benutzt.
Lizenz: GPL (mit Ausnahmen für kommerzielle proprietäre Nutzung)
Webseite: <https://icsharpcode.github.io/SharpZipLib/>

5.2. Technischer Aufbau

Das Programm besteht aus separaten, weitgehend autonomen Modulen. So ist beispielsweise die grafische Benutzeroberfläche (GUI) von der restlichen Logik strikt getrennt. Aber auch die Logik ist in verschiedene Bereiche aufgeteilt. Jedes Modul läuft zur Beschleunigung und für den flüssigeren Ablauf in einem eigenen nebenläufigen Thread.

5.2.1. Modularität

Für das Einlesen und Ausgeben von verschiedenen Dateiformaten gibt es jeweils eine abstrakte Hauptklasse `BaseImporter` bzw. `BaseExporter`. Abgeleitete Klassen müssen nur den eigentlichen Einlese- bzw. Ausgabevorgang implementieren. So können schnell und einfach neue Formate unterstützt werden, ohne dass man sich Gedanken über die Interfaceprogrammierung oder Weiterverarbeitung der Rohdaten machen muss.

Implementiert sind hier das `vk4`-Format, das `DD+`-Format und als Ausgabeformat `arff`. `Vk4` ist das native, proprietäre Dateiformat des verwendeten Mikroskops, `DD+` eine Eigenentwicklung des `DigiDak+`-Projektes und `arff` ein sehr weit verbreitetes Dokumentenformat zum Speichern von Merkmalsvektoren.

Ähnlich ist der Aufbau für die einzelnen Algorithmen. Auch hier existiert eine abstrakte Hauptklasse namens `BaseAlgorithm`. Diese enthält alle wichtigen Grundeigenschaften und -funktionen, die jedes Verfahren mindestens haben muss. Jede Implementierung dieser Basisklassen muss eine Reihe von Anforderungen erfüllen. So müssen beispielsweise Name und Beschreibung des jeweiligen Algorithmus ausgefüllt werden sowie welche Eingabedaten dieser benötigt und welches Format das Ergebnis der Berechnung ergibt. Mögliche Formate hierfür sind unter anderem Vektoren, Matrizen, Bilder oder Merkmale. Die wichtigste zu implementierende Methode ist `Run`. Diese enthält den Quellcode des eigentlichen Verfahrens und ist komplett vom Rest des Programms entkoppelt. Alle Eingabeparameter, wie Vorberechnungen oder Benutzeroptionen, werden ihr per Parameter übergeben. Die Rückgabe ist das Ergebnis der Berechnung.

Dank der außerordentlichen guten Reflection-API von C# müssen neue Klassen, die diese Basisklassen erweitern, nirgendwo extra registriert werden. Sobald eine neue Implementierung vorhanden ist, wird diese automatisch erkannt und kann sofort benutzt werden.

5.2.2. Parallelisierung

Im Einzelnen braucht keiner der vorgestellten Algorithmen wirklich lange zur Berechnung. Da aber viele verschiedene Algorithmen mit jeweils unterschiedlichen Parametern auf sehr viele Scans angewandt werden müssen, kann die Auswertung doch sehr viel Zeit in Anspruch nehmen. Deswegen ist es wichtig, auch im Rahmen einer Testimplementierung halbwegs effiziente Verfahren zu benutzen. Heutige CPUs sind sehr schnell, meist besitzen sie sogar mehr als nur einen Rechenkern. Um ihre volle Leistung ausnutzen zu können, müssen dafür aber mehrere Berechnungen parallel ablaufen. Dies verkompliziert einiges sehr stark, aber nur so lässt sich die Verarbeitungsdauer um ein Vielfaches reduzieren.

In diesem Fall ist die Parallelisierung jedoch noch recht leicht zu erreichen. Da jeder Scan, für sich betrachtet, unabhängig von den anderen ist, können alle jeweils gleichzeitig abgearbeitet werden. Genau das kann man sich hier zunutze machen und mit etwas Aufwand sogar noch einen Schritt weiter gehen, denn teilweise sind auch die unterschiedlichen Algorithmen komplett voneinander unabhängig und können parallel berechnet werden.

Um ein hohes Maß an Parallelität, also eine gute Auslastung der CPU-Kerne, zu erreichen, ist es wichtig, die Prozesse zur rechten Zeit zu starten und möglichst für jeden Kern mindestens einen Prozess in Ausführung zu haben. Da einige Algorithmen von vorhergehenden anderen teilweise abhängig sind, müssen diese in der richtigen Reihenfolge abgearbeitet werden. Auch müssen Zwischenergebnisse so lange im Speicher gehalten werden, bis sie nicht mehr von weiteren abhängigen Prozessen benötigt werden. Um dies zu erreichen, wurde ein einfacher Garbage-Collector (GC) entwickelt, der zum richtigen Zeitpunkt die entsprechenden Daten freigibt. Da die Zwischenergebnisse viel Speicherplatz verbrauchen, dürfen diese sich nicht stauen. Nachfolgende Prozesse müssen höher priorisiert werden, damit die Zwischenergebnisse verworfen werden können. Dies wurde durch die Benutzung eines eigens entworfenen Prozess-Scheduler erreicht. Dieser sorgt für eine faire Abarbeitung der noch anstehenden Prozesse, so dass der Speicherverbrauch minimal bleibt und es dennoch zu einer guten Auslastung aller Kerne kommt.

5.3. Benutzeroberfläche

Die Benutzeroberfläche ist in drei Teilbereiche untergliedert. Die Standardansicht, nachdem ein Projekt geöffnet wurde, ist in Abbildung 5.1 zu sehen. Zum einen gibt es eine tabellarische Auflistung aller importierten Scans im oberen linken Bereich der Applikation. Rechts daneben befindet sich ein Vorschauenfenster der momentan selektierten Faser. Die untere Hälfte nimmt die sogenannte Verarbeitungspipeline ein.

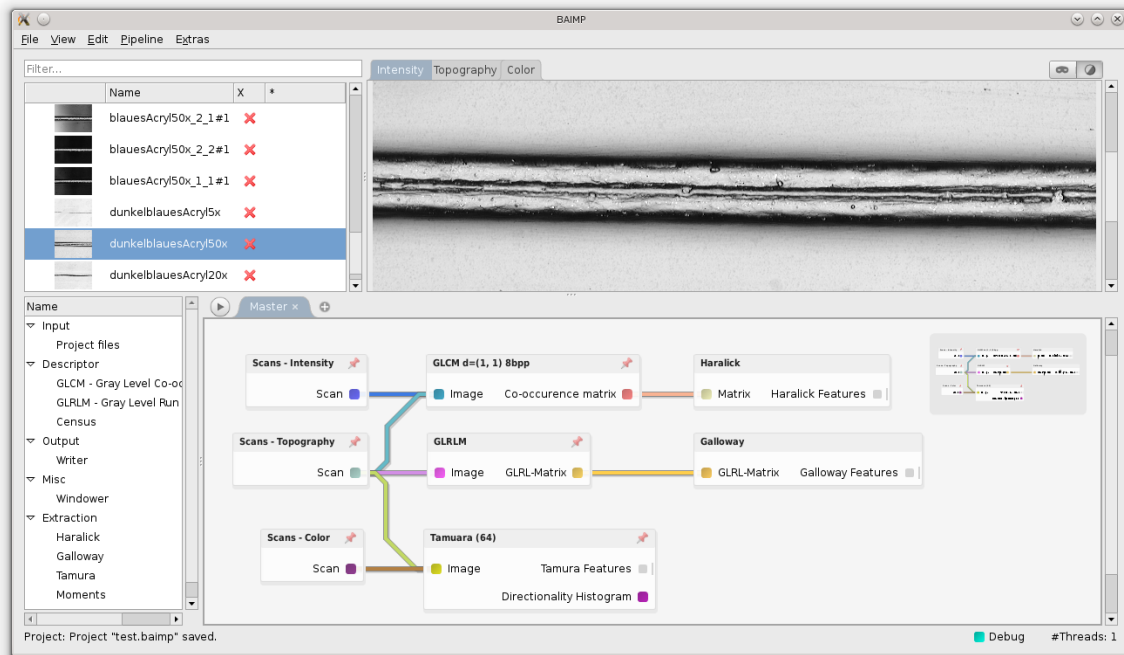


Abbildung 5.1.: Hauptbildschirm eines geöffneten Projekts. Importiert wurden verschiedene Fasern und es ist eine einfache Verarbeitungspipeline zu sehen

i. Faserverwaltung

Durch Doppelklick auf einen Scan in der Liste können dessen Eigenschaften editiert werden. Name, Fasertyp, Vergrößerung der Aufnahme und vieles mehr sind dort änderbar. Neben dem Namen ist zur besseren Übersicht eine kleine Vorschau zu sehen. Auf der rechten Seite befindet sich entweder ein rotes Kreuz oder ein grünes Häkchen. Rot bedeutet, dass noch nicht alle notwendigen Eigenschaften für die entsprechende Faser gesetzt sind und vom Nutzer noch weitere Aktionen erforderlich sind. Dementsprechend sind alle fertigen Fasern mit einem Häkchen versehen und können später ausgewertet werden.

ii. Vorschau

Im Vorschaubereich werden die momentan ausgewählten Fasern zoom- und verschiebbar angezeigt, bei multipler Auswahl werden diese verkleinert im Gitter angeordnet. Am oberen Rand befinden sich einige Aktionsbuttons. Jede Faser kann mehrere Scans enthalten. In der dargestellten Vorschau in Abbildung 5.1 sind es drei: Intensität, Topographie und ein Farbbild. Durch Klick auf den entsprechenden Reiter kann zwischen den Scantypen gewechselt werden,

auch hier ist die gleichzeitige Auswahl mehrerer Einträge für einen besseren visuellen Vergleich möglich.

Mit dem letzten Button oben rechts kann zwischen einer Graustufendarstellung und einer Grün-nach-rot-Darstellung gewechselt werden. Der Button links davon blendet die Fasermaske ein bzw. aus. Eine Fasermaske beschreibt, welcher Teil des Gesamtscans die eigentliche Faser ist. Der nicht maskierte Bereich ist der Hintergrund.

Es ist möglich, den Benutzer diese Maske manuell erstellen zu lassen. Der Maskeneditor ist direkt im Vorschauenfenster eingebaut und wird wie ein digitales Malprogramm bedient. Man zeichnet dort einfach die entsprechenden Stellen der Faser mit einem Pinsel nach. Diese Maskeninformation wird dann bei der Verarbeitung benutzt, um den Hintergrund und die Faser zu segmentieren.

iii. Verarbeitungspipeline

Die Verarbeitungspipeline ist die wichtigste Komponente. Je nach Ansichtstyp kann diese deshalb auch den kompletten Arbeitsbereich der Anwendung einnehmen. Eine laufende Verarbeitung ist in Abbildung 5.2 zu sehen.

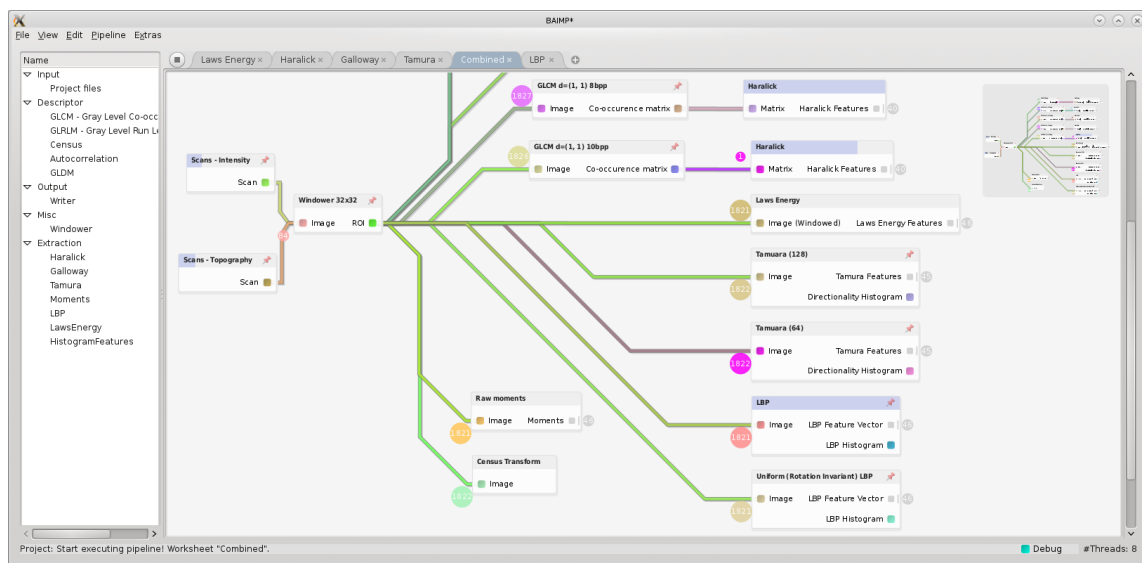


Abbildung 5.2.: Eine sich in der Verarbeitung befindliche Pipeline

Auf der linken Seite befinden sich alle implementierten Algorithmen. Diese werden per Drag-'n'-Drop auf die rechte Seite, in das sogenannte Arbeitsblatt, gezogen. Es kann mehrere Arbeitsblätter pro Projekt geben, alle mit ihren eigenen Namen und unterschiedlichen Algorithmen. Diese werden als Reiter am oberen Rand angezeigt. Jeder Algorithmus wird innerhalb des Arbeitsblattes als Knoten dargestellt. Jeder Knoten hat unterschiedliche Ein- und Ausgänge, abhängig vom dahinterliegenden Algorithmus. Jeder Ein- und Ausgang hat einen bestimmten

Typ, zum Beispiel Bild oder Matrix. Mit der Maus kann man die verschiedenen Knoten miteinander verbinden. Dadurch wird die Ausgabe des ersten Knotens mit der Eingabe des zweiten verbunden. Die Knoten werden automatisch auf Kompatibilität geprüft und nur dann ist eine Verbindung zulässig. Alle grau dargestellten Ausgänge sind Endpunkte und können nicht weiter verbunden werden. Dies sind die extrahierten Merkmale.

Das Stecknadel-Symbol in der Überschrift eines jeden Knoten ist ein Button. Ist er liegend, also von der Seite dargestellt, so werden die Zwischenergebnisse des entsprechenden Knoten schnellstmöglich verworfen, um Arbeitsspeicher freizugeben. Ist die Stecknadel von oben zu sehen, dann werden die Zwischenergebnisse nicht verworfen. Diese können dann mit einem Klick auf den korrespondierenden Ausgang nach oder auch schon während der Verarbeitung betrachtet werden. Das kann zur Fehlerfindung oder aber auch für ein besseres Verständnis sehr hilfreich sein.

Mit einem Doppelklick auf einen Knoten öffnet sich ein Optionsmenü. Dort sind mögliche algorithmusspezifische Einstellungen vorzunehmen.

Hat man seine Verarbeitungspipeline fertiggestellt, kann man nun auf den Play-Button am oberen linken Rand klicken. Daraufhin wird jeder importierte Scan nach und nach geladen und an die entsprechenden Knoten weitergegeben. Diese wiederum geben ihre Zwischenergebnisse an die Nachfolgeknoten weiter, bis alle Aufgaben abgearbeitet sind. In Abbildung 5.2 ist eine solche Pipeline dargestellt.

Die Zahlen vor den Eingängen sind die Größe des momentanen Eingabepuffers, also Zwischenergebnisse vom vorherigen Knoten, die noch nicht weiterverarbeitet werden konnten. Die grauen Zahlen am Ausgang sind die Anzahl der Merkmalsvektoren, die bereits berechnet wurden.

Der blaue Balken hinter dem Namen des Knotens ist der Fortschritt einer einzigen Berechnung. Da die Berechnungen parallel durchgeführt werden, können auch einzelne Knoten gleichzeitig verarbeitet werden. Aus diesem Grund sind dann dort entsprechend mehrere Fortschrittsbalken gleichzeitig untereinander zu sehen.

5.4. Experimenteller Aufbau

In Tabelle 5.3 ist die Anzahl der verschiedenen Fasern des hier verwendeten Testsets zu sehen. Verwendet werden hier sechs verschiedene Fasertypen. Jede Faser wurde mit 20-facher und 50-facher Vergrößerung aufgenommen. Daraus ergeben sich dementsprechend zwölf Klassen, und insgesamt $122 \text{ Fasern} \cdot 2 \text{ Vergrößerungsstufen} \cdot 2 \text{ Scantypen} = 488$ Instanzen.

Das Ergebnis der richtig klassifizierten Instanzen bei zwölf Klassen ist im Mittel bei zufälliger Eingabe $1/12$, also $8,3\bar{3}\%$. Alle Ergebnisse mit signifikant höherer Rate sind als Erfolg anzusehen. Ein typisches Signifikanzniveau ist 5% .

Die Faserscans haben jeweils eine Auflösung von 2048×1536 Bildpunkten. Dabei überdeckt die Faser den gesamten Scanbereich von links nach rechts. Durch die längliche Form einer Faser verdeckt sie jedoch nicht den oberen und unteren Bereich des Bildes. Aus diesem Grund unterscheidet sich die Anzahl der Unterteilungen (Fenster) für jeden Scan stark. Da die Klassifikation auf die Fenster angewandt wird, ergibt sich daraus, dass auch die Anzahl der Klasseninstanzen für jede Faser sehr unterschiedlich ist.

Acryl	Alpakawolle	Angoraziege	Lammwolle	Polyester	Schafwolle
8	40	8	8	40	18

Abbildung 5.3.: Verwendete Fasertypen und deren Anzahl im Testset mit je 20- und 50-facher Vergrößerung

Die Auswertung wurde aufgrund ihres Umfanges auf drei verschiedenen Rechnern durchgeführt. Ihre Konfiguration ist in Tabelle 5.4 einzusehen.

Modell	Prozessor	Arbeitsspeicher	Massenspeicher
Thinkpad X220	Intel Core i7-2640M 2x 2.8-3.5 GHz, 4MB L3	2x 4GB DDR3-1333MHz	SSD Crucial M500 240 GB
Thinkpad X240	Intel Core i7-4600U 2x 2.1-3.3 GHz, 4MB L3	1x 8GB DDR3L-1600Mhz	Samsung SSD 840 EVO 250 GB
-	AMD FX-6300 6x 3.50 GHz, 8MB L3	2x 8GB DDR3-1333Mhz	Samsung SSD 840 PRO 128 GB

Abbildung 5.4.: Konfiguration der verwendeten Computersysteme

Maßgeblich für die Verarbeitungsdauer ist hier fast ausschließlich die CPU-Geschwindigkeit und zum Teil dessen Übertragungsrate zum Hauptspeicher. Da sich die Bearbeitung der Merk-

malsextraktion und Klassifikation problemlos aufteilen lässt, können Multiprozessor und Mehrkernprozessoren fast uneingeschränkt verwendet werden.

Zu bedenken ist, dass selbst bei diesem relativ kleinen Testset viel Arbeitsspeicher benötigt wird. Je mehr parallel abgearbeitet werden soll, desto mehr Arbeitsspeicher ist notwendig. Die zuvor vorgestellte Anwendung benötigt rund 1 GB pro CPU-Kern. Um auch bei größeren Verarbeitungspipelines auf keine Probleme zu stoßen, sind eher 2 GB angebracht.

6. Evaluierung

In diesem Kapitel werden einige der vorgestellten Methoden an realen Testdaten ausgewertet. Hierbei wird nicht nur die Güte mit Hilfe statistischer Größen betrachtet, sondern auch weitere Qualitätskriterien aufgestellt, um externe Faktoren ebenfalls vergleichen zu können.

6.1. Anforderungen

Zu den in Unterkapitel 4.1 genannten Anforderungen einzelner Merkmale kommen folgende für einen Merkmalsvektor hinzu:

1. Genauigkeit
Wird mit Hilfe der Fläche unter der ROC-Kurve und der statistischen Präzision angegeben.
2. Reproduzierbarkeit
Zwei unter sehr ähnlichen Bedingungen aufgenommene Scans sollen nahezu den gleichen Vektor erzeugen.
3. Unterscheidungskraft
Der gesamte Merkmalsraum soll gut abgedeckt sein oder zumindest eine ausreichend große Variation bieten.
4. Größe des Vektors
Die Anzahl der betrachteten Features muss ausreichend groß und dennoch möglichst linear unabhängig sein. Bei zu vielen Merkmalen kann es zur Überanpassung kommen. Auch benötigt ein sehr großer Merkmalsvektor signifikant mehr Rechenzeit und es lassen sich nur schwer Aussagen über die Güte einzelner Merkmale treffen. Ein gutes Mittelmaß ist hier notwendig.

Die Reproduzierbarkeit ergibt sich aus den beiden Punkten Robustheit und Invarianz aus Abschnitt 4.1. Da sie jedoch mit die wichtigste Eigenschaft eines guten Merkmalsvektors ist, wird sie hier nochmal zur Verdeutlichung aufgeführt.

Im Gegenzug dazu steht die Unterscheidungskraft im direkten Widerspruch zur Invarianz. Soll ein Merkmal gegenüber bestimmten Transformationen invariant sein, so verkleinert sich im Allgemeinen der Merkmalsraum. Ähnlich ist es mit der Robustheit. Auch hier werden

Informationen, die als Bildstörung oder Rauschen interpretiert werden können, vernachlässigt.

All diese Anforderungen zu erfüllen, ist dementsprechend ein Wunschtraum. Schon allein die Messung, wie nah man an einem Idealzustand ist, kann mitunter unmöglich sein. Aus diesem Grund dienen diese Anforderungen hier als Richtlinie. Im weiteren Verlauf dieses Auswertungskapitels kann dementsprechend nur grob auf deren Erfüllung eingegangen werden.

6.2. Ergebnisse

In den folgenden Abschnitten werden die Klassifizierungsergebnisse der umgesetzten Algorithmen vorgestellt. Dazu gehören die Tamura Texturmerkmale (4.4.1), Haralick Merkmale der Co-occurrence Matrix (4.4.2), Galloway Merkmale der GLRLM (4.4.4), LBP (4.4.6), Autokorrelation (4.4.8) und Laws Energiefilter (4.4.9). Wenn vorhanden, werden diese mit unterschiedlichen Parametern getestet und verglichen. Während des Extraktionsprozesses sind so rund 6 Millionen Merkmalsvektoren entstanden.

Jede Methode wird mit insgesamt sechs verschiedenen Klassifikatoren ausgewertet. Diese sind mit ihren Einstellungen in Tabelle 6.1 zu sehen. Die dort genannten Abkürzungen werden in den weiteren Auswertungstabellen und -grafiken benutzt.

Weiterhin wurde eine Kreuzvalidierung mit zehn Faltungen eingesetzt, um Trainings- und Testsets zu erzeugen. Diese wurde für jede Methode und jeden Klassifikator jeweils zehn Mal durchgeführt und die Ergebnisse gemittelt.

(1)	trees.J48	-C 0.25 -M 2
(2)	trees.RandomForest	-I 10 -K 0 -S 1 -num-slots 1
(3)	functions.SMO	-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\"
(4)	lazy.IBk	-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"
(5)	bayes.NaiveBayes	
(6)	rules.DecisionTable	-X 1 -S \"BestFirst -D 1 -N 5\"

Tabelle 6.1.: Verwendete Klassifikatoren und deren Parameter in *Weka*

Da die Merkmalsextraktion und auch besonders die darauffolgende Klassifikation teilweise mehrere Tage in Anspruch nehmen kann, konnten hier nicht immer alle Parameter ausgiebig getestet werden. Auch beschränkt sich die Klassifizierung nur auf die Grundalgorithmen. Merkmalsreduktion und Kombinationen werden hier nicht ausgewertet. Damit stellen die hier vorgestellten Ergebnisse durchaus eine untere Grenze des Möglichen dar.

Um Platz zu sparen, werden in den Auswertungstabellen folgende Abkürzungen verwendet:

AUC	Area under ROC	Fläche unter der ROC-Kurve
PC	Percentage Correct	Prozentual richtig klassifizierte Instanzen
I		Intensitätsbild
T		Topographiebild

6.2.1. Tamura

Eine Übersicht der Ergebnisse ist in der Abbildung 6.1 zu sehen. Dort ist für sechs verschiedene Fenstergrößen jeweils ein Graph abgebildet. Die Y-Achse ist dabei der gemittelte Wert der prozentual richtig klassifizierten Fasern aller sechs auszuwertenden Klassifikatoren. Die X-Achse ist hingegen in fünf verschiedene Histogrammgrößen für den Tamura-Deskriptor eingeteilt.

Die Tabellen mit den Gesamtergebnissen sind im Anhang 8 zu finden.

Die Datensets mit Topographieinformationen sind bei einer Vergrößerung von 20 fast immer am besten klassifiziert worden. Bei 50-facher Vergrößerung sind diese jedoch wesentlich schlechter, hier sind offenbar die Intensitätsbilder bedeutend besser geeignet. Warum so starke Unterschiede zwischen 20- und 50-facher Vergrößerung auftreten, ist hier nicht offensichtlich und eine Erklärung dafür kann an dieser Stelle nicht gegeben werden.

Auch sehr gut schneiden besonders große Fenster mit kleinen Histogrammgrößen ab. Mit steigender Histogrammgröße werden auch mehr Klassen falsch zugeordnet.

Verglichen mit anderen hier aufgeführten Ergebnissen liegt diese hier dennoch mit im oberen Bereich. Bei einer Histogrammgröße von 128 Bins und einer Fenstergröße von 48x48 Pixeln liegt der Anteil der korrekt klassifizierten Fasern bei über 70%. Im Schnitt kann man mit einer Rate von gut $\frac{2}{3}$ rechnen.

Verglichen mit Resultaten aus anderen Veröffentlichungen sind diese Ergebnisse recht gut. Bei der Klassifizierung von medizinischen Röntgenaufnahmen wurden zum Beispiel nur 27% bis 35% der Aufnahmen richtig klassifiziert [MHK12].

In einer Veröffentlichung aus dem Jahr 2008 wurde als einziges Merkmal speziell die *Directionality* ausgewertet. Bei der Klassifizierung von 112 Texturbildern wurden ungefähr 25% dieser Texturen richtig klassifiziert [IZL08].

Andererseits konnte in einer Publikation aus dem Jahr 2011 eine Präzision von über 80% erreicht werden [bAFJbA11]. Dazu muss aber gesagt werden, dass dort nur drei Klassen gegeben waren - Aufnahmen von Papier, Jelutong-Holz und Baumwolle.

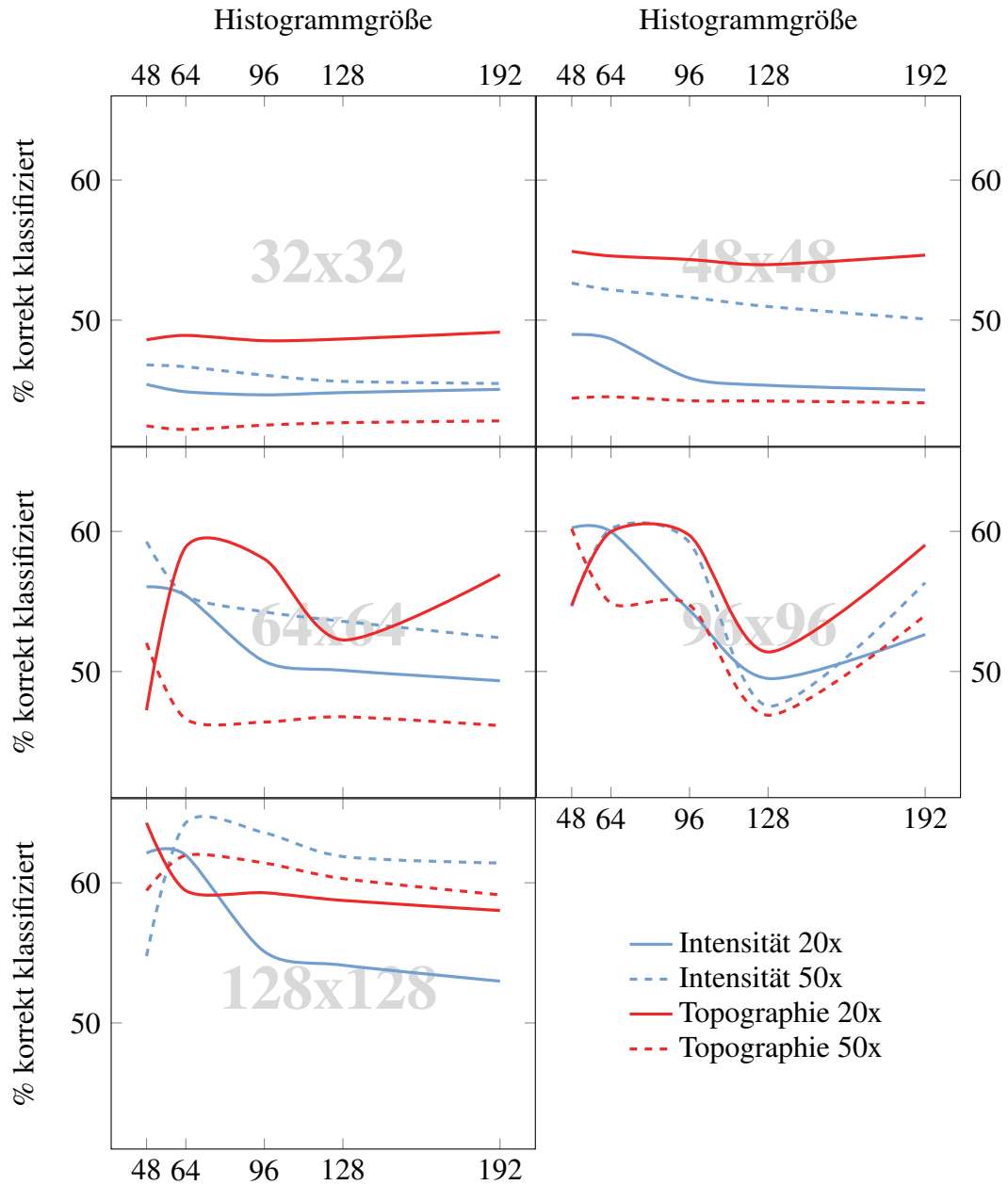


Abbildung 6.1.: Gemittelte Klassifikationsergebnisse aller sechs Klassifikatoren der Tamura-Merkmale. Jeweils ein Diagramm für eine Fenstergröße; von links nach rechts, von oben nach unten: 32x32, 48x48, 64x64, 96x96, 128x128 Pixel

6.2.2. Haralick

Die Gesamtauswertung der Haralick-Merkmale befindet sich wegen ihres großen Umfangs im Anhang 8. Ausgewertet wurden hier die Abhängigkeit des Klassifizierungsergebnisses von unterschiedlichen Fenstergrößen, Abstandsvektoren, GLCM Größen (Bit-Tiefe), Klassifikatoren, Aufnahmevergrößerungsfaktoren und der Unterschied zwischen Intensitäts- und Topographiebildern.

Da es sich hierbei um sehr viele Faktoren und dementsprechend sehr viele Daten handelt, wurden die folgenden Graphen vereinfacht. Wie in den Auswertungstabellen im Anhang zu sehen, erzielte durchschnittlich stets der RandomForest-Klassifikator die besten Ergebnisse. Die meisten anderen lieferten sehr ähnliche Werte, mit Ausnahme von NaiveBayes. Dessen Ergebnisse sind signifikant schlechter. Aus diesem Grund werden die Ergebnisse nicht von allen Klassifikatoren gemittelt, sondern zur Vereinfachung nur die des RandomForest ausgewertet.

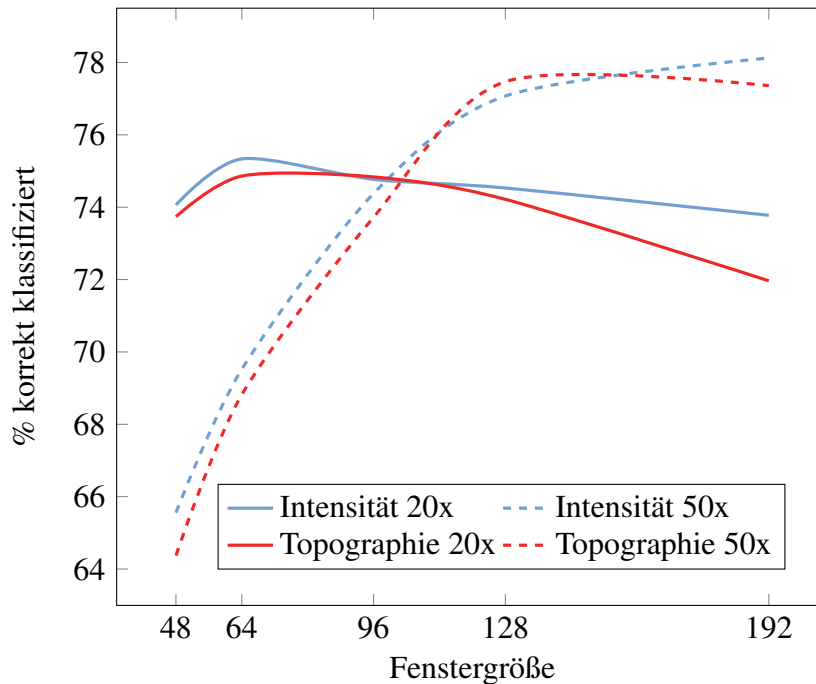


Abbildung 6.2.: Vergleich der Auswirkungen des Klassifizierungsergebnisses der Haralick-Merkmale zwischen den verschiedenen Datensätzen bei unterschiedlichen Fenstergrößen

Unter der Annahme, dass das Resultat sich bei verschiedenen Abstandsvektoren und Bit-Tiefen nicht signifikant ändert, sind in Abbildung 6.2 die Ergebnisse in Abhängigkeit von einzig der Fenstergröße zu sehen. Die Prozentwerte der korrekt klassifizierten Fasern sind dabei der Durchschnitt von allen drei getesteten Abstandsvektoren und beiden Bit-Tiefen je Datensatz.

Hier ist deutlich zu sehen, dass höhere Vergrößerungsfaktoren auch einen größeren Fensterausschnitt bedingen. Da bei geringerer Vergrößerung die entstehenden Faserscans ebenfalls kleiner sind, ist auch die Abbildung der Texturprimitiven kleiner. Dementsprechend benötigt man auch nur kleine Ausschnitte.

Auch wenn der GLCM-Deskriptor prinzipiell globale Eigenschaften erschließen kann, so sind dennoch, anders als beispielsweise bei der GLRLM, die lokal gewonnen Informationen aus einem Teilbereich komplett unabhängig von den umliegenden Pixeln. Damit sollten kleine Fenster, solange sie für das Erfassen der Primitiven groß genug sind, keine Nachteile mit sich bringen.

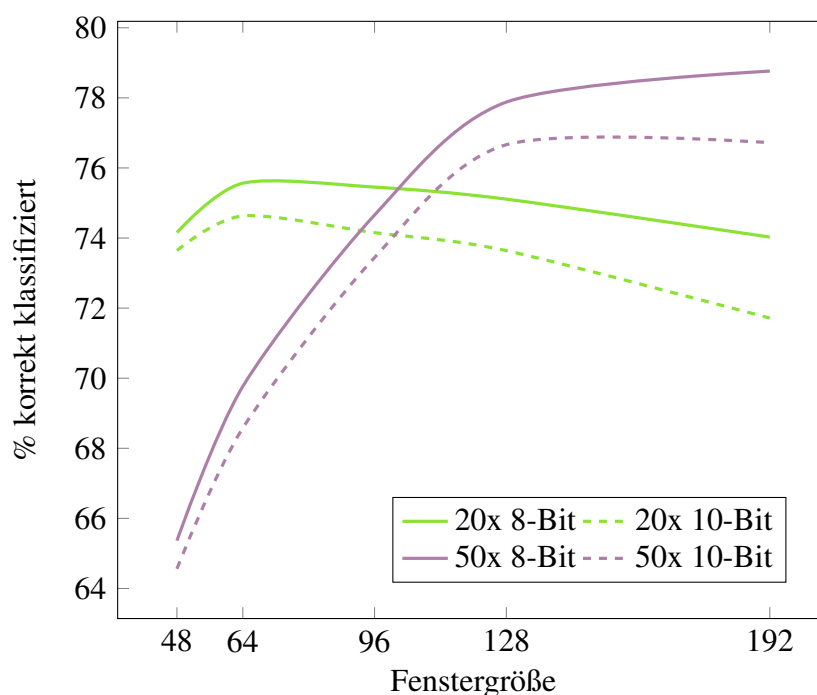


Abbildung 6.3.: Veranschaulichung der Auswirkung auf das Klassifizierungsergebnis der Haralick-Merkmale bei unterschiedlichen Größen der zugrundeliegenden GLCM

Aus den vorliegenden Ergebnissen ist abzulesen, dass bei 20-facher Vergrößerung eine Fenstergröße von 64x64 Pixeln die beste Wahl darstellt. Daraus kann geschlossen werden, dass 64x64 Pixel ausreichend sind, um die Primitiven der Fasern gut abzudecken. Nach der im letzten Abschnitt aufgestellten Annahme sollte bei doppelter Vergrößerung, und damit einer Vervierfachung der Größe der Primitiven, nun auch ein viermal so großes Fenster notwendig sein.

Wie im Graphen von Abbildung 6.2 zu sehen, decken sich hier die experimentellen Ergebnisse mit dieser Vermutung. Hier ist der 50-fache Vergrößerungsfaktor 2,5-mal so groß wie der des

anderen Testsets mit 20-facher Vergrößerung. Die benötigte Fenstergröße sollte dementsprechend $64 \times 64 \cdot 2,5$ sein. Und tatsächlich ist abzulesen, dass die besten Ergebnisse bei 50-facher Vergrößerung zwischen 128×128 und 192×192 Pixeln zu finden sind.

Um die anfängliche Annahme zu bestätigen, dass das Klassifizierungsergebnis nicht von der Bit-Tiefe abhängig ist, werden in Abbildung 6.3 die Ergebnisse von verschiedenen Bit-Tiefen verglichen. Da die Berechnungsdauer in Abhängigkeit von der Bit-Rate exponentiell wächst, wurden hier nur 8- und 10-Bit GLCM berechnet. Stichproben bei der Auswertung haben gezeigt, dass weder mehr noch weniger Grauwerte eine Verbesserung mit sich brachten. In einem Analyse-Paper über GLCM und Haralick von D. Clausi [Cla02] wurde sogar festgestellt, dass schon ab 6-Bit keine Verbesserung zu erwarten ist. In den dort ausgewerteten Texturdatensätzen verschlechterte sich das Ergebnis sogar signifikant, je mehr Grauwerte für die Auswertung verwendet wurden.

Die hier im Graphen dargestellten Ergebnisse sind dabei die gemittelten Werte der verschiedenen Intensitäts- und Topographiebilder sowie der Abstandsvektoren. Wie in Abbildung 6.2 zu sehen, sind die Ergebnisse der beiden Scandaten ohnehin nahezu identisch.

Zwar liegen die Topographie- und Intensitätsbilder in einer sehr hohen Grauwert-Auflösung vor, so dass der Aufbau einer großen Matrix problemlos möglich wäre, aber dennoch muss dies nicht zwangsläufig zu einem besseren Ergebnis führen. Wie im Graphen zu sehen, ist das Resultat nicht signifikant abhängig von der Anzahl der Bit-Auflösung der GLCM. Wenn überhaupt eine Aussage dazu machbar ist, dann sind in diesem Fall mehr Informationen im Deskriptor eher schlecht für das Endergebnis.

Der wahrscheinlichste Grund dafür sind Sensorrauschen oder Schmutz- und Staubpartikel. Weder das Mikroskop ist perfekt, noch können die Sensordaten fehlerfrei digitalisiert werden. Jedoch beim Herunterrechnen auf eine geringere Anzahl an Grauwerten fallen diese leichten Messfehler dann weit weniger ins Gewicht. Auch wenn danach nur noch weniger Graustufen übrig sind, sind diese dann im Verhältnis zueinander näher am Original.

Die zweite Annahme, dass die Klassifizierung (fast) unabhängig vom Abstandsvektor ist, können ebenfalls mit Hilfe der Auswertungsergebnisse bestätigt werden. Zur besseren Visualisierung sind die entsprechenden Werte noch einmal in Abbildung 6.4 zu sehen. Wieder ist hier das arithmetische Mittel über die Ergebnisse der Topographie- und Intensitätsbilder genommen worden.

Da sich im vorherigen Abschnitt herausgestellt hat, dass die 8-Bit Matrizen etwas effektiver sind, wurden auch nur diese hier betrachtet.

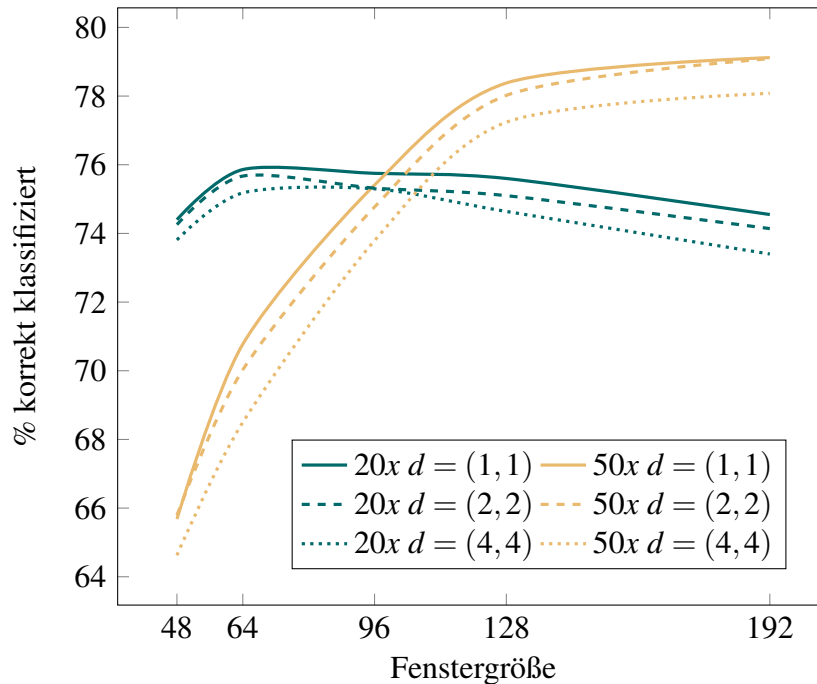


Abbildung 6.4.: Auswirkung auf das Klassifizierungsergebnis der Haralick-Merkmale mit unterschiedlichen Abstandsvektoren

Wie man deutlich erkennen kann, unterscheiden sich die Klassifizierungsraten nicht signifikant bei unterschiedlichen Abstandsvektoren. Die Abstandsvektoren sind nur in Kombination zu gebrauchen, um die Gesamtklassifizierung robuster zu machen.

Die Ergebnisse mit einer Präzision von 70% bis knapp über 80% decken sich mit Resultaten aus anderen Veröffentlichungen. So wurden beispielsweise in einer Veröffentlichung von F. Ershad 80% bis 86% aller Aufnahmen von Steintexturen richtig klassifiziert [Ers12]. Die leicht besseren Resultate lassen sich durch die geringe Anzahl von nur drei Klassen erklären.

Bei der Klassifizierung von Radar-Texturaufnahmen zur Erkennung von Eisbergen und Meereis wurde durch Kombination verschiedener Co-occurrence Matrizen eine Präzision von maximal 91% erreicht [ST96]. Je nach Kombinationsverfahren sank diese jedoch bis auf ca. 51% ab. Auch Auswertungen aus komplett anderen Bereichen erzielten ähnliche Raten. So konnten Gesichter in einer Publikation aus dem Jahr 2009 immerhin mit einer Wahrscheinlichkeit von bis zu 75% richtig erkannt werden [ED09].

6.2.3. Galloway

In der Tabelle 6.2 sind die Klassifizierungsergebnisse der Galloway-Merkmale einer GLRLM mit $d = (1, 0)$ (siehe Abschnitt 4.4.4) zu sehen. Zudem ist die Bit-Tiefe auf 8 Bit heruntergerechnet worden.

			[1]		[2]		[3]		[4]		[5]		[6]	
			PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC
32x32	I	20x	62.68%	0.79	65.87%	0.88	51.11%	0.69	59.77%	0.74	35.29%	0.72	56.41%	0.79
		50x	54.33%	0.72	57.19%	0.82	55.06%	0.72	48.91%	0.68	28.12%	0.69	55.92%	0.81
	T	20x	59.05%	0.77	63.30%	0.86	53.75%	0.70	53.16%	0.70	30.07%	0.74	55.89%	0.79
		50x	44.48%	0.66	48.70%	0.75	40.72%	0.58	39.60%	0.61	17.95%	0.63	47.37%	0.72
48x48	I	20x	68.62%	0.83	73.07%	0.92	52.70%	0.72	68.79%	0.80	35.53%	0.77	61.80%	0.84
		50x	61.81%	0.78	65.23%	0.88	57.26%	0.71	59.20%	0.75	34.11%	0.72	59.02%	0.83
	T	20x	66.87%	0.82	71.47%	0.91	56.37%	0.74	62.99%	0.76	34.98%	0.79	61.02%	0.83
		50x	53.59%	0.73	57.74%	0.82	40.26%	0.59	48.43%	0.67	15.23%	0.65	52.32%	0.76
64x64	I	20x	72.05%	0.85	76.41%	0.93	53.74%	0.73	72.85%	0.83	36.10%	0.81	61.72%	0.85
		50x	66.54%	0.82	70.47%	0.91	58.08%	0.74	65.84%	0.79	41.19%	0.76	62.08%	0.86
	T	20x	68.96%	0.83	73.55%	0.92	55.31%	0.75	64.88%	0.78	39.19%	0.82	60.23%	0.84
		50x	56.95%	0.77	62.18%	0.86	45.15%	0.66	54.39%	0.71	23.94%	0.71	54.00%	0.79
96x96	I	20x	73.01%	0.86	77.53%	0.94	58.53%	0.78	75.33%	0.85	40.84%	0.85	66.91%	0.89
		50x	73.21%	0.86	76.95%	0.94	61.06%	0.77	74.27%	0.84	49.86%	0.82	66.43%	0.88
	T	20x	69.46%	0.84	74.79%	0.93	55.41%	0.76	64.84%	0.78	42.34%	0.84	61.34%	0.87
		50x	64.68%	0.82	69.40%	0.90	48.87%	0.71	63.52%	0.77	32.51%	0.73	58.19%	0.83
128x128	I	20x	73.14%	0.87	77.51%	0.94	59.54%	0.78	75.88%	0.85	42.60%	0.86	66.99%	0.89
		50x	76.27%	0.89	80.58%	0.96	61.50%	0.80	80.29%	0.88	53.55%	0.86	69.03%	0.91
	T	20x	70.75%	0.85	75.19%	0.93	55.66%	0.77	66.40%	0.80	44.87%	0.85	61.84%	0.87
		50x	69.57%	0.84	73.66%	0.92	51.79%	0.75	68.34%	0.80	39.66%	0.80	61.52%	0.85
192x192	I	20x	73.70%	0.86	78.35%	0.94	54.19%	0.71	76.49%	0.86	51.75%	0.82	65.16%	0.88
		50x	76.91%	0.89	80.30%	0.96	61.26%	0.81	79.65%	0.88	52.44%	0.88	66.69%	0.91
	T	20x	69.44%	0.84	74.44%	0.93	55.53%	0.74	65.10%	0.79	50.17%	0.84	61.98%	0.86
		50x	67.68%	0.84	72.46%	0.93	59.28%	0.79	68.90%	0.81	42.72%	0.79	59.34%	0.86
<i>Mittelwert</i>			66.41%	0.82	70.68%	0.90	54.25%	0.73	64.91%	0.78	38.13%	0.78	60.55%	0.84

Tabelle 6.2.: Klassifizierungsergebnisse der Galloway-Merkmale mit sechs verschiedenen Fenstergrößen

Die Rate der korrekt klassifizierten Fasern liegt im Vergleich zu den anderen hier vorgestellten Algorithmen im oberen Mittelfeld. Die Gesamtauswertung kann in Tabelle 6.2 eingesehen werden. Wie dort anhand der letzten Zeile leicht zu sehen ist, erwiesen sich die beiden Klassifikatoren J48 und RandomForest in diesem Testszenario als am besten geeignet. Beide klassifizieren die gegebenen Fasern durchschnittlich zu $\frac{2}{3}$ korrekt.

Weiterhin kann man im Diagramm 6.5 leicht ablesen, dass sich hier die Intensitätsinformationen als am Besten herausgestellt haben. Diese erzielen bei gleichem Vergrößerungsfaktor ein signifikant besseres Gesamtergebnis.

Offensichtlich sind größere Fenster bei diesen Merkmalen von Vorteil. Kleiner als 64x64 Pixel sollten sie weder bei der 20- noch bei der 50-fachen Vergrößerung sein. Ein lokales Maximum wird meist bei einer Fenstergröße von 128x128 Pixel erreicht, dies scheint dementsprechend ein guter Richtwert zu sein.

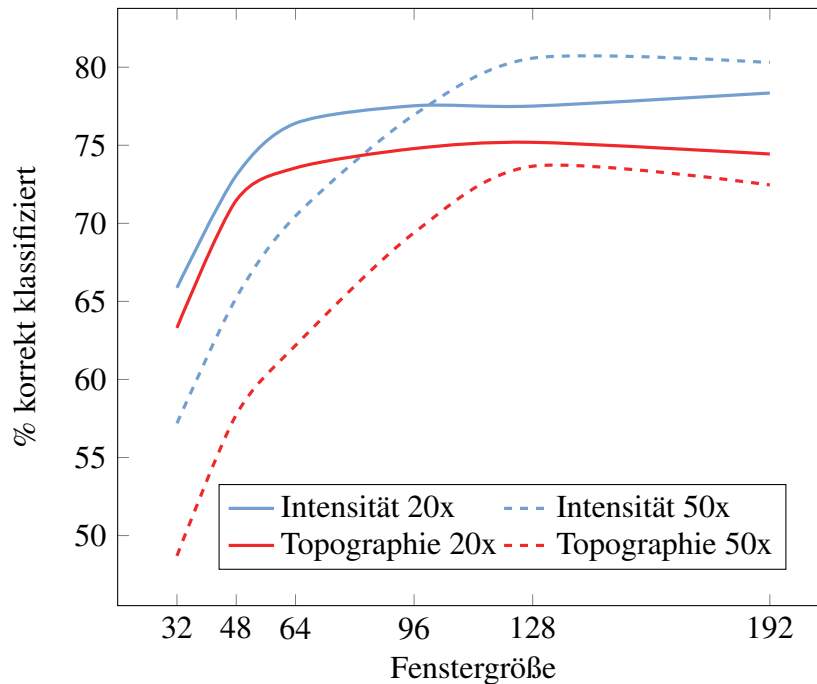


Abbildung 6.5.: Galloway-Ergebnisse der RandomForest-Klassifizierung in Abhängigkeit von der Fenstergröße

Erstaunlich ist hier, dass selbst bei 20-facher Vergrößerung Fenster mit einer Größe von 192x192 Pixeln mitunter die besten Ergebnisse lieferten, besonders bei den Intensitätsdaten. Anzumerken ist, dass es recht wenige dieser großen Ausschnitte gibt, besonders in sehr schmalen Fasern. Offenbar haben diese aber eine weit größere Aussagekraft als die vielen kleinen Bereiche. Vorstellbar ist, dass bei gerichteten Texturen die Lauflänge in entsprechender Richtung besonders lang ist. Durch die Fensterung trennt man diese Lauflängen auf und stützt diese auf eine einheitliche Länge. Anstatt nun viele unterschiedliche, sehr große Lauflängen zu haben, hat man nun in jedem Fenster genau die gleiche Lauflänge - und zwar die der Fenstergröße. Damit ähnelt das Problem dem Leck-Effekt aus dem Bereich der Signalanalyse [Lyo09].

Eine Möglichkeit, dieses Problem zu umgehen, wäre, vorher die Richtung (falls vorhanden) der Textur zu bestimmen, zum Beispiel mit Hilfe des Tamura-Merkmals *Directionality*, und diese als Laufrichtung 90° gedreht für die GLRLM zu verwenden. Damit sollten die meisten Lauflängen wesentlich kürzer sein und das Abschneiden weniger oft vorkommen.

Alternativ kann man auch nicht-quadratische Fenstergrößen verwenden. Damit grenzt man die Problematik ein, dass man aus kleinen Fasern nur wenige Ausschnitte erstellen kann. Mit diesen größeren, langgezogenen, Bereichen kann man nun wieder wesentlich größere Lauflängen erfassen.

Auch die hier gezeigten Ergebnisse stimmen mit denen anderer Veröffentlichungen überein. Die richtige Klassifizierung von Infrarotaufnahmen gelang beispielsweise in einem Paper aus dem Jahr 2013 in über 61% der Fälle [PS13]. Die Präzision in einem Datensatz von Iris-Aufnahmen mit 20 verschiedenen Klassen betrug in einer Publikation von 2013 im Durchschnitt 57% und war damit im gleichen Datensatz um rund 20 Prozentpunkte schlechter als die Haralick-Merkmale einer Co-occurrence Matrix [BSSP13].

Als 2-Klassen-Problem zur Unterscheidung von gut- und bösartigen Tumoren konnte sogar eine Klassifizierungsrate von rund 98% erreicht werden. In diesem Fall war diese etwas besser als die Haralick-Merkmale mit 93% [PR11].

6.2.4. Local Binary Pattern

Die kompletten Auswertungsergebnisse der LBP-Merkmale befinden sich im Anhang in der Tabelle 8.10. Ein Überblick kann sich in der Abbildung 6.6 verschafft werden. Dort sind die Resultate der verschiedenen Datensätze in Abhängigkeit von den Fenstergrößen zu sehen. Dies sind nur die Ergebnisse des IBk-Klassifikators, der, wie man aus der Gesamtauswertung auslesen kann, mit Abstand der am besten geeignete Klassifikator für die LBP-Merkmale ist.

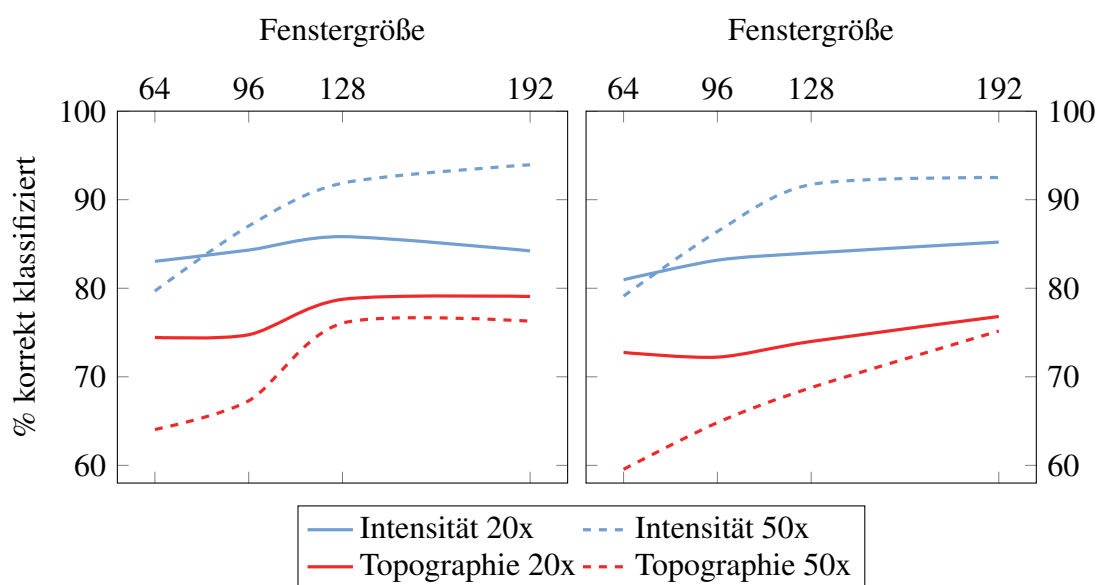


Abbildung 6.6.: Klassifizierungsraten in Abhängigkeit der Fenstergröße des IBk-Klassifikators für die LBP-Merkmale.
Links: Original LBP; Rechts: Rotationsinvariantes LBP

Hier stellt man schnell fest, dass die Ergebnisse der Faserklassifikation im Vergleich zu den anderen vorgestellten Methoden sehr effektiv sind. Je nach Einstellung werden über 90% richtig klassifiziert.

Auch hier sind offenbar die Intensitätsbilder den Topographieinformationen vorzuziehen, besonders bei der 50-fachen Vergrößerung ist der Unterschied in der Klassifizierung signifikant.

Festzustellen ist außerdem, dass sich Ergebnisse der rotationsinvarianten LBP-Methode gegenüber der originalen nicht verbessern. Es ist kein signifikanter Unterschied zu erkennen. Da die hier benutzten Faserscans in den meisten Fällen recht genau ausgerichtet wurden, sind kaum Rotationen vorhanden. Für die Robustheit des Merkmalsvektors sollte diese Variante dennoch in der Praxis nicht vernachlässigt werden.

Auch in diesem Fall bewegen sich die Ergebnisse in dem Rahmen, wie sie auch in anderen Veröffentlichungen zu finden sind. So wurden beispielsweise aus einem Datenset mit 25 verschiedenen Texturklassen rund 53% der Texturen richtig klassifiziert [ME13]. Im gleichen Datenset schnitt die rotationsinvariante LBP-Version mit 83% wesentlich besser ab. Die uniforme LBP-Variante ist mit 54% im Vergleich zur normalen nur unwesentlich besser.

Wesentlich bessere Ergebnisse mit der uniformen Variante konnten in einer anderen Publikation aus dem Jahr 2013 erreicht werden. Mit einem Testset aus insgesamt 14 Klassen wurden dort 85% bis 90% aller Texturen richtig klassifiziert [GZZ10]. Auf einem anderen Datensatz mit 61 Klassen wurden immerhin noch ganze 66% korrekt klassifiziert.

6.2.5. Autokorrelation

In Tabelle 6.3 sind die Ergebnisse der Autokorrelation zu sehen. Wie zu erwarten war, ist die Fehlerquote relativ hoch, da nur drei Merkmale extrahiert worden sind.

Durchschnittlich am besten abgeschnitten haben hier J48 und DecisionTable. Beide haben alle Datensets mit rund 41% bis 57% richtig klassifiziert.

Die besten Ergebnisse erzielte die Autokorrelation mit einer Fenstergröße von 192x192 Pixeln für Intensitätsbilder mit 50-facher Vergrößerung.

Ebenfalls anzumerken ist, dass, je kleiner die Fenstergröße ausfiel, desto besser wurde die 20-fache Vergrößerung im Gegensatz zur 50-fachen klassifiziert. Dies ist nicht weiter überraschend, denn zum einen gibt es einige sehr schmale Fasern, die bei 20-facher Vergrößerung in nur wenige bis gar keine größeren Regionen aufgeteilt werden können. Dementsprechend kann es passieren, dass die Anzahl der Instanzen bei dieser Fenstergröße einfach zu klein ist. Zum anderen werden bei zu großen Fenstern oft nicht die Primitive erfasst, sondern weitaus größere Strukturen. Das kommt bei schmalen Fasern und geringen Vergrößerungsfaktoren besonders zum Tragen.

			[1]		[2]		[3]		[4]		[5]		[6]	
			PC	AUC	PC	AUC	PC	AR	PC	AUC	PC	AUC	PC	AUC
32x32	I	20x	49.39%	0.68	45.43%	0.69	43.96%	0.56	41.67%	0.61	44.59%	0.61	47.77%	0.68
		50x	41.84%	0.61	36.23%	0.61	40.74%	0.53	30.81%	0.55	40.39%	0.56	42.91%	0.63
	T	20x	46.14%	0.63	40.62%	0.64	43.98%	0.54	35.97%	0.57	43.52%	0.60	45.62%	0.66
		50x	41.16%	0.61	35.26%	0.61	40.53%	0.53	30.62%	0.55	39.37%	0.54	42.08%	0.65
48x48	I	20x	51.22%	0.72	47.80%	0.73	44.86%	0.63	44.85%	0.64	45.12%	0.69	49.07%	0.73
		50x	45.75%	0.66	40.82%	0.66	40.26%	0.55	35.68%	0.58	38.76%	0.54	44.26%	0.66
	T	20x	46.46%	0.68	42.69%	0.68	42.14%	0.60	39.38%	0.60	40.61%	0.63	46.05%	0.70
		50x	41.95%	0.65	38.07%	0.65	40.26%	0.50	33.69%	0.57	38.79%	0.55	44.48%	0.69
64x64	I	20x	51.38%	0.75	49.57%	0.77	47.41%	0.68	47.32%	0.67	29.73%	0.68	49.26%	0.77
		50x	48.43%	0.70	44.22%	0.70	38.66%	0.56	39.40%	0.61	36.90%	0.58	44.31%	0.69
	T	20x	44.91%	0.70	42.07%	0.70	41.38%	0.63	39.93%	0.61	19.22%	0.60	46.91%	0.74
		50x	42.86%	0.67	39.44%	0.67	38.66%	0.50	36.10%	0.59	36.44%	0.58	44.94%	0.70
96x96	I	20x	54.99%	0.78	53.36%	0.81	52.68%	0.72	51.78%	0.70	37.19%	0.76	52.40%	0.82
		50x	51.55%	0.75	49.45%	0.75	37.23%	0.55	46.57%	0.66	34.70%	0.60	47.58%	0.74
	T	20x	49.12%	0.73	44.94%	0.74	47.64%	0.69	42.71%	0.64	26.55%	0.67	51.06%	0.77
		50x	49.17%	0.72	44.61%	0.71	36.00%	0.52	41.75%	0.63	35.60%	0.60	49.92%	0.74
128x128	I	20x	55.96%	0.79	53.92%	0.81	53.34%	0.73	52.57%	0.71	39.08%	0.79	52.87%	0.82
		50x	54.30%	0.78	52.53%	0.79	48.85%	0.69	50.23%	0.69	23.64%	0.63	51.64%	0.78
	T	20x	49.67%	0.74	44.77%	0.73	49.15%	0.70	41.94%	0.64	35.97%	0.71	51.61%	0.78
		50x	52.11%	0.76	47.37%	0.75	49.53%	0.70	44.87%	0.65	20.67%	0.63	52.41%	0.80
192x192	I	20x	55.11%	0.78	53.06%	0.81	53.43%	0.72	53.20%	0.71	39.56%	0.80	52.96%	0.82
		50x	57.21%	0.79	54.93%	0.81	54.84%	0.75	52.92%	0.71	31.17%	0.74	55.62%	0.83
	T	20x	49.06%	0.72	43.74%	0.72	49.10%	0.69	42.78%	0.65	37.26%	0.72	51.46%	0.77
		50x	53.65%	0.77	47.49%	0.76	55.37%	0.75	45.43%	0.66	26.89%	0.70	55.29%	0.83
Mittelwert			49.31%	0.72	45.52%	0.72	45.42%	0.63	42.59%	0.63	35.07%	0.65	48.85%	0.74

Tabelle 6.3.: Klassifizierungsergebnisse der Autokorrelation mit sechs verschiedenen Fenstergrößen

6.2.6. Laws Energiefilter

Die Ergebnisse der Laws Energiefilter sind im Anhang in Tabelle 8.11 einzusehen.

Maximal konnten hier ungefähr $\frac{3}{4}$ aller Fasern korrekt klassifiziert werden. Damit liegt das Ergebnis im Mittelfeld aller evaluierten Algorithmen.

Auch in diesem Fall zeigte sich, dass die Intensitätsinformationen besser geeignet zu sein scheinen. Interessant ist, dass die Datensets mit 20-facher Vergrößerung teilweise ein leicht besseres Resultat liefern als die mit den 50-fachen Vergrößerungen.

Ein Vergleich mit Ergebnissen aus der Literatur ist an dieser Stelle leider nicht möglich. Scheinbar wird diese Methode nur in Kombination mit anderen verwendet. Aus diesem Grund ließen sich keine Klassifizierungsraten der Laws Energiefilter finden.

7. Zusammenfassung und Ausblick

In dieser Arbeit wurden unterschiedliche Methoden zur Klassifizierung von Fasern vorgestellt. Dazu wurde eine ausführliche Literaturrecherche betrieben und die wichtigsten Grundalgorithmen beschrieben. Für den interessierten Leser ist jeweils die passende weiterführende Literatur mit angegeben worden.

Um einen Eindruck von der Genauigkeit der beschriebenen Methoden für unseren Anwendungsfall zu erhalten, wurden einige davon umgesetzt. Dafür ist ein eigenständiges grafisches Experimentierprogramm entstanden. Mit dessen Hilfe war es möglich, in kurzer Zeit Merkmalsvektoren von Algorithmen mit vielen verschiedenen Parametern zu generieren.

Entstanden sind während der Auswertungsphase mehrere Millionen Merkmalsvektoren. Bei ihrer Analyse haben sich die LBP-Merkmale als besonders effektiv herausgestellt. Je nach Datensatz, Parameter und Klassifikator konnten bis zu 90% der Faserscans richtig klassifiziert werden. Auch sehr gute Resultate erzielten die Haralick- und Galloway-Merkmale. Beide verwenden unterschiedliche, aber dennoch sehr ähnliche Deskriptoren, die GLCM bzw. die GLRLM. Hier lag die Rate der richtig klassifizierten Fasern bei rund $\frac{3}{4}$. Ähnliche Ergebnisse erzielten auch Laws Energiefilter. Etwas schlechter hingegen fielen die Ergebnisse der Tamura-Texturmerkmale aus. Hier wurden die Klassen zwischen 50% bis 60% richtig den Fasern zugeordnet.

Die Resultate der Autokorrelation sind sogar noch schwächer. Dies war jedoch durch den sehr kleinen Merkmalsvektor zu erwarten.

Um diese Resultate besser mit denen anderer Veröffentlichungen vergleichen zu können, lag ein weiterer Fokus in dieser Arbeit auf der Erstellung eines Konzepts für eine nachvollziehbare Datenerfassung und deren Digitalisierung. Denn im Allgemeinen ist es sehr schwierig, die Ergebnisse anderer Publikationen nachzuvollziehen oder gar nachzustellen. Umso wichtiger ist es, jeden Schritt gut zu dokumentieren.

Zudem kann dieses Konzept beispielsweise in weiterführenden Arbeiten benutzt werden, um die Ergebnisse mit den hier gewonnenen besser vergleichen zu können oder aber um die Ergebnisse dieser Arbeit verifizieren zu können.

7.1. Ausblick

Eine der nächsten offensichtlichen Aufgaben ist die Umsetzung der bereits erwähnten Algorithmen. Hier findet man problemlos auch noch zahlreiche weitere, bei denen sich eine nähere Betrachtung sicherlich lohnen würde.

Besonders Methoden aus dem Bereich der Signalverarbeitung sind vielversprechend. Diese können zudem leicht mit den jetzt schon umgesetzten Algorithmen kombiniert werden. So wurde beispielsweise erfolgreich die Gabor-Transformation als Grundlage für Laws Energiefilter genutzt [RP13].

Auch im Bereich der Klassifikation muss noch einiges getan werden. Bei der Auswertung in dieser Arbeit wurden nur wenige Klassifikatoren benutzt und diese auch nur mit Standardparametern ausgewertet. Möchte man eine höhere Genauigkeit erzielen, so können unter Umständen andere Parameter durchaus behilflich sein. Die Kombinationsmöglichkeiten sind jedoch sehr groß. Das Experimentieren mit verschiedenen Werten kann dementsprechend sehr viel Zeit in Anspruch nehmen. Aber auch wenn man einen guten Klassifikator mit entsprechenden Parametern findet, ist keinesfalls sichergestellt, dass dieser allgemeine Gültigkeit besitzt. Unterschiedliche Datensätze können zu unterschiedlichen Ergebnissen führen.

Aus diesem Grund besteht eine weitere Aufgabe darin, die Anzahl der Testdaten zu vergrößern. Hierbei muss nicht einmal unbedingt die Gesamtanzahl der Scans gesteigert werden, aber zumindest die Varianz pro Testset sollte erhöht werden. Weitere unterschiedliche Fasern aus verschiedenen Quellen sind notwendig. Möglicherweise sind auch weitere Fasertypen hinzuzufügen. Um eine Überanpassung an mögliche spezielle Eigenschaften oder Fehler des verwendeten Mikroskops zu vermeiden, sollten auch Scans mit anderen Mikroskopen erstellt werden. Immerhin können in der Praxis zu untersuchende Fasern auch nicht mit ein und demselben Mikroskop eingescannt werden, wie mit dem, mit dem die Datenbank erstellt wurde. Es ist nicht auszuschließen, dass beispielsweise gerätespezifisches Rauschen die Ergebnisse beeinflussen kann.

Desweiteren sind einige der vorgestellten Merkmalsvektoren auch mit anderen Methoden klassifizierbar. So ist zum Beispiel der Vektor beim LBP sehr groß und eignet sich nur in Maßen für die Auswertung mit üblichen Klassifikatoren. Hier macht es mehr Sinn, den Vektor als Verteilung oder Funktion anzusehen und diese direkt miteinander zu vergleichen. Möglich wären hierzu unter anderem die Verwendung der Maximum-Likelihood-Methode oder ein Chi-Quadrat-Test.

Für eine bessere Vergleichbarkeit ist es natürlich wichtig, sich in der Auswertung jeden Extraktionsalgorithmus einzeln anzugucken. Hat man jedoch das Ziel, die Präzision der Klassifizierung

zu maximieren, so muss man auch Kombinationen der Grundalgorithmen evaluieren. Hier läuft man jedoch schnell Gefahr, dass die Dimension des Merkmalsvektors überhand nimmt. Eine entsprechende Vorselektion ist in diesem Fall wahrscheinlich unabdingbar.

7.2. Fazit

Diese Arbeit konnte zeigen, dass sich durchaus unterschiedliche Fasertypen algorithmisch miteinander vergleichen lassen. Mit einer entsprechenden Datenbank lässt sich einigermaßen sicher automatisch der Fasertyp einer Faserprobe feststellen.

Jedoch gibt es einige schwerwiegende Einschränkungen. So müssen zum Beispiel die Fasern alle möglichst genau gleich digitalisiert werden. Es gibt unzählige, teilweise schwer definierbare Parameter in jedem Schritt der Klassifikation. Die Auswirkungen bei nur kleinen Abweichungen können für das Ergebnis verheerend sein.

Deshalb muss man sich hier ernsthaft die Frage stellen, ob man einer „Maschine“ eine kriminologische Entscheidung überlassen will. Denn auch durch Kombinationen und spezielle Erweiterungen der Grundalgorithmen konnte bis jetzt niemand eine hundertprozentig richtige Klassifizierung mit praxisnahen Daten erreichen.

Literaturverzeichnis

- [AD81] H. Abels and H. Degen. *Handbuch des statistischen Schaubilds: Konstruktion, Interpretation und Manipulation von graphischen Darstellungen*. Verlag Neue Wirtschafts-Briefe, 1981.
- [AKV12] Christian Arndt, Christian Kraetzer, and Claus Vielhauer. First approach for a computer-aided textile fiber type determination based on template matching using a 3D laser scanning microscope. In *Proceedings of the on Multimedia and security, MMSec '12*, page 57–66, New York, NY, USA, 2012. ACM.
- [Alg10] Yasser Alginahi. *Preprocessing Techniques in Character Recognition*, July 2010. Published By InTech.
- [Aly05] Mohamed Aly. *Survey on Multiclass Classification Methods*, 2005.
- [AR11] A.PADMA and R.SUKANESH. Automatic Classification and Segmentation of Brain Tumor in CT Images using Optimal Dominant Gray level Run length Texture Features. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2(10), 2011.
- [AS81] Narendra Ahuja and B. J. Schachter. Image Models. *ACM Comput. Surv.*, 13(4):373–397, December 1981.
- [bAFJbA11] H. bin Adam, M. Fauzi bin Hassan, M. Jais bin Gimin, and A. bin Abdullah. Material surface analysis for robot labeling. In *Pattern Analysis and Intelligent Robotics (ICPAIR), 2011 International Conference on*, volume 1, page 136–138, June 2011.
- [Bal08] Peter J Ball. *Blood and Fibre Forensics*, 2008.
- [Ban08] Isaac N. Bankman, editor. *Handbook of Medical Image Processing and Analysis*. Academic Press, second edition, December 2008.
- [BdKGM06] Charles E. H. Berger, Jan A. de Koeijer, Wendy Glas, and Henk T. Madhuizen. Color Separation in Forensic Image Processing. *Journal of Forensic Sciences*, 51(1):100–102, 2006.
- [Bel61] R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. 'Rand Corporation. Research studies. Princeton University Press, 1961.

- [BJx02] Ye Bin and Peng Jia-xiong. Improvement and Invariance Analysis of Zernike Moments using as a Region-Based Shape Descriptor. In *Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI '02*, page 120–, Washington, DC, USA, 2002. IEEE Computer Society.
- [BL97] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271, December 1997.
- [Blu04] Scott Blunsden. Texture classification using non-parametric markov random fields. Technical report, 2004.
- [Bra00] R.N. Bracewell. *The Fourier Transform and Its Applications*. Electrical engineering series. McGraw-Hill Higher Education, 2000.
- [BS10] D. Bushenko and R. Sadykhov. Method of Extended Objects Identification on Low-Contrast Images, 2010.
- [BSSP13] Kulkarni S. B., Shirodkar S., Hegadi R. S., and Kulkarni U. P. *Iris Recognition using Fusion of Gray Level Co-occurrence Matrix and Gray Level Run Length Matrix*,, page 241–246. Elsevier, 2013.
- [Bur06] Textile Economics Bureau. Fiber Organon, 2006.
- [CBLC04] G. Castellano, L. Bonilha, L.M. Li, and F. Cendes. Texture analysis of medical images, July 2004.
- [CC85] R. Chellappa and S. Chatterjee. Classification of textures using Gaussian Markov random fields. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(4):959–963, Aug 1985.
- [CCB85] R. Chellappa, S. Chatterjee, and R. Bagdazian. Texture synthesis and compression using Gaussian-Markov random field models. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(2):298–303, March 1985.
- [CDLL98] A. Chambolle, R.A. De Vore, Nam-Yong Lee, and B.J. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *Image Processing, IEEE Transactions on*, 7(3):319–335, March 1998.
- [CJ83] George R. Cross and Anil K. Jain. Markov Random Field Texture Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(1):25–39, Jan 1983.

- [Cla02] David A. Clausi. *An analysis of co-occurrence texture statistics as a function of grey level quantization*, volume 28, page 45–62. 2002.
- [Coc69] John Cocke. *Programming Languages and Their Compilers: Preliminary Notes*. Courant Institute of Mathematical Sciences, New York University, 1969.
- [CP00] Chi-Ho Chan and G.K.-H. Pang. Fabric defect detection by Fourier analysis. *Industry Applications, IEEE Transactions on*, 36(5):1267–1276, 2000.
- [CSG90] A. Chu, C.M. Sehgal, and J.F. Greenleaf. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11(6):415–419, 1990.
- [CT00] W. J. Chisum and B. Turvey. Evidence Dynamics: Locard’s Exchange Principle & Crime Reconstruction. *Journal of Behavioral Profiling*, 1(1), 2000.
- [CZ01] D.A. Clausi and Yongping Zhao. Rapid determination of co-occurrence texture features. In *Geoscience and Remote Sensing Symposium, 2001. IGARSS ’01. IEEE 2001 International*, volume 4, page 1880–1882 vol.4, 2001.
- [CZ02] David A. Clausi and Yongping Zhao. Rapid Extraction of Image Texture by Co-occurrence Using a Hybrid Data Structure. *Comput. Geosci.*, 28(6):763–774, July 2002.
- [Dau80] John G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847–856, 1980.
- [Dau85] John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, Jul 1985.
- [Dau92] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1992.
- [DJL92] R.A. DeVore, B. Jawerth, and B.J. Lucier. Image compression through wavelet transform coding. *Information Theory, IEEE Transactions on*, 38(2):719–746, March 1992.
- [DL94] R.A. DeVore and B.J. Lucier. Classifying the smoothness of images: theory and applications to wavelet image processing. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, page 6–10 vol.2, Nov 1994.

- [DV05] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, 14(12):2091–2106, 2005.
- [DVSK11] Jana Dittmann, Claus Vielhauer, Gunter Saake, and Stefan Kiltz. Projektantrag „DigiDak+ Sicherheits- und Forschungskolleg - digitale Formspuren“, 2011.
- [ED09] A. Eleyan and Hasan Demirel. Co-occurrence based statistical approach for face recognition. In *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*, page 611–615, Sept 2009.
- [Ers12] Shervan Fekri Ershad. Texture Classification Approach Based on Combination of Edge & Co-occurrence and Local Binary Pattern. *CoRR*, abs/1203.4855, 2012.
- [Fan05] Q. Fan. *Chemical Testing of Textiles*. Woodhead publishing in textiles. Taylor & Francis, 2005.
- [Flu05] Jan Flusser. *Moment Invariants in Image Analysis*, 2005.
- [Fod02] Imola Fodor. *A Survey of Dimension Reduction Techniques*, 2002.
- [FS05] Jan Flusser and Tomáš Suk. Moment invariants for recognizing symmetric objects. In *Proceedings of the 11th international conference on Computer Analysis of Images and Patterns, CAIP'05*, page 9–16, Berlin, Heidelberg, 2005. Springer-Verlag.
- [Fu82] K.S. Fu. *Syntactic pattern recognition and applications*. Prentice-Hall advanced reference series: Computer science. Prentice-Hall, 1982.
- [Gal75] Mary M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4(2):172–179, June 1975.
- [Geo79] M. A. Georgeson. *Spatial Fourier Analysis and Human Vision*, chapter 2. Lawrence Earlbaum Associates, 2006 edition, 1979.
- [Gib50] James J. Gibson. *The perception of the visual world / James J. Gibson*. Houghton Mifflin Boston, 1950.
- [GMH⁺12] Markus Gipp, Guillermo Marcus, Nathalie Harder, Apichat Suratane, Karl Rohr, Rainer König, and Reinhard Männer. Haralick's Texture Features Computed by GPUs for Biological Applications, 2012.

- [GV11] S. Gruhn and C. Vielhauer. Surface classification and detection of latent fingerprints: Novel approach based on surface texture parameters. In *Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on*, page 678–683, 2011.
- [GZZ10] Zhenhua Guo, D. Zhang, and D. Zhang. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *Image Processing, IEEE Transactions on*, 19(6):1657–1663, June 2010.
- [Haa10] Alfred Haar. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [HAMD12] Mario Hildebrandt, Christian Arndt, Andrey Makrushin, and Jana Dittmann. Computer-aided fiber analysis for crime scene forensics, 2012.
- [Har79] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [HDS73] Rm Haralick, Dinstein, and K. Shanmugan. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, page 610–621, 1973.
- [Hou09] M.M. Houck. *Identification of Textile Fibers*. Woodhead Publishing Series in Textiles. Woodhead Publishing Limited, 2009.
- [HS10] M.M. Houck and J.A. Siegel. *Fundamentals of Forensic Science*. Academic Press, 2010.
- [Hut79] John E Hutchinson. *Fractals and self similarity*. University of Melbourne.[Department of Mathematics], 1979.
- [IEE90] IEEE. IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology. *IEEE Std 610.4-1990*, 1990.
- [IZL08] M.M. Islam, Dengsheng Zhang, and Guojun Lu. A geometric method to compute directionality features for texture images. In *Multimedia and Expo, 2008 IEEE International Conference on*, page 1521–1524, June 2008.
- [Jan02] Barbara Zitova Tomas Suk Jan Flusser. Moments and Moment Invariants in Pattern Recognition, 2002.
- [Jen86] J.R. Jensen. *Introductory digital image processing: A remote sensing perspective*. Jan 1986.

- [JKP94] George H. John, Ron Kohavi, and Karl Pflieger. Irrelevant Features and the Subset Selection Problem. In *MACHINE LEARNING: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL*, page 121–129. Morgan Kaufmann, 1994.
- [Jol02] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.
- [JSC⁺10] Chen J., Shan S., He C., Zhao G., Pietikäinen M., and Chen X & Gao W. WLD: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1705–1720, 2010.
- [JSG⁺08] Chen J., Shan S., Zhao G., Chen X., and Gao W & Pietikäinen M. A robust descriptor based on Weber’s law. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, Alaska*, page 7 p, 2008.
- [KCC89] James M Keller, Susan Chen, and Richard M Crownover. Texture description and segmentation through fractal geometry . *Computer Vision, Graphics, and Image Processing*, 45(2):150–166, 1989.
- [Keya] Keyence. Keyence VK-X105. http://www.keyence.com/products/measure-sys/3d-measure/vk-x100_x200/models/vk-x105/index.jsp
Letzer Zugriff: 05. Juni 2014.
- [Keyb] Keyence. Keyence VK-X110. http://www.keyence.com/products/measure-sys/3d-measure/vk-x100_x200/models/vk-x110/index.jsp
Letzer Zugriff: 05. Juni 2014.
- [KP06] Zoltan Kato and Ting-Chuen Pong. A Markov random field image segmentation model for color textured images . *Image and Vision Computing*, 24(10):1103–1114, 2006.
- [KS01] Diane Kelly and Terry Shepard. A case study in the use of defect classification in inspections. In *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research, CASCON ’01*, page 7–. IBM Press, 2001.
- [KSW85] J. N. Kapur, Prasanna K. Sahoo, and A. K. C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, page 273–285, 1985.
- [Law80] K. I. Laws. *Textured image segmentation*. PhD thesis, University Southern California, 1980.

- [Lew10] M. Lewin. *Handbook of Fiber Chemistry, Third Edition*. International Fiber Science and Technology. Taylor & Francis, 2010.
- [LP96] Simon X. Liao and Mirosław Pawlak. On Image Analysis by Moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996. http://zernike.uwinnipeg.ca/~s_liao/pdf/pami96.pdf(2012.07.03,20:33).
- [Lyo09] Douglas Lyon. The Discrete Fourier Transform, Part 4: Spectral Leakage. *Journal of Object Technology*, 8(7):23–34, 2009.
- [Man82] Benoit B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman, 1 edition, August 1982.
- [ME13] Rakesh Mehta and Karen Egiazarian. Rotated Local Binary Pattern (RLBP) - Rotation Invariant Texture Descriptor. In Maria De Marsico and Ana L. N. Fred, editors, *ICPRAM*, page 497–502. SciTePress, 2013.
- [MHK12] S.M Mohammadi, M.S Helfroush, and K. Kazemi. *Novel shape-texture feature extraction for medical X-ray image classification*, page 659–676. 2012.
- [MOPS00] Topi Mäenpää, Timo Ojala, Matti Pietikäinen, and Maricor Soriano. Robust Texture Classification by Subsets of Local Binary Patterns. In *ICPR*, page 3947–3950, 2000.
- [MR68] J. A. McLaughlin and J. Raviv. Nth-Order Autocorrelations in Pattern Recognition. *Inf. Comput.*, 12(2):121–142, February 1968.
- [Nai12] William Henry Nailon. *Texture Analysis Methods for Medical Image Characterization*, chapter 4. InTech, March 2012.
- [NC10] Ferrante Neri and Andrea Caponio. A differential evolution for optimisation in noisy environment. *Int. J. Bio-Inspired Comput.*, 2010.
- [Nee86] H.L. Needles. *Textile Fibers, Dyes, Finishes, and Processes: A Concise Guide*. Pollution Technology Review. Noyes Publications, 1986.
- [OD92] Philippe P. Ohanian and Richard C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, 1992.
- [OPM00] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In David Vernon, editor, *ECCV (1)*, volume 1842 of *Lecture Notes in Computer Science*, page 404–420. Springer, 2000.

- [Ots79] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, page 62–66, 1979.
- [Oul03] Infotech Oulu. The local binary pattern approach to texture analysis - Extensions AND APPLICATIONS, 2003.
- [PA97] Arie Pikaz and Amir Averbuch. An Efficient Topological Characterization of Gray-Levels Textures, Using a Multiresolution Representation. *CVGIP: Graphical Model and Image Processing*, 59(1):1–17, 1997.
- [Pea01] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2:559–572, 1901.
- [Pea04] K. Pearson. *Mathematical Contributions to the Theory of Evolution*. London Univ. Drapers' Company research memoirs: Biometric series. 1904.
- [PFTV92] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Power Spectra Estimation Using the FFT. In *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, chapter 13, page 542–551. Cambridge University Press, 2nd edition, 1992.
- [PONH94] Matti Pietikainen, Timo Ojala, Jarkko Nisula, and Jouni Heikkinen. Experiments with two industrial problems using texture classification based on feature distributions, 1994.
- [PR11] A. Padma and R. Sukanesh. Automatic Classification and Segmentation of Brain Tumor in CT Images using Optimal Dominant Gray level Run length Texture Features. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 2(10), 2011.
- [Pri81] M. B. Priestley. *Spectral analysis and time series / M.B. Priestley*. Academic Press London ; New York, 1981.
- [PS13] T Bharathi P. and P. Subashini. Texture Feature Extraction of Infrared River Ice Images using Second-Order Spatial Statistics. *World Academy of Science*, 7, February 2013.
- [Qui92] J. Ross Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, October 1992.
- [RG99] J. Robertson and Michael Grieve. Forensic examination of fibres. 1999.

- [RP13] Preeti P. Rege and Anuradha C. Phadke. Classification of Architectural Distortion from Other Abnormalities in Mammograms. *International Journal of Application or Innovation in Engineering & Management*, 2(2):42–48, February 2013.
- [RR09] Mahshid Roumi and Mahshid Roumi. Implementing Texture Feature Extraction Algorithms on FPGA, 2009.
- [RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, December 2000.
- [RYS95] N. Ramesh, J.H. Yoo, and I.K. Sethi. Thresholding Based on Histogram Approximation. page 271–279, October 1995.
- [SF02] Tomas Suk and Jan Flusser. Blur and Affine Moment Invariants, 2002.
- [Sk178] J. Sklansky. Image Segmentation and Feature Extraction. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(4):237–247, 1978.
- [SL99] Yan Solihin and C. G. Leedham. Integral Ratio: A New Class of Global Thresholding Techniques for Handwriting Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, page 761–768, August 1999.
- [SS04] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation, 2004.
- [ST96] Leen-Kiat Soh and C. Tsatsoulis. Texture representation of SAR sea ice imagery using multi-displacement co-occurrence matrices. In *Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.'*, *International*, volume 1, page 112–114 vol.1, May 1996.
- [Sub99] SWGMAT Fiber Subgroup. Forensic Fiber Examination Guidelines, 1999.
- [SWG04] Fiber Subgroup SWGMAT. A Forensic Fiber Examiner Training Program, 2004.
- [TH03] Du-Ming Tsai and Tse-Yun Huang. Automated surface inspection for statistical textures. *Image and Vision Computing*, 21(4):307–323, 2003.
- [The83] Charles W. Therrien. An estimation-theoretic approach to terrain image segmentation. *Computer Vision, Graphics, and Image Processing*, 22(3):313–326, 1983.
- [Tit37] E.C. Titchmarsh. *Introduction to the Theory of Fourier Integrals*. Clarendon Press, 1937.

- [TJ90] M. Tuceryan and A.K. Jain. Texture segmentation using Voronoi polygons. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):211–216, Feb 1990.
- [TJ93] Mihran Tuceryan and Anil K. Jain. Handbook of pattern recognition & computer vision. chapter Texture analysis, page 235–276. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1993.
- [TLL95] Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. Neural network studies, 1. Comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995.
- [TMY78] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural Features Corresponding to Visual Perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, 1978.
- [TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.
- [Tur86] M R Turner. Texture Discrimination by Gabor Functions. *Biol. Cybern.*, 55(2-3):71–82, November 1986.
- [Uns95] M. Unser. Texture classification and segmentation using wavelet frames. *Image Processing, IEEE Transactions on*, 4(11):1549–1560, Nov 1995.
- [VDO85] L. Van Gool, P. Dewaele, and A. Oosterlinck. Survey: Texture Analysis Anno 1983. 29(3):336–357, March 1985.
- [Vie05] Claus Vielhauer. *Biometric User Authentication for IT Security: From Fundamentals to Handwriting (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006 edition, October 2005.
- [Wec80] Harry Wechsler. Texture analysis – a survey. *Signal Processing*, 2(3):271–282, July 1980.
- [WoCC84] J.W. Weaver, American Association Textile of Chemists, and Colorists. *Analytical Methods for a Textile Laboratory*. American Association of Textile Chemists and Colorists, 1984.
- [WR75] J. S. Weszka and A. Rosenfeld. A comparative study of texture measures for terrain classification. *NASA STI/Recon Technical Report N*, 76:13470, March 1975.

- [Yar] L. Yaroslavsky. Local adaptative filters for image restauration and enhancement. In Marie-Odile Berger, Rachid Deriche, Isabelle Herlin, Jérôme Jaffré, and Jean-Michel Morel, editors, *ICAOS '96*, volume 219 of *Lecture Notes in Control and Information Sciences*, page 31–39. Springer Berlin Heidelberg.
- [Zha04] M. Zhao. *Image Thresholding Technique Based on Fuzzy Partition and Entropy Maximization*. School of Electrical and Information Engineering, Graduate School of Engineering, University of Sydney, Kapitel 3, 2004.
- [ZPY01] Yang Xue Zhi, G.K.-H. Pang, and N. H C Yung. Fabric defect detection using adaptive wavelet. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 6, page 3697–3700 vol.6, 2001.
- [ZW94] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the third European conference on Computer Vision (Vol. II)*, ECCV '94, page 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

8. Anhang

Tamura		[1]		[2]		[3]		[4]		[5]		[6]		∅	
Histogrammgr.: 32		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC		
48x48	I	20x	46,94%	0,47	46,38%	0,46	43,97%	0,44	40,14%	0,40	44,70%	0,45	50,40%	0,50	45,42%
		50x	48,20%	0,48	46,21%	0,46	49,23%	0,49	38,35%	0,38	47,46%	0,47	51,39%	0,51	46,81%
	T	20x	52,67%	0,53	50,83%	0,51	44,37%	0,44	44,25%	0,44	46,08%	0,46	53,34%	0,53	48,59%
		50x	46,69%	0,47	42,78%	0,43	40,65%	0,41	36,04%	0,36	39,85%	0,40	48,74%	0,49	42,46%
64x64	I	20x	46,41%	0,46	45,61%	0,46	43,96%	0,44	39,23%	0,39	44,34%	0,44	49,78%	0,50	44,89%
		50x	47,68%	0,48	46,05%	0,46	49,07%	0,49	38,63%	0,39	47,20%	0,47	51,43%	0,51	46,68%
	T	20x	52,43%	0,52	53,16%	0,53	44,26%	0,44	45,51%	0,46	43,98%	0,44	54,14%	0,54	48,91%
		50x	44,01%	0,44	44,73%	0,45	40,62%	0,41	36,27%	0,36	38,82%	0,39	48,75%	0,49	42,20%
64x64	I	20x	46,18%	0,46	45,47%	0,45	43,96%	0,44	39,20%	0,39	43,54%	0,44	49,66%	0,50	44,67%
		50x	46,83%	0,47	45,17%	0,45	49,03%	0,49	37,81%	0,38	46,91%	0,47	50,68%	0,51	46,07%
	T	20x	52,48%	0,52	50,84%	0,51	44,14%	0,44	44,63%	0,45	45,31%	0,45	53,81%	0,54	48,54%
		50x	46,62%	0,47	42,66%	0,43	40,66%	0,41	36,37%	0,36	39,99%	0,40	48,76%	0,49	42,51%
96x96	I	20x	46,62%	0,47	45,87%	0,46	43,96%	0,44	39,21%	0,39	43,12%	0,43	50,21%	0,50	44,83%
		50x	46,38%	0,46	44,59%	0,45	49,02%	0,49	37,33%	0,37	46,08%	0,46	50,39%	0,50	45,63%
	T	20x	52,88%	0,53	50,90%	0,51	44,10%	0,44	45,01%	0,45	45,23%	0,45	53,81%	0,54	48,66%
		50x	46,94%	0,47	42,83%	0,43	40,63%	0,41	37,04%	0,37	39,93%	0,40	48,75%	0,49	42,69%
128x128	I	20x	47,03%	0,47	46,35%	0,46	43,96%	0,44	39,69%	0,40	42,93%	0,43	50,42%	0,50	45,06%
		50x	45,84%	0,46	44,56%	0,45	49,03%	0,49	37,34%	0,37	45,70%	0,46	50,41%	0,50	45,48%
	T	20x	53,70%	0,54	51,93%	0,52	44,05%	0,44	45,77%	0,46	45,11%	0,45	54,30%	0,54	49,14%
		50x	46,92%	0,47	43,15%	0,43	40,64%	0,41	37,44%	0,37	39,91%	0,40	48,80%	0,49	42,81%
∅		48,17%	0,48	46,50%	0,47	44,47%	0,45	39,76%	0,40	43,81%	0,44	50,90%	0,51	45,60%	

Tabelle 8.1.: Klassifizierungsergebnisse der Tamura Texturmerkmale mit einer Histogrammgröße von 32 Bins.

Tamura		[1]		[2]		[3]		[4]		[5]		[6]		∅	
Histogrammgr.: 48		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC		
48x48	I	20x	52,44%	0,74	52,42%	0,78	45,17%	0,64	46,13%	0,65	43,52%	0,67	54,12%	0,78	48,97%
		50x	54,52%	0,73	53,44%	0,78	54,40%	0,70	46,00%	0,66	51,58%	0,75	55,94%	0,80	52,65%
	T	20x	60,27%	0,81	60,01%	0,84	46,70%	0,68	53,82%	0,70	48,98%	0,76	59,70%	0,85	54,91%
		50x	48,38%	0,70	47,65%	0,73	40,26%	0,52	41,17%	0,62	39,99%	0,62	49,11%	0,75	44,43%
64x64	I	20x	52,50%	0,73	51,90%	0,77	44,70%	0,63	45,90%	0,65	43,08%	0,67	53,85%	0,78	48,66%
		50x	53,77%	0,73	52,56%	0,77	54,10%	0,70	45,52%	0,65	51,31%	0,74	55,72%	0,80	52,16%
	T	20x	60,16%	0,81	59,91%	0,83	45,69%	0,67	54,16%	0,70	47,90%	0,76	59,68%	0,85	54,58%
		50x	48,40%	0,70	47,54%	0,73	40,26%	0,51	42,43%	0,62	39,52%	0,62	49,02%	0,75	44,53%
64x64	I	20x	51,93%	0,73	51,78%	0,77	44,87%	0,63	45,54%	0,64	27,83%	0,68	53,26%	0,78	45,87%
		50x	52,92%	0,71	51,57%	0,76	53,49%	0,69	44,15%	0,64	52,68%	0,73	54,97%	0,79	51,63%
	T	20x	60,59%	0,81	60,00%	0,83	44,86%	0,66	54,42%	0,70	46,36%	0,73	59,74%	0,85	54,33%
		50x	48,31%	0,70	46,87%	0,72	40,26%	0,51	41,43%	0,62	39,53%	0,61	49,12%	0,76	44,25%
96x96	I	20x	51,57%	0,73	51,80%	0,77	45,08%	0,63	45,25%	0,64	24,89%	0,68	53,53%	0,78	45,35%
		50x	52,11%	0,71	50,35%	0,75	53,42%	0,69	43,46%	0,64	51,84%	0,72	54,64%	0,78	50,97%
	T	20x	60,40%	0,81	60,16%	0,84	44,85%	0,66	54,32%	0,70	44,32%	0,73	59,71%	0,85	53,96%
		50x	48,08%	0,70	46,81%	0,72	40,26%	0,51	41,86%	0,62	39,41%	0,61	48,94%	0,75	44,23%
128x128	I	20x	51,79%	0,73	52,50%	0,77	45,08%	0,63	45,07%	0,64	22,30%	0,68	53,38%	0,77	45,02%
		50x	51,64%	0,70	50,31%	0,75	53,40%	0,69	43,32%	0,64	47,21%	0,71	54,58%	0,78	50,08%
	T	20x	61,32%	0,81	60,63%	0,84	44,80%	0,66	55,01%	0,71	46,15%	0,73	59,95%	0,84	54,64%
		50x	48,06%	0,70	46,86%	0,72	40,26%	0,51	41,55%	0,62	38,85%	0,61	49,02%	0,75	44,10%
∅			53,46%	0,74	52,75%	0,77	46,10%	0,63	46,53%	0,65	42,36%	0,69	54,40%	0,79	49,27%

Tabelle 8.2.: Klassifizierungsergebnisse der Tamura Texturmerkmale mit einer Histogrammgröße von 48 Bins.

Tamura		[1]		[2]		[3]		[4]		[5]		[6]		∅	
Histogrammgr.: 64		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC		
48x48	I	20x	57,42%	0,78	57,35%	0,83	49,73%	0,69	52,55%	0,70	44,31%	0,74	55,59%	0,83	52,83%
		50x	59,16%	0,77	58,68%	0,82	55,75%	0,72	52,24%	0,70	50,38%	0,79	60,02%	0,84	56,04%
	T	20x	64,78%	0,85	65,73%	0,89	50,95%	0,74	61,45%	0,76	49,92%	0,81	62,70%	0,89	59,26%
		50x	53,09%	0,75	53,18%	0,79	37,21%	0,59	47,05%	0,66	41,85%	0,68	50,98%	0,78	47,23%
64x64	I	20x	57,29%	0,77	56,92%	0,83	49,13%	0,69	51,78%	0,69	41,66%	0,73	55,47%	0,83	52,04%
		50x	58,71%	0,77	58,38%	0,82	55,10%	0,71	52,07%	0,70	48,59%	0,78	59,65%	0,83	55,42%
	T	20x	64,77%	0,85	65,90%	0,89	50,97%	0,74	61,84%	0,76	47,12%	0,80	62,73%	0,89	58,89%
		50x	52,45%	0,75	52,35%	0,78	37,18%	0,58	46,86%	0,66	40,05%	0,67	50,71%	0,78	46,60%
64x64	I	20x	57,03%	0,78	56,91%	0,83	48,67%	0,69	51,09%	0,69	35,38%	0,74	55,25%	0,83	50,72%
		50x	57,54%	0,76	56,95%	0,81	54,58%	0,70	50,54%	0,69	47,02%	0,76	58,95%	0,83	54,26%
	T	20x	64,57%	0,85	65,67%	0,89	50,10%	0,74	61,03%	0,75	44,29%	0,80	62,49%	0,89	58,03%
		50x	52,34%	0,75	52,63%	0,78	37,21%	0,58	47,48%	0,66	37,63%	0,67	51,06%	0,78	46,39%
96x96	I	20x	56,92%	0,77	57,00%	0,83	48,80%	0,69	51,41%	0,69	31,45%	0,74	54,91%	0,83	50,08%
		50x	57,29%	0,75	56,90%	0,80	54,49%	0,70	49,63%	0,68	45,47%	0,75	57,72%	0,81	53,58%
	T	20x	58,37%	0,80	58,92%	0,84	47,20%	0,68	54,00%	0,71	37,72%	0,73	57,26%	0,85	52,25%
		50x	52,69%	0,74	52,75%	0,78	38,65%	0,56	46,66%	0,66	36,87%	0,64	53,02%	0,78	46,77%
128x128	I	20x	56,71%	0,77	56,46%	0,83	49,01%	0,69	50,67%	0,69	26,73%	0,74	56,46%	0,83	49,34%
		50x	55,59%	0,74	55,03%	0,79	54,48%	0,70	48,20%	0,67	43,95%	0,75	57,21%	0,81	52,41%
	T	20x	65,28%	0,85	65,59%	0,89	49,98%	0,74	60,80%	0,75	37,23%	0,79	62,57%	0,89	56,91%
		50x	51,88%	0,75	51,65%	0,77	37,15%	0,58	47,24%	0,66	37,91%	0,66	50,99%	0,78	46,14%
∅			57,69%	0,78	57,75%	0,82	47,82%	0,68	52,23%	0,70	41,28%	0,74	56,79%	0,83	52,26%

Tabelle 8.3.: Klassifizierungsergebnisse der Tamura Texturmerkmale mit einer Histogrammgröße von 64 Bins

Tamura		[1]		[2]		[3]		[4]		[5]		[6]		∅	
Histogrammgr.: 96		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC		
48x48	I	20x	57,44%	0,79	58,24%	0,84	53,61%	0,74	53,03%	0,71	48,61%	0,79	56,98%	0,85	54,65%
		50x	63,69%	0,83	64,87%	0,87	58,40%	0,78	60,28%	0,75	54,57%	0,81	62,42%	0,87	60,71%
	T	20x	63,59%	0,86	65,07%	0,90	54,96%	0,78	61,96%	0,77	56,10%	0,85	59,75%	0,89	60,24%
		50x	61,31%	0,81	62,78%	0,85	44,95%	0,67	57,33%	0,73	44,13%	0,74	57,41%	0,84	54,65%
64x64	I	20x	57,24%	0,79	58,45%	0,84	53,15%	0,73	53,80%	0,72	49,49%	0,80	56,16%	0,85	54,72%
		50x	63,79%	0,82	64,40%	0,87	58,04%	0,76	58,93%	0,74	53,54%	0,81	62,44%	0,87	60,19%
	T	20x	63,03%	0,86	64,60%	0,90	54,66%	0,78	61,38%	0,76	56,33%	0,85	59,84%	0,89	59,97%
		50x	61,86%	0,82	63,02%	0,86	45,02%	0,66	57,56%	0,73	44,01%	0,73	57,78%	0,84	54,88%
64x64	I	20x	58,02%	0,80	58,62%	0,85	52,49%	0,73	53,58%	0,71	46,88%	0,80	56,63%	0,85	54,37%
		50x	63,05%	0,82	63,50%	0,86	57,27%	0,76	58,23%	0,74	51,40%	0,80	61,76%	0,86	59,20%
	T	20x	63,02%	0,86	64,83%	0,90	54,52%	0,78	61,68%	0,76	54,25%	0,85	59,94%	0,89	59,71%
		50x	61,43%	0,82	62,54%	0,85	45,01%	0,66	58,29%	0,73	43,73%	0,73	57,51%	0,84	54,75%
96x96	I	20x	53,25%	0,76	53,55%	0,81	47,34%	0,68	48,17%	0,67	41,27%	0,73	53,44%	0,82	49,50%
		50x	53,14%	0,74	52,93%	0,78	39,97%	0,59	46,92%	0,66	38,70%	0,66	53,60%	0,80	47,54%
	T	20x	56,18%	0,79	56,58%	0,83	47,79%	0,69	51,55%	0,69	41,04%	0,74	55,19%	0,84	51,39%
		50x	52,31%	0,73	52,40%	0,77	38,68%	0,58	46,37%	0,66	38,77%	0,66	52,75%	0,79	46,88%
128x128	I	20x	56,63%	0,79	56,84%	0,84	51,84%	0,72	52,55%	0,71	42,47%	0,78	55,54%	0,84	52,65%
		50x	60,26%	0,80	60,46%	0,84	56,83%	0,75	55,64%	0,72	45,49%	0,77	59,37%	0,84	56,34%
	T	20x	63,84%	0,86	64,88%	0,90	54,48%	0,78	61,96%	0,76	49,19%	0,85	59,83%	0,89	59,03%
		50x	60,45%	0,81	61,29%	0,85	44,99%	0,66	58,45%	0,73	41,30%	0,72	57,55%	0,85	54,01%
∅		59,68%	0,81	60,49%	0,85	50,70%	0,71	55,88%	0,72	47,06%	0,77	57,79%	0,85	55,27%	

Tabelle 8.4.: Klassifizierungsergebnisse der Tamura Texturmerkmale mit einer Histogrammgröße von 96 Bins

Tamura		[1]		[2]		[3]		[4]		[5]		[6]		∅	
Histogrammgr.: 128		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC		
48x48	I	20x	57,93%	0,80	58,69%	0,85	53,89%	0,73	53,07%	0,71	50,38%	0,80	57,39%	0,85	55,23%
		50x	68,59%	0,86	68,99%	0,90	60,18%	0,81	65,39%	0,79	57,38%	0,85	66,46%	0,90	64,50%
	T	20x	62,04%	0,85	63,83%	0,89	55,37%	0,79	60,14%	0,75	55,15%	0,85	59,46%	0,89	59,33%
		50x	69,48%	0,87	71,39%	0,91	51,81%	0,75	69,07%	0,80	47,72%	0,81	63,24%	0,89	62,12%
64x64	I	20x	58,15%	0,80	57,98%	0,85	53,41%	0,72	52,82%	0,71	49,44%	0,80	56,78%	0,85	54,76%
		50x	69,25%	0,86	69,54%	0,90	60,02%	0,80	64,87%	0,78	55,28%	0,85	66,74%	0,90	64,28%
	T	20x	62,19%	0,85	63,93%	0,89	55,33%	0,79	59,94%	0,75	55,45%	0,85	59,82%	0,89	59,44%
		50x	69,46%	0,87	70,85%	0,91	52,03%	0,75	69,86%	0,81	46,29%	0,81	63,27%	0,89	61,96%
64x64	I	20x	58,30%	0,80	58,80%	0,85	52,70%	0,72	54,20%	0,72	49,00%	0,80	57,53%	0,85	55,09%
		50x	68,40%	0,86	69,02%	0,90	59,59%	0,79	64,50%	0,78	53,35%	0,85	66,56%	0,90	63,57%
	T	20x	62,37%	0,85	64,29%	0,89	55,18%	0,78	60,72%	0,76	53,29%	0,86	59,87%	0,89	59,29%
		50x	68,85%	0,88	70,13%	0,90	51,81%	0,74	69,21%	0,80	45,11%	0,81	63,36%	0,89	61,41%
96x96	I	20x	57,47%	0,80	57,98%	0,85	52,49%	0,72	52,75%	0,71	47,57%	0,80	56,48%	0,85	54,12%
		50x	66,50%	0,85	66,99%	0,89	59,70%	0,79	61,82%	0,76	50,81%	0,84	65,40%	0,89	61,87%
	T	20x	62,70%	0,85	63,24%	0,89	55,23%	0,79	60,62%	0,76	50,80%	0,86	59,86%	0,89	58,74%
		50x	68,37%	0,87	69,53%	0,90	51,83%	0,74	68,86%	0,80	39,91%	0,80	63,31%	0,89	60,30%
128x128	I	20x	56,94%	0,79	56,77%	0,84	52,29%	0,72	52,89%	0,71	43,41%	0,79	55,56%	0,84	52,98%
		50x	66,12%	0,84	66,57%	0,88	59,29%	0,79	61,67%	0,76	50,93%	0,82	63,90%	0,88	61,41%
	T	20x	62,75%	0,85	63,43%	0,89	55,07%	0,78	60,78%	0,76	46,51%	0,85	59,55%	0,89	58,02%
		50x	67,89%	0,87	70,05%	0,90	51,84%	0,74	68,84%	0,80	33,06%	0,79	63,18%	0,89	59,14%
∅		64,19%	0,84	65,10%	0,88	54,95%	0,76	61,60%	0,76	49,04%	0,82	61,39%	0,88	59,38%	

Tabelle 8.5.: Klassifizierungsergebnisse der Tamura Texturmerkmale mit einer Histogrammgröße von 128 Bins

Haralick		[1]		[2]		[3]		[4]		[5]		∅		
Fenstergröße: 48x48		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC			
(1,1)	8	I	20x	69,94%	0,84	74,54%	0,93	59,16%	0,79	72,24%	0,82	31,83%	0,77	58,47%
			50x	60,95%	0,77	67,10%	0,88	56,06%	0,74	61,99%	0,76	30,93%	0,69	53,20%
		T	20x	67,88%	0,82	74,26%	0,92	56,45%	0,78	73,21%	0,83	19,54%	0,71	54,71%
	50x		58,20%	0,76	64,26%	0,87	49,35%	0,71	62,63%	0,76	29,80%	0,69	49,87%	
	10	I	20x	68,71%	0,83	73,30%	0,92	59,19%	0,79	70,49%	0,81	33,63%	0,77	58,24%
			50x	59,81%	0,77	65,42%	0,87	54,65%	0,72	60,29%	0,75	30,06%	0,68	51,89%
T		20x	68,48%	0,82	73,97%	0,92	57,18%	0,78	72,39%	0,82	18,26%	0,73	54,70%	
	50x	58,59%	0,77	65,08%	0,87	49,21%	0,71	61,50%	0,75	29,39%	0,68	49,58%		
(2,2)	8	I	20x	69,53%	0,83	75,05%	0,93	59,14%	0,79	72,78%	0,83	29,58%	0,75	58,03%
			50x	61,17%	0,77	66,86%	0,88	54,73%	0,73	61,80%	0,76	30,40%	0,69	52,57%
		T	20x	67,71%	0,82	73,46%	0,92	56,45%	0,78	72,75%	0,83	19,54%	0,71	54,58%
	50x		57,95%	0,76	64,72%	0,86	50,09%	0,71	62,00%	0,75	30,12%	0,69	50,05%	
	10	I	20x	68,99%	0,83	73,66%	0,92	58,51%	0,78	71,02%	0,82	31,36%	0,75	57,68%
			50x	58,99%	0,76	65,16%	0,87	52,20%	0,71	59,92%	0,74	28,23%	0,67	50,31%
T		20x	68,54%	0,83	73,69%	0,92	56,57%	0,78	72,14%	0,82	17,82%	0,72	54,33%	
	50x	58,17%	0,76	64,54%	0,86	49,61%	0,70	61,23%	0,75	28,96%	0,68	49,52%		
(4,4)	8	I	20x	69,79%	0,84	74,38%	0,93	58,17%	0,78	73,70%	0,83	27,44%	0,73	57,45%
			50x	59,32%	0,76	65,30%	0,87	53,44%	0,72	61,64%	0,75	27,02%	0,68	50,97%
		T	20x	66,96%	0,82	73,24%	0,92	56,11%	0,77	72,56%	0,83	19,23%	0,71	54,19%
	50x		57,13%	0,75	63,95%	0,86	50,42%	0,71	61,02%	0,75	29,18%	0,68	49,63%	
	10	I	20x	69,51%	0,83	73,44%	0,92	59,06%	0,79	71,98%	0,82	29,18%	0,73	57,76%
			50x	57,33%	0,75	63,50%	0,85	51,90%	0,70	58,84%	0,74	24,73%	0,65	48,94%
T		20x	68,44%	0,83	73,80%	0,92	56,48%	0,78	72,40%	0,82	17,80%	0,72	54,32%	
	50x	56,89%	0,75	63,67%	0,86	49,66%	0,70	60,00%	0,74	27,36%	0,68	48,71%		
∅			63,71%	0,79	69,43%	0,89	54,74%	0,75	66,69%	0,79	26,72%	0,71	53,32%	

Tabelle 8.6.: Klassifizierungsergebnisse der Haralick-Merkmale mit 48x48 Pixel großen Fenstern, drei unterschiedlichen Abstandsvektoren und mit jeweils 8- und 10-Bit Graustufen

Haralick		[1]		[2]		[3]		[4]		[5]		[6]		∅
Fenstergröße: 64x64		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	
(1,1)	8	I	20x	71,53%	0,85	76,18%	0,94	58,37%	0,80	73,92%	0,84	35,08%	0,79	63,02%
			50x	65,64%	0,81	71,18%	0,90	57,46%	0,77	67,54%	0,80	34,09%	0,69	59,18%
		T	20x	69,20%	0,83	75,53%	0,93	57,10%	0,80	74,00%	0,84	28,35%	0,75	60,84%
	50x		62,85%	0,80	70,37%	0,90	49,75%	0,74	68,89%	0,80	38,14%	0,70	58,00%	
	10	I	20x	69,87%	0,84	74,35%	0,93	58,55%	0,79	71,94%	0,83	34,64%	0,78	61,87%
			50x	64,50%	0,80	69,20%	0,90	56,60%	0,75	64,90%	0,78	33,83%	0,68	57,81%
T		20x	69,77%	0,85	75,46%	0,93	57,34%	0,80	73,27%	0,83	30,10%	0,76	61,19%	
	50x	63,38%	0,80	69,44%	0,90	49,71%	0,73	68,33%	0,80	35,16%	0,69	57,20%		
(2,2)	8	I	20x	71,58%	0,85	76,26%	0,94	58,51%	0,80	73,86%	0,84	32,28%	0,77	61,00%
			50x	64,53%	0,80	70,57%	0,90	57,11%	0,76	66,62%	0,79	33,18%	0,68	60,48%
		T	20x	69,09%	0,83	75,07%	0,93	58,22%	0,81	74,02%	0,84	28,05%	0,75	56,86%
	50x		63,00%	0,79	69,48%	0,89	52,14%	0,74	68,43%	0,80	37,60%	0,70	55,09%	
	10	I	20x	70,33%	0,84	74,33%	0,93	58,62%	0,79	71,78%	0,83	32,57%	0,76	61,07%
			50x	63,35%	0,79	69,26%	0,89	55,97%	0,75	64,05%	0,77	33,08%	0,68	57,04%
T		20x	69,50%	0,84	74,62%	0,93	58,03%	0,81	73,51%	0,83	29,57%	0,76	58,80%	
	50x	62,33%	0,79	68,55%	0,89	50,16%	0,73	66,69%	0,79	35,07%	0,69	56,50%		
(4,4)	8	I	20x	72,14%	0,86	76,09%	0,94	55,58%	0,78	74,82%	0,84	33,88%	0,76	62,50%
			50x	63,12%	0,79	69,20%	0,89	56,04%	0,75	65,41%	0,78	32,70%	0,68	59,46%
		T	20x	68,43%	0,83	74,26%	0,93	57,29%	0,80	73,88%	0,84	27,00%	0,74	57,81%
	50x		61,63%	0,79	67,80%	0,89	52,53%	0,74	66,90%	0,79	36,02%	0,70	54,59%	
	10	I	20x	71,35%	0,85	74,80%	0,93	54,88%	0,78	73,35%	0,84	33,35%	0,76	62,53%
			50x	61,57%	0,78	67,73%	0,88	54,91%	0,73	63,07%	0,77	32,90%	0,67	56,04%
T		20x	69,38%	0,84	74,23%	0,93	55,94%	0,79	73,52%	0,83	28,16%	0,75	60,25%	
	50x	60,75%	0,78	67,29%	0,88	50,51%	0,73	65,33%	0,78	34,15%	0,68	55,61%		
∅			66,62%	0,82	72,14%	0,91	55,47%	0,77	69,92%	0,81	32,87%	0,72	58,31%	

Tabelle 8.7.: Klassifizierungsergebnisse der Haralick-Merkmale mit 64x64 Pixel großen Fenstern, drei unterschiedlichen Abstandsvektoren und mit jeweils 8- und 10-Bit Graustufen

Haralick		[1]		[2]		[3]		[4]		[5]		[6]		∅		
Fenstergröße: 96x96		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC			
(1,1)	8	I	20x	71,35%	0,85	75,64%	0,93	57,57%	0,79	73,20%	0,84	35,20%	0,81	60,84%	0,86	62,30%
			50x	70,79%	0,85	75,48%	0,93	60,77%	0,79	73,58%	0,83	25,62%	0,72	65,21%	0,87	61,91%
		T	20x	69,55%	0,84	75,85%	0,94	59,36%	0,82	76,67%	0,86	36,59%	0,82	60,58%	0,87	63,10%
	50x		69,85%	0,84	75,34%	0,93	54,38%	0,77	75,97%	0,85	31,10%	0,74	60,29%	0,83	61,16%	
	10	I	20x	69,56%	0,84	73,63%	0,93	58,51%	0,79	69,99%	0,82	33,92%	0,81	61,17%	0,86	61,13%
			50x	70,93%	0,84	74,99%	0,92	60,57%	0,79	71,73%	0,82	25,50%	0,73	65,45%	0,87	61,53%
T		20x	69,37%	0,84	73,99%	0,93	60,30%	0,82	74,17%	0,84	37,25%	0,82	59,18%	0,86	62,38%	
		50x	69,04%	0,83	73,93%	0,92	54,38%	0,77	74,39%	0,84	25,32%	0,73	60,15%	0,83	59,54%	
(2,2)	8	I	20x	70,78%	0,85	74,76%	0,93	58,22%	0,79	72,41%	0,83	35,37%	0,80	60,89%	0,86	62,07%
			50x	70,10%	0,84	74,91%	0,92	58,67%	0,78	72,19%	0,83	24,68%	0,72	64,54%	0,87	60,85%
		T	20x	69,80%	0,84	75,87%	0,94	60,75%	0,82	76,31%	0,86	37,19%	0,82	59,83%	0,86	63,29%
	50x		68,86%	0,83	74,63%	0,92	56,43%	0,78	75,02%	0,84	30,22%	0,74	59,96%	0,83	60,85%	
	10	I	20x	69,55%	0,84	73,88%	0,93	58,79%	0,79	69,75%	0,82	33,79%	0,80	61,18%	0,87	61,16%
			50x	69,55%	0,84	74,31%	0,92	59,30%	0,79	70,60%	0,81	24,16%	0,73	64,01%	0,86	60,32%
T		20x	69,30%	0,84	74,34%	0,93	60,69%	0,82	74,49%	0,84	37,35%	0,82	59,83%	0,86	62,67%	
		50x	67,46%	0,83	72,73%	0,91	54,78%	0,77	72,55%	0,82	25,02%	0,73	58,80%	0,82	58,56%	
(4,4)	8	I	20x	71,37%	0,86	75,80%	0,94	57,47%	0,79	73,55%	0,84	39,31%	0,81	62,79%	0,88	63,38%
			50x	68,22%	0,83	73,76%	0,92	57,18%	0,77	70,70%	0,82	23,52%	0,72	63,03%	0,86	59,40%
		T	20x	69,43%	0,84	74,80%	0,93	59,47%	0,81	75,19%	0,85	37,01%	0,81	59,04%	0,85	62,49%
	50x		68,09%	0,83	73,82%	0,92	57,57%	0,78	74,62%	0,84	28,47%	0,74	59,41%	0,82	60,33%	
	10	I	20x	71,49%	0,85	74,91%	0,93	58,32%	0,80	72,19%	0,83	36,49%	0,80	62,90%	0,88	62,72%
			50x	67,96%	0,83	72,82%	0,91	57,86%	0,78	69,11%	0,81	22,44%	0,72	61,46%	0,84	58,61%
T		20x	69,17%	0,84	74,17%	0,93	59,07%	0,81	75,06%	0,85	36,47%	0,82	59,41%	0,85	62,23%	
		50x	66,41%	0,82	71,88%	0,91	55,04%	0,77	72,25%	0,82	24,32%	0,73	58,27%	0,82	58,03%	
		∅		69,50%	0,84	74,43%	0,93	58,14%	0,79	73,15%	0,83	31,10%	0,77	61,18%	0,85	61,25%

Tabelle 8.8.: Klassifizierungsergebnisse der Haralick-Merkmale mit 96x96 Pixel großen Fenstern, drei unterschiedlichen Abstandsvektoren und mit jeweils 8- und 10-Bit Graustufen

Haralick		[1]		[2]		[3]		[4]		[5]		[6]		∅		
Fenstergröße: 128x128		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC			
(1,1)	8	I	20x	71,13%	0,86	75,50%	0,94	57,57%	0,78	72,62%	0,83	37,59%	0,82	61,22%	0,87	62,04%
			50x	74,33%	0,87	77,84%	0,94	62,91%	0,83	76,24%	0,85	31,70%	0,80	67,20%	0,89	67,53%
		T	20x	69,47%	0,84	75,69%	0,94	59,34%	0,82	75,87%	0,85	40,35%	0,83	60,09%	0,86	62,63%
	50x		73,66%	0,86	78,92%	0,94	57,27%	0,79	81,43%	0,88	40,78%	0,81	63,39%	0,86	68,59%	
	10	I	20x	68,98%	0,84	72,69%	0,93	58,83%	0,79	69,78%	0,82	37,39%	0,82	60,46%	0,87	60,59%
			50x	72,50%	0,86	76,98%	0,94	63,60%	0,83	74,00%	0,84	32,07%	0,80	65,64%	0,88	65,56%
T		20x	68,33%	0,84	73,36%	0,93	59,29%	0,82	73,43%	0,84	40,74%	0,83	59,22%	0,86	60,91%	
		50x	71,75%	0,85	76,82%	0,94	57,16%	0,79	78,67%	0,86	34,93%	0,79	62,16%	0,86	65,19%	
(2,2)	8	I	20x	71,07%	0,86	75,30%	0,94	58,02%	0,79	72,40%	0,83	37,58%	0,81	59,77%	0,86	61,50%
			50x	73,82%	0,87	77,84%	0,94	64,09%	0,83	75,80%	0,85	30,59%	0,80	66,64%	0,89	66,74%
		T	20x	69,57%	0,84	74,89%	0,93	59,28%	0,82	74,87%	0,85	40,32%	0,83	59,56%	0,86	62,38%
	50x		73,08%	0,86	78,20%	0,94	58,53%	0,80	79,85%	0,87	39,74%	0,81	62,41%	0,86	68,00%	
	10	I	20x	69,65%	0,84	73,52%	0,93	59,06%	0,79	70,21%	0,82	37,33%	0,82	60,40%	0,86	60,72%
			50x	72,72%	0,87	76,77%	0,94	64,15%	0,83	74,02%	0,84	30,93%	0,80	64,94%	0,87	65,11%
T		20x	68,74%	0,84	73,34%	0,93	59,17%	0,81	73,14%	0,84	40,46%	0,83	59,53%	0,86	61,25%	
		50x	71,64%	0,85	76,06%	0,93	57,96%	0,79	76,65%	0,85	35,19%	0,79	61,90%	0,86	64,98%	
(4,4)	8	I	20x	71,23%	0,85	75,45%	0,94	56,83%	0,78	74,33%	0,85	40,57%	0,81	61,55%	0,88	62,64%
			50x	72,51%	0,86	77,07%	0,94	64,07%	0,83	75,18%	0,84	29,62%	0,79	64,73%	0,88	65,64%
		T	20x	68,27%	0,84	73,83%	0,93	58,80%	0,81	74,51%	0,85	39,84%	0,82	59,10%	0,86	61,52%
	50x		71,47%	0,85	77,41%	0,94	59,52%	0,80	79,40%	0,87	36,89%	0,81	61,20%	0,85	66,81%	
	10	I	20x	70,59%	0,85	74,73%	0,93	58,64%	0,79	72,44%	0,83	40,02%	0,81	61,61%	0,88	62,11%
			50x	71,69%	0,86	75,94%	0,93	63,50%	0,83	73,78%	0,84	29,62%	0,79	62,66%	0,87	63,92%
T		20x	68,57%	0,84	74,20%	0,93	58,80%	0,81	74,55%	0,85	39,85%	0,82	59,31%	0,86	60,91%	
		50x	71,58%	0,85	77,41%	0,94	59,56%	0,80	79,30%	0,87	36,97%	0,81	61,40%	0,85	64,67%	
		∅		71,10%	0,85	75,82%	0,94	59,83%	0,81	75,10%	0,85	36,71%	0,81	61,92%	0,87	63,83%

Tabelle 8.9.: Klassifizierungsergebnisse der Haralick-Merkmale mit 128x128 Pixel großen Fenstern, drei unterschiedlichen Abstandsvektoren und mit jeweils 8- und 10-Bit Graustufen

		[1]		[2]		[3]		[4]		[5]		[6]				
		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC			
64x64	I	20x	R	67,25%	0,81	71,88%	0,93	77,87%	0,92	83,04%	0,90	39,52%	0,80	58,75%	0,83	
		50x	R	66,11%	0,80	71,38%	0,92	77,26%	0,92	80,96%	0,88	44,50%	0,82	59,74%	0,84	
	T	20x	R	64,84%	0,79	71,62%	0,91	75,41%	0,91	79,68%	0,88	47,06%	0,85	63,15%	0,87	
		50x	R	67,49%	0,80	71,12%	0,91	74,54%	0,90	79,11%	0,87	52,06%	0,87	64,74%	0,87	
	I	20x	R	60,29%	0,75	67,28%	0,89	73,45%	0,90	74,45%	0,83	39,76%	0,79	58,22%	0,83	
		50x	R	62,49%	0,77	66,97%	0,89	72,91%	0,90	72,75%	0,82	40,75%	0,79	58,06%	0,82	
	T	20x	R	54,34%	0,72	62,28%	0,86	67,90%	0,87	64,05%	0,77	27,78%	0,71	52,47%	0,78	
		50x	R	54,00%	0,72	60,83%	0,84	66,39%	0,86	59,55%	0,74	35,38%	0,77	53,78%	0,79	
	∅				62,10%	0,77	67,92%	0,89	73,22%	0,90	74,20%	0,84	40,85%	0,80	58,61%	0,83
	96x96	I	20x	R	68,28%	0,82	73,48%	0,94	81,79%	0,94	84,31%	0,91	45,69%	0,85	62,96%	0,87
50x			R	68,70%	0,82	72,32%	0,94	80,84%	0,94	83,17%	0,90	53,68%	0,88	63,11%	0,88	
T		20x	R	72,44%	0,84	78,22%	0,95	81,72%	0,94	87,03%	0,92	51,44%	0,87	66,47%	0,89	
		50x	R	73,28%	0,83	76,67%	0,94	81,01%	0,93	86,41%	0,92	54,96%	0,90	68,96%	0,90	
I		20x	R	64,50%	0,78	69,04%	0,91	76,85%	0,93	74,75%	0,83	45,47%	0,82	59,30%	0,85	
		50x	R	65,49%	0,79	67,84%	0,91	75,25%	0,92	72,22%	0,82	47,80%	0,83	58,22%	0,84	
T		20x	R	59,62%	0,75	67,48%	0,89	73,81%	0,90	67,29%	0,79	27,03%	0,73	54,43%	0,80	
		50x	R	61,66%	0,77	67,54%	0,88	71,30%	0,89	64,83%	0,77	34,27%	0,77	58,55%	0,82	
∅				66,75%	0,80	71,57%	0,92	77,82%	0,92	77,50%	0,86	45,04%	0,83	61,50%	0,86	
128x128		I	20x	R	69,56%	0,81	74,67%	0,94	83,49%	0,95	85,83%	0,92	50,60%	0,87	63,39%	0,88
	50x		R	70,44%	0,82	74,93%	0,94	82,61%	0,95	83,98%	0,91	55,32%	0,89	65,86%	0,89	
	T	20x	R	77,57%	0,87	82,42%	0,97	86,58%	0,96	91,87%	0,95	55,65%	0,91	72,37%	0,92	
		50x	R	80,16%	0,88	81,81%	0,97	86,46%	0,96	91,72%	0,95	65,63%	0,93	73,33%	0,93	
	I	20x	R	65,14%	0,79	71,74%	0,92	81,24%	0,94	78,75%	0,86	35,76%	0,83	60,40%	0,85	
		50x	R	65,12%	0,79	70,08%	0,92	77,58%	0,93	73,99%	0,83	50,47%	0,84	58,73%	0,85	
	T	20x	R	66,67%	0,80	74,17%	0,93	81,83%	0,94	76,06%	0,86	26,57%	0,76	57,23%	0,84	
		50x	R	67,11%	0,80	72,20%	0,92	76,20%	0,92	68,78%	0,80	41,54%	0,83	61,92%	0,85	
	∅				70,22%	0,82	75,25%	0,94	82,00%	0,94	81,37%	0,89	47,69%	0,86	64,15%	0,88
	192x192	I	20x	R	71,25%	0,83	76,63%	0,95	84,90%	0,95	84,22%	0,90	52,53%	0,86	64,96%	0,88
50x			R	72,14%	0,84	75,57%	0,95	84,76%	0,95	85,21%	0,91	54,73%	0,87	65,31%	0,88	
T		20x	R	78,45%	0,87	84,26%	0,97	88,59%	0,97	93,94%	0,96	53,36%	0,92	74,49%	0,94	
		50x	R	79,27%	0,88	82,49%	0,97	87,76%	0,97	92,50%	0,95	63,45%	0,93	72,77%	0,93	
I		20x	R	65,37%	0,79	71,93%	0,92	81,15%	0,94	79,07%	0,87	36,03%	0,83	60,94%	0,85	
		50x	R	65,73%	0,80	71,06%	0,91	79,79%	0,94	76,81%	0,85	48,51%	0,84	61,80%	0,86	
T		20x	R	67,71%	0,81	73,72%	0,93	81,59%	0,94	76,28%	0,86	26,48%	0,76	58,61%	0,84	
		50x	R	67,88%	0,81	73,76%	0,93	77,43%	0,93	75,17%	0,85	39,33%	0,82	63,80%	0,87	
∅				70,98%	0,83	76,18%	0,94	83,25%	0,95	82,90%	0,89	46,80%	0,85	65,34%	0,88	

Tabelle 8.10.: Klassifikationsergebnisse der LBP-Merkmale.
Zeilen mit einem R sind die rotationsinvarianten Merkmale.

Laws		[1]		[2]		[3]		[4]		[5]		[6]		∅ PC		
Energiefilter		PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC	PC	AUC			
15x15	I	20x	54,00%	0,70	59,50%	0,82	49,58%	0,61	44,08%	0,62	17,18%	0,67	56,06%	0,79	46,73%	
		50x	50,22%	0,68	57,04%	0,80	45,13%	0,63	40,36%	0,61	43,28%	0,71	54,51%	0,79	48,42%	
	T	20x	47,35%	0,64	53,55%	0,75	47,55%	0,52	37,65%	0,58	16,35%	0,61	52,41%	0,73	42,48%	
		50x	44,79%	0,65	51,51%	0,76	43,03%	0,52	34,06%	0,57	41,00%	0,58	51,06%	0,77	44,24%	
30x30	I	20x	61,85%	0,77	67,63%	0,89	51,91%	0,69	52,43%	0,69	23,28%	0,72	56,96%	0,82	52,34%	
		50x	58,34%	0,75	64,87%	0,86	52,57%	0,72	47,03%	0,66	37,28%	0,73	58,51%	0,82	53,10%	
	T	20x	53,54%	0,71	59,29%	0,82	46,85%	0,56	44,37%	0,63	13,80%	0,65	55,97%	0,79	45,64%	
		50x	54,19%	0,73	59,95%	0,83	42,11%	0,53	38,89%	0,61	39,26%	0,63	55,77%	0,81	48,36%	
60x60	I	20x	67,68%	0,83	74,39%	0,93	52,93%	0,73	61,78%	0,76	30,58%	0,75	59,93%	0,84	57,88%	
		50x	66,74%	0,82	73,35%	0,92	57,02%	0,77	60,07%	0,75	51,13%	0,79	63,17%	0,87	61,91%	
	T	20x	59,65%	0,77	66,22%	0,89	50,01%	0,65	51,87%	0,69	18,67%	0,68	57,34%	0,83	50,63%	
		50x	60,48%	0,78	66,17%	0,88	44,56%	0,59	45,14%	0,65	37,81%	0,69	58,58%	0,83	52,12%	
∅				56,57%	0,74	62,79%	0,85	48,60%	0,63	46,48%	0,65	30,80%	0,68	56,69%	0,81	50,32%

Tabelle 8.11.: Klassifizierungsergebnisse der Laws-Energiemerkmale mit drei unterschiedlichen Fenstergrößen

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe.

Magdeburg, den 13.06.2014