Fortschritt-Berichte VDI

VDI

Reihe 10

Informatik/
Kommunikation

Dipl.-Ing. Arne Ehlers,
Hannover

Nr. 855

Object Detection using Feature Mining in a Distributed Machine Learning Framework

tnt
Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Object Detection using Feature Mining in a Distributed Machine Learning Framework

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)

genehmigte

Dissertation

von

Dipl.-Ing. Arne Ehlers

geboren am 24. November 1976 in Itzehoe.

2016

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Referent: Prof. Dr.-Ing. B. Rosenhahn
Korreferent: Prof. Dr.-Ing. E. Reithmeier
Vorsitzender: Prof. Dr.-Ing. J. Ostermann
Tag der Promotion: 09.11.2016

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Fortschritt-Berichte VDI

Reihe 10

Informatik/
Kommunikation

Dipl.-Ing. Arne Ehlers,
Hannover

Nr. 855

Object Detection using Feature Mining in a Distributed Machine Learning Framework

tnt

Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Ehlers, Arne
Object Detection using Feature Mining in a Distributed Machine Learning Framework
Fortschr.-Ber. VDI Reihe 10 Nr. 855. Düsseldorf: VDI Verlag 2017.
162 Seiten, 72 Bilder, 10 Tabellen.
ISBN 978-3-18-385510-0, ISSN 0178-9627,
€ 62,00/VDI-Mitgliederpreis € 55,80.

Keywords: Object Detection – Feature Mining – Fractal Features – Data Augmentation – Machine Learning – Adaptive Boosting – Distributed Computing

This dissertation addresses the problem of visual object detection based on machine-learned classifiers. A distributed machine learning framework is developed to learn detectors for several object classes creating cascaded ensemble classifiers by the Adaptive Boosting algorithm. Methods are proposed that enhance several components of an object detection framework:
At first, the thesis deals with augmenting the training data in order to improve the performance of object detectors learned from sparse training sets.
Secondly, feature mining strategies are introduced to create feature sets that are customized to the object class to be detected. Furthermore, a novel class of fractal features is proposed that allows to represent a wide variety of shapes.
Thirdly, a method is introduced that models and combines internal confidences and uncertainties of the cascaded detector using Dempster's theory of evidence in order to increase the quality of the post-processing.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

© VDI Verlag GmbH · Düsseldorf 2017

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9627

ISBN 978-3-18-385510-0

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Acknowledgement

The present thesis originates from my work at the Institut für Informationsverarbeitung (TNT) of the Gottfried Wilhelm Leibniz Universität Hannover. Doing research at the institute has been a great experience. I have learned a lot about computer vision, machine learning, and object detection. But all this would not have been possible without the help of many people I would like to thank at this point.

I would like to express my deep gratitude to Prof. Dr.-Ing. Bodo Rosenhahn for giving me the opportunity to work in his group and for the supervision of my dissertation. His guidance, advises and feedback largely contributed to making this thesis a success.

I also like to thank Prof. Dr.-Ing. Eduard Reithmeier for being the second supervisor of my thesis and Prof. Dr.-Ing. Jörn Ostermann for being the chair of my defense committee.

Special thanks go to Florian Baumann. We made up a superb team by complementing each other perfectly in many tasks of research and development. I very much enjoyed our corporate work on many papers and participating several times together in conferences.

I would like to thank all colleagues at the institute for their help and support and their contribution to create such a brilliant atmosphere at the TNT. Especially, I would like to mention Matthias Reso, Stephan Preihs, Kai Cordes, Björn Scheuermann, Hanno Ackermann, and Marco Munderloh. Thank you very much for giving suggestions, criticisms, and encouragement in discussions and as well for the great time we spent together.

A very special thanks goes to my parents Hilde and Walter Ehlers who enabled me to study and always supported me.

This work has been partially funded by the DFG within the excellence cluster REBIRTH. The author gratefully acknowledges the support.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Contents

Abbreviations **VII**

Symbols and Notation **IX**

Abstract/Kurzfassung **XII**

1 Introduction **1**

2 Related Work and Data Sets **16**
- 2.1 Machine Learning for Visual Object Detection 17
  - 2.1.1 Feature Provision 17
  - 2.1.2 Learning Algorithms 19
- 2.2 Data Sets and Benchmarks 21

3 Fundamentals **28**
- 3.1 Common Features 29
  - 3.1.1 Haar-like Features 29
  - 3.1.2 Histograms of Oriented Gradients 31
  - 3.1.3 From Features to Classifiers 33
- 3.2 Supervised Machine Learning 34
  - 3.2.1 Adaptive Boosting 35
  - 3.2.2 Viola and Jones Detection Framework 38
  - 3.2.3 Margin Analysis 50
  - 3.2.4 Variants of Boosting Algorithms 51
- 3.3 Data Analysis 52
  - 3.3.1 Cluster Analysis 53
  - 3.3.2 Principal Component Analysis 56
- 3.4 Detector Performance Measures 58

4 Distributed Machine Learning Framework **62**

5 Learning from Sparse Training Data **68**
- 5.1 Training Data Augmentation 70
- 5.2 Experimental Results 71
  - 5.2.1 Experiments on Face Detection 71
  - 5.2.2 Experiments on Cell Data Set 74

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5.3 Discussion 74

6 Fractal Integral Paths 76

6.1 Boosted Fractal Integral Paths 79

6.1.1 Fractals 79

6.1.2 Fractal Features 80

6.1.3 Fractal Properties 80

6.1.4 Construction of Fractals 82

6.1.5 Feature Types 82

6.2 Experimental Results 83

6.2.1 Face Detection 83

6.2.2 Microscopic Cell Detection 87

6.2.3 Training and Computing Time 89

6.3 Discussion 89

7 Multi-Feature Mining for Detector Learning 90

7.1 2Rec Features 93

7.2 Keypoint HOG Features 95

7.3 Experimental Results 99

7.3.1 Face Detection 99

7.3.2 Lateral Car Detection 110

7.3.3 Pedestrian Detection 113

7.3.4 Insights into the Training Process 118

7.4 Discussion 120

8 Non-Maximum Suppression using Dempster's Theory of Evidence 121

8.1 Merging Multiple Detections based on Dempster's Theory 123

8.1.1 Cascaded Classifier 123

8.1.2 Dempster-Shafer Theory of Evidence 124

8.1.3 Joint Confidence based on Dempster-Shafer 125

8.1.4 Confidence-based Detection Merging 125

8.2 Experimental Results 126

8.2.1 Face Detection 127

8.2.2 Lateral Car Detection 129

8.3 Discussion 129

9 Conclusion 131

A Lindenmayer Systems Defining Fractal Integral Paths 135

A.1 L-System Defining Gosper Curve 136

A.2 L-System Defining E-Curve 136

Bibliography 138

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Abbreviations

1D	one dimensional
2D	two dimensional
3D	three dimensional
AdaBoost	Adaptive Boosting
ADAS	Advanced Driver Assistance Systems
AEB	Autonomous Emergency Braking
Bagging	Bootstrap Aggregating
CNN	Convolutional Neural Network
DS	Dempster-Shafer Theory of Evidence
Euro NCAP	European New Car Assessment Programme
$\mathbf{F_1}$	F-measure
FP	False Positive
FPGA	Field-Programmable Gate Array
FPPI	False Positives Per Image
FPPW	False Positives Per Window
FPR	False Positive Rate
GMM	Gaussian Mixture Model
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HOG	Histograms of Oriented Gradients
ICA	Independent Component Analysis
KPCA	Kernel Principal Component Analysis
KPHOG	Keypoint HOG
L-System	Lindenmayer System
MPI	Message Passing Interface
NMS	Non-Maximum Suppression

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

OpenCV	Open Source Computer Vision library
OpenMP	Open Multi-Processing
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
SIFT	Scale-Invariant Feature Transform
SLIC	Simple Linear Iterative Clustering
SMD	Surface-Mount Device
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Symbols and Notation

General Symbols

p	probability
$\mathbf{x}$	training sample
y	label for a training sample
I	image matrix
μ_I	arithmetic mean (pixel) of an image I
N_{Pixel}	number of pixels in an image or detection window
u, v	image coordinates
X	random variable
$\mathrm{Var}(X)$	variance of random variable X
$\mathrm{E}(X)$	expected value of random variable X
i, j, n, r, s	control variables
N_{TP}	number of true positive classifications
N_{FP}	number of false positive classifications
N_{P}	number of positives in test set
N_{N}	number of negatives in test set

Symbols for Adaptive Boosting

$\mathcal{S}$	set of training examples
$N_{\mathcal{S}}$	number of training examples
$\mathcal{X}$	set of training samples
$\mathcal{Y}$	set of possible training labels
N_R	number of training rounds or weak classifiers
W	distribution over the training set $\mathcal{S}$
h	weak hypothesis or weak classifier
ϵ_r	training error in round r
β_r	mapping factor for round r
H	strong hypothesis
J	loss function

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Symbols for Viola and Jones Learning Scheme

γ	feature
ρ	polarity of a weak classifier
θ	decision threshold of weak classifier
$N_{\mathcal{P}}$	number of positive training examples
$N_{\mathcal{N}}$	number of negative training examples
w	weight of training example
e_i	indicator function for classification of i-th training example
$\sum_{r=1}^{N_R} \alpha_r h_r(\mathbf{x})$	linear combination
α_r	weighting factor for weak classifier of round r
$\mathcal{P}$	set of positive training examples
$\mathcal{N}$	set of negative training examples
$w_{r,\mathcal{P}}$	sum of weights of positive training set $\mathcal{P}$ in round r
$w_{r,\mathcal{N}}$	sum of weights of positive training set $\mathcal{N}$ in round r
$\hat{w}_{r,\mathcal{P}}$	sum of positive example weights before in sorted list including the current example in round r
$\hat{w}_{r,\mathcal{N}}$	sum of negative example weights before in sorted list including the current example in round r

Symbols for Viola and Jones Cascade Scheme

t	minimum acceptable true positive rate or detection rate per stage
T	true positive rate or detection rate of a cascaded classifier
f	maximum acceptable false positive rate per stage
F	false positive rate of a cascaded classifier
F_{Target}	target false positive rate of a cascaded classifier
τ	decision threshold of strong classifier
N_S	number of stages in a cascaded classifier
N_{Rs}	number of weak classifiers in s-th cascade stage
h_{r_s}	weak classifier of round r in s-th cascade stage
H_s	strong classifier or ensemble of weak classifiers in s-th cascade stage
α_{rs}	weighting factor for r-th weak classifier in s-th cascade stage
τ_s	decision threshold of the strong classifier or ensemble of weak classifiers in s-th cascade stage

Symbols for Learning from Sparse Training Data

Ψ	mean of the training samples
$\overline{\mathbf{x}}$	difference vector of samples to mean training data Ψ

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

$\overline{X}$	zero-mean data matrix of shifted training samples $\overline{\mathbf{x}}$
C	covariance matrix
ϕ	non-linear transformation into a higher dimensional space
k	kernel function $k(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$
λ	parameter controlling strength of projection into PCA-space
U	matrix of left-singular vectors
Σ	diagonal matrix of singular values
V	matrix of right-singular vectors
Λ	negative training example to be projected

Symbols for Fractal Integral Paths

$\mathcal{G} = (\mathcal{V},\omega,\mathcal{R})$	grammar defining a Lindenmayer system
$\mathcal{V}$	alphabet of a Lindenmayer system
ω	string of symbols from $\mathcal{V}$ defining the initial state
$\mathcal{R}$	set of production rules
X, Y, F, $+$, $-$	symbols used in alphabet $\mathcal{V}$

Symbols for Dempster-Shafer Theory of Evidence

$\emptyset$	empty set
Ω	frame of discernment or hypotheses set
$\wp(\Omega)$	power set
A	hypothesis, $A \in \wp(\Omega)$
m	mass function
$\otimes$	Dempster's rule of combination

Symbols for Non-Maximum Suppression using Evidence Theory

D_i	i-th candidate detection
ι, κ	scale in u and v-dimension
D'_j	refined merged detection for j-th cluster L_j of candidate detections
$\mathcal{L}$	partition of the set of candidate detections
L	subset or cluster of candidate detections
Γ_j	confidence of j-th detection cluster

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Abstract

An important task in visual recognition systems, aiming on the extraction and interpretation of information in images or videos, is the detection of objects. In this process, all instances of a specified object class are requested to be localized in the visual input data. Object detection is essential for many applications that require a more comprehensive scene understanding, like advanced driver assistance systems or self-driving cars. The utilized object detectors are often created by machine learning algorithms that follow the paradigm of learning from examples. In a computational expensive training process, the algorithms learn the characteristics and visual appearance of the object class from training examples but the created detector has to work very fast and efficiently. Frequently, the object characteristics are not directly extracted from the observations but from a feature representation of the input data that gives a guiding principle on how to identify distinctive structures.

This thesis addresses the problem of visual object detection based on machine-learned classifiers. A distributed machine learning framework is developed to learn detectors for several object classes creating cascaded ensemble classifiers by the Adaptive Boosting algorithm. Methods are proposed that enhance several components of an object detection framework to improve its performance:

At first, the thesis deals with augmenting the training data in order to improve the performance of object detectors learned from sparse training sets. This problem frequently arises in industrial applications when highly specialized detectors are learned for e.g. quality assurance.

Secondly, methods are proposed to enhance the feature set that is utilized in the detector learning and its application. Feature mining strategies are introduced in order to create feature sets that are customized to the object class to be detected. By adapting to distinctive object structures, more representative features are assembled in a set of manageable size that enables an efficient detector learning. Furthermore, a novel class of fractal features is proposed that allows to represent a wide variety of shapes.

Thirdly, improvements are proposed to the post-processing that is performed after applying the learned detector to further work up its output. Commonly, this involves the assignment of confidences, merging detections that are very close to each other and dropping detections having low confidence. A method is introduced that models and combines internal confidences and uncertainties of the cascaded detector using Dempster's theory of evidence in order to increase the quality of the post-processing.

Keywords: Object Detection, Feature Mining, Fractal Features, Data Augmentation, Machine Learning, Adaptive Boosting, Distributed Computing

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Kurzfassung

Die Objektdetektion ist eine wichtige Teilaufgabe im maschinellen Sehen, welches die Extraktion von Informationen aus Bildern oder Videos und deren Interpretation zum Ziel hat. Hierbei sollen sämtliche Instanzen einer Objektklasse in den visuellen Eingangsdaten lokalisiert werden. Die Detektion von Objekten ist eine elementare Voraussetzung für weitergehende Verfahren wie Fahrerassistenzsysteme oder selbstfahrende Autos, die eine umfassendere Wahrnehmung ihrer Umgebung erfordern. Die eingesetzten Objektdetektoren sind häufig durch maschinelle Lernalgorithmen erstellt worden, die dem Paradigma des Lernens anhand von Beispielen folgen. Der Algorithmus lernt hierbei in einem rechenintensiven Trainingsprozess das charakteristische Aussehen der Objektklasse anhand von Trainingsbeispielen. Der erstellte Detektor hingegen muss sehr schnell und effizient arbeiten. Häufig werden die Objektcharakteristiken nicht direkt aus den wahrgenommenen Eingangsdaten sondern aus einer Merkmalsdarstellung extrahiert, die Richtlinien zur Identifizierung markanter Strukturen vorgibt.

Diese Dissertation befasst sich mit der visuellen Objektdetektion durch maschinell gelernte Klassifikatoren. Ein verteiltes maschinelles Lernsystem ist entwickelt worden, um mit Hilfe des Adaptive Boosting Algorithmus Ensemble-Klassifikatoren für unterschiedliche Objektklassen anzulernen. Es werden Verfahren zur Verbesserung verschiedener Komponenten eines Objektdetektionssystems vorgestellt, um die Detektionsleistung des Gesamtsystems zu erhöhen:

Als Erstes beschäftigt sich diese Arbeit mit der Anreicherung der Trainingsdaten, um die Leistung von Detektoren zu steigern, welche auf kleinen Trainingsmengen angelernt werden. Diese Problematik tritt häufiger bei industriellen Anwendungen auf, wenn hoch spezialisierte Detektoren beispielsweise für die Qualitätssicherung erstellt werden sollen.

Der zweite Beitrag der Dissertation stellt Verfahren zur Verbesserung der Merkmalsmengen vor, die beim Anlernen eines Detektors und während der Detektion genutzt werden. Es werden Methoden zur gezielten Generierung von Merkmalsmengen entwickelt. Hierdurch können die Merkmalsmengen an die Charakteristiken der zu detektierenden Objektklasse angepasst werden, sodass eine Menge von aussagekräftigeren Merkmalen entsteht, die gleichzeitig überschaubar ist und somit ein effizientes Anlernen erlaubt. Weiterhin wird eine neue Klasse von Fraktalmerkmalen vorgestellt, die vielfältige Strukturen repräsentieren kann.

Drittens werden Verbesserungen für die Detektionsnachverarbeitung entwickelt. Üblicherweise werden den Detektionen in diesem Schritt Konfidenzen zugewiesen, nah beieinander gelegene Detektionen verschmolzen und Detektionen mit niedriger Konfidenz verworfen. Ein Verfahren wird vorgestellt, dass interne Konfidenzen und Unsicherheiten der Detektorkaskade mit Hilfe der Evidenztheorie modelliert und kombiniert, um die Qualität der Nachverarbeitung zu erhöhen.

Stichworte: Objektdetektion, Merksmalsextraktion, Fraktalmerkmale, Datenaugmentation, Maschinelles Lernen, Adaptive Boosting, Verteiltes Rechnen

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter

1

Introduction

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

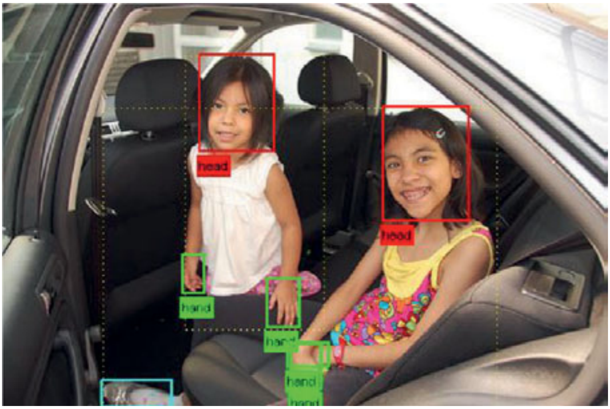


Figure 1.1: Example image of detected persons and body parts, image from [51].

Motivation

Object detection is the process of finding all instances of a requested object class and specifying their size and location. It is an essential task in computer vision that has the goal to give machines the ability to perceive their environment in a way humans do. Object detection is widely used in today's machines and electronic devices. Digital cameras e.g. use face detectors to control the auto-focus [121] or to perform smile detection [152]. Furthermore, the detection of a person's body parts as illustrated in Figure 1.1 enables novel ways in Human-Computer Interaction (HCI). Examples are current smart TVs that can be controlled by hand gestures [87] or detectors that enable to scroll a display by eye movement while reading [117]. Several detectors for body parts might be combined for the analysis of scenes where humans interact with each other. An example application is the automatic detection of emergency events in crowded scenes by intelligent surveillance systems [67].

Object detectors are as well the basis for various Advanced Driver Assistance Systems (ADAS). Many of today's new cars already provide detectors that are able to recognize traffic signs. The European New Car Assessment Programme (Euro NCAP) announced for 2016 to extend their car safety rating by the evaluation of Autonomous Emergency Braking (AEB) systems that detect pedestrians and intervene to prevent or weaken a collision [49].

But the development of autonomously driving cars requires a comprehensive scene understanding that involves the detection of several different objects like the road or lane, other cars, and pedestrians [48, 153] as illustrated in Figure 1.2.

Another field of application for object detectors is industrial production utilizing object detection for quality assurance after the assembly line. Defective products are automatically picked out when e.g. a visual inspection system detects a fault

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

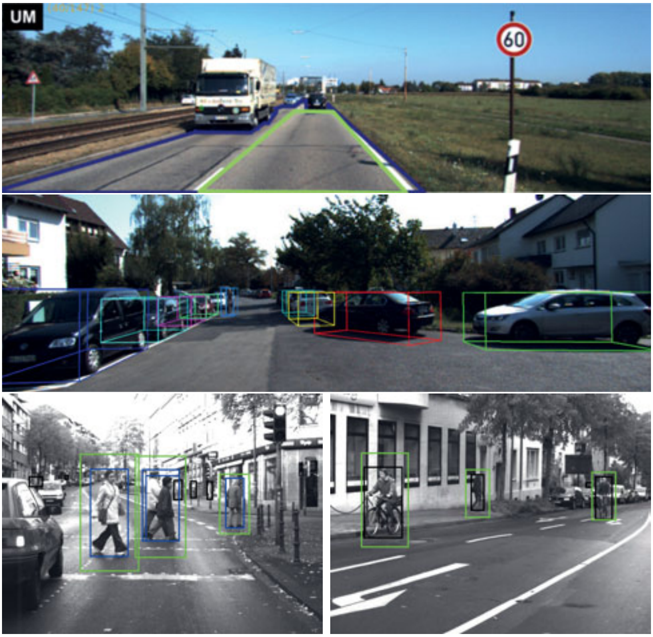


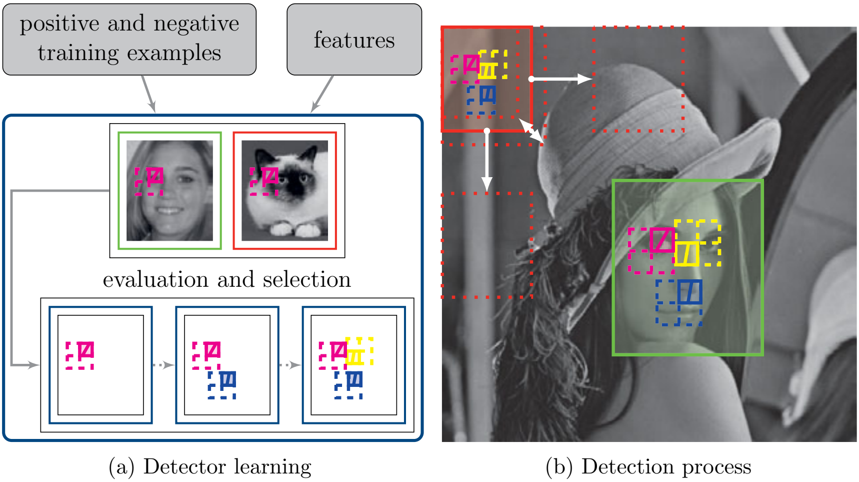
Figure 1.2: Example images of objects to be detected in street traffic. From top to bottom: Road and lane detection [60]. Car detection [63]. Pedestrian detection [47] (see Chapter 7).

[101].

Instead of directly designing a detector for each class of objects to be recognized, often machine learning algorithms are utilized to learn classifiers from training examples. These algorithms autonomously construct a classifier in an in general computational expensive and time-consuming learning phase.

Figure 1.3 presents the basic work flow of the detector learning and the detection process. Machine learning algorithms for binary decisions commonly require a positive and a negative training set of sensor information (e.g. camera images) showing and explicitly not showing the object, respectively. In addition, a feature set is provided that gives the learning algorithm a guiding principle on how to distinguish between objects. This principle, implied by the feature set, might be e.g. to analyze edges [126] or to exploit intensity differences between image regions [145]. The most simplest case is to consider pixel intensities as features. In that case, the size of the set of features is equal to the number of pixels in an example image. But single pixel values provide no information about their spatial neighborhood that is required e.g. to exploit edges. As a consequence, more complex features are commonly necessary that utilize spatial image neighborhoods. But such a feature set swiftly grows to

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.
]



(a) Detector learning (b) Detection process

Figure 1.3: Detector learning and detection process. (a) A face detector is learned from training examples [1]. The feature set is evaluated on the examples and three features are selected that split the training set in its positive and negative part with a comparatively small error. (b) The learned detector is utilized to find a face. In a *sliding window* approach, the detector window is scaled and shifted over the image and the content of the window is classified.

huge sizes if any possible feature manifestation is considered.

The following example demonstrates this circumstance: A face detector should be learned from training images of the size 24×24 pixels. The utilized feature type is computed on quadratic image neighborhoods of different sizes that are allowed to overlap. Starting from the largest neighborhood, the set of possible feature locations consists of 1 feature of size 24×24 and $2 \cdot 2 = 4$ features of size 23×23 down to $24 \cdot 24 = 576$ features of size 1×1. Hence, even for such small training images, $1 \cdot 1 + 2 \cdot 2 + \ldots + 24 \cdot 24 = 4900$ possible feature locations have to be evaluated. Despite the fact that the smaller neighborhoods provide only little context information and might be omitted, such feature sets can easily contain up to hundreds of thousands or millions of elements if multiple structures (e.g. edges at different directions) should be utilized per location and larger training examples are exploited.

The detector learning aims on identifying characteristic structures at specific locations with respect to the size of the training images. In this way, the size of the training images as well specifies the detector window that defines the spatial neighborhood that is analyzed in the detection process. A machine learning algorithm might evaluate the complete feature set on every training example in order to select the features utilized in the learned detector and to determine decision rules that classify the detector input with respect to the feature responses. Hence, the size

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

of the feature set is crucial for the computational costs and time required for the detector learning.

Figure 1.3a illustrates the detector learning process that has the goal to find features and associated decision rules that are able to split the training set in its positive and negative part with a preferably small error. A positive training example for faces is presented in the green frame while an example of the negative set is marked by a red border. The magenta structure in the detector window represents a feature that has been learned in a real detector training and demonstrates that the eye region is especially important for face detection. In the following, two additional features are learned and printed in blue and yellow. The detector learning is assumed to be finished after three learned features so that the bottom right frame presents the completed so-called *ensemble classifier*. This means that an ensemble of features and decision rules classifies together if the detector window contains or not contains an object.

The detection process is shown in Figure 1.3b. A so-called *sliding window* is applied to find all instances of the object class in an image. For this, the detector window is slided over the image and the ensemble classifier is evaluated on each position. In order to find objects of different sizes, the detector window has to be scaled as well. The green rectangle represents the position and scale of the detector window at which the learned ensemble classifier has found a face.

A frequent requirement on a learned classifier is its efficient computation allowing real-time detection. For this, it is very important that the utilized features can be efficiently calculated. Although a learned detector consists of significantly less features than present in the feature set, it has to be considered that these features have to be computed many times during the detection process. Getting back to the previous example of a face detector learning, that detector would operate with a detector window of the size 24×24 pixels. It can be also assumed that this matches the smallest scale on which the sliding window is applied. Supposed that the sliding window is shifted in steps of two pixels on this scale, the number of evaluated detector windows is $((512 - 24)/2)^2 = 59536$ solely on this scale for an image of size 512×512 pixels as presented in Figure 1.3b.

The strategy of learning from examples has the inherent difficulty that appropriate training and feature sets are crucial for the detection performance of the learned detector. The set of positive training examples should preferably comprise the complete variety of appearances of the object class. Likewise, the negative set is required to provide a huge diversity of non-objects that can occur in the detection process. The feature set has to represent the structures that are characteristic of the object class. But the determination of this intrinsic structures is the purpose of the learning algorithm so that it is in general difficult to restrict the feature set a priori to suitable features.

Another difficulty emerges from the high computational costs of the detector learning. The approach to alleviate the aforementioned problems by significantly expand-

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

ing the training and feature sets would quickly raise the computational costs such that the detector training becomes impractical in reasonable time.

The cascaded object detector proposed by Viola and Jones [145] that is learned by the Adaptive Boosting (AdaBoost) [56] algorithm has proofed to be a very efficient method in several applications. Because of its clear structure and high efficiency and hence enabled scope for enhancements, this work has been chosen to build upon the Viola and Jones detector.

Contributions

The structure of the thesis and its chapters follow the work flow of the developed object detection framework but for convenience the contributions are listed in a slightly different order.

The training of an object detector by a machine learning algorithm is as previously described in general a computational expensive task. The utilization of larger training sets tend to result in a higher performance of the learned detector. Likewise, the quality of a detector can often be improved by enlarging the feature set and in this way increasing the variability of available features. But both of these approaches also strongly raise the computational costs and the time required for the detector learning. This work pursues two strategies to respond to these disadvantages that are listed in the following together with the remaining contributions:

- First, many subtasks in the used machine learning algorithm AdaBoost can be performed in **parallel** so that a distributed machine learning framework is developed based on the Message Passing Interface (MPI) to utilize a huge amount of computing power. In this way, the overall computational costs are not decreased but the required time can be considerably reduced.
- Second, **feature mining** is performed to create customized feature pools that have a smaller size but provide a higher feature variability [44]. The saved computational costs and time can then be reinvested in an additional customized pool of complementary features to further increase the detection performance. Experiments show the benefit of a mixed learning of object detectors from two complementary feature types.
- In order to enrich the variety of shapes that the feature set resembles, a **novel class of features** is proposed that utilizes **space filling curves**, a type of fractals [46]. These fractals can be represented by special variants of integral images so that the new features can be smoothly incorporated into classifier learning together with rectangular Haar-like features [115].
- The problem of **sparse training data** is addressed that is not uncommon in learning highly specialized detectors for e.g. industrial production. Sparse training sets containing only a small amount of positive samples often result in poor classification performance if a classier is learned from these examples.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

From a model of the positive object space obtained by a Principal Component Analysis (PCA) [116], it can be observed that negative training images projected into the objects PCA space are often far away from the object class. This broad boundary between the object classes in training can yield to a high classification error of boosted classifiers on the test set. Hence, the basic idea of the contribution [43] is to narrow the boundary by augmenting the training set based on a obtained model.

- In order to find objects in an image, object detectors commonly apply a sliding window that evaluates the image at various different positions and scales. This frequently results in multiple detections of the same object at slightly shifted and scaled positions that have to be reduced to a single detections in a post-processing step denoted Non-Maximum Suppression (NMS). A method for **merging multiple detections** is proposed [45] that utilizes Dempster-Shafer Theory of Evidence (DS) [38, 137] to **exploit confidences** obtained from internal variables of the cascaded detector. The evidence theory combines in the process confidence and uncertainty information to compute an overall confidence for detections that gives an appropriate measure to distinguish the reliability of detections. In a second step, this confidence measure is employed to improve the accuracy of a merged object's position.

Structure of the Thesis

Figure 1.4 presents an overview on the structure of the thesis. The contents of the individual chapters are summarized in the following:

Chapter 2: The chapter starts with an overview of related work in the context of visual object detection using machine learning techniques. An introduction is given on different strategies that are pursued to obtain features which represent the visual appearance of objects. Then, concepts are discussed to learn classifiers from examples by means of these features. Chapter 2 ends with an introduction of the data sets and benchmarks that are utilized in this work and a description of their specific properties.

Chapter 3: In this chapter, fundamentals of the thesis are introduced. A detailed description of feature types is given that are well established in the task of visual object detection and are taken as a basis for the feature mining methods proposed in Chapter 7. The AdaBoost algorithm and the Viola and Jones detector are introduced from which the object detection framework of this work is evolved. Unsupervised methods for data analysis used in this thesis are described, namely cluster algorithms and the PCA. At the end of the chapter, common performance measures for object detectors are briefly discussed.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 4: This chapter presents the distributed machine learning framework that has been developed to utilize the huge amount of computing power that is more than helpful for learning capable object detectors. The structure and parallel work flow of the framework is briefly described. Further on, the requirements on the software architecture are discussed and an insight in the implementation of the framework is given.

Chapter 5: The problem of sparse training data is addressed that is not uncommon in learning highly specialized detectors for e.g. industrial production. A method to augment the training data is developed that rests upon a statistical analysis of the positive object class. Chapter 5 is based on an already published conference article that appeared in [43].

Chapter 6: In this chapter, a new class of features is introduced that utilizes fractals to resemble a larger variety of shapes. Fractal structures are generated by space filling curves that are incorporated into the objection detection framework by special variants of integral images. This method has been published in a conference article that appeared in [46].

Chapter 7: Feature mining approaches are introduced in Chapter 7 that aim at the construction of more effective features while reducing the computational costs of the classifier learning. Two complementary feature types are developed that are customized to the object class by a generic approach and thus exploit on the one hand coarse structures and on the other hand fine details. The obtained sets of customized features provide a high variability in feature shapes but are comparatively small and enable a fast learning. Parts of Chapter 7 have been published in a conference article [44].

Chapter 8: This chapter presents an improvement of the Non-Maximum Suppression that is commonly performed in a post-processing step of object detectors. A method for merging multiple detections is described that utilizes Dempster-Shafer Theory of Evidence to combine confidence and uncertainty information obtained from the cascaded detector to compute a confidence measure for detections that is applied to refine the determined object position. Chapter 8 is based on previously published conference article that appeared in [45].

Chapter 9: The thesis is concluded and the main contributions are summarized.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

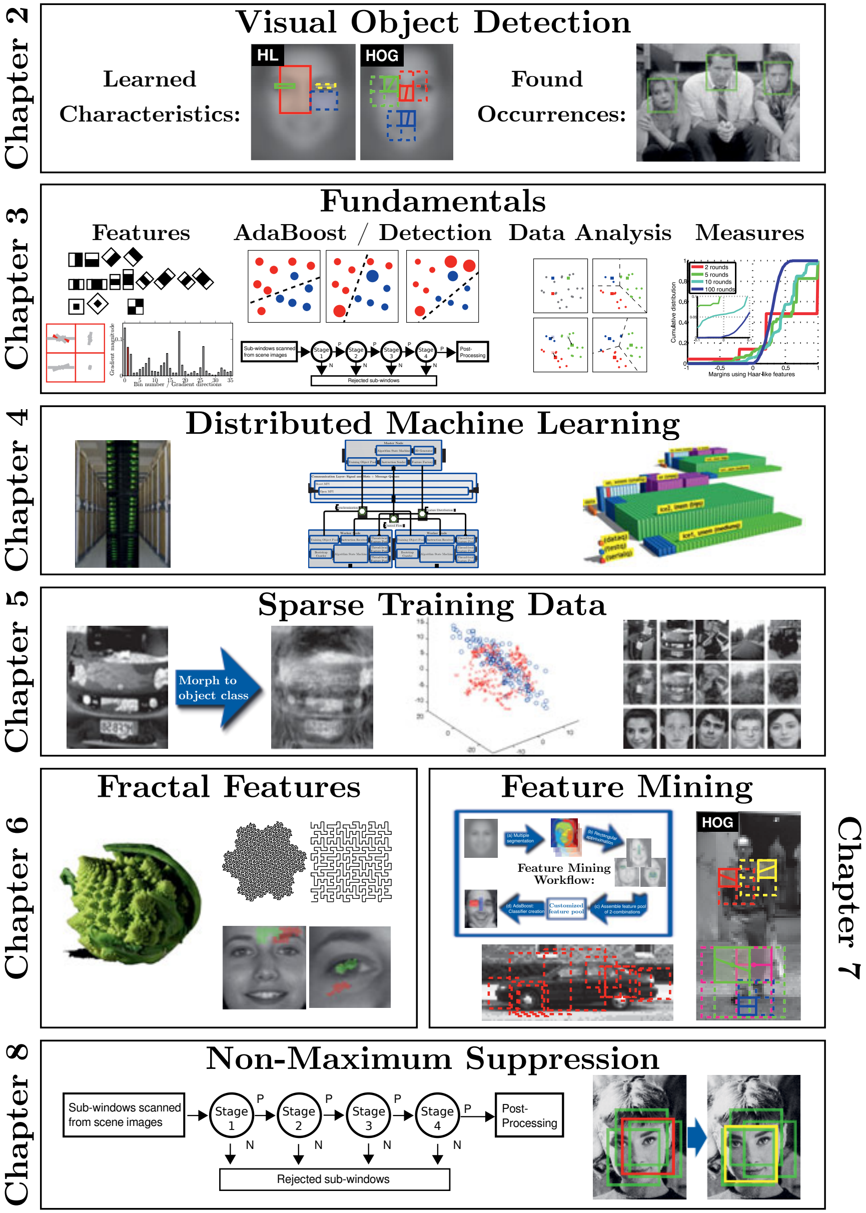


Figure 1.4: Structure of the thesis.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Papers of the Author

Papers of the author that are relevant to main contributions of the thesis:

[43] **Arne Ehlers** and Florian Baumann and Ralf Spindler and Birgit Glasmacher and Bodo Rosenhahn, *PCA Enhanced Training Data for Adaboost*, 14th International Conference on Computer Analysis of Images and Patterns (CAIP), 2011

In this paper we propose to enhance the training data of boosting-based object detection frameworks by the use of principal component analysis (PCA). The quality of boosted classifiers highly depends on the image databases exploited in training. We observed that negative training images projected into the objects PCA space are often far away from the object class. This broad boundary between the object classes in training can yield to a high classification error of the boosted classifier in the testing phase. We show that transforming the negative training database close to the positive object class can increase the detection performance. In experiments on face detection and the analysis of microscopic cell images, our method decreases the amount of false positives while maintaining a high detection rate. We implemented our approach in a Viola & Jones object detection framework using AdaBoost to combine Haar-like features. But as a preprocessing step our method can easily be integrated in all boosting-based frameworks without additional overhead.

[44] **Arne Ehlers** and Florian Baumann and Bodo Rosenhahn, *Exploiting Object Characteristics using Custom Features for Boosting-based Classification*, 18th Scandinavian Conference on Image Analysis (SCIA), 2013

Typical feature pools used to train boosted object detectors contain various redundant and unspecific information which often yield less discriminative detectors. In this paper we introduce a feature mining algorithm taking domain specific knowledge into account. Our proposed feature pool contains rectangular shaped features generated from an image clustering algorithm applied on the mean image of the object training set. A combination of two such spatially separated rectangular regions yields a set of features which have a similar evaluation time like classical Haar-like features, but are much smarter (automatically) selected and more discriminative since image correlations can be more consequently exploited. Overall, training is faster and results in more selective detectors showing improved precision. Several experiments demonstrate the gain when using our proposed feature set in contrast to standard features.

[45] **Arne Ehlers** and Björn Scheuermann and Florian Baumann and Bodo Rosenhahn, *Cleaning Up Multiple Detections Caused by Sliding Window Based Object Detectors*, 18th Iberoamerican Congress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP), 2013

Object detection is an important and challenging task in computer vision. In

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

cascaded detectors, a scanned image is passed through a cascade in which all stage detectors have to classify a found object positively. Common detection algorithms use a sliding window approach, resulting in multiple detections of an object. Thus, the merging of multiple detections is a crucial step in post-processing which has a high impact on the final detection performance. First, this paper proposes a novel method for merging multiple detections that exploits intra-cascade confidences using Dempster's Theory of Evidence. The evidence theory allows hereby to model confidence and uncertainty information to compute the overall confidence measure for a detection. Second, this confidence measure is applied to improve the accuracy of the determined object position. The proposed method is evaluated on public object detection benchmarks and is shown to improve the detection performance.

[46] **Arne Ehlers** and Florian Baumann and Bodo Rosenhahn, *Boosted Fractal Integral Paths for Object Detection*, 10th International Symposium on Visual Computing (ISVC), 2014

In boosting-based object detectors, weak classifiers are often build on Haar-like features using conventional integral images. That approach leads to the utilization of simple rectangle-shaped structures which are only partial suitable for curved-shaped structures, as present in natural object classes such as faces. In this paper, we propose a new class of fractal features based on space-filling curves, a special type of fractals also known as Peano curves. Our method incorporates the new feature class by computing integral images along these curves. Therefore space-filling curves offer our proposed features to describe a wider variety of shapes including self-similar structures. By introducing two subtypes of fractal features, three-point and four-point features, we get a richer representation of curved and topology separated but correlated structures. We compare AdaBoost using conventional Haar-like features and our proposed fractal feature class in several experiments on the well-known MIT+CMU upright face test set and a microscopy cell test set.

Other papers of the author resulting from side research projects:

[12] Florian Baumann and Katharina Ernst and **Arne Ehlers** and Bodo Rosenhahn, *Symmetry Enhanced Adaboost*, 6th International Symposium on Visual Computing (ISVC), 2010

This paper describes a method to minimize the immense training time of the conventional Adaboost learning algorithm in object detection by reducing the sampling area. A new algorithm with respect to the geometric and accordingly the symmetric relations of the analyzed object is presented. Symmetry enhanced Adaboost (SEAdaboost) can limit the scanning area enormously, depending on the degree of the objects symmetry, while it maintains the detection rate. SEAdaboost allows to take advantage of the symmetric characteristics of an object by concentrating on corresponding symmetry features

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

during the detection of weak classifiers. In our experiments we gain 39 % reduced training time (in average) with slightly increasing detection rates (up to 2.4 % and up to 6 % depending on the object class) compared to the conventional Adaboost algorithm.

[131] Björn Scheuermann and **Arne Ehlers** and Hamon Riazy and Florian Baumann and Bodo Rosenhahn, *Ego-Motion Compensated Face Detection on a Mobile Device*, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011

In this paper we propose face tracking on a mobile device by integrating an inertial measurement unit into a boosting based face detection framework. Since boosting based methods are highly rotational variant, we use gyroscope data to compensate for the camera orientation by virtual compensation of the camera ego-motion. The proposed fusion of inertial sensors and face detection has been tested on Apple's iPhone 4. The tests reveal that the proposed fusion provides significant better results with only minor computational overhead compared to the reference face detection algorithm.

[13] Florian Baumann and **Arne Ehlers** and Karsten Vogt and Bodo Rosenhahn, *Cascaded Random Forest for Fast Object Detection*, 18th Scandinavian Conference on Image Analysis (SCIA), 2013

A Random Forest consists of several independent decision trees arranged in a forest. A majority vote over all trees leads to the final decision. In this paper we propose a Random Forest framework which incorporates a cascade structure consisting of several stages together with a bootstrap approach. By introducing the cascade, 99 % of the test images can be rejected by the first and second stage with minimal computational effort leading to a massively speeded-up detection framework. Three different cascade voting strategies are implemented and evaluated. Additionally, the training and classification speed-up is analyzed. Several experiments on public available datasets for pedestrian detection, lateral car detection, and unconstrained face detection demonstrate the benefit of our contribution.

[16] Florian Baumann and Jie Liao and **Arne Ehlers** and Bodo Rosenhahn, *Motion Binary Patterns for Action Recognition*, 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2014

In this paper, we propose a novel feature type to recognize human actions from video data. By combining the benefit of Volume Local Binary Patterns and Optical Flow, a simple and efficient descriptor is constructed. Motion Binary Patterns (MBP) are computed in spatio-temporal domain while static object appearances as well as motion information are gathered. Histograms are used to learn a Random Forest classifier which is applied to the task of human action recognition. The proposed framework is evaluated on the well-known, publicly available KTH dataset, Weizman dataset and on the IXMAS dataset

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

for multi-view action recognition. The results demonstrate state-of-the-art accuracies in comparison to other methods.

[14] Florian Baumann and **Arne Ehlers** and Bodo Rosenhahn and Jie Liao, *Computation Strategies for Volume Local Binary Patterns applied to Action Recognition*, 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2014

Volume Local Binary Patterns are a well-known feature type to describe object characteristics in the spatio-temporal domain. Apart from the computation of a binary pattern further steps are required to create a discriminative feature. In this paper we propose different computation methods for Volume Local Binary Patterns. These methods are evaluated in detail and the best strategy is shown. A Random Forest is used to find discriminative patterns. The proposed methods are applied to the well-known and publicly available KTH dataset and Weizman dataset for single-view action recognition and to the IXMAS dataset for multi-view action recognition. Furthermore, a comparison of the proposed framework to state-of-the-art methods is given.

[15] Florian Baumann and Fangda Li and **Arne Ehlers** and Bodo Rosenhahn, *Thresholding a Random Forest Classifier*, 10th International Symposium on Visual Computing (ISVC), 2014

The original Random Forest derives the final result with respect to the number of leaf nodes voted for the corresponding class. Each leaf node is treated equally and the class with the most number of votes wins. Certain leaf nodes in the topology have better classification accuracies and others often lead to a wrong decision. Also the performance of the forest for different classes differs due to uneven class proportions. In this work, a novel voting mechanism is introduced: each leaf node has an individual weight. The final decision is not determined by majority voting but rather by a linear combination of individual weights leading to a better and more robust decision. This method is inspired by the construction of a strong classifier using a linear combination of small rules of thumb (AdaBoost). Small fluctuations which are caused by the use of binary decision trees are better balanced. Experimental results on several datasets for object recognition and action recognition demonstrate that our method successfully improves the classification accuracy of the original Random Forest algorithm.

[17] Florian Baumann and Liu Wei and **Arne Ehlers** and Bodo Rosenhahn, *Sequential Boosting for Learning a Random Forest Classifier*, IEEE Winter Conference on Applications of Computer Vision (WACV), 2015

This paper introduces a novel tree induction algorithm called sequential Random Forest (sRF) to improve the detection accuracy of a standard Random Forest classifier. Observations have shown that the overall performance of a forest is strongly influenced by the number of training samples. The main idea is to sequentially adapt the number of training samples per class so that each

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

tree better complements the existing trees in the whole forest. Further, we propose a weighted majority voting with respect to a class and tree specific error rate for decreasing the influence of poorly performing trees. The sRF algorithm shows competing results in comparison to state-of-the-art approaches using two datasets for object recognition, two standard machine learning datasets, and three datasets for human action recognition.

[18] Florian Baumann and Karsten Vogt and **Arne Ehlers** and Bodo Rosenhahn, *Probabilistic Nodes for Modelling Classification Uncertainty for Random Forest*, 14th IAPR International Conference on Machine Vision Applications, (MVA), 2015

In this paper, we propose to enhance the original Random Forest algorithm by introducing probabilistic nodes. Platt Scaling is used to interpret the decision of each node as a probability and was initially developed for calibrating Support Vector Machines. Nowadays it is used to calibrate the output probabilities of decision trees, boosted trees, or Random Forest classifiers. In comparison to these approaches, we integrate the Platt Scaling calibration method into the decision process of every node within the ensemble of decision trees. Regarding the original Random Forest, the nodes serve as a guide to predict the path through the tree until reaching a leaf node. In this paper, we interpret the decision as a probability and incorporate more information into the decision process. The proposed approach is evaluated using two well-known machine learning datasets as well as object recognition datasets.

[123] Christoph Reinders and Florian Baumann and Björn Scheuermann and **Arne Ehlers** and Nicole Mühlpforte and Alfred Effenberg and Bodo Rosenhahn, *On-The-Fly Handwriting Recognition using a High-Level Representation*, 16th International Conference on Computer Analysis of Images and Patterns (CAIP), 2015

Automatic handwriting recognition plays a crucial role because writing with a pen is the most common and natural input method for humans. Whereas many algorithms detect the writing after finishing the input, this paper presents a handwriting recognition system that processes the input data during writing and thus detects misspelled characters on the fly from their origin.

The main idea of the recognition is to decompose the input data into defined structures. Each character can be composed out of the structures point, line, curve, and circle. While the user draws a character, the digitized points of the pen are processed successively, decomposed into structures, and classified with the help of samples. The intermediate classification allows a direct feedback to the user as soon as the input differs from a given character.

[19] Florian Baumann and Jie Liao and **Arne Ehlers** and Bodo Rosenhahn, *Recognizing Human Actions using novel Space-time Volume Binary Patterns*, Neurocomputing Journal, 2016

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In this paper, we propose a novel feature type, namely Motion Binary Pattern (MBP) and different computation strategies for the well-known Volume Local Binary Pattern (VLBP). MBPs are a combination of VLBPs and Optical Flow. By combining the benefit of both methods, a simple and efficient descriptor is constructed. Motion Binary Patterns are computed in the spatio-temporal domain while the motion in consecutive frames is described. Finally, a feature descriptor is constructed by a histogram computation. Volume Local Binary Patterns are a feature type to describe object characteristics in the spatio-temporal domain. But apart from the computation of such a pattern further steps are required to create a discriminative feature. These steps are evaluated in detail and the best strategy is shown. For MBPs and VLBPs, a Random Forest classifier is learned and applied to the task of human action recognition. The proposed novel feature type and VLBPs are evaluated on the well-known, publicly available KTH dataset, Weizman dataset and on the IXMAS dataset for multi-view action recognition. The results demonstrate challenging accuracies in comparison to state-of-the-art methods.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 2

Related Work and Data Sets

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

This chapter begins with an introduction in feature provision strategies for learning visual object detectors. An overview on machine learning techniques is presented to explain the context of the methods proposed in this thesis. Further on, the data sets and benchmarks are introduced that are utilized in this work to learn object detectors based on the proposed contributions and to evaluate their performance.

2.1 Machine Learning for Visual Object Detection

The creation of classifiers for visual object detection is commonly performed by learning from examples. For this, a machine learning algorithm is provided with labeled training data that consists of examples of the objects to distinguish and assigned class labels. The task of the machine learning algorithm is to learn the underlying concept or characteristics of the object classes from the training examples in order to distinguish between them.

The object classes are often not directly learned from the observations (e.g. images in case of visual object detection) but with the help of features that give the learning algorithm a guiding principle on how to determine their characteristics in order to reduce the learning effort and the required amount of training data. This principle, implied by the features, might be e.g. to analyze edges [126] or to exploit intensity differences between image regions [145]. Hence, features are essentially functions that, given some feature parameter, map from the domain of observations into the codomain of feature values, the *feature space*. The goal of the machine learning algorithm is then to find decision boundaries in the feature space to distinguish between the object classes.

As illustrated by an example in Chapter 1, sets of more complex features can swiftly reach tremendous sizes if any possible feature manifestation is considered. So this would consequently lead to constraints reducing the variability of a feature type because very huge feature sets cannot be processed during classifier learning in reasonable time. But this problem as well gives the motivation for the development of different strategies to create and provide feature pools that are small but offer a huge variety of suitable features.

2.1.1 Feature Provision

The provision of features that are appropriate for the specific object class can have a large impact on the achievable performance of a detection system. A variety of strategies exist for feature provision to which four categories are specified as follows:

Deep learning Feature extraction is completely incorporated into the machine learning process by deep learning. High-level features are thus evolved from low-level by way of mid-level features. Recently, Convolutional Neural Networks (CNNs)

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

performing deep learning show great success in many applications since the required tremendous computational resources and training data are available. Examples for deep learning methods to which a performance comparison is given in Section 7.3 are in the domain of face detection [93, 52] and for pedestrian detection [136].

Expert feature design A contrary strategy is pursued by expert feature design. Human experts thereby develop a model of the object class and features appropriate to that specific domain. This approach is frequently taken in domains where especially an efficient and computationally inexpensive object detector is desired. An example of such a domain and approach is pedestrian detection in which Zhang et al. [159] recently present state-of-the-art performance.

Feature selection Dollár et al. [40] address two related research areas. *Feature selection* methods that create a subset by choosing reasonable features from a large feature pool and *feature mining*, i.e. deriving a model of an own entire feature space. Feature selection strategies are divided by Blum et al. [23] into three categories: (i) *Embedded methods*, such as AdaBoost [56] in which the feature selection mechanism is incorporated into the machine learning algorithm. (ii) *Filter methods* assign a score to features in order to filter out irrelevant information. In this way, Koller and Sahami [81] propose a method for selecting a subset of features by eliminating features providing little or no additional information. More recently, Appel et al. [7] develop a method for pruning low-performing features for speed-up during Boosting of decision trees that guarantees identical performance to a classical training. (iii) *Wrapper methods* evaluate the quality of a feature subset by the performance of a classifier learned from them.

Feature mining But Guyon and Elisseeff [66] argue that extracting a feature pool directly from the data set usually yields to a better performance and a more discriminative power. Feature mining addresses this task of exploiting domain knowledge in constructing a customized feature set.

The methods in the previously specified categories are not mutually exclusive such that [44] applies AdaBoost as an embedded feature selection method to a feature pool constructed by feature mining.

Following these lines of research, Chapter 7 proposes a generic approach for different object domains that incorporates feature mining strategies to construct a customized mixed pool of complementary features that are able to utilize coarse object characteristics as well as fine object details.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

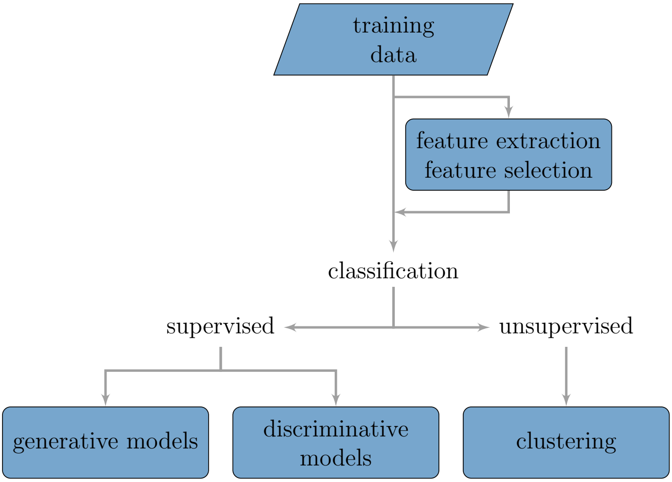


Figure 2.1: Overview on categories in machine learning techniques.

2.1.2 Learning Algorithms

A good flowchart defining categories for methods in pattern recognition is presented by Webb in [151, p. 28]. Figure 2.1 shows a simplified variant of the flowchart that is adjusted to give an overview on categories for machine learning methods used in this thesis. Classification approaches are divided into two categories:

- *Unsupervised* methods exploit the training data without the prior knowledge of associated class labels. These methods analyze the training data or a feature space derived from it to create a partition of clusters. Section 3.3.1 briefly introduces several clustering algorithms that are utilized in this work.
- *Supervised* methods learn from labeled training data so that a class label is assigned to each example in the training data specifying its class membership. These methods learn the unknown concept of the classes from their assigned training examples. After learning, the algorithms predict for an unseen example to which of the predefined classes it belongs.

Supervised methods can be further divided into *generative* and *discriminative* models.

- Generative approaches aim to learn a model of the observations given by the training data and the associated class labels. The joint probability $p(\mathbf{x},y) = p(\mathbf{x}|y) \cdot p(y)$ of the training samples $\mathbf{x}$ and the labels y is estimated. They are generative because synthetic data points in the input space can be generated by sampling from the model. The class label of unseen data is predicted by

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

using Bayes rules to compute the posterior probabilities

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y) \cdot p(y)}{p(\mathbf{x})}$$

and selecting the class label y having the highest probability. The probability $p(\mathbf{x})$ for input data to occur is class independent in doing so and hence irrelevant for classification. The joint probabilities are typically estimated by choosing a model for the class densities and optimizing its parameter to fit to the observations. Examples of supervised, generative models are Naive Bayes classifier [125], Gaussian Mixture Models (GMMs) [36], and Hidden Markov Models (HMMs) [11].

- Discriminative models learn to distinguish between object classes by directly modeling the posterior probabilities $p(y|\mathbf{x})$ from the training examples. Examples of supervised, discriminative approaches are Support Vector Machines (SVMs) [32], Neural Networks [106], Random Forests [26], and AdaBoost.

As described in Section 2.1.1, AdaBoost is categorized as an embedded method that also involves the *feature selection* block presented in Figure 2.1. Although not used in the context of features in this work, the PCA introduced in Section 3.3.2 can be categorized as a *feature extraction* method or unsupervised learning.

Deep learning approaches, briefly described in Section 2.1.1, that are often performed by CNNs bypass the feature selection / feature extraction block and work directly on the training data.

This work applies besides cluster algorithms and PCA mainly AdaBoost as supervised, discriminative method for classifier learning. AdaBoost is in the following briefly compared with the related Bootstrap Aggregating (Bagging) [25] and Random Forest methods:

Adaptive Boosting The AdaBoost algorithm creates an *ensemble classifier*, i.e. a combination of multiple classifiers that give a collective decision. In a repetitive procedure, AdaBoost adds classifiers to the ensemble that minimize a training error with respect to a distribution on the training set that is maintained by the algorithm. The distribution is adapted after each round based on the classification result of the currently added classifier giving higher influence to misclassified training examples. In this way, the algorithm prefers in the following rounds classifiers that correctly classify these examples and consequently concentrates on harder training examples. The final learned ensemble of classifiers gives a weighted majority decision in which the influence of each classifier is based on its achieved training error. A detailed description of AdaBoost is given in Section 3.2.1.

Bootstrap Aggregating Bagging proposed by Breiman [25] in 1996 is as well as AdaBoost a meta-algorithm for machine learning in the sense that a learning

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

method is performed several times to create an ensemble classifier. Bagging repetitively generates new smaller training sets of the same size by uniformly sampling with replacement from the original training set. The sampling procedure ensures that the generated training sets, denoted bootstrap samples, are independent and identically distributed (i.i.d.). In this way, multiple versions of a predictor can be learned from the bootstrap samples and combined in an aggregated predictor showing improved accuracy. When predicting a class, the results of the multiple classifiers are aggregated in a plurality vote.

Random Forest In 2001, Breiman [26] proposes Random Forests that combine the concept of Bagging with the methods of Ho [71, 72] and Amit [5] to learn decision trees from randomly selected features. A Random Forest is an ensemble of decision trees that are learned from different bootstrap samples.

In addition, each tree is grown in a procedure that utilizes randomly selected subsets of the complete feature set. Until a stopping criteria is fulfilled or every leaf node of the tree is pure (i.e. all training examples assigned to the node are of the same class), a tree is grown by selecting a random feature subset of the same size for each non-pure leaf node and finding for each of these nodes the feature in the subset that achieves the best split of the assigned training examples. For this, a splitting function measures the level of impurity of the split. Each current non-pure leaf node is then divided into two child nodes and its assigned training examples are split according to the found feature and passed to its child nodes that are the new leaf nodes to process.

The Random Forest classifies in a plurality vote in which each tree votes for its most likely class with equal weight.

In own works, Baumann et al. [13, 15, 17, 18] propose to improve the performance of Random Forests classifiers by incorporating concepts from Boosting algorithms. Short introductions in the developed methods are given by the abstracts of papers of the author presented in Chapter 1.

2.2 Data Sets and Benchmarks

In the following, a brief description of the data sets is given that are utilized in this thesis. A distinction has to be made between pure data sets, that only provide images of an object class, and benchmarks that contain a test set with corresponding ground truth information for evaluation. Some data sets include as well a negative training set of non-object images and benchmarks can also contain a training set. Benchmarks may provide an evaluation tool for an uniform detector assessment or define a evaluation methodology. Table 2.1 gives an overview of the properties of the utilized data sets.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 2.1: Properties of the utilized data sets. The number of positive and negative training examples is presented as well as the properties of the provided test sets.

Data set	Training		Testing		Evaluation tool
	#Pos	#Neg	#Images	#Objects	
AT&T faces	400	-	-	-	no
BioID faces [22]	1521	-	-	-	no
MUCT faces [110]	3755	-	-	-	no
GENKI-4K faces [1]	4000	-	-	-	no
MIT+CMU faces [141]	-	-	130	511	no
FDDB faces [74]	-	-	2845	5171	yes
Cryo cells	250	350	-	-	no
UIUC cars [3]	550	500	108	139	yes
Daimler pedestrians [47]	15660	6744 [1]	21790	56492	yes [2]

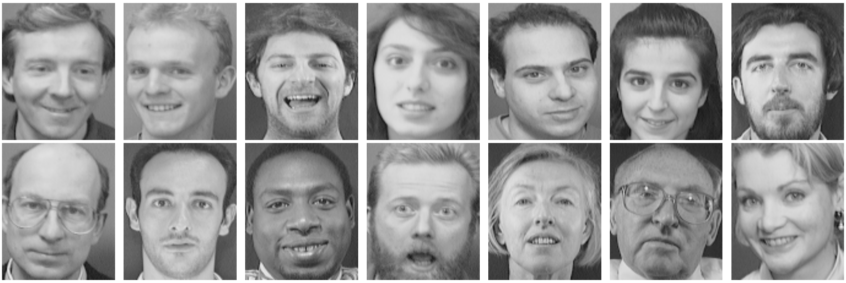


Figure 2.2: Examples of the AT&T face data set. Credits are given to AT&T Laboratories Cambridge.

AT&T Face Database The AT&T face database contains ten different images each of 40 people varying in the lighting, facial expressions, and facial details (glasses / no glasses). The images were taken against a dark homogeneous background with the people in an upright, frontal position.

The field of view of the shown faces is fairly homogeneous such that no additional alignment is performed. Credits are given to AT&T Laboratories Cambridge. Figure 2.2 presents some example images taken from the database.

BioID Face Database The BioID face database [22] consists of 1521 gray images presenting 23 different test persons. The data set shows a variety of illumination,

---

[1] The negative training set consists of scene image containing no objects such that it can be used to extract negative training samples.

[2] The Daimler benchmark itself does not provide an evalution tool but the tool of the CalTech benchmark [42] has been extended to evaluate it.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

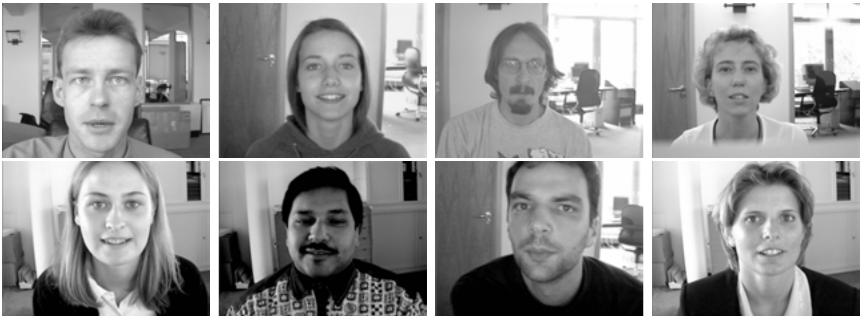


Figure 2.3: Example images taken from the BioID face database [22]

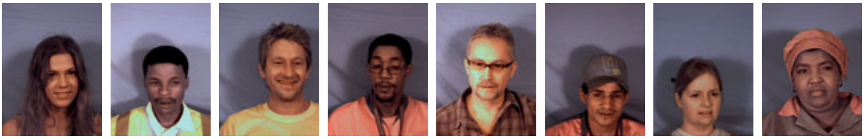


Figure 2.4: Example images of the MUCT face data set [110]

background, and face size. It provides as well manually marked landmarks including eye coordinates such that a automatic process can align the images to the eyes for adding them to a face detection training set. Figure 2.3 shows some example images of the BioID database.

MUCT Landmarked Face Database The MUCT landmarked face database [110] contains 3755 faces with 76 manual landmarks. The images present 276 persons of varying age and ethnicity and are acquired under ten different lightning setups from five view angles. For the application in the face training set, the faces are automatically aligned to the provided eye landmarks. Figure 2.4 presents example images of the MUCT face database.

MPLAB GENKI Database, GENKI-4K Subset The MPLAB GENKI database, GENKI-4K subset [1] contains 4000 images of faces under different facial expressions, lightning conditions, and geographical locations. All images are manually aligned to the eye positions before adding them along with a mirrored copy to a face training set. Hence, the positive face training set derived from the GENKI database consists of 8000 face images. Examples taken from the GENKI-4K subset are shown in Figure 2.5.

MIT+CMU Frontal Face Dataset The MIT+CMU frontal face dataset A+C [141] consists of 130 gray-scale images containing 511 faces. The data set provides

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

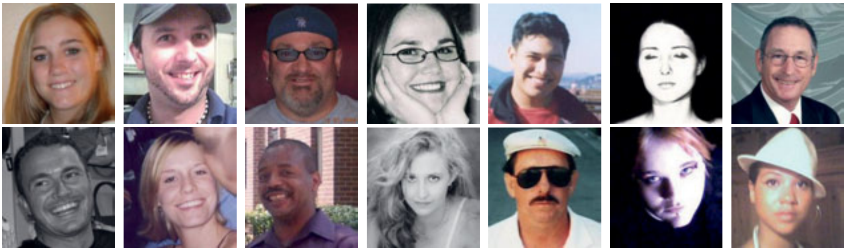


Figure 2.5: Example images of the MPLab GENKI Database, GENKI-4K Subset [1].

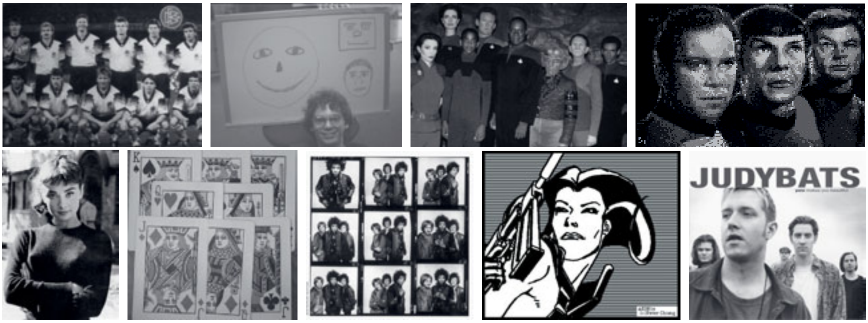


Figure 2.6: Example images taken from MIT/CMU frontal face data sets A,C [141].

ground truth coordinates of the contained faces in form of six facial landmarks including the eyes. The eye coordinates and distance is utilized to compute the position and scale of face bounding boxes that are matched against detector bounding boxes to evaluate detection performance. The MIT+CMU data set does not provide an evaluation tool. A Receiver Operating Characteristic (ROC) curve for evaluation is thus created by adjusting an internal threshold controlling the detector's selectivity. Despite its age, this test set is still challenging. The image database is partially noisy and blurred and contains several difficult samples like low-resolution images, comics, line drawings, and a binary raster image. Figure 2.6 presents some example images of the data set.

Face Detection Data Set and Benchmark The FDDB face detection benchmark [74] consists of 2845 images containing 5171 faces. An evaluation tool to measure the detection performance is provided as well. The evaluation procedure requires multiple detections to be merged to a single detection in advance and a confidence value has to be assigned. A performance curve is then generated by traversing all confidence values and calculating the True Positive Rate (TPR) and total False Positives considering only merged detections that have a confidence that

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

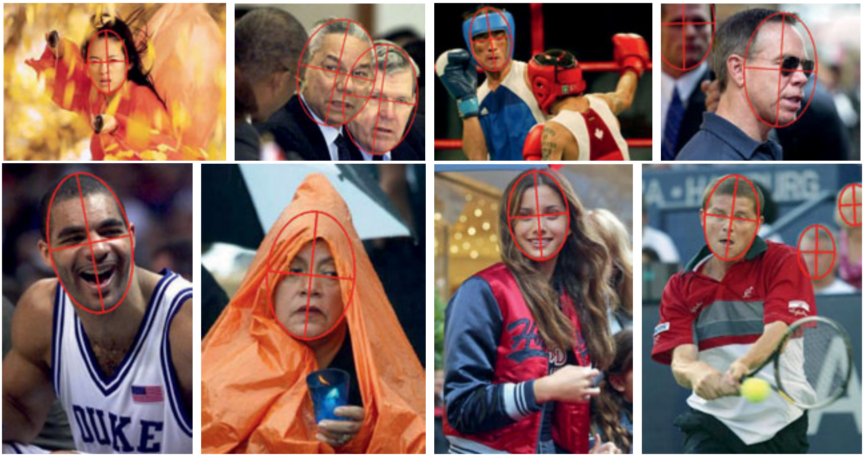


Figure 2.7: Example images of the Face Detection Data Set and Benchmark (FDDB) with visualized face ground truth. The images are taken from the project website [74].

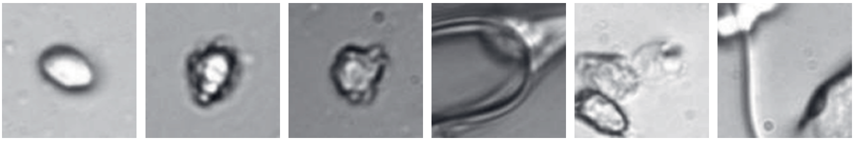


Figure 2.8: Examples of the cryo cell data set. The first three images show unimpaired cells while the last three images present non-cells or corrupted cells [43, 46].

is greater or equal than the currently selected confidence. The authors supply results on the project web page [74] for different face detectors generated by this tool in form of files that contain the point coordinates of the performance curves. Figure 2.7 presents some example images in which the provided ground truth face positions are visualized.

Cryo Cell Data Set The cell data set is acquired during cryo-conservation and as a result ice fronts are forming around the cells. The goal is to detect (and track) the cells in videos. 250 images of cells and 350 images of non-cells are collected to gain a reasonable database. Figure 2.8 shows some examples of this data set.

UIUC Image Database for Car Detection The UIUC image database for car detection [3] consists of 1050 gray-scale training images containing 550 car and 500 non-car images. This database also provides test images and an evaluation tool

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

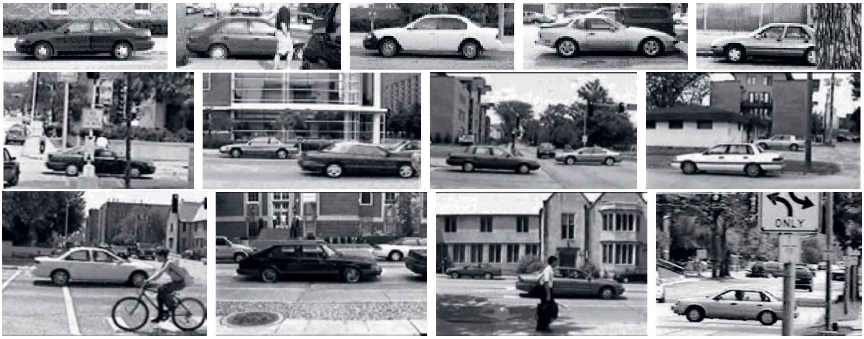


Figure 2.9: The UIUC dataset for lateral car detection [3]. The top row shows examples of the positive training set. The other images are taken from the multi-scale test set.

to automatically calculate precision and recall of an applied detector. To create a complete performance curve, an internal threshold of the detector controlling its selectivity is adjusted multiple times followed by an evaluation of the detector output by the provided tool. The multi-scale test set is evaluated that consists of 108 test images that contain 139 cars at different scales. Example images of the positive training set and the multi-scale test set are presented in Figure 2.9.

Daimler Monocular Pedestrian The Daimler mono pedestrian detection benchmark dataset [47] offers a training set as well as a test set to evaluate the detection performance. Example images are shown in Figure 2.10. The positive training set consists of 15660 gray-scale pedestrian images. 6744 scene images that do not contain pedestrians are provided to bootstrap the negative training set. The Daimler mono pedestrian benchmark data set provides as well a test set of a 27-minute drive through urban traffic captured from a moving vehicle. It is composed of 21790 images and 56492 labeled bounding boxes of comprised pedestrians. This test set can be evaluated as well by the Caltech pedestrian detection benchmark [42]. But the authors of the Caltech benchmark change the evaluation methodology of the Daimler test set and provide adapted pedestrian annotations. A discussion of the selected evaluation methodology is given in [42] and details of the changes to the Daimler test set can be found on the project web page [42]. In order to present a comparison to several other methods, the evaluation by the Caltech benchmark is selected in this thesis. Example images of the positive training set and the test set are presented in Figure 2.10.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

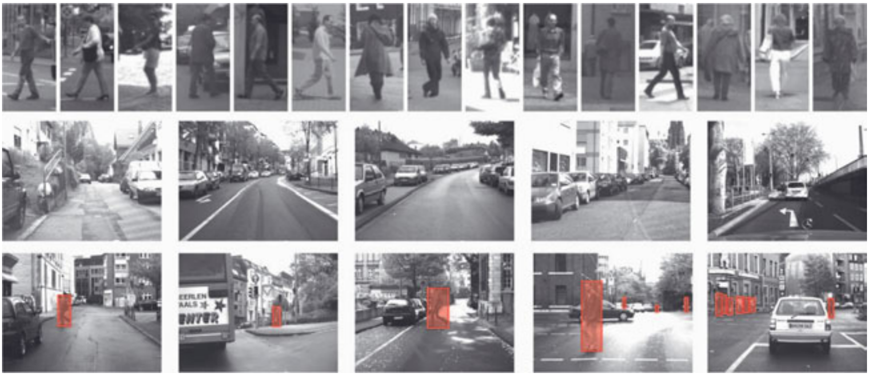


Figure 2.10: Example images of Daimler mono pedestrian detection benchmark. The top row shows examples of the positive training set. The other images are from the test set of a 27-minute drive through urban traffic. The images are taken from [47].

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 3

Fundamentals

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

This chapter presents fundamentals the thesis builds on. Common feature types are introduced that are well established in machine learning for visual object detection. The supervised machine learning algorithm AdaBoost and the Viola and Jones object detection framework are described in detail. Unsupervised methods for data analysis are presented by means of cluster algorithms and the Principal Component Analysis (PCA). Further, an introduction in performance measures for detectors is given.

3.1 Common Features

In this section, two feature types are introduced that demonstrate good performance and are widely used in several object detection tasks. The Haar-like features and Histograms of Oriented Gradients (HOG) descriptors are explained as well as methods for their efficient computation using the integral image representation.

3.1.1 Haar-like Features

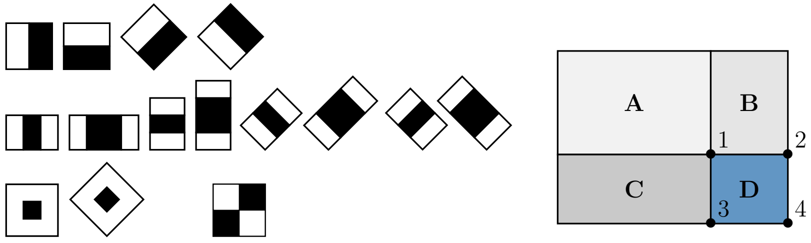
A very popular feature class are Haar-like features that, inspired by Haar basis functions, have been introduced by Papageorgiou et al. [115]. Haar-like features represent rectangular structures such that the feature is computed as the difference of the sum of pixel intensities in two coherent rectangular regions. Commonly used templates of Haar-like features consist of two, three, or four rectangles. The positively and negatively weighted rectangular regions have the same size in many of these templates. If the sizes differ, the sum of pixel intensities have to be multiplied by a factor to compensate for the difference. These feature templates are scaled independently in both dimensions and shifted in the detection window to form an overcomplete feature set. Lienhart and Maydt [96] enriched conventional Haar-like feature sets by templates that are rotated by 45°. Figure 3.1a presents, except for the last template, the Haar-like features that are defined by Lienhart and Maydt [96] and are as well utilized in the face detector of Open Source Computer Vision library (OpenCV) [24]. The last template in Figure 3.1a is defined by Viola and Jones [145] as *four-rectangle-feature*.

Integral images One reason for the success of Haar-like features is their very efficient computation using integral images as intermediate image representation [145]. An integral image I_{Int} is computed in a pre-processing step from an image I by summing up its pixel intensities from top left to bottom right:

$$I_{\text{Int}}(u,v) = \sum_{u'=0}^{u'\leq u} \sum_{v'=0}^{v'\leq v} I(u',v') \tag{3.1}$$

The sum of pixel intensities of an arbitrary upright rectangular image region, as required in the computation of conventional Haar-like features, can then be deter-

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b)

Figure 3.1: (a) Haar-like feature templates that are, except for the last template, defined by Lienhart and Maydt [96] and are as well utilized in the OpenCV [24] face detector. The last template is the *four-rectangle-feature* defined by Viola and Jones [145]. A Haar-like feature is computed by subtracting the sum of pixel intensities of the white rectangles from the sum of pixel intensities belonging to black rectangles. (b) Computing the sum of pixel intensities of arbitrary rectangular areas in integral images. The pixel sum in rectangular region **D** in the original image can be calculated from the pixel values at the corner points 1 to 4 in the corresponding integral image.

mined by accessing only four points in the integral image. Figure 3.1b presents an example in which the sum of pixel intensities of the rectangle **D** is computed from the pixel values at its corner points in the integral image representation:

$$\mathbf{D} = I_{\mathrm{Int}}(u_4,v_4) - I_{\mathrm{Int}}(u_3,v_3) - I_{\mathrm{Int}}(u_2,v_2) + I_{\mathrm{Int}}(u_1,v_1) \tag{3.2}$$

The corner point (u_4,v_4) in the integral image represents the sum of pixel intensities of all rectangles **A**,**B**,**C** and **D**. Hence, the pixel values at the points (u_2,v_2) and (u_3,v_3) have to be subtracted from $I_{\mathrm{Int}}(u_4,v_4)$ and $I_{\mathrm{Int}}(u_1,v_1)$ has to the be added since the sum of pixel intensities in the rectangle **A** has been subtracted twice. Since Haar-like features consist of adjacent rectangular regions that share corner points, a Haar-like feature formed by two and three rectangles can be computed from 6 and 8 pixel values, respectively, and the *four-rectangle-feature* requires nine memory accesses.

Additionally, Viola and Jones [146] suggest to perform a variance normalization of the detection window in the detector learning and its application to achieve partial invariance to changing lighting conditions. The variance of a random variable X is defined in Equation (3.3) and can be transformed as in Equation (3.4) because of the linearity of expected values $\mathrm{E}(X)$:

$$\begin{aligned} \mathrm{Var}(X) &= \mathrm{E}[(X - \mathrm{E}(X))^2] && (3.3)\\ &= \mathrm{E}(X^2) - (\mathrm{E}(X))^2 && (3.4) \end{aligned}$$

Selecting the pixel of an image I as random variable and assuming a discrete uniform

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

distribution, the variance of an image or detection window can be written as:

$$\begin{aligned}\mathrm{Var}(I) &= \frac{N_{\mathrm{Pixel}}}{N_{\mathrm{Pixel}}-1}\left(\frac{1}{N_{\mathrm{Pixel}}}\sum I(u,v)^2 - \left(\frac{1}{N_{\mathrm{Pixel}}}\sum I(u,v)\right)^2\right) \\ &= \frac{1}{N_{\mathrm{Pixel}}-1}\sum I(u,v)^2 - \frac{N_{\mathrm{Pixel}}}{N_{\mathrm{Pixel}}-1}{\mu_I}^2\end{aligned}, \tag{3.5}$$

where N_{Pixel} denotes the number of pixels in the image or detection window and μ_I is the arithmetic mean of its pixels. In the variance normalization, μ_I can be computed using the integral image. But for the efficient computation of the sum of squared pixels, an additional integral image has to be calculated from the image squared.

Another rotated integral image is required to compute the set of rotated Haar-like features. A detailed description of the computation procedure is given in [96].

3.1.2 Histograms of Oriented Gradients

Dalal and Triggs [35] introduce the Histograms of Oriented Gradients (HOG) descriptor in human detection. They learn a linear SVM based on the HOG descriptor that exploits gradient information. It is inspired by orientation histograms [54], Scale-Invariant Feature Transform (SIFT) descriptors [100] and shape contexts [20]. The principle of their descriptor is to characterize object appearances only by the distribution of local intensity gradient magnitudes and directions and without the information of their precise location. Hence, the HOG descriptor is organized as concatenated gradient histograms computed on a dense, overlapping grid of image regions. Figure 3.2 visualizes the HOG descriptor for training examples of face, car, and pedestrian data sets.

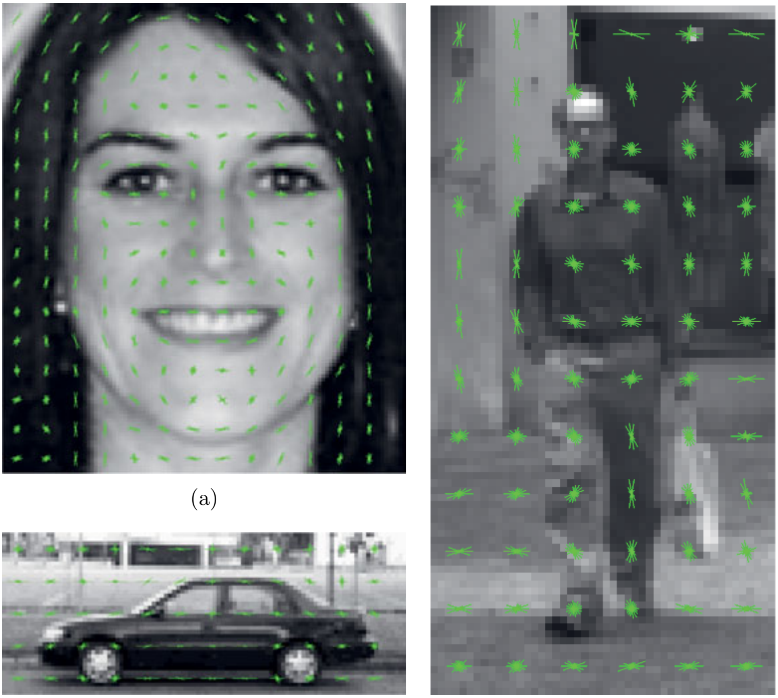
Dalal and Triggs [35] structure the extraction of their descriptor into five steps and evaluate different methods in each case:

1. **Gamma normalization** can be performed in a first step but yields only modest performance increase such that Dalal and Triggs [35] omit normalization for their default detector.
2. **Gradient computation** is suggested to be performed without pre-smoothing the image. The gradients are computed by centered [-1,0,1] one dimensional (1D) point derivatives.
3. **Orientation binning:** The grid of image regions is set up from so-called cells that have a size of 8 × 8 pixels for the default detector. The gradient direction is computed for each pixel and votes weighted by the gradient magnitude are accumulated into orientation bins identified by the gradient direction. It is suggested to vote for nine bins of unsigned gradient directions in human detection.
4. **Normalization and descriptor blocks:** For the compensation of local variations in illumination and contrast, normalization is performed on the basis of

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

blocks that consist of 2 × 2 cells in the default detector.

5. **Descriptor structure:** The HOG descriptor consists of the concatenated histograms that are normalized per block. The cells cover the complete detection window in a grid and the blocks are suggested to be overlapping such that the histogram of each cell is contained under different normalizations in four block histograms.


(a)

(b)

(c)

Figure 3.2: HOG descriptors visualized on training examples. The example images belong to positive training sets for (a) face detection [1], (b) lateral car detection [3] and (c) pedestrian detection [47].

Integral histograms Similar to Haar-like features, HOG descriptors can be computed more efficiently using an integral image representation [89, 120]. For each histogram bin selectable in the gradient computation in HOG cells, an integral histogram is constructed that adds up the magnitudes of gradients that have an orientation associated with the respective bin. A histogram bin prior to normalization referring to an arbitrary rectangular image region can then be calculated from the

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

four corner points of the rectangle in the integral histogram.

3.1.3 From Features to Classifiers

As described in Section 2.1, features are essentially functions that, controlled by some feature parameter, map the domain of observations into the feature space, the codomain of feature values. The domain of observations are images for visual object detection and in case of Haar-like features, described in Section 3.1.1, the codomain of feature values is the set of real numbers $\mathbb{R}$.

In order to create a classifier based on a feature, a machine learning algorithm has to find a decision boundary in the corresponding feature space and to create a decision rule for classification with respect to that boundary. The process of classifier learning is illustrated in Figure 3.3 by an example taken from a true face detector learning. For this, the response of a generalized Haar-like feature is analyzed that is learned for a face detector presented in Chapter 7. The construction of the generalized Haar-like feature is described in detail in Section 7.1 but at this point the sole important property is that the feature maps as well to the set of real numbers $\mathbb{R}$. This feature γ is visualized in the upper left corner of Figure 3.3 and basically computes, similar to Haar-like features, the difference between mean pixel intensities of rectangular image regions. The feature has been selected as the first in the learning of a face detector from the initial training set of 8004 positive and 2916 negative examples and demonstrates that the difference between the dark eye region and the brighter region of nose and forehead is characteristic of faces. Additionally, the top row of Figure 3.3 presents samples $\mathbf{x}$ of the positive and negative training sets together with a marker that is used to highlight the feature value $\gamma(\mathbf{x})$ for each sample in the diagram below.

The diagram presents the class-conditional densities of the positive and negative set as a function of the feature value assuming that both classes have a Gaussian distribution parametrized by the sample mean and sample variance computed from the feature responses on the examples of the respective classes. This illustrated method to learn a model of the observations given by the training data and the associated class labels is an example of a generative approach as discussed in Section 2.1.2. On the basis of the model of Gaussian distributions, the blue dashed line in the diagram can be observed as decision boundary for the positive and negative class and a decision rule can be formulated to create a classifier based on the feature. The blue solid line in the diagram marks the decision threshold θ that has been learned in the true face detector training by the AdaBoost algorithm that belongs to the discriminative approaches (see Section 2.1.2) and is described in detail in Section 3.2.1. This decision boundary learned by AdaBoost is able to classify $93.18\,\%$ of the training set correctly but it can be observed from the markers in the diagram that the most right negative and the most left positive example in the top row are misclassified.

The difference in the decision boundaries learned by the Gaussian model and

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

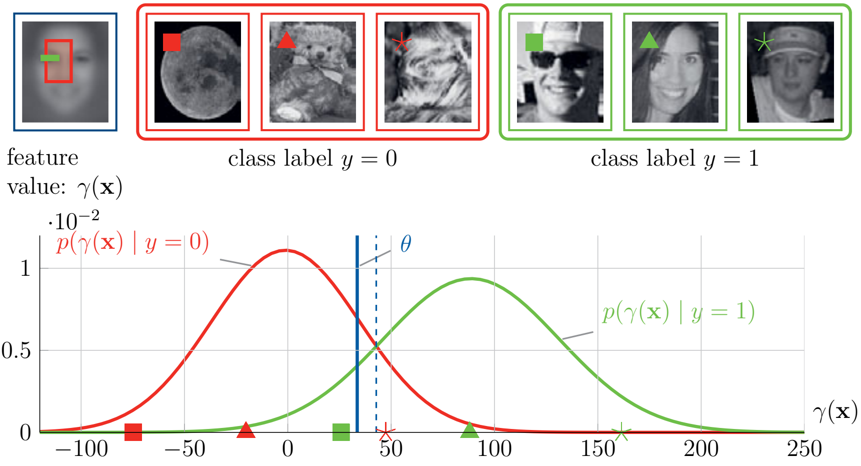


Figure 3.3: Construction of a classifier based on a feature. A generalized Haar-like feature that is learned in a true face detector training is presented in the top left corner. This feature basically computes the difference between mean pixel intensities of rectangular image regions and returns a real number as feature value. The rest of the top row shows samples of the negative and positive training set [1] together with a marker that is used to highlight the corresponding feature value in the diagram below. In a simple approach, the two classes are assumed to have Gaussian distributions that are parametrized by the sample mean and sample variance computed from the respective feature responses. The diagram presents the class-conditional densities of the positive and negative set and the decision boundary obtained from this model as blue dashed line as well as the boundary learned by AdaBoost in the true training as blue solid line.

AdaBoost is a result of the different approaches, on the one hand a generative method that assumes Gaussian distributions to learn the boundary and on the other hand a discriminative method that finds the boundary based on the splitting of the training examples.

3.2 Supervised Machine Learning

This section introduces Adaptive Boosting (AdaBoost) and the Viola and Jones framework that utilizes this machine learning algorithm for visual object detection. AdaBoost is a powerful technique to learn strong classifiers and enables the Viola and Jones framework together with additional contributions to create very successful object detectors that are real-time capable.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

3.2.1 Adaptive Boosting

In 1984, Valiant [143] develops a model of learning in which a hypothesis or prediction rule is to be learned from randomly chosen examples of a labeled training set that represents an unknown concept. In case of learning classifiers for visual object detection, the concept to be identified by the learning algorithm is the appearance of one or multiple object classes. The learned hypothesis is demanded to correctly classify with high probability new instances of the object class.

Following this methodology, Michael Kearns formulates in 1988 [79] the basic idea of boosting algorithms. In his manuscript, Kearns states the question if the ability of an efficient algorithm to learn under some assumptions [143] a hypothesis that is just a little better than random guessing implies the existence of an algorithm that creates a hypothesis of arbitrary accuracy. This problem is denoted *hypothesis boosting problem*. Further, Kearns declares an algorithm that outputs the first-mentioned weak hypothesis as *weak learning* algorithm while the algorithm that creates a hypothesis of arbitrary accuracy is denoted *strong learning* algorithm. The idea of combining several weak hypotheses to a strong hypothesis is later very well illustrated in the horse race example of Freund and Schapire [57]. The weak hypotheses in Kearns' manuscript [79] correspond in the example to simple rules of thumb that an expert on horse racing might give for the result of a race. One very simple rule might be to bet on the horse that recently won the most races. Obviously, not every race is won by the same horse. But it is as well convincing that the above rule has a better probability to predict the actual winning horse than random guessing. An expert on horse racing might be able to give several of such simple rules of thumb. Kearns' question is in this context, if it is possible to combine these rules and evaluate them together to form a strong hypothesis that predicts the actual winning horse with a high probability.

Schapire [127] proofed Kearns' implication in 1990 by constructing a recursive strong learning algorithm that runs a weak learner several times. Accordingly, a learning algorithm that uses a different learning algorithm as a subroutine is called a *boosting* algorithm. Following Schapire's proof [127], further boosting algorithms have been developed, e.g. [55]. Freund and Schapire [56] proposed in 1996 the Adaptive Boosting (AdaBoost) algorithm that adaptively adjusts to the errors of the learned weak hypotheses by generating distributions over the training set during the boosting process. Algorithm 1 presents the pseudo code of the AdaBoost algorithm in its first multi-class extension and is described in detail in the following. As a boosting algorithm that follows the paradigm of learning from examples, AdaBoost requires the input of a labeled training set $\mathcal{S} = \{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{N_\mathcal{S}},y_{N_\mathcal{S}})\}$ of $N_\mathcal{S}$ training examples, each consisting of a training sample $\mathbf{x}_i \in \mathcal{X}$ and an assigned label $y_i \in \mathcal{Y} = \{1,\ldots,j\}$ specifying the class of the sample, together with a generic weak learning algorithm that is denoted by **WeakLearn**. The algorithm is round-based such that **WeakLearn** is called several times in a non-recursive loop and a third input parameter is the requested number of rounds N_R.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Algorithm 1: Adaptive Boosting algorithm AdaBoost.M1 in pseudo code.

```
Input:
    (a) Set $\mathcal{S} = \{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{N_\mathcal{S}},y_{N_\mathcal{S}})\}$ of $N_\mathcal{S}$ training examples,
        each consisting of a training sample $\mathbf{x}_i \in \mathcal{X}$ (e.g. an image) and an
        assigned label $y_i \in \mathcal{Y} = \{1,\ldots,j\}$ specifying the class of the sample.
    (b) weak learning algorithm WeakLearn
    (c) parameter $N_R$ specifying the number of training rounds

Initialize the distribution $W_r(i)$ over the training set $\mathcal{S}$ as $W_1(i) = \frac{1}{N_\mathcal{S}}$ for all $i$.

For training round $r = 1$ to $N_R$:

    1. Determine hypothesis $h_r : \mathcal{X} \to \mathcal{Y}$ by executing WeakLearn
       given the distribution $W_r$.
    2. Compute the error of $h_r$: $\epsilon_r = \sum_{i:h_r(\mathbf{x}_i)\neq y_i} W_r(i)$.
       If $\epsilon_r > \frac{1}{2}$, then set $N_R = r - 1$ and abort loop.
    3. Set $\beta_r = \frac{\epsilon_r}{1-\epsilon_r}$.
    4. Update distribution $W_r$:
           $W_{r+1}(i) = \frac{W_r(i)}{Z_r} \times \begin{cases} \beta_r & \text{if } h_r(\mathbf{x}_i) = y_i \\ 1 & \text{otherwise} \end{cases}$
       with $Z_r$ as normalization constant that ensures $W_{r+1}$ to be a distribution.

Output the final strong hypothesis

    $H(\mathbf{x}) = \arg\max_{y\in\mathcal{Y}} \sum_{r:h_r(\mathbf{x})=y} \log \frac{1}{\beta_r}.$
```

One key element of Adaptive Boosting is that the boosting algorithm governs a distribution over the training set that is controlled by the weak learning algorithm and adapted at the end of each learning round. The mechanism for adapting the weight distribution is illustrated in Figure 3.4. The initial distribution W_1 of the first round over the set of $N_\mathcal{S}$ training examples is defined uniform: $W_1(i) = \frac{1}{N_\mathcal{S}}$ for all i.

In each training round r, AdaBoost provides **WeakLearn** with the actual distribution W_r. **WeakLearn** is run with the task to compute a hypothesis $h_r : \mathcal{X} \to \mathcal{Y}$ that classifies a subset of the training set correctly which is associated with a high probability with respect to W_r. More specifically, **WeakLearn** is instructed to find the weak hypothesis h_r that minimizes the training error

$$\epsilon_r = \sum_{i:h_r(\mathbf{x}_i)\neq y_i} W_r(i) \tag{3.6}$$

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

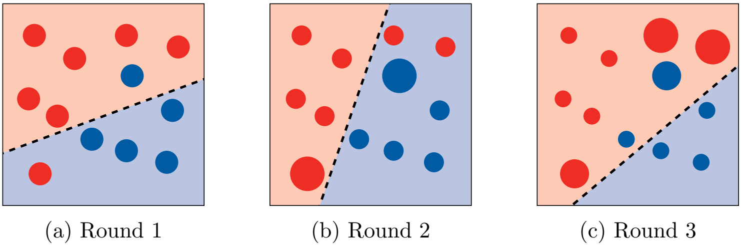

(a) Round 1 (b) Round 2 (c) Round 3

Figure 3.4: Illustration of AdaBoost's adaption of the weight distribution in the first three training rounds for a two dimensional (2D) feature space. The algorithm starts with an uniform distribution and learns in each round a decision boundary presented as dashed line. The weights are adapted so that misclassified training examples get higher influence in the following rounds resulting in different decision boundaries. The strong hypothesis is given as a linear combination of all learned decision boundaries.

with respect to the distribution W_r. The training error defined in Equation (3.6) is simply the sum of weights handled by W_r that are assigned to training examples for which the hypothesis h_r predicts a wrong class label.

If the minimum achievable training error ϵ_r of some round r is greater than $1/2$, the series of training rounds is aborted and the algorithm combines only the weak hypotheses h_1 to h_{r-1} that were already computed to a final strong hypothesis H. Freund and Schapire show in [56] that the training error of H drops to zero exponentially fast if $\epsilon_r \leq 1/2$ holds true for all $r = 1 \dots N_R$ and the weak hypotheses consistently achieve a training error only slightly better than $1/2$. In the case of binary classification ($k = 2$), this means that the weak hypotheses need to be only slightly better than random guessing. Based on the training error ϵ_r achieved by the actual weak hypothesis h_r, a scaling term $\beta_r = \frac{\epsilon_r}{1-\epsilon_r}$ is computed that maps the allowed training error into the interval [0,1). β_r is then used to adapt the weight distribution W over the training set such that training examples that are misclassified by the actual weak hypothesis get a higher probability with respect to W in the following training round:

$$W_{r+1}(i) = \frac{W_r(i)}{Z_r} \times \begin{cases} \beta_r & \text{if } h_r(\mathbf{x}_i) = y_i \\ 1 & \text{otherwise} \end{cases} \tag{3.7}$$

with Z_r as normalization constant that ensures W_{r+1} to be a distribution. More precisely, Equation (3.7) reduces the weights of correctly classified training examples by multiplying them with β_r followed by a normalization over all training example weights. The training error of a weak hypothesis has in this way a direct influence on the strength of the weight adaption. Then the AdaBoost algorithm continues in the next training round and runs **WeakLearn** to compute the subsequent hypothesis h_{r+1} that minimizes the training error with respect to the updated distribution W_{r+1}.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In this way, AdaBoost assigns higher weights to training examples that are difficult to classify for **WeakLearn** and thus focuses on learning these hard examples.

After all requested training rounds have been accomplished, the algorithm combines all weak hypotheses that were computed to a final strong hypothesis

$$H(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{r:h_r(\mathbf{x})=y} \log \frac{1}{\beta_r}. \tag{3.8}$$

The strong hypothesis decides in a weighted majority vote and outputs the class label y that maximizes the sum of weights assigned to weak hypotheses predicting that label. The weight of a weak hypothesis h_r is set to $\log\left(\frac{1}{\beta_r}\right)$ such that a higher influence on the majority decision is given to weak hypotheses that achieved a lower training error.

3.2.2 Viola and Jones Detection Framework

Viola and Jones proposed a very successful framework for real-time face detection [145, 146] in 2001 that utilizes the AdaBoost algorithm in the classifier learning. The Viola and Jones framework integrates several methods that contribute to its success. The key contributions of the framework are:

- Utilization of Haar-like features and their fast computation using integral images (see Section 3.1.1)
- Application of AdaBoost in detector learning
- Definition of a efficient weak learning algorithm
- Introduction of a cascaded detector structure

A detailed description of Haar-like features is given in Section 3.1.1. The last three contributions are thoroughly discussed in the following.

Boosting Algorithm

The object detector is learned by the AdaBoost variant for binary classification in a slightly modified version of the original algorithm [57]. The pseudo code in Algorithm 2 presents the AdaBoost algorithm in the Viola and Jones framework. The following description focuses on these modifications and differences to the multi-class variant of AdaBoost presented in Section 3.2.1.

The AdaBoost algorithm in the Viola and Jones framework requires as well as the previously described multi-class variant as inputs a labeled training set $\mathcal{S}$, a weak learning algorithm and the number of requested training rounds. For each training sample $\mathbf{x}_i \in \mathcal{X}$, the assigned label is defined as $y_i \in \mathcal{Y} = \{0,1\}$ with $y_i = 0$ for a negative and $y_i = 1$ for a positive example. But the distribution over the training set $\mathcal{S}$ is initialized slightly different so that the probabilities of all positive and all negative examples each add up to 1/2 while all positive and negative examples, respectively have the same probability.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Algorithm 2: Viola and Jones Boosting algorithm in pseudo code.

```
Input:
    (a) Set $\mathcal{S}$ of $N_{\mathcal{S}}$ training examples,
        each consisting of a training sample $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{0,1\}$
        with $y_i = 0$ for a negative and $y_i = 1$ for a positive example.
    (b) weak learning algorithm that selects weak hypothesis $h(\mathbf{x}) = h(\mathbf{x},\gamma,\rho,\theta)$,
        with $\gamma$, $\rho$ and $\theta$ as parameter that are determined to minimize the
        classification error on the training set $\mathcal{S}$.
    (c) parameter $N_R$ specifying the number of training rounds

Initialize the distribution over the training set $\mathcal{S}$ as the weights
    $w_{1,i} = \frac{1}{2N_{\mathcal{N}}},\frac{1}{2N_{\mathcal{P}}}$ for $y_i = 0,1$ respectively, where $N_{\mathcal{N}}$ are the number of
    negative and $N_{\mathcal{P}}$ are the number of positive training examples.

For training round $r = 1$ to $N_R$:

    1. Determine best weak hypothesis $h_r(\mathbf{x}) = h(\mathbf{x},\gamma_r,\rho_r,\theta_r)$ that minimizes
       the error $\epsilon_r = \min_{\gamma,\rho,\theta} \sum_i w_{r,i} |h(\mathbf{x}_i,\gamma,\rho,\theta) - y_i|$
       with respect to the distribution defined by the weights $w_{r,i}$.
    2. Set $\beta_r = \frac{\epsilon_r}{1-\epsilon_r}$.
    3. Update the distribution of weights:
           $w_{r+1,i} = w_{r,i} \beta_r^{1-e_i}$
       with $e_i = 0$ if the example $\mathbf{x}_i$ is classified correctly and $e_i = 1$ otherwise.
    4. Normalize the weights to be a distribution:
           $w_{r+1,i} = \frac{w_{r+1,i}}{\sum_{j=1}^{N_{\mathcal{S}}} w_{r+1,j}}$

Output the final strong hypothesis

    $H(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{r=1}^{N_R} \alpha_r h_r(\mathbf{x}) \geq \frac{1}{2} \sum_{r=1}^{N_R} \alpha_r \\ 0 & \text{otherwise} \end{cases}$
    with $\alpha_r = \log \frac{1}{\beta_r}$ .
```

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In each training round, the AdaBoost algorithm runs Viola and Jones' weak learning algorithm to compute the best weak hypothesis $h_r : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the training error

$$\epsilon_r = \min_{\gamma,\rho,\theta} \sum_i w_{r,i} |h(\mathbf{x}_i,\gamma,\rho,\theta) - y_i| \tag{3.9}$$

with respect to the distribution defined by the weights $w_{r,i}$. Hence, the best weak hypothesis is found to be $h_r(\mathbf{x}) = h(\mathbf{x},\gamma_r,\rho_r,\theta_r)$ with minimizers γ_r, ρ_r and θ_r. Viola and Jones propose an efficient weak learning algorithm that computes $h_r(\mathbf{x})$ and determines its minimizers. This weak learning algorithm is described in detail in the next paragraph.

In binary classification, a weak learning algorithm is able to find in any case a hypothesis with training error $\epsilon_r \leq 1/2$, because otherwise the inverse hypothesis $1 - h_r(\mathbf{x})$ would satisfy the error bound. Thus, the abort condition of the multi-class algorithm can be omitted.

The update rules for the weight distribution use a slightly different notation benefiting from the simpler case of binary classification but work exactly the same as in the multi-class variant. The weights are first adapted such that the impact of the correctly classified training examples is reduced:

$$w_{r+1,i} = w_{r,i} \beta_r^{1-e_i} \tag{3.10}$$

with $e_i = 0$ if the example $\mathbf{x}_i$ is classified correctly and $e_i = 1$ otherwise. Followed by a normalization that ensures the weights to be a distribution:

$$w_{r+1,i} = \frac{w_{r+1,i}}{\sum_{j=1}^{N_\mathcal{S}} w_{r+1,j}} \tag{3.11}$$

After the last training round, the AdaBoost algorithm outputs a final strong hypothesis

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{r=1}^{N_R} \alpha_r h_r(\mathbf{x}) \geq \frac{1}{2} \sum_{r=1}^{N_R} \alpha_r \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

with $\alpha_r = \log \frac{1}{\beta_r}$.

Hence, the strong hypothesis that predicts the correct class label with high accuracy is a linear combination of N_R weak hypotheses that performs a weighted majority decision. The weights α_r of the weak hypotheses give higher influence in the weighted majority vote to hypotheses that caused a smaller training error. The strong hypothesis predicts a positive object class if the weak hypotheses voting for the positive class are associated with at least half of the sum of all hypothesis weights.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Weak Learning Algorithm

Viola and Jones' weak learner computes a hypothesis that is characterized by a feature γ, polarity ρ and threshold θ. The weak hypothesis is defined as

$$h(\mathbf{x},\gamma,\rho,\theta) = \begin{cases} 1 & \text{if } \rho\gamma(\mathbf{x}) < \rho\theta \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

so that the feature response $\gamma(\mathbf{x})$ is compared to a threshold θ and the polarity $\rho \in \{-1,1\}$ controls the direction of the inequality. The objective of the weak learning algorithm is in each training round r to determine the parameter γ_r, polarity ρ_r and threshold θ_r that minimizes the training error given by Equation (3.9). The feature γ_r is in this process selected from a feature set that has to be provided to the weak learner. Viola and Jones propose to exploit an overcomplete set of Haar-like features as described in Section 3.1.1. The weak learning algorithm selects the error minimizing feature, threshold and polarity in an efficient procedure that is described in the following with an example of learning a strong hypothesis for two training rounds.

Hypothesis Learning Example

The setup for the strong hypothesis learning by the Viola and Jones framework is defined as follows:

Input: The training set is assumed to consist of a positive set $\mathcal{P} = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\}$ containing four training samples and a negative set $\mathcal{N} = \{\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7,\mathbf{x}_8,\mathbf{x}_9\}$ of five samples.

The weak learner is provided with a example feature set consisting of only two features γ^a and γ^b.

The requested number of training rounds is set to $N_R = 2$.

Initialization: The training example weights $w_{r,i}$ are initialized according to Algorithm 2 so that

$$w_{1,1} = w_{1,2} = w_{1,3} = w_{1,4} = \frac{1}{8}$$

and

$$w_{1,5} = w_{1,6} = w_{1,7} = w_{1,8} = w_{1,9} = \frac{1}{10}.$$

Training: In the following, the sum of weights assigned to all positive training examples in round r is denoted

$$w_{r,\mathcal{P}} = \sum_{i:\mathbf{x}_i \in \mathcal{P}} w_{r,i} \tag{3.14}$$

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

and the sum of all negative example weights of round r is given by

$$w_{r,\mathcal{N}} = \sum_{i:\mathbf{x}_i \in \mathcal{N}} w_{r,i} \tag{3.15}$$

Hence, the initialization yields $w_{1,\mathcal{P}} = 0.5$ and $w_{1,\mathcal{N}} = 0.5$.

The weak learning algorithm creates for each feature in the feature set a list that is sorted by the feature response $\gamma(\mathbf{x})$ to the training samples. These lists are presented in Table 3.1 for the two training rounds of the hypothesis learning example.

For each element in the lists, the algorithm computes the sum of positive example weights before and including the current example $\hat{w}_{r,\mathcal{P}}$ and the sum of negative example weights before and including the current example $\hat{w}_{r,\mathcal{N}}$. The training error that the corresponding feature γ generates, if a decision threshold θ that splits the training set is put between the current and next feature value, is then computed as

$$\epsilon = \min\left(\hat{w}_{r,\mathcal{P}} + (w_{r,\mathcal{N}} - \hat{w}_{r,\mathcal{N}}), \hat{w}_{r,\mathcal{N}} + (w_{r,\mathcal{P}} - \hat{w}_{r,\mathcal{P}})\right). \tag{3.16}$$

The first term represents here the error that is generated if all training examples after the threshold are classified as positive. The second contrary computes the error when labeling all training examples before the threshold as positive. Hence, the polarity ρ of a potential weak hypothesis is defined by which term minimizes ϵ.

The training error ϵ_r generated by the best weak hypothesis h_r of round r is thus given by the minimum ϵ over all sorted lists. The feature creating the list that minimizes ϵ is selected as γ_r by the weak learning algorithm and the other two minimizers of $h(\mathbf{x})$ can be obtained as well from this list. In this way, the weak learner computes the best weak hypothesis h_r by a single pass through all lists.

This learning procedure is in the following illustrated by the previously defined learning example with the help of Table 3.1.

Round 1: The sorted lists created for both features γ^a and γ^b of the feature set are presented on top of each other in Table 3.1. The first column shows the sorted feature values and the second column lists the training sample for which the feature value is assumed. Positive training samples are printed in blue and negative samples in red. The following five columns present the values that the weak learning algorithm evaluates in the first round when it passes the sorted lists. The weak learner also considers the sum of all positive example weights $w_{r,\mathcal{P}}$ and the sum of all negative example weights $w_{r,\mathcal{N}}$ that are independent of the feature and thus listed in the headline of the training round. When passing the list of the first feature γ^a, the minimum training error of this feature is found to be 0.325 in the upper half of the columns M and M' that are shorthand symbols for the minimizing terms in Equation (3.16). This value is stored as potential minimum error together with the threshold and polarity that are required to define the weak classifier based on this feature. After that the next list is evaluated and, in this case of only two features, the minimum training error of the first round is found as $\epsilon_1 = 0.3$ for the feature

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 3.1: Weak learning example with two features for two rounds. M and M' are shorthand symbols for the minimizing terms of ϵ. Positive training examples are printed in blue while negative examples are red. The minimum training error ϵ_r and decision threshold θ_r found by the weak learner are marked green. The weights of the misclassified examples that contribute to the error are marked gray.

$M = \hat{w}_{r,\mathcal{P}} + (w_{r,\mathcal{N}} - \hat{w}_{r,\mathcal{N}})$, $M' = \hat{w}_{r,\mathcal{N}} + (w_{r,\mathcal{P}} - \hat{w}_{r,\mathcal{P}})$											
		Round r=1: $w_{1,\mathcal{P}} = 0.5$, $w_{1,\mathcal{N}} = 0.5$					**Round r=2:** $w_{2,\mathcal{P}} = 0.357$, $w_{2,\mathcal{N}} = 0.643$				
Feature γ^a											
$\gamma^a(\mathbf{x})$	$\mathcal{X}$	$w_{1,i}$	$\hat{w}_{1,\mathcal{P}}$	$\hat{w}_{1,\mathcal{N}}$	M	M'	$w_{2,i}$	$\hat{w}_{2,\mathcal{P}}$	$\hat{w}_{2,\mathcal{N}}$	M	M'
10	$\mathbf{x}_8$	0.1	0	0.1	0.4	0.6	0.167	0	0.167	0.476	0.524
20	$\mathbf{x}_3$	0.125	0.125	0.1	0.525	0.475	0.089	0.089	0.167	0.565	0.435
30	$\mathbf{x}_6$	0.1	0.125	0.2	0.425	0.575	0.167	0.089	0.334	0.398	0.602
40	$\mathbf{x}_9$	0.1	0.125	0.3	0.325	0.675	0.167	0.089	0.501	0.231	0.769
50	$\mathbf{x}_1$	0.125	0.25	0.3	0.45	0.55	0.089	0.178	0.501	0.32	0.68
60	$\mathbf{x}_7$	0.1	0.25	0.4	0.35	0.65	0.071	0.178	0.572	0.249	0.751
70	$\mathbf{x}_4$	0.125	0.375	0.4	0.475	0.525	0.089	0.267	0.572	0.338	0.662
80	$\mathbf{x}_2$	0.125	0.5	0.4	0.6	0.4	0.089	0.356	0.572	0.427	0.573
90	$\mathbf{x}_5$	0.1	0.5	0.5	0.5	0.5	0.071	0.356	0.643	0.356	0.644
Feature γ^b											
$\gamma^b(\mathbf{x})$	$\mathcal{X}$	$w_{1,i}$	$\hat{w}_{1,\mathcal{P}}$	$\hat{w}_{1,\mathcal{N}}$	M	M'	$w_{2,i}$	$\hat{w}_{2,\mathcal{P}}$	$\hat{w}_{2,\mathcal{N}}$	M	M'
11	$\mathbf{x}_3$	0.125	0.125	0	0.625	0.375	0.089	0.089	0	0.732	0.268
22	$\mathbf{x}_9$	0.1	0.125	0.1	0.525	0.475	0.167	0.089	0.167	0.565	0.435
33	$\mathbf{x}_6$	0.1	0.125	0.2	0.425	0.575	0.167	0.089	0.334	0.398	0.602
44	$\mathbf{x}_1$	0.125	0.25	0.2	0.55	0.45	0.089	0.178	0.334	0.487	0.513
55	$\mathbf{x}_8$	0.1	0.25	0.3	0.45	0.55	0.167	0.178	0.501	0.32	0.62
66	$\mathbf{x}_2$	0.125	0.375	0.3	0.575	0.425	0.089	0.267	0.501	0.409	0.591
77	$\mathbf{x}_4$	0.125	0.5	0.3	0.7	0.3	0.089	0.356	0.501	0.498	0.502
88	$\mathbf{x}_7$	0.1	0.5	0.4	0.6	0.4	0.071	0.356	0.572	0.427	0.573
99	$\mathbf{x}_5$	0.1	0.5	0.5	0.5	0.5	0.071	0.356	0.643	0.356	0.644

γ^b and marked green in Table 3.1. The decision threshold can be read off the table as to be $\gamma^b(\mathbf{x}_4) = 77 < \theta_1 \leq \gamma^b(\mathbf{x}_7) = 88$. Because the minimum is obtained from column M', the polarity is determined as $\rho_1 = 1$ together with the threshold $\theta_1 = (77 + 88)/2 = 82.5$ so that the best weak hypothesis of the first round is according to Equation (3.13) defined as

$$h_1(\mathbf{x}) = \begin{cases} 1 & \text{if } \gamma^b(\mathbf{x}) < 82.5 \\ 0 & \text{otherwise} \end{cases}.$$

Hence, the error $\epsilon_1 = 0.3$ is the sum of the weights $w_{1,6}$, $w_{1,8}$ and $w_{1,9}$ of the misclassified examples $\mathbf{x}_6$, $\mathbf{x}_8$ and $\mathbf{x}_9$ marked in gray. The scaling factor is computed as

$$\beta_1 = \frac{\epsilon_1}{1 - \epsilon_1} = \frac{0.3}{1 - 0.3} = \frac{3}{7} = 0.429$$

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

based on the error and used to weight the weak hypothesis h_1 in the strong hypothesis H and to adapt the training examples after Equations (3.10) and (3.11).

Thus, the weights of the misclassified examples remain

$$w'_{1,6} = w'_{1,8} = w'_{1,9} = \frac{1}{10}$$

while the weights of the correctly classified examples are decreased before normalization to

$$w'_{1,1} = w'_{1,2} = w'_{1,3} = w'_{1,4} = \frac{1}{8} \cdot \beta_1 = \frac{1}{8} \cdot \frac{3}{7} = \frac{3}{56}$$

and

$$w'_{1,5} = w'_{1,7} = \frac{1}{10} \cdot \beta_1 = \frac{1}{10} \cdot \frac{3}{7} = \frac{3}{70} \ .$$

The sum of the adapted weights is

$$\sum_i w'_{1,i} = 3 \cdot \frac{1}{10} + 4 \cdot \frac{3}{56} + 2 \cdot \frac{3}{70} = \frac{84 + 60 + 24}{280} = \frac{168}{280} = \frac{3}{5}$$

and the example weights are normalized before the second training round to

$$w_{2,6} = w_{2,8} = w_{2,9} = \frac{\frac{1}{10}}{\sum_i w'_{1,i}} = \frac{1}{10} \cdot \frac{5}{3} = \frac{1}{6} \approx 0.167$$

$$w_{2,1} = w_{2,2} = w_{2,3} = w_{2,4} = \frac{\frac{3}{56}}{\sum_i w'_{1,i}} = \frac{3}{56} \cdot \frac{5}{3} = \frac{5}{56} \approx 0.089$$

and

$$w_{2,5} = w_{2,7} = \frac{\frac{3}{70}}{\sum_i w'_{1,i}} = \frac{3}{70} \cdot \frac{5}{3} = \frac{1}{14} \approx 0.071 \ .$$

Round 2: The sum of weights of all positive and negative training examples is computed for the second round to

$$w_{2,\mathcal{P}} = w_{2,1} + w_{2,2} + w_{2,3} + w_{2,4} = 4 \cdot \frac{5}{56} = \frac{5}{14} \approx 0.357$$

and

$$w_{2,\mathcal{N}} = w_{2,6} + w_{2,8} + w_{2,9} + w_{2,5} + w_{2,7} = 3 \cdot \frac{1}{6} + 2 \cdot \frac{1}{14} = \frac{9}{14} \approx 0.643 \ .$$

The weak learner passes the sorted lists for the second round again and computes the last four columns of Table 3.1 in this process. Due to the adapted example weights, the minimum training error is achieved in this round by feature γ^a and found as $\epsilon_2 = 0.231$ in column M such that the polarity is set to $\rho_1 = -1$.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The best weak hypothesis of the second round is thus obtained from Table 3.1 as

$$h_2(\mathbf{x}) = \begin{cases} 1 & \text{if } -1 \cdot \gamma^a(\mathbf{x}) < -1 \cdot 45 \\ 0 & \text{otherwise} \end{cases} .$$

The scaling factor of the second round is computed as

$$\beta_2 = \frac{\epsilon_2}{1-\epsilon_2} = \frac{0.231}{1-0.231} \approx 0.300$$

but the example weights are not adapted because the last requested training round $N_R = 2$ is finished.

Output: The learned weak hypotheses are weighted in the final strong hypothesis by factors $\alpha_r = \log \frac{1}{\beta_r}$ that give higher influence to weak hypotheses that achieved lower training errors. The final strong hypothesis is computed according to Equation (3.12) as the weighted majority decision

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } 0.368 \cdot h_1(\mathbf{x}) + 0.529 \cdot h_2(\mathbf{x}) \geq (0.368 + 0.529)/2 = 0.4485 \\ 0 & \text{otherwise} \end{cases}$$

Memory consumption The selection of the best weak hypothesis in the learning algorithm is efficient, because the sorted lists need only to be created in the first training round. This creation requires the computation of all features for all training examples and sorting the results. The sorted lists are represented in Table 3.1 by the first two columns. The example weights can be determined from a global map of feature identifiers to feature labels and currently associated weights that is stored only once per training round but is accessed by all sorted lists. All other columns in Table 3.1 contain temporary values that are calculated while the weak learner passes the list and don't have to be stored. Besides the sorted lists and the map of feature values, only the current minimum training error and the parameter of the corresponding weak hypothesis are saved during the weak learning algorithm.

But the training of object detectors that show a high performance in real scenarios readily involves hundreds of thousands or up to millions of features and tens of thousands of training examples. Roughly 25000 examples are considered in each training round of the face detectors learned for this work. The evaluated feature set contains in some cases up to almost four million features.

Based on eight bytes to store a feature identifier and value, all sorted lists would consume $4000000 \cdot 25000 \cdot (4 \text{ bytes} + 4 \text{ bytes}) = 800$ gigabytes. This requirement easily exceeds the memory capacity of conventional computer workstations but deleting the sorted lists from memory after use and recomputing them in the following training round massively increases the computational costs.

A strategy to solve this problem that is pursued in this thesis is to parallelize the classifier learning on a distributed computing cluster. The weak learning algorithm

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

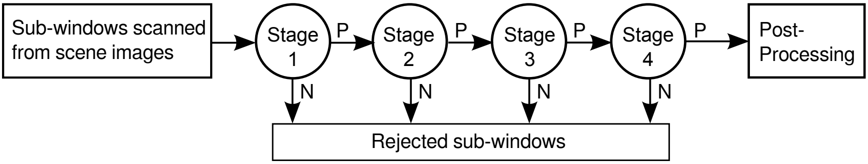


Figure 3.5: Classifier cascade: Evaluated sub-window have to be positively classified (P) and passed by all cascade stages to be considered as a found object. Each cascade stage can reject a sub-window if it is negatively classified (N) and thus prevents its processing by the following stages.

allows the parallelization in terms of features such that each computing node only evaluates a subset of training examples. In this way, not only the computational demands but also the memory consumption is distributed in the computing cluster.

As part of this thesis, a distributed machine learning framework has been developed that is briefly described in Chapter 4.

Cascaded Detector

Viola and Jones furthermore propose to arrange strong hypotheses in a classifier cascade. The motivation is to take into account that a sliding window based detector in general is confronted with a very large amount of negative sub-windows while only a small quantity of sub-windows show the object to be detected. Because of this ratio, it is most important for a fast object detector to rapidly handle negative samples. Another insight is that it is possible to learn strong hypotheses that can reject a moderate percentage of negative samples while preserving nearly all positive samples very efficiently, meaning that they only consist of relatively few weak hypotheses. Such strong hypotheses are utilized in a classifier cascade as illustrated in Figure 3.5.

Every evaluated sub-window has to be positively classified and passed by all strong hypotheses, in the following denoted cascade stages, to be considered as a found object. Each cascade stage definitely rejects a sub-window if it is negatively classified and thus prevents its processing by the following stages. Sub-windows that have passed several stages are in general more similar to the positive object class, making the classification problem of later stages harder. The complexity of strong hypotheses increases in later stages for this reason. But the classifier cascade is very efficient because most sub-windows are rejected in earlier stages and only few have to be processed by the more complex stages.

In practice, roughly at least half of all negative sub-windows are rejected in each stage allowing for real-time object detection.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Cascade learning The training procedure for a classifier cascade has to reflect the concept that only negative samples that are miss-classified by all previous stages are presented to the following stage. Hence, a strong hypothesis in a classifier cascade is never confronted with negative samples that are correctly classified by one of its preceding stages. So it is not necessary for such a strong hypothesis to learn to distinguish these rejected negative samples from the positive object class. As a consequence, examples in the negative training set that are correctly classified after learning a cascade stage can be removed from the set before training the next stage. Then the negative training set has to be replenished with examples that are pertinent to the stage to be learned. The so far learned cascaded classifier is therefore applied to unseen negative examples and only False Positives (FPs) are added to the negative training set. This procedure is called a *bootstrap* strategy [140]. The negative training set for later cascade stages consists due to the bootstrap strategy of examples that are mostly very similar to the positive examples and thus hard to learn. The complexity of later cascade stages increases as a result.

Algorithm 3 presents Viola and Jones' method to learn a classifier cascade. During learning, Algorithm 2 is run several times to create a cascade of strong hypotheses that satisfies a desired overall detection performance. Hence, the input of the algorithm contains in addition to the training examples the maximum acceptable false positive rate f per stage and minimum detection rate per stage t. These input parameter control the classification performance of each single cascade stage and thus have an effect on their complexity. An additional input parameter is the requested overall False Positive Rate (FPR) of the cascade F_{Target} controlling the number of learned stages.

In this way, the outer loop of Algorithm 3 continues to learn further cascade stages as long as F_{Target} is not fulfilled on condition that each stage maintains a minimum detection rate of t. This means that the miss rate in each stage is restricted but the loss of True Positive (TP) detections is expected to grow for longer classifier cascades.

The inner loop of the algorithm consists of the round-based training of Algorithm 2 with the difference that the training is not finished after a certain number of rounds but a specified classification performance has to be accomplished.

More precisely, the so far trained classifier cascade, including the currently learned stage, is evaluated on a validation set of unseen examples after each round of learning a strong hypothesis. The training of the currently learned stage is completed if its decision threshold τ can be adjusted such that the stage contributes to the performance of the cascade by fulfilling the required minimum detection rate t and maximum acceptable false positive rate f per stage. If a stage is finished but the False Positive Rate (FPR) achieved on the validation set by the complete cascade does not satisfy F_{Target}, the bootstrapping is performed to update the negative training set before the next cascade stage is learned. Viola and Jones propose a slightly different bootstrap strategy such that the negative set is erased and completely replenished by bootstrapped examples.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Algorithm 3: Viola and Jones cascade algorithm in pseudo code.

```
Input:
    (a) Training set of positive examples $\mathcal{P}$ and negative examples $\mathcal{N}$
    (b) Parameter $f$, $t$ defining per stage the acceptable maximum false
        positive rate and minimum detection rate, respectively.
    (c) Parameter $F_{\text{Target}}$ specifying the desired overall false positive rate

Initialize
    (a) the false positive rate of the learned cascade $F_0 = 1.0$
    (b) the true positive or detection rate of the learned cascade $T_0 = 1.0$
    (c) the number of the current cascade stage $i = 0$

While $F_i > F_{\text{Target}}$:
    1. Set $i = i + 1$
    2. Set $N_{R_i} = 0$ and $F_i = F_{i-1}$.
    3. While $F_i > f \cdot F_{i-1}$:
        i. Set $N_{R_i} = N_{R_i} + 1$
        ii. Learn a classifier with $N_{R_i}$ features from the training data $\mathcal{P}$ and $\mathcal{N}$.
        iii. Evaluate performance of current cascade by determining
             $F_i$ and $T_i$ on a validation set.
        iv. Decrease decision threshold $\tau_i$ of the $i$-th strong classifier until the
            current classifier cascade has a detection rate of at least $t \times T_{i-1}$
            (this also affects $F_i$).
    4. Set $\mathcal{N} = \emptyset$
    5. If $F_i > F_{\text{Target}}$ then evaluate the current cascade detector on the set of
       non-face images and put any false detections into the set $\mathcal{N}$.

Output:
    Learned classifier cascade given by a set $\{H_s(\mathbf{x}) : s \in 1 \dots N_S = i\}$
    of strong classifiers $H_s(\mathbf{x})$ (or stages) that each contain $N_{Rs}$ weak classifiers
    with
        $H_s(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{r=1}^{N_{Rs}} \alpha_{rs} h_{r_s}(\mathbf{x}) \geq \tau_s \\ 0 & \text{otherwise} \end{cases}$
    and $\alpha_{rs} = \log \frac{1}{\beta_{rs}}$ .
```

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The output of Algorithm 3 is the learned classifier cascade represented by a set $\{H_s(\mathbf{x}) : s \in 1 \dots N_S\}$ of N_S stages or strong hypotheses $H_s(\mathbf{x})$ that each contain N_{Rs} weak classifiers with

$$H_s(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{r=1}^{N_{Rs}} \alpha_{rs} h_{r_s}(\mathbf{x}) \geq \tau_s \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

and $\alpha_{rs} = \log \frac{1}{\beta_{rs}}$.

Figure 3.6 presents a flowchart illustrating the cascade learning process of the object detection framework that has been developed as part of this thesis.

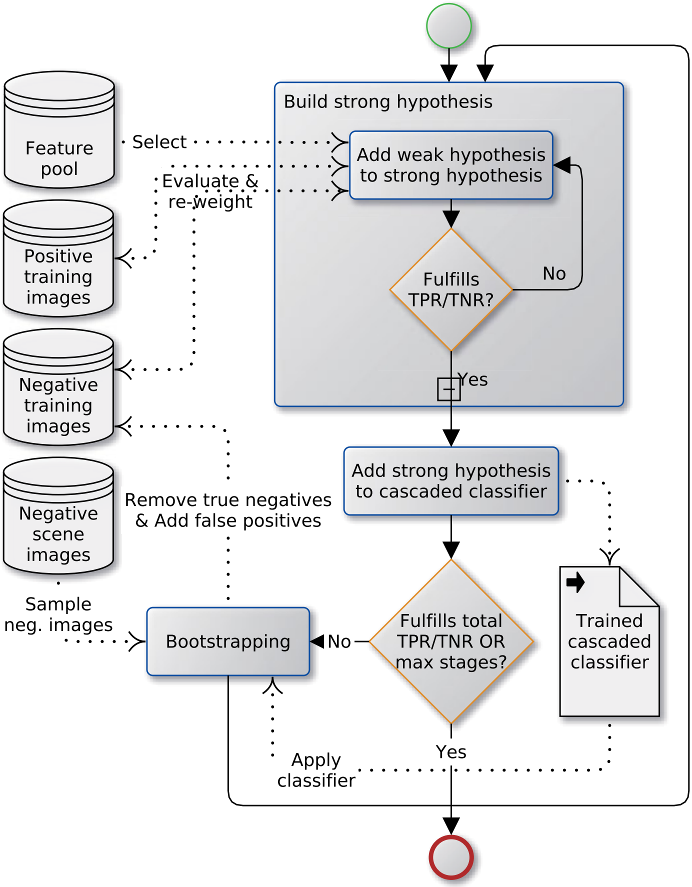


Figure 3.6: Flowchart illustrating the process of learning a cascaded classifier.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

3.2.3 Margin Analysis

Shortly after the publication of the AdaBoost algorithm research has been started to examine its classification performance. As the consideration of only the training error is not sufficient to estimate the test error of AdaBoost, Schapire et al. [130] proposed the margin as a measure of the confidence in the algorithms classification. They defined the classification margin of a training example $(\mathbf{x}_i, y_i)$ as the difference between the sum of the weights of the weak classifiers voting for the correct object class and the maximal sum of weights assigned to an incorrect class

$$\mathrm{margin}(\mathbf{x}_i, y_i) = \sum_{r : h_r(\mathbf{x}_i) = y_i} \alpha'_r - \arg \max_{y \in \mathcal{Y} \setminus y_i} \sum_{r : h_r(\mathbf{x}_i) = y} \alpha'_r \tag{3.18}$$

with $\alpha'_r = \dfrac{\alpha_r}{\sum_{j=1}^{N_R} \alpha_j}$.

As the weights α'_r are normalized to sum up to one, the margin is defined in the range $[-1, 1]$ and a positive value implies a correct decision. Hence a large positive margin represents a confident correct classification.

Evaluated on the complete training set a margin distribution can be derived. Schapire et al. observed that, due to its adaption of the training example weights, AdaBoost proceeds very aggressive in reducing the amount of training examples having a small margin. For this reason the AdaBoost learning algorithm can reduce the test error even after the training error has reached zero.

In the last decade, much research has been done in estimating the test error subject to the margin and in finding a boundary based on the minimum margin and other training parameter. But more recent research [90] indicates as well that considering the minimum margin is not sufficient to estimate the generalization capabilities of a boosted classifier and that the complete margin distribution has to be taken into account.

The distribution of margins over a complete training set can be visualized as a cumulative distribution giving a more detailed description of the training success than the training error by itself. Given the restricted domain [-1,1], the cumulative distribution represents for each value in this range the fraction of training examples whose margin is at most equal to the value. Figure 3.7 presents an example of a cumulative distribution. By definition of the margin, a value of -1 means that all weak hypotheses h_r of the boosted classifier vote with their associated weight α'_r for a wrong label and a margin of 1 is achieved when all hypotheses vote correctly. Hence, a margin of 0 represents the decision boundary if the weighted majority of the strong hypothesis classifies a training example correctly. From the cumulative margin distribution in Figure 3.7, it can be observed that about 14 percent of the training set is misclassified by the strong hypothesis learned after 2 training rounds. This fraction of wrongly classified examples is reduced to nearly zero after learning 100 rounds. The alteration of the curve shapes with increasing rounds demonstrates that AdaBoost focuses in learning to reduce the amount of examples having a small

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

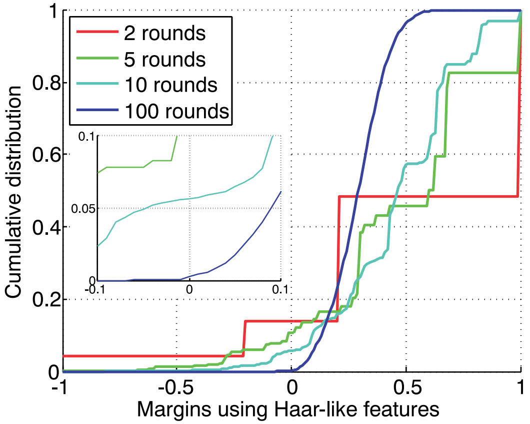


Figure 3.7: Cumulative margin distribution of face training set after 2, 5, 10, and 100 rounds of training.

margin. The fraction of examples with a very high margin also decreases as a side effect such that the curves intersect. But this allows the AdaBoost algorithm to continue to learn even after the training error has reached zero.

3.2.4 Variants of Boosting Algorithms

Since the introduction of AdaBoost, much research has been done to analyze its abilities as a learning algorithm and to identify strengths and weaknesses in order to propose improved variants. This section gives a brief overview on several directions of advancement. Schapire presents in [128] a summary on different perspectives that have been applied to provide an understanding of AdaBoost.

One approach is to analyze its learning capabilities with respect to the margins of the training examples. The margin analysis is described in Section 3.2.3 and used in this thesis to evaluate the training success in some experiments.

Another approach is to view AdaBoost as a procedure to minimize a loss function. In this way, a perspective on the learning algorithm is given that is auxiliary for the comparison and development of boosting variants. A slightly different notation is used for the AdaBoost algorithm in this section in order to provide a more convenient formula for the loss function. Differing from Section 3.2.2, the label of a training example for binary classification is specified as $y_i \in \mathcal{Y} = \{-1, +1\}$ with $y_i = -1$ for a negative and $y_i = +1$ for a positive example. Since the weak hypothesis is defined as $h_r : \mathcal{X} \rightarrow \mathcal{Y} = \{-1, +1\}$, the training error to be minimized by the weak

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

hypothesis is set slightly different as

$$\epsilon_r = \min_{\gamma,\rho,\theta} \sum_i w_{r,i} \cdot \frac{1}{2} \cdot |h(\mathbf{x}_i,\gamma,\rho,\theta) - y_i| \tag{3.19}$$

The strong hypothesis for predicting the class label then computes the sign of the weighted combination of the weak hypotheses

$$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{r=1}^{N_R} \alpha_r h_r(\mathbf{x})\right) = \operatorname{sign}\left(F(\mathbf{x})\right). \tag{3.20}$$

AdaBoost can be interpreted as a forward stagewise additive model that approximates the solution by sequentially adding new basis functions without adjusting the parameters and coefficients of the previously added basis functions [70]. The basis functions are the weak hypotheses and the additive model minimizes an exponential loss function [59, 70, 128]:

$$J(F) = \sum_i e^{-y_i F(\mathbf{x}_i)} \tag{3.21}$$

Following this line of research, Friedman et al. propose LogitBoost [59] that directly minimizes the logistic loss:

$$J(F) = \sum_i \ln\left(1 + e^{-2y_i F(\mathbf{x}_i)}\right) \tag{3.22}$$

The loss functions can be minimized with the help of another perspective on boosting algorithms as iterative functional gradient descent algorithms [104]. By this, e.g. the AnyBoost algorithms [104] minimize a cost functional by gradient descent that chooses linear combinations of elements, the weak hypotheses, of an inner product function space. Gentle AdaBoost [59] proposes an improvement to the gradient descent of AdaBoost by adaptive Newton steps obtained from the minimization of a weighted squared error at each step. As a consequence, the strength of the gradient descent and thus the weight assigned to a single hypothesis is bounded. Exceeding hypothesis weights resulting from outliers in the training data might otherwise have a detrimental effect on the generalization error [129].

An alternative optimization strategy is to choose loss functions that can be formulated as linear programs. Demiriz et al. [37] introduce LPBoost that directly maximizes the minimum margin by solving a linear programming problem. More recently, Warmuth et al.[150] propose an improved variant that additionally satisfies a logarithmic iteration bound by adding a relative entropy regularization.

3.3 Data Analysis

This section presents an introduction in unsupervised machine learning techniques that are jointly utilized in this thesis together with the supervised learning strategies

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

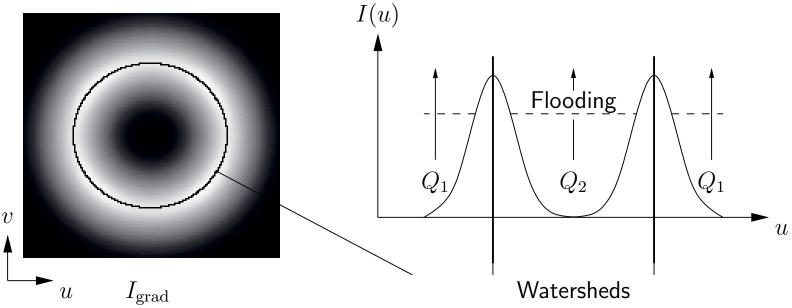


Figure 3.8: Illustration of the basic principle of watershed segmentation. A gradient image I_{grad}, shown in the left figure, is interpreted as a topographic relief. The relief is flooded from water sources Q located at the local minima of the relief as illustrated in the right figure that represents the horizontal cut through the center of the left image. Barriers, denoted as watersheds, are build at positions where waters originated from different sources meet. Images taken from [27].

presented in Section 3.2. The methods for data analysis, that are described in the following, explore the data in an unsupervised process in order to discover knowledge about the structure of the data. Section 3.3.1 presents cluster algorithms and Section 3.3.2 briefly describes the Principal Component Analysis (PCA).

3.3.1 Cluster Analysis

As part of this work, a distributed machine learning framework is developed that follows the training procedure explained in Section 3.2.2. The weak hypotheses are learned from sets of features that represent image regions or gradient informations similar to the common features described in Section 3.1. Feature mining strategies are introduced in Chapter 7 in order to create feature sets that are adapted to characteristic structures of the object class to be detected. This process involves the image segmentation of the training data that is performed by the cluster algorithms presented in the following. Additionally, cluster algorithms are utilized for the Non-Maximum Suppression in the post-processing of the object detector.

Watershed The watershed method was first formulated in 1978 by Digabel and Lantuéjoul [39, 84] followed by numerous improvements and modifications. In this work, the watershed algorithm of Fernand Meyer [108] is utilized for image segmentation. The basic principle of the algorithm can be very intuitively described and is illustrated in Figure 3.8:

The gradient image of a gray scale image is computed and interpreted as a topographic relief so that pixel values in the gradient image are considered as altitude

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

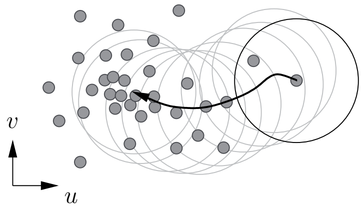


Figure 3.9: Basic principle of the meanshift algorithm. Starting from a data point, the mean in a local window is computed and the window is shifted to that mean. This procedure is repeated until the mean converges to a local area of highest density. Image taken from [27].

information. In the following, the relief is flooded starting from water sources placed at each local minima of the relief. At the positions where waters originated from different sources meet, barriers are put into place that separate the water basins. These barriers are called watersheds and represent the boundaries of segmented image regions that are oriented along edges in the original image. This procedure might result in an oversegmentation if too many local minima are found as water sources. A gradient image might be as well obtained from a binary edge image. The strength of segmentation can then be controlled by the sensitivity of the edge detector. The gradient image is derived from the edge image by a distance transform that replaces the value of each pixel by its distance to the nearest edge pixel. The complement to the transformed image represents the topographic relief that is flooded by the watershed method.

Meanshift The meanshift algorithm was originally proposed in 1975 [61] as mode-seeking procedure that finds the maxima of a density function. The principal concept of the meanshift algorithm is illustrated in Figure 3.9. For the application as cluster algorithm, the data points to be clustered are assumed to be sampled from a probability density function. Starting from a data point, the algorithm searches for a mode, a local maximum of the density function. For that reason, the weighted mean is computed in a local window around the data point given by a kernel function and the window is shifted to this mean. In this way, the window is moved in the direction of the maximum increase in density. This procedure is repeated until the window converges to a local maxima of the density function. The size of the window is in the process controlled by a bandwidth parameter of the kernel. A cluster is then given by the set of all data points that converge to the same mode. An advantage of meanshift clustering over k-means, that is described in following, is that the number of clusters needs not to be specified in advance. The selected bandwidth parameter steers the method such that a smaller bandwidth tends to result in a higher number of defined clusters.

Comaniciu and Meer [31] proposed in 2002 a method for image segmentation based

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

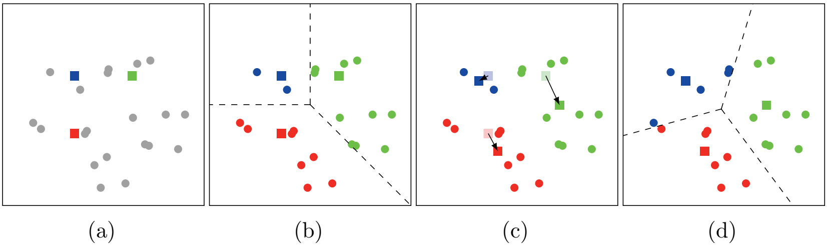

(a) (b) (c) (d)

Figure 3.10: Iterations of the standard k-means algorithm. (a) k cluster centers (colored squares) are randomly selected as initialization. (b) Data points are assigned to the cluster center that is in the shortest distance. The line segments of the Voronoi diagram based on the cluster centers are shown as dashed lines. (c) All cluster centers are shifted to the mean coordinates of all data points belonging to the corresponding cluster. (d) The data points are reassigned based on the updated cluster centers. Step (c) and (d) are repeated until the cluster assignment converges.

on the meanshift algorithm.

K-means The standard algorithm of k-means clustering, also referred to as Lloyd's algorithm, was developed by Stuart P. Lloyd in 1957 but it was first published by Lloyd in 1982 [98]. A more efficient version of the algorithm was proposed by Hartigan and Wong in 1979 [69].

In the case of a discrete set of data points, the objective of k-means clustering is to assign each data point to one of k clusters such that the sum of all squared distances between data points and the center or mean of their associated cluster is minimized. The data points are in this way partitioned to minimize the intra-class variance. The number k of requested clusters is in the process an input parameter of the algorithm.

Several improved variants of the algorithm exist that e.g. enhance the initial selection of the cluster means. In the following, the standard algorithm that randomly initializes the means is briefly described and illustrated in Figure 3.10.

Starting from an initial selection of means (see Figure 3.10a), repeatedly all data points are assigned to the mean that is in the shortest distance (see Figures 3.10b and 3.10d) and the means are updated to be the centroids of the data points assigned to each cluster (see Figure 3.10c). These steps are repeated until the cluster assignment and thus the means converge. Due to the random initialization, the algorithm is not deterministic.

Regarding images, the data points can represent pixel and the distances to the cluster centers might be computed as distance of pixel values in color space and/or the spatial distance of the pixel's position in the image.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

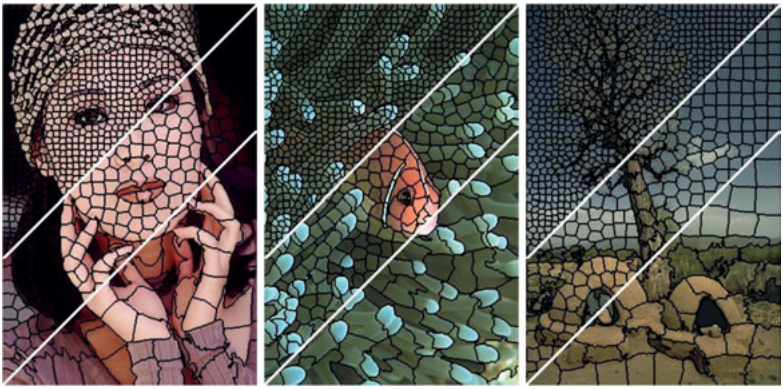


Figure 3.11: Examples of image segmentations created by the SLIC superpixel algorithm [2]. The SLIC algorithm has been parametrized to segment each image into superpixels of approximately 64, 256, and 1024 pixels size from top left to bottom right. The images are taken from [2].

Superpixel The concept of superpixels is originally developed by Ren and Malik [124] as a local, coherent and structure preserving over-segmentation. The Simple Linear Iterative Clustering (SLIC) [2] superpixel algorithm is closely related to k-means clustering such that it assigns the image pixel in an iterative procedure to a priorly specified number of superpixels. The assignment is based on a distance measure that is defined in the five-dimensional space of the pixels' color vector in CIELAB color space and the pixel position in image coordinates. Figure 3.11 presents examples of superpixel segmentations created by the SLIC algorithm.

A disadvantage of the iterative procedure in SLIC is that even after convergence superpixels can be separated into not connected small fragments. Hence, SLIC has to perform a post-processing that enforces connectivity but does not provide an optimal solution.

The superpixel algorithm of Schick et al. [132] uses a different optimization procedure so that the problem of not connected fragments does not occur. Furthermore, the algorithm allows to transparently control the compactness of the superpixels.

In this work, the algorithm of Schick et al. [132] is applied for superpixel segmentation.

3.3.2 Principal Component Analysis

The Principal Component Analysis (PCA) is utilized in this thesis for a statistical analysis of the positive object class. A method is proposed in Chapter 5 to augment sparse training data with respect to the analysis.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

A PCA [116] transforms (possibly) correlated variables into linearly uncorrelated variables called principal components by applying an orthogonal transformation. The transformation is designed in such a fashion that the first axis (principal component) represents the highest amount of data variation, whereas the other following orthogonal axes are sorted in a decreasing order, depending on the amount of variance. A PCA is simply computed as a singular value decomposition of a data matrix or by an eigendecomposition of a covariance matrix generated from a data set that for the method proposed in Chapter 5 consists of the positive training images. The eigenvectors are derived in this work from a singular value decomposition of the covariance matrix. Since that covariance matrix is symmetric and positive semidefinite, the obtained eigenvectors are identical to those provided by a PCA and also the order of the corresponding eigenvalues is the same.

Similar to the notations in [142], n-dimensional data points $\mathbf{x}_1 \ldots \mathbf{x}_{N_{\mathcal{P}}} \in \mathcal{P}$ are assumed and the mean of the data points is computed as

$$\Psi = \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} \mathbf{x}_i \tag{3.23}$$

Further, $\overline{\mathbf{x}}_i = \mathbf{x}_i - \Psi$ is defined to be the difference to the mean vector that shifts the data in this way towards the origin. The matrix $\overline{X} = [\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_{N_{\mathcal{P}}}]$ contains the difference vectors of the data, so that the covariance matrix of the data points is given as

$$\begin{aligned} C &= \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} \overline{\mathbf{x}}_i \overline{\mathbf{x}}_i^T && (3.24) \\ &\propto \overline{X}\,\overline{X}^T && (3.25) \end{aligned}$$

A singular value decomposition $C = U \Sigma V^T$ of the covariance matrix C allows to compute the principal components of the data. Note that the computation of $\overline{X}^T \overline{X}$ can be much more efficient if less data points are available than the dimension of the data (see [142] for details). This happens e.g. when images are encoded as vector and only a few images (e.g. a couple hundred) are available. Then U needs to be multiplied by $\overline{X}$ and rescaled to get the eigenvectors and the first s eigenvectors can be used for approximation of the space.

Figure 3.12 shows two simple examples in which the missing parts of a corrupted face and cell image (not contained in the training data) has been reconstructed with the help of the PCA of a data set by performing a subspace projection. As can be seen, the position of the nose and mouth as well as the cell boundaries have been reconstructed fairly well.

There exist many extensions and modifications about PCA-methods for data clustering. In Chapter 5, Kernel Principal Component Analysis (KPCA) [134] methods are applied as well. The idea for KPCA is to employ a mapping $\phi(\overline{\mathbf{x}})$ on the data to lift the input to a higher dimensional space, in which the subspace can be approximated more easily. Since the higher dimensional space can become very large, they

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

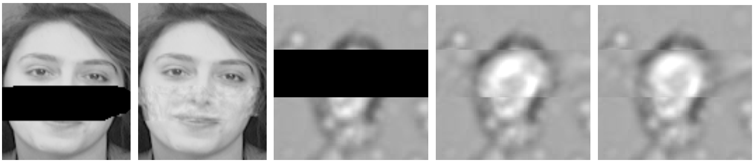


Figure 3.12: Missing data estimation of face image (on the left) and a cell image (on the right). No blending was performed and it is shown that the mouth and nose as well as the cell structure is well approximated. The reconstruction of the cell to the right is based on a Kernel Principal Component Analysis (KPCA). For this kind of data, it seems to approximate the cell boundaries slightly better [43].

key idea behind KPCA is to avoid the explicit computation of ϕ and to work with a kernel $k(\overline{\mathbf{x}}_i,\overline{\mathbf{x}}_j) = \langle \phi(\overline{\mathbf{x}}_i),\phi(\overline{\mathbf{x}}_j) \rangle = \phi(\overline{\mathbf{x}}_i)^T \phi(\overline{\mathbf{x}}_j)$. The covariance matrix C then becomes

$$C \;=\; \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} \phi(\overline{\mathbf{x}}_i)\phi(\overline{\mathbf{x}}_i)^T \tag{3.26}$$

which is again split and normalized after a Singular Value Decomposition (SVD), $C = U\Sigma V^T$. In this thesis, two standard kernels are used, namely

$$k_1(\mathbf{x}_i,\mathbf{x}_j) = -\exp((\mathbf{x}_i - \mathbf{x}_j)^2/(\sigma)^2)$$

with $\sigma = 1$ and

$$k_2(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2.$$

Depending on the selected kernel and its non-linear mapping, the KPCA allows to structure the data set along principal components that are not available in the linear approach of the PCA.

3.4 Detector Performance Measures

This section discusses commonly used measures to evaluate the performance of object detectors for the binary classification problem.

True Positive Rate (TPR) or recall The TPR is also referred to as recall or sensitivity. It represents the proportion of positive samples that are correctly classified

$$\text{True Positive Rate} = \text{Recall} = \frac{N_{\text{TP}}}{N_{\text{P}}} \tag{3.27}$$

with N_{TP} denoting the number of True Positive (TP) detections and N_{P} the total number of positive objects that have to be detected in a test set.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

On the contrary, the quota of misclassified positive samples is denoted as miss rate and given by

$$\text{Miss rate} = 1 - TPR = 1 - \frac{N_{\text{TP}}}{N_{\text{P}}} \tag{3.28}$$

Precision The precision measures the fraction of all positively classified samples that are correctly classified as such

$$\text{Precision} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}} \tag{3.29}$$

with the number of False Positive (FP) detections given as N_{FP}.

False Positive Rate (FPR) The FPR measures the percentage of negative samples that are wrongly classified as positive

$$\text{False Positive Rate} = \frac{N_{\text{FP}}}{N_{\text{N}}} \tag{3.30}$$

with N_{N} denoting the total number of negatives in the test set. But the determination of N_{N} is problematic in the task of visual detection of objects in images. Often a sliding window is used to extract samples from a scene image in order to classify them. Hence, the number of negatives provided to the classifier depends on the parametrization of the sliding window. Furthermore, N_{N} is complicated to compute in practice because test sets in general specify some neighborhood in space and scale around the true object position in which a detection is considered as True Positive (TP). But the number of samples in this neighborhood depends as well on the parametrization of the sliding window and has to be subtracted from the total number of sliding window samples for a exact computation of N_{N}. Commonly, the number of samples in these neighborhoods is considerably smaller than the total number of samples such that the latter is chosen to be N_{N} to simplify matters. A FPR that is computed with respect to the sliding window samples is also denoted as False Positives Per Window (FPPW).

An alternative measure that avoids the problematic dependency on the sliding window is False Positives Per Image (FPPI) representing the ratio of N_{FP} to the number of scene images in the test set.

F-measure ($\mathbf{F_1}$) The F-measure or $\mathbf{F_1}$ score is the harmonic mean of precision and recall:

$$\mathbf{F_1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot N_{\text{TP}}}{N_{\text{P}} + N_{\text{TP}} + N_{\text{FP}}} \tag{3.31}$$

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Detector performance curves In order to evaluate the performance of a detector on a test set, it is often not sufficient to consider a single value as quality measure. Commonly, a performance curve is generated instead that allows a more detailed analysis of the detection characteristic. A performance curve can be created by adjusting an internal parameter controlling the detector's selectivity. In this way, the detection rate of a detector can be raised or lowered having the side effect that in general the FPR increases or decreases as well. The detection results of multiple detector runs with different parametrization are in doing so consolidated in a single curve. Some benchmarks provide evaluation tools that do not allow multiple detector runs in order to create a performance curve. Instead, these benchmarks specify a evaluation methodology that commonly requires that the detector generates a list of detections with an assigned confidence value. A performance curve is then generated by traversing all confidence values and identifying TPs and FPs considering only detections that have a confidence that is greater or equal than the currently selected confidence.

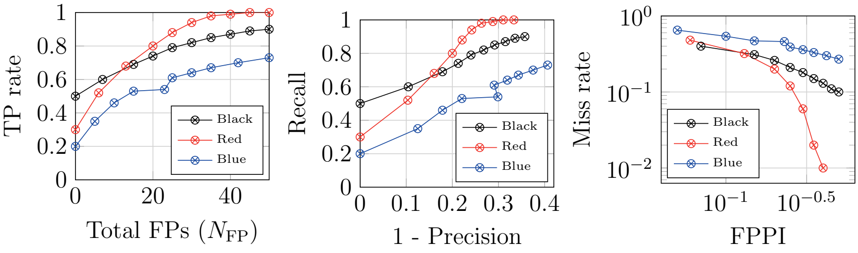
An example of such curves is the Receiver Operating Characteristic (ROC) that presents the True Positive Rate as a function of the False Positive Rate. The comparison of sliding window based detectors using ROC curves holds the problem that the FPR and more specific N_N is highly dependent on the parameter that controls the sample density of the sliding window as previously described. Hence, the FPR, also denoted as False Positives Per Window (FPPW), is frequently substituted with False Positives Per Image (FPPI) or the total number N_FP of FPs.

Figure 3.13a shows an example of the latter variant of a detector performance curve and illustrates the advantages over a single-valued quality measure. Though the blue curve is obviously showing the least performance, the decision if the black or red curve demonstrates a better performance is application specific. If False Positive detections are not permitted, the black curve should be preferred, while the red curve is more appropriate when a high True Positive Rate is desired and False Positive detections are acceptable. It is difficult to represent this distinction in a single-valued measure.

Figures 3.13b and 3.13c present the same detection results as in Figure 3.13a but use different performance measures. Precision and recall are utilized in Figure 3.13b showing a similar appearance but the blue curve demonstrates that performance curves based on the precision can be less smooth. Figure 3.13c illustrates the detection results in the same way as the Caltech pedestrian detection benchmark [42] presenting the miss rate as a function of False Positives Per Image.

Log-average miss rate The Caltech pedestrian detection benchmark uses the log-average miss rate to represent an entire detector performance curve in the variant shown in Figure 3.13c by a single reference value. It is essentially the average of miss rates corresponding to nine FPPI rates that are evenly spaced in log-space in the interval 10^{-1} to 10^{0} [42].

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b) (c)

Figure 3.13: Detector performance curves representing the same detection results. The results are artificial examples supposing that 100 object have to be detected in 100 images. Different numbers of TP and FP detections are assumed for the three detectors and the performance measures are computed accordingly. The performance curves present (a) the True Positive Rate as a function of the total number N_{FP} of FPs, (b) Recall as a function of 1 - Precision, and (c) the miss rate as a function of False Positives Per Image.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter

4

Distributed Machine Learning Framework

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

This chapter gives a brief introduction in the distributed machine learning framework that has been developed as part of this work.

One topic of this thesis is the research on new feature types that are applied in the learning of object detectors. Hence, Chapter 7 introduces two complementary feature types and demonstrates the benefit of learning a mixed detector that utilizes them. But a mixed learning makes specific demands on the software architecture of a machine learning framework written in C++ as the feature objects should be runtime polymorph. Especially in distributed systems that have no shared memory, a sophisticated design is required to dynamically dispatch the methods provided by the feature objects.

The machine learning framework is written in C++ and transparently handles different feature types in a distributed system as the objective in development has been to build a generic framework that is able to process a wide range of scalar and vector-valued feature types.

In the following, the structure of the developed distributed framework is presented and the mechanisms to organize a massive parallel object detector learning are described. Furthermore, a brief insight into the class architecture of the framework is given in terms of runtime polymorphism, efficient computation and distributed computing.

The distributed machine learning framework is able to parallelize the training of an object detector on a computer cluster. The MPI and in particular the OpenMPI [62] library is used for inter-node communication. OpenMPI is accessed by the Boost.MPI [65] library that provides are more convenient interface to MPI and above all applies Boost.Serialization [122] to support the transmission of complex user-defined data types and C++ Standard Library types. The inter-node communication is controlled by a communication layer that implements a signal and slots mechanism using message queues. In that way, program objects are allowed to asynchronously communicate among distributed computing nodes. A hybrid model of parallel programming is implemented such that shared memory multiprocessing on each computing node is accomplished by the Open Multi-Processing (OpenMP) API [112]. The topology of the computing cluster designates one node the elevated position as master node that coordinates the machine learning task and assigns working packages to the worker nodes. To simplify communication, messages are only transmitted from the master node to the worker nodes and vice versa but not between worker nodes. Figure 4.1 presents the structure of a cluster consisting of the master node and two worker nodes.

The master node mainly controls the machine learning task and does hardly any computation. Each node possesses an algorithm state machine that implements the machine learning algorithm and an instruction object that enables every software component to communicate with the master node. The machine learning task is parallelized on the worker nodes applying multiple threads but a shared memory. In this way, all threads on each machine share one training object pool that are

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

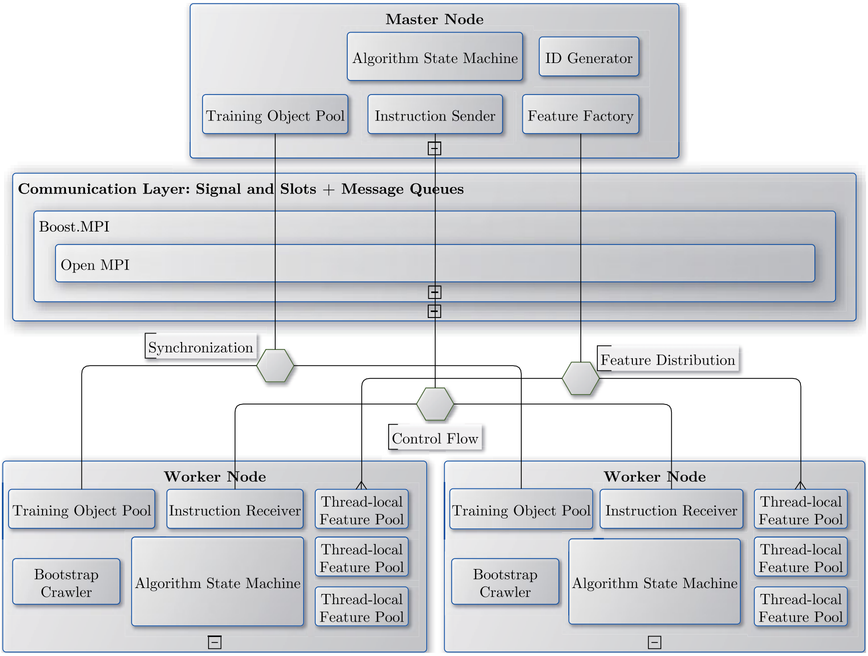


Figure 4.1: Simplified structure of a computing cluster consisting of the master node and two worker nodes. All inter-node messages are handled by a communication layer incorporating Boost.MPI and OpenMPI. Each node contains an algorithm state machine, an automatically synchronized training object pool and an instruction handler. Each algorithm state machine is thread-parallelized having thread-local feature pools that are directly supplied with features by the feature factory of the master node.

automatically synchronized with the pool on the master node. The parallelization of the machine learning task is performed on feature level such that every thread possesses its own feature pool that is directly provided with features by the feature factory of the master node. A global ID generator in the master node takes care that every object in the distributed memory system is clearly recognizable.

The framework is designed with a strong emphasis on generic programming transparently handling different feature types. To maintain program efficiency, static typing is applied where possible and dynamic typing when needed [107]. From the feature base class four intermediate base classes are inherited that differ in the type of the return value computed by the feature. To support a wide range of possible features, the specified return types are integer and double scalar as well as vector of integers and vector of doubles. The implemented features classes are then in-

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

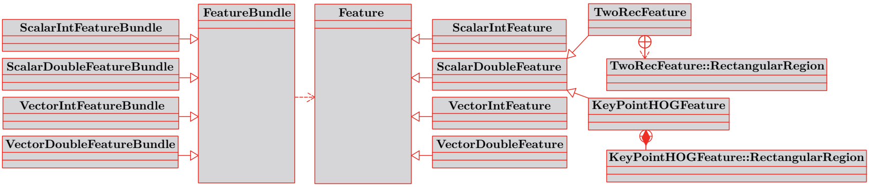


Figure 4.2: Class diagram showing the organization of feature objects in `FeatureBundle`. `FeatureBundle` contain only features of homogeneous type to the effect that their return value is of the same type. `FeatureBundles` as well as feature objects are therefore derived from four base classes each that declare virtual methods having the specific return type. The `TwoRecFeature` and `KeypointHOGFeature` have the same return value type and base class. Both provide the rectangular image region on which they are computed as a subclass object.

herited from one of these intermediate base classes. The usage of dynamic typing is reduced by the introduction of feature bundles that are objects containing only homogeneous feature types to the effect that their return value is of the same type. Thus four different subclasses of feature bundles are inherited as well representing the return types of their contained features. In that way, feature computation can be performed by a feature bundle calling a virtual method of the containing features that they have inherited from their intermediate base class. Dynamic typing is applied on the level of feature bundles by a virtual method that is declared in the feature bundle base class and triggers the computation of all features contained in a bundle. This virtual method returns a Boost.Variant [58] data type. Boost.Variant is a generic union container that can include any kind of type taken from a predefined set of possible types. Here the predefined types as well represent the different return types of the feature classes. The explicit value that a Boost.Variant contains can be checked and retrieved at run-time realizing dynamic typing. In that way, the computation of features is independent of the explicit feature class transparently handled by calling this virtual method of the feature bundle class. The computational overhead is little as dynamic typing is only required once per feature bundle. Figure 4.2 shows the classes involved in the computation of the features types used in this work. The previously described virtual methods have been omitted in the class diagram because their signature is too long for presentation.

For a generic framework that transparently processes arbitrary feature object types, run-time polymorphism of these objects is essential. This can be achieved in C++ by accessing objects of a subclass type by means of pointers to a common base class and calling virtual functions declared in that base class. The application of pointer is obviously problematic in a distributed memory environment but the Boost.Serialization library allows to solve this problem. Boost.Serialization is able to follow the base class type pointers and to serialize the derived subclass object without slicing it to its base class. By this, polymorph object types can be transmitted

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

via MPI among cluster nodes without knowing the exact data type at compile time. Such polymorph objects are organized by the machine learning framework in object pools that access them by means of pointers to their common base class `PoolObject`. To allow deep copies of these run-time polymorph data types, a clone interface is implemented. The `PoolObject` base class declares a virtual method `PoolObject* clone()` that is overwritten in the derived classes by a method having a covariant return type: `DerivedClass* clone()` . In that way, the overwritten `clone()` methods ensure that complete deep copies of objects are created which exact data type is unknown at compile time. In addition, the `PoolObject` base class inherits an object of type `ObjectId` that is set by the ID generator of the master node to clearly identify every `PoolObject` in the computing cluster. The architecture of the object pool together with the `PoolObject` base class and as an example its derived `TrainingImage` subclass is presented in Figure 4.3. The `TrainingImage` subclass implements the clone interface and therefore overwrites the virtual `clone()` method with its own having a covariant return type.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

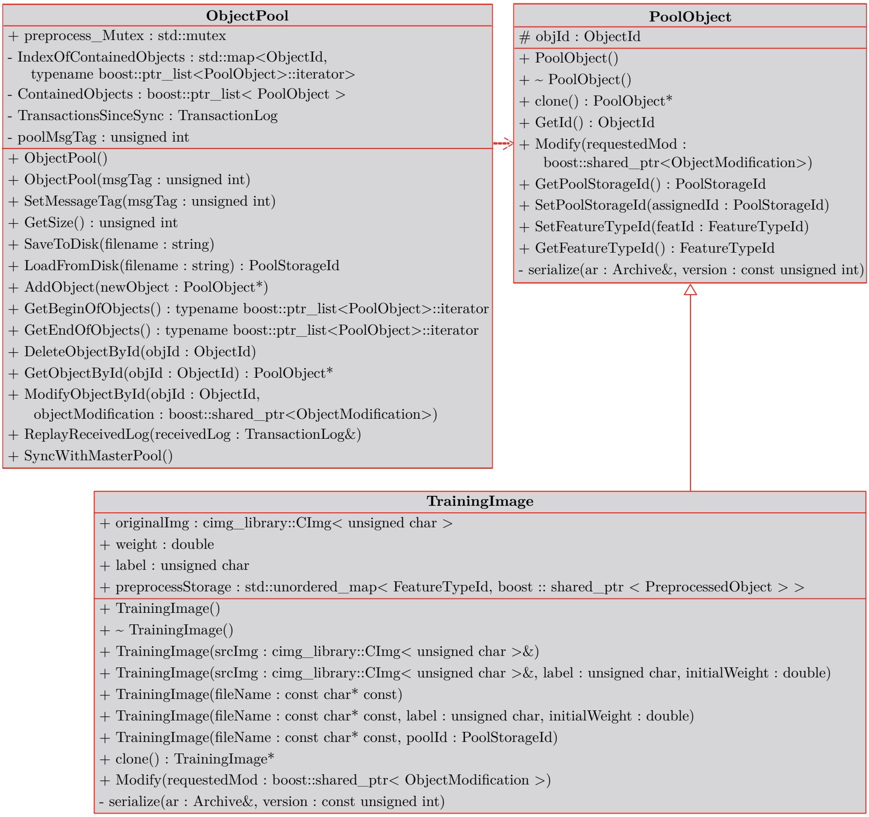

ObjectPool
+ preprocess_Mutex : std::mutex
- IndexOfContainedObjects : std::map<ObjectId, typename boost::ptr_list<PoolObject>::iterator>
- ContainedObjects : boost::ptr_list< PoolObject >
- TransactionsSinceSync : TransactionLog
- poolMsgTag : unsigned int
+ ObjectPool()
+ ObjectPool(msgTag : unsigned int)
+ SetMessageTag(msgTag : unsigned int)
+ GetSize() : unsigned int
+ SaveToDisk(filename : string)
+ LoadFromDisk(filename : string) : PoolStorageId
+ AddObject(newObject : PoolObject*)
+ GetBeginOfObjects() : typename boost::ptr_list<PoolObject>::iterator
+ GetEndOfObjects() : typename boost::ptr_list<PoolObject>::iterator
+ DeleteObjectById(objId : ObjectId)
+ GetObjectById(objId : ObjectId) : PoolObject*
+ ModifyObjectById(objId : ObjectId, objectModification : boost::shared_ptr<ObjectModification>)
+ ReplayReceivedLog(receivedLog : TransactionLog&)
+ SyncWithMasterPool()
PoolObject
# objId : ObjectId
+ PoolObject()
+ ~ PoolObject()
+ clone() : PoolObject*
+ GetId() : ObjectId
+ Modify(requestedMod : boost::shared_ptr<ObjectModification>)
+ GetPoolStorageId() : PoolStorageId
+ SetPoolStorageId(assignedId : PoolStorageId)
+ SetFeatureTypeId(featId : FeatureTypeId)
+ GetFeatureTypeId() : FeatureTypeId
- serialize(ar : Archive&, version : const unsigned int)
TrainingImage
+ originalImg : cimg_library::CImg< unsigned char >
+ weight : double
+ label : unsigned char
+ preprocessStorage : std::unordered_map< FeatureTypeId, boost :: shared_ptr < PreprocessedObject > >
+ TrainingImage()
+ ~ TrainingImage()
+ TrainingImage(srcImg : cimg_library::CImg< unsigned char >&)
+ TrainingImage(srcImg : cimg_library::CImg< unsigned char >&, label : unsigned char, initialWeight : double)
+ TrainingImage(fileName : const char* const)
+ TrainingImage(fileName : const char* const, label : unsigned char, initialWeight : double)
+ TrainingImage(fileName : const char* const, poolId : PoolStorageId)
+ clone() : TrainingImage*
+ Modify(requestedMod : boost::shared_ptr< ObjectModification >)
- serialize(ar : Archive&, version : const unsigned int)

Figure 4.3: Class diagram presenting the object pool architecture. `TrainingImages` as well as feature objects are organized in `ObjectPool` components. They are therefore derived from the base class `PoolObject` that declares besides others a clone interfaces and unique `ObjectId`. To ensure run-time polymorphism, the `ObjectPool` handles pointers to the `PoolObject` base class.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 5

Learning from Sparse Training Data

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

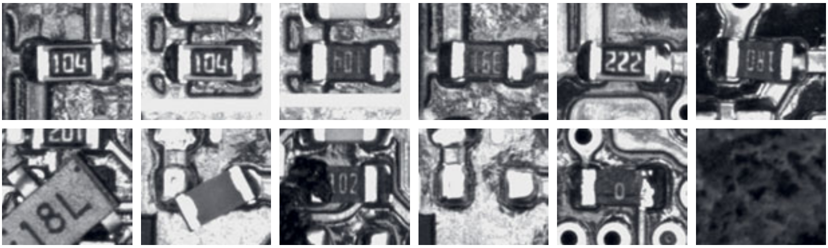


Figure 5.1: Example images of SMDs. Top row: Good components that are properly soldered. Bottom row: Examples of missing or defective components that are wrongly placed or have faulty solder joints.

This chapter deals with the problem of sparse training data that is not uncommon e.g. in industrial applications. Especially classifiers for quality assurance are confronted with the objective to detect very rare occurrences with a high TPR. But also a high precision is desired since every production sample that is wrongly classified as defective results in an economic damage. An example application in quality assurance is the visual inspection of Surface-Mount Devices (SMDs) placed on circuit boards shown in Figure 5.1 in which boards with defective components have to be picked out after the assembly line.

Sparse training sets containing only a small amount of positive samples, that represent the object class to be detected, often result in poor classification performance if a classifier is learned from these examples. The Principal Component Analysis (PCA) described in Section 3.3.2 allows to obtain a model of the positive object space. In case of sparse training sets, it can be observed that negative training images projected into the objects PCA-space are often far away from the object class. This broad boundary between the object classes in training can yield to a high classification error of boosted classifiers on the test set.

Hence, the basic idea of the following approach is to narrow the boundary by augmenting the training set based on a model obtained from a PCA in order to improve the detection performance. This chapter is based on a published conference article [43].

Prior Work

The face detection method of Turk and Pentland [142] demonstrated that an object space, learned by a PCA from a training set of face images, provides a suitable model for face detection and recognition. The idea behind a PCA-space of objects is to learn a global subspace from training data which span the object variations as principal components in a high-dimensional vector space. In order to detect or recognize objects, a input image is projected into the learned subspace and distances to the training data in that space are evaluated [142]. Subsequently, other variants

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

to learn an object space like Independent Component Analysis (ICA) [10] or KPCA [134], described in Section 3.3.2, have also been proposed for face detection and recognition.

In the case of boosting-based object detectors, other approaches exploit the representation in an object space by means of features that are evaluated in that space [157, 4]. But a drawback of these approaches are the computational costs since all input images need to be projected in the object space for the detection process.

Contribution

The method proposed in this chapter is inspired by the capabilities of an evaluation in object space as presented in the prior work but intends to avoid the projection in the object space during detection. So the key contribution described in Section 5.1 is to modify and augment the training data of the non-objects in such a way that they are closer to the PCA-space of objects and therefore to cause a much smaller margin (see Section 3.2.3) at the start of training. This is in some way contrary to approaches in semi-supervised learning [88] in which the negative training class is raised at the boundary being opposite to the object class. Overall, it allows to train a much more selective classifier, especially if only a sparse amount of training data is available. Since the training data is enhanced prior to the classifier learning, the method is self-contained such that it is also applicable for e.g. other boosting variants presented in Section 3.2.4.

5.1 Training Data Augmentation

Often, negative training examples are not always well chosen to differ between objects and non-objects. This is mainly due to the fact that the non-object space is significantly larger and more complex than the positive examples. Basically, the non-objects can be seen as the complementary space to the learned PCA object space. Therefore, its variability is hard to reflect in the training data. The idea is to bring the training data close to the PCA-space. This can simply be done by projecting the negative training examples onto the trained PCA-space and then shifting it back towards the non-object space with a scale $\lambda \in [0 \ldots 1]$:

Let $U_s = U(1:n,1:s), s \leq n$ be the upper left matrix of U stemming from $C = U\Sigma V^T$ and let $\Lambda \in \mathcal{N}$ be an example of the non-object class. The shift of Λ towards Λ_s being closer to the object space can simply be done by computing

$$\begin{aligned} \text{Proj} &= U_s^T \cdot (\Lambda - \Psi) && (5.1) \\ \text{Rec} &= U_s \cdot \text{Proj} + \Psi && (5.2) \\ \Lambda_s &= \text{Rec} + \lambda(\Lambda - \text{Rec}) && (5.3) \end{aligned}$$

Note that U is an unitary matrix and thus U_s describes the inverse projection of U_s^T. In case of the more efficient computation mentioned in Section 3.3.2 this property

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

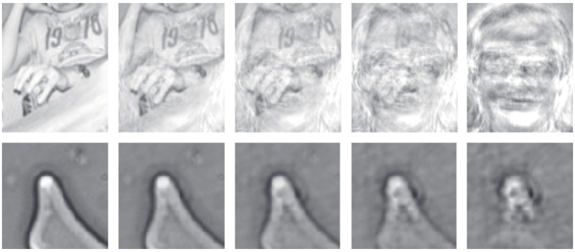


Figure 5.2: Morphing a non-object towards the PCA-learned object space. Top row: Example morph for a face space. Bottom row: Example morph for a cell space [43].

is not given and instead a pseudoinverse has to be used. Obviously, $\lambda = 0$ yields the projection on the PCA-space, whereas $\lambda = 1$ leads to the training example itself. So λ steers the amount on how much the example is shifted towards the object space. In the following experiments, projection strength parameter $\lambda = 0.3, 0.5$, and 0.7 are applied.

Figure 5.2 shows two non-object examples which are morphed with different weighting factors towards their object space, namely a non-face towards face space in the top row and a defective cell towards cell space in the bottom row. The middle images are non-objects which are much better suited to learn a boundary between the positive and negative classes in the boosting framework.

5.2 Experimental Results

First, experiments are conducted for face detection using the AT&T Face Database. The second set of experiments is performed on microscopic images from the cryo cell data set. Both data sets are described in Section 2.2 and divided into a training and validation set using a 67/33 ratio. Crowther and Cox [33] illustrated that especially for small bases a split containing only a small part for validation is not recommendable. They suggested to select a ratio between 50/50 and 70/30.

Figure 5.3 shows in the top row example images of non-faces and non-cells. The middle row shows morphed images towards the trained PCA-space. These images are then used for training to find a more selective classifier. The bottom row shows positive example images of faces and cells of the used databases.

5.2.1 Experiments on Face Detection

Using the approach described in Section 5.1, four different training sets are generated containing the object/non-object examples and morphed non-object examples with $\lambda = 0.3, 0.5$ and 0.7. A strong classifier is learned for all four data sets by

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

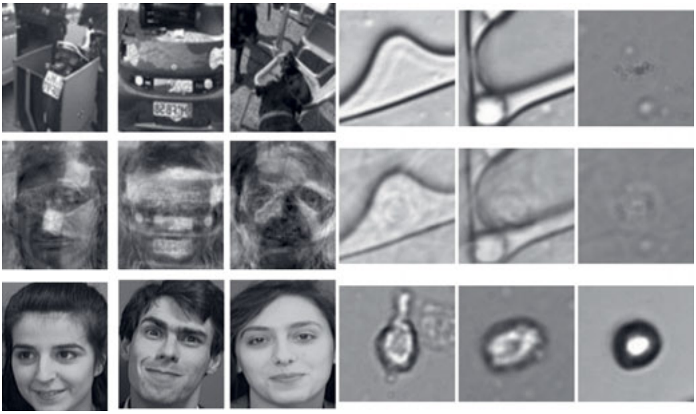


Figure 5.3: Top row: Example images of non-faces and non-cells. Middle row: Morphed images towards the trained PCA-space. These images are either used to find a more selective classifier. Bottom row: Example faces and cells of the used data sets [43].

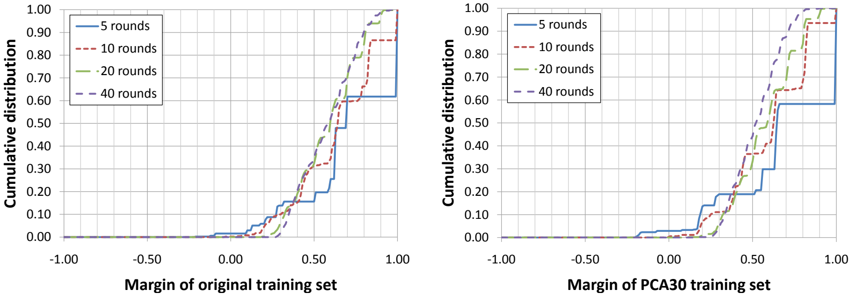
AdaBoost as described in Section 3.2.1 using Haar-like features that are introduced in Section 3.1.1. The impact of the classification confidences during training on the test error is subject to the margin theory that is briefly described in Section 3.2.3. In the following, margin distributions are evaluated while learning classifiers from the original and an augmented training set in order to analyze the training success.

Margin Analysis

Figure 5.4a presents the cumulative margin distributions after different training rounds for the original training set consisting of face and non-face images. In Figure 5.4b, margin distributions are shown when learning a classifier from the face data set using PCA-enhanced non-faces with $\lambda = 0.3$. The impact of the learning process of AdaBoost on the margin distributions is clearly noticeable in both figures. The amount of training examples having a small margin is in both cases strongly reduced during training. Roughly after 10 rounds, the training error reaches zero as all examples images have a positive margin and hence are correctly classified. Then the AdaBoost algorithm further concentrates on the training examples that are hard to classify and continues to reduce the number of training examples that are correctly classified with only low confidence.

But it is also observable that it is more difficult to classify the morphed training set. After 5 training rounds the boosted classifier for the morphed set makes almost twice as much wrong decisions compared to the classifier boosted on the original training set. Also about 15 % and 30 % of the training examples on the morphed set

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b)

Figure 5.4: Cumulative margin distributions (a) of the original face training set after 5, 10, 20, and 40 rounds, (b) of the morphed training set. The negative object class has been morphed using $\lambda = 0.3$. [43]

have a margin smaller than 0.2 and 0.56, respectively. In comparison, for the original set the smallest 15 % have a margin below 0.32 and the margins of the smallest 30 % do not exceed 0.62.

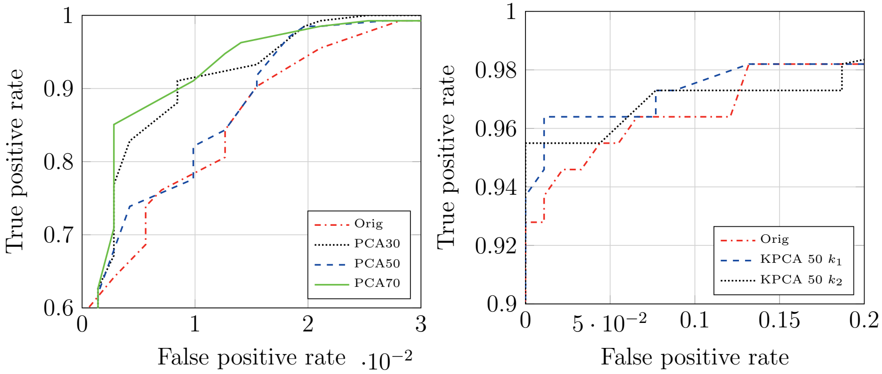
After 40 training rounds the boosted classifier for the morphed training set has caught up in the lower region of the margin distribution. The minimum margin amounts roughly to 0.26 in both cases and the progress of the cumulative distributions is similar showing only a slightly steeper slope for the morphed training set.

As discussed in Section 3.2.3, the margin distribution in training has been found to be an indicator for the quality of a classifier in terms of its test error. Hence, the result of the PCA-enhanced training to achieve a similar margin distribution starting from an adverse one can be interpreted as a higher training success. So the AdaBoost classifiers learned from PCA-enhanced training data can be expected to achieve superior performance in the test phase.

Detection Performance

In the following, the experimental results for the test set are presented. Figure 5.5a shows the ROC curves of multiple classifiers varying the decision threshold for detecting faces as described in Section 3.4. The red curve represents the classifier based on the original data set, whereas the other curves show the performance of the classifiers using PCA-enhanced images for training. Overall, the curves show that the classifiers which have been learned from an augmented training set are more selective in detecting faces so that good TPRs are achieved while maintaining a lower FPR.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a)

(b)

Figure 5.5: (a) ROC curve for the face data set using different thresholds of boosting with the original data (red) and using PCA-enhanced non-faces with different λ-values (0.3, 0.5, and 0.7). The PCA-enhanced data reveals a much more selective performance. (b) ROC curve for the cell data set.

5.2.2 Experiments on Cell Data Set

For the cell data set, a KPCA-method [134] is used for augmenting the training data in order to demonstrate that the method on PCA-enhancement of training data is not restricted to a specific method for subspace learning. The KPCA utilizes a selectable kernel function as described in Section 3.3.2. For the following experiments, two standard kernel functions are selected:

$$k_1(\mathbf{x}_i,\mathbf{x}_j) = -\exp((\mathbf{x}_i - \mathbf{x}_j)^2/(\sigma)^2) \tag{5.4}$$

with $\sigma = 1$ and

$$k_2(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^2 \tag{5.5}$$

The KPCA-enhanced training data leads for both kernels to an increased performance of the detection rate which is shown in the ROC curves in Figure 5.5b. E.g. for a TPR of 96.3 %, the KPCA-enhanced training data with k_1 yields a classifier that produces a FPR of 1 %, whereas the original data produces a FPR of 6 %. The KPCA-enhanced training data with k_2 yields similar performance in producing a TPR of 95.5 % with no FPs.

5.3 Discussion

This chapter introduces an approach to enhance (sparse) training data for boosting-based object detectors to achieve a higher detection performance. Using PCA, the

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

negative training examples are shifted in PCA-space near to the positive training class. The trained classifier achieves a lower classification error being more selective in detection. Experiments on face detection and microscopic cell images showed that the method decreases the FPR of the boosted classifier. The variable strength of the transformation allows for a trade-off between TPR and FPR. But in all experiments, the proposed method managed to significantly lower the amount of FPs without reducing the TPR.

The PCA is a linear method that, having this restriction, might not be able to efficiently structure some data sets of high complexity. The kernel functions of the KPCA incorporate non-linearity into the method and hence a KPCA with an appropriate kernel function might be more effective for such data sets.

PCA methods are computational expensive if big data sets should be analyzed. For image data, the dimensionality of the data points is commonly very high since each image is represented by a high-dimensional vector. In case of only few data points, the trick to reduce the dimension, reported in Section 3.3.2, can be applied in order to reduce the computational efforts for PCA. For data sets consisting of many high-dimensional data points, an approach can be to first cluster the data set (see Section 3.3.1) and then to perform the PCA only on the set of cluster centroids.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter

6

Fractal Integral Paths

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

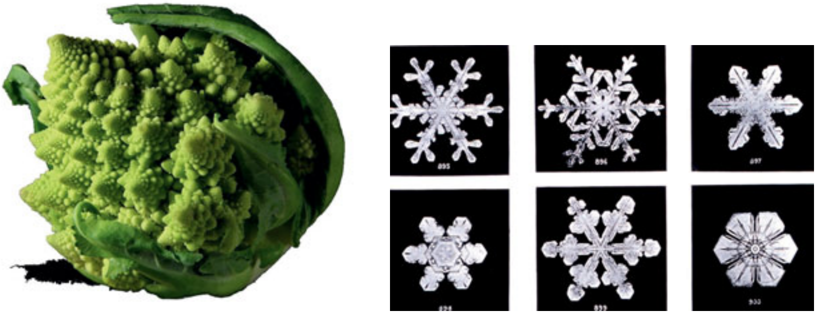

(a) (b)

Figure 6.1: Fractal structures in: (a) Romanesco broccoli. This picture is retouched for presentation in this work. The original version is part of Wikimedia Commons and has been released into public domain by its author, Jon Sullivan. (b) Snowflakes. Image is part of Wikimedia Commons and in the public domain due to its age.

In this chapter and the following Chapter 7, new feature types are developed for machine-learned object detection. A widely used feature type for object detection is the Haar-like feature described in Section 3.1.1. But this approach leads to the utilization of simple rectangle-shaped structures which are only partial suitable for curved-shaped structures. Additionally, these rectangular features are often designed for small detector basis windows and training sets that mostly contain low-resolution object details. So that features having rough shapes are sufficient to describe its gross characteristics.

In contrast, the new feature type of this chapter is designed to represent fine object details that have a wider variety of shapes. It is inspired by fractal structures that are present in natural objects e.g. in the romanesco broccoli or snowflakes shown in Figures 6.1a and 6.1b.

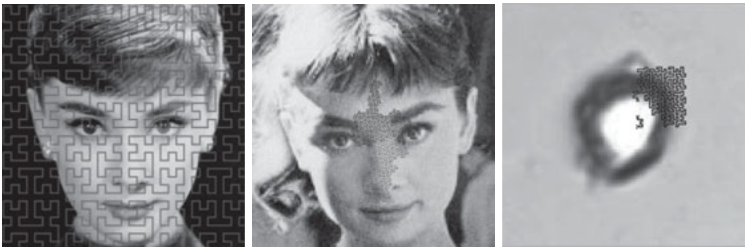
Hence, a new class of fractal features denoted by *Fractal Integral Paths* is proposed that bases on space-filling curves, a special type of fractals also known as Peano curves. This chapter has been previously published as a conference article [46].

Prior Work

Research is done in various parts of object detection frameworks to improve its performance. Zhang et al. [156] present a well structured survey of advances in face detection. They categorize developments into variations of the learning algorithm as discussed in Section 3.2.4 and advances in the extraction of features (see Section 3.1). In the advances of feature extraction, many different types of features based on histograms [149], binary patterns [158], or edges [126] have been developed.

But a notable field of research is also the improvement of the Haar-like features,

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b) (c)

Figure 6.2: (a) Illustration of the Peano-Hilbert curve traversing a face image [141]. (b) Example of a selected fractal feature in training. Image taken from [141]. (c) Microscopic cell with selected fractal feature [46].

e.g. the extension by rotated Haar-like features [96] introduced in Section 3.1.1. Pham et al. developed polygonal Haar-like features [118] to increase the variety of shapes that a feature can represent. In this way, many improved Haar-like features rely on combinations of multiple rectangular structures in order to maintain the fast feature computation using integral images.

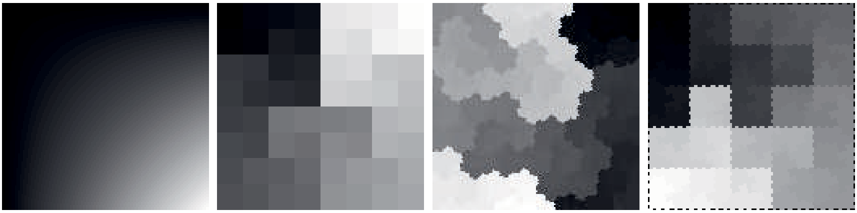
In contrast, the proposed method replaces the rectangular structures of conventional Haar-like features by a new class of fractal features that are able to adopt to curved-shaped structures. But conforming with Haar-like features, the fractal features as well utilize an intermediate image representation to allow for an efficient computation.

Contribution

This chapter proposes and evaluates a new integral image representation based on space-filling curves to explore fine non-rectangular structures. Three types of fractal features are introduced based on the Peano-Hilbert-, Gosper-, or E-Curve. Preceding the feature extraction, integral images traversing along these fractal curves are calculated. In that way, only two memory references are required to represent complex fractal structures by computing the sum of pixel intensities covered by that structure. Similarly to Haar-like features, the difference between pixel intensities in two image regions builds a feature. Utilizing three points on the fractal curve, a feature can exploit two adjacent fractal structures. The end point of the first path segment is here as well the starting point of the second path segment. Four-point features represent non-cohesive image regions defining separated fractal path segments.

The novel fractal feature class is evaluated on the well-established MIT+CMU frontal face dataset introduced in Section 2.2 and compared to standard Haar-like features. In addition to face detection, the fractal features are applied to microscopic cell data, see Section 2.2 and Figures 2.8 and 6.2c. In Figure 6.2a is exemplary shown

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) Conventional (b) Hilbert (c) Gosper (d) E-Curve

Figure 6.3: Normalized integral images traversing different paths on a homogeneous image [46].

how the Peano-Hilbert fractal curve traverses an image plane. Figure 6.2b presents a feature formed by the Gosper curve and selected by AdaBoost for face detection.

To summarize, the contributions in this chapter are:

- A novel fractal feature class is developed.
- Three-point and four-point features allow for a richer representation.
- The new method is evaluated in the field of face and microscopic cell detection.

6.1 Boosted Fractal Integral Paths

Boosted fractal integral paths are based on the Viola and Jones object detection framework introduced in Section 3.2.2. The novel class of fractal features is described in detail in this section.

6.1.1 Fractals

Following [9], a fractal is "a rough or fragmented geometric shape that can be split into parts, each of which is (at least approximately) a reduced-size copy of the whole". Such a property is also called self-similarity so that a pattern observed in one scale can often be found on other scales. There exist many examples in nature which demonstrate the beauty and importance, but also frequent appearance of fractals, e.g. in crystals, in snow flakes, or plants such as the romanesco broccoli (see Figures 6.1a and 6.1b). Hence, it is assumed that fractals can provide a good description for structures in all natural images including the test sets of faces and microscopic cells employed in this work.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

6.1.2 Fractal Features

So the motivation is to take advantage of that common appearance of fractal structures by the feature set provided to the learning algorithm. Hence, the proposed class of features utilizes a special type of fractal curves to compute fractal integral images along these curves. Conventional integral images are constructed as described in Section 3.1.1 by summing up pixel intensities from the upper left to the lower right corner of an image.

Similarly, the fractal integral images integrate the pixel intensities of a 2D image plane along the fractal curve. Figure 6.3 displays the conventional integral image used for Haar-like features and the fractal integral images following three different fractal curves. These integral images have been normalized and computed on a homogeneous image. Precalculated integral images provide the benefit that only few references into the integral image are required to compute the sum of pixel intensities in a region of the original image. In that way, pixel sums of arbitrary rectangular regions can be calculated by accessing four points in the conventional integral image (see Section 3.1.1). In contrast, differences between two pixels in the fractal integral image represent the sum of pixels covered by diverse fractal structures having a huge variability of shapes including self-similar structures. According to Haar-like features, a fractal feature is as well calculated as the difference of the sums of pixel intensities in two image regions. These regions are defined by sampling numerous positions on the fractal integral path and build the feature set for the AdaBoost machine learning algorithm.

6.1.3 Fractal Properties

To appropriately construct the integral image, it is desirable that the followed fractal path traverses every pixel in the image exactly once. This property is given by space-filling curves also referred to as Peano curves. One member of this type of fractal curves is the Peano-Hilbert curve. Like other space-filling curves, it has the property of creating a 1D representation of a 2D image while preserving its proximity relationship better than a raster scan. Thus, the Peano scan is examined for texture analysis and image compression due to its improved autocorrelation [6, 34, 83, 119, 114]. The integral image shown in Figure 6.3b illustrates the proximity of the Peano-Hilbert curve as each quadrant is completely traversed before the next quadrant is entered.

Several space-filling curves are known but not all of them are suited for fractal integral paths. Fractals that base on tree structures as H tree fractals cannot be used. The difference of two pixels of an integral image computed on this tree is not as required in any case the sum of pixels of the original image along the fractal path that connects these two points. The Z-order curve has applications similar to the Peano scan, but the Peano-Hilbert curve is preferred in this work due to its better preservation of proximity. The E-Curve is favored over the Moore curve

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

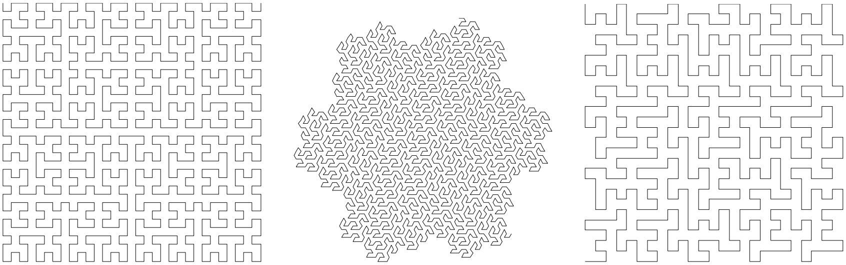

(a) Hilbert curve (b) Gosper curve (c) E-Curve

Figure 6.4: Fractal curves used to traverse image plane and to define fractal structures exploited in feature computation [46].

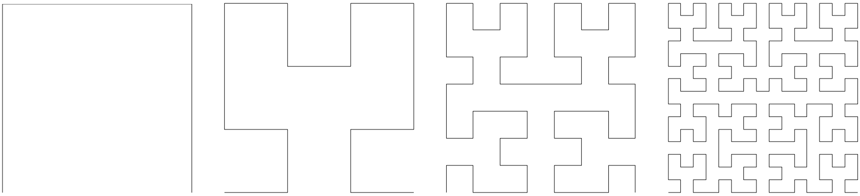

(a) 1. iteration (b) 2. iteration (c) 3. iteration (d) 4. iteration

Figure 6.5: First four iterations of the Peano-Hilbert curve. [46]

because its shape has compared to the Moore curve a stronger difference to the Peano-Hilbert curve. Space-filling curves that are closed curves like the Sierpinski curve are not suited as well. They could be split, defining a start and an end point, but the property of the conventional integral image that the start and end point are at different sides of the image plane would be lost. The Gosper curve is chosen as a third fractal for the proposed method.

Figure 6.4 illustrates the selected fractal curves. The Gosper curve is not a space-filling curve in terms of the definition given above as it traverses approximately every sixth pixel of the 2D image a second or third time. But due to the locality of the curve, this yields only some slightly enlarged features in which the corresponding pixels are weighted two or three times. As the feature selection process is performed by the AdaBoost algorithm with respect to the minimization of the classification error (see Sections 3.2.1 and 3.2.2) the disparity of those fractal features is acceptable. The Gosper curve is selected because of its different shape containing angles that are multiples of 60 degrees. This leads also to an non-square outer boundary. Hence, an inner part of the curve is clipped as the fractal path has discontinuities at its boundary. Similar to the disparities in features, this discontinuities are also tolerable due to the feature selection process.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

6.1.4 Construction of Fractals

Fractal curves can be build using a Lindenmayer System (L-System). The biologist Aristid Lindenmayer defined in 1968 a mathematical model to simulate the growth of multi-cellular organisms [97]. He developed a system of string replacement rules which are applied in parallel to recursively create an output string. The L-System grammar is defined by a tuple $\mathcal{G} = (\mathcal{V},\omega,\mathcal{R})$, where

- $\mathcal{V}$ is the alphabet of the system,
- ω is a string of symbols from $\mathcal{V}$ and defines the initial state,
- $\mathcal{R}$ defines a set of production rules. Each rule consists of the predecessor, a string of symbols from $\mathcal{V}$, and the successor, the string of symbols from $\mathcal{V}$ the predecessor is replaced by.

In contrast to a formal grammar the production rules of a L-System are in parallel applied in each iteration of the system. The alphabet $\mathcal{V}$ consists of constant symbols that are not substituted by the production rules and variables that are replaced and thus can be found on the left hand side of the rules. In this application of constructing a small set of space-filling curves, context-free L-Systems are used in which the production rules only refer to an individual symbol and do not take its neighboring symbols into account. Interpreted as turtle graphics, the output of a L-System can be used to construct fractals. In turtle graphics, a so-called turtle bot draws by executing a queue of simple instructions, like draw line, turn left, and turn right. The length of one line segment and the angle to turn is often given as global parameter that can be a function of the iteration depth. Thus, each constant in the alphabet of the L-System represents a command for the turtle bot.

The Peano-Hilbert curve for example can be described by the following L-System:

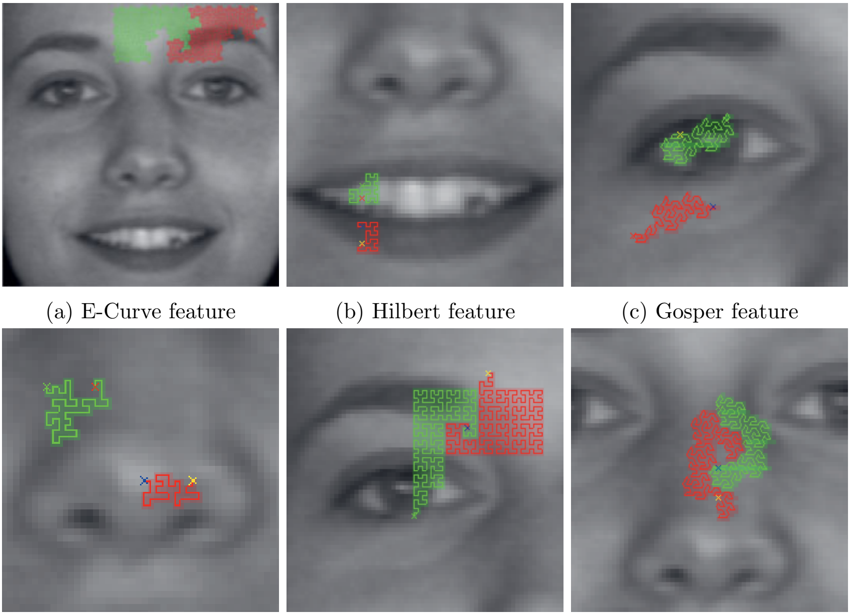
- $\mathcal{V} = \{X,Y,+,-\}$,
- $\omega = X$,
- $\mathcal{R} : \begin{cases} X \rightarrow +YF - XFX - FY+ \\ Y \rightarrow -XF + YFY + FX- \end{cases}$,

where F instructs the turtle bot to draw a line of length 1, $+$ to rotate anticlockwise by 90° and $-$ to rotate clockwise by 90°. X and Y are variables and thus do not represent commands to the bot. Figure 6.5 shows the first four iterations of the Peano-Hilbert curve starting at an initial angle of 0°. Similarly, the Gosper curve and the E-Curve can be constructed by slightly more complex L-Systems that are specified in Appendix A. The first iterations of the Gosper curve and E-Curve are, respectively, presented in Figure A.1 and A.2

6.1.5 Feature Types

In order to describe diverse structures, two different types of features are implemented for each fractal, three- and four-point features. Their specific name refers

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) E-Curve feature (b) Hilbert feature (c) Gosper feature

(d) E-Curve feature (e) Hilbert feature (f) Gosper feature

Figure 6.6: Examples of fractal features found in training [46]. Images from [22].

to the property that the calculation of these fractal features requires only three and four memory references, respectively. Three-point features represent two adjacent integral path segments and thus give preferences to cohesive regions. Similarly, the Haar-like features presented in Section 3.1.1 only describe connected areas but require in the case of a two-rectangle feature six memory references. Additionally, four-point features are defined that represent separated regions that better conform to diverse structures.

6.2 Experimental Results

The proposed fractal features are evaluated for face detection and the detection of microscopic cells.

6.2.1 Face Detection

The face detectors are learned from 1022 gray level images showing 340 individuals taken from the BioID face database [22], the MUCT face database [110], and the

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

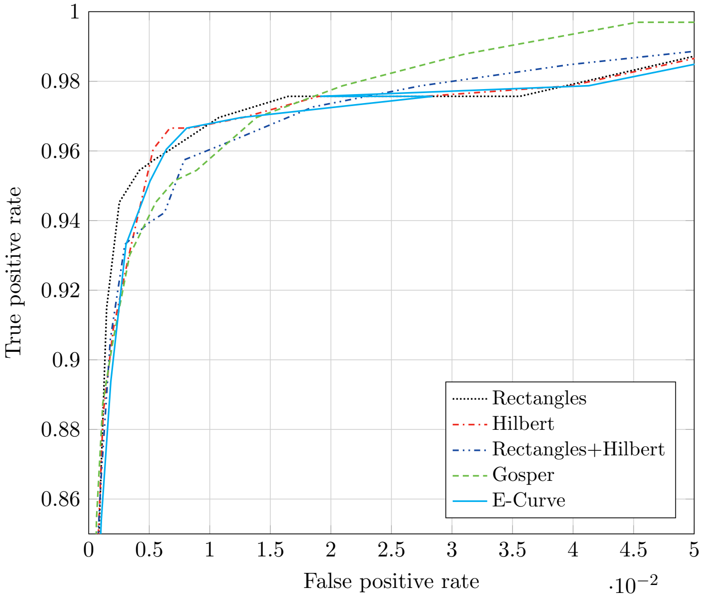


Figure 6.7: ROC curves showing the detection performance on the MIT+CMU frontal face dataset. Classifiers consisting of fractal features are compared to conventional Haar-like features (Rectangles) and a mixed classifier of Peano-Hilbert fractals and Haar-like features.

AT&T database of faces that are introduced in Section 2.2.

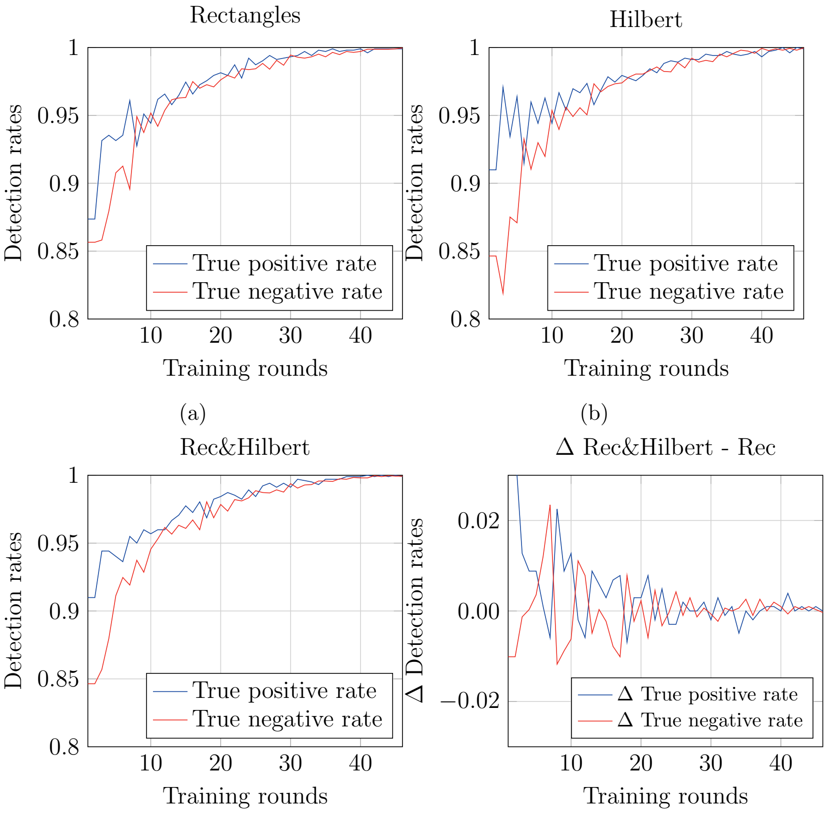
Example images taken from these data sets are presented in Figures 2.2, 2.3, and 2.4. In an automatic process, the faces are localized with respect to the given eye positions. The extracted images are then aligned and zoomed to a common scale resulting in a final patch resolution of 128 × 128 pixels.
The three types of fractal features and, for comparison, Haar-like features have been trained for 46 rounds. Figure 6.6 presents some fractal features selected in the training process. These boosted classifiers are evaluated on the MIT+CMU frontal face dataset A and C [141].

Figure 6.7 presents the detection performances in ROC curves. The fractal features and Haar-like features (Rectangles) show different characteristics. On the one hand the Haar-like features demonstrate better results in the high precision range and are at some point outperformed by the Hilbert fractals. On the other hand the fractal features achieve higher TPRs.

In a second experiment, the characteristics of rectangle and fractal features are combined. Hence, the Hilbert fractals are selected, as they achieved the best results

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

on the microscopic cell test set (see Table 6.1), and incorporated into a combined rectangle-fractal framework. The training success of the combined framework is compared in Figure 6.8 to the corresponding homogeneous frameworks. The combined framework shows less fluctuations in the detection rates during training, indicating that the different characteristics of fractal and rectangle features stabilize the combined training. Figure 6.8d demonstrates despite some fluctuations the overall improvement of the combined classifier compared to a pure Haar-like classifier.



(a) (b)

(c) (d)

Figure 6.8: (a)-(c) Detection rates showing training success in face detection vs. amount of training rounds. (d) Difference in detection rates of Rectangles+Hilbert and Rectangles only vs. amount of rounds, presenting the benefit of mixed training.

Additionally, experiments are conducted on degraded and modified versions of the training data to give an analysis of the strengths and weaknesses of the different

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

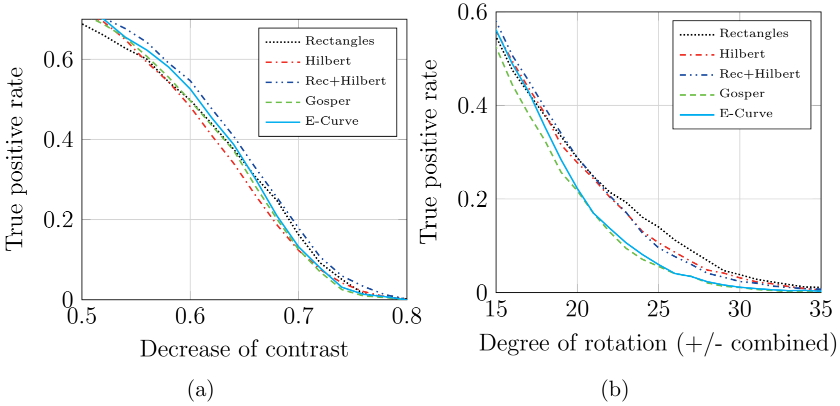


Figure 6.9: (a) Detection rate on training face images with lowered contrast. (b) Detection rate on rotated training face images. For visual clarity a middle section of the x-axis, showing most differences, is presented in both figures.

feature classes. Figure 6.9a illustrates the detection rates achieved on the positive training set when its contrast is degraded. Controlled by a parameter between 0 and 1 the appearance of each training image is transformed, respectively, between the original image having the expected contrast and a homogeneous mean image with minimal contrast. In Figure 6.9b the influence of rotation is shown. The training images are therefor rotated up to 45° prior to the detection process. It can be observed that especially in case of low contrast the application of fractal features can improve the detection performance. In case of strong rotations the rectangle features perform better. These observations indicate that fractal features fit closer to curve-shaped object structures. The closer fitting can result in an improved robustness to contrast changes but can also lead to a higher sensitivity to rotations.

The overall performance of the tested detection framework is not as high compared to very sophisticated face detectors as the focus of the research is on the impact of features and the comparison of the new feature class to conventional Haar-like features. Hence, a basic, non-cascaded boosting framework is intentionally selected and additional pre- or post-processing steps are relinquished like e.g. canny pruning in OpenCV [24] to increase the detector's performance. Another reason for the performance gap is the discrepancy in the properties of the training set and the MIT+CMU data set. The new feature class is designed and trained to adapt to various fine structures present in higher image resolutions that are common these days. In contrast, the well-known MIT+CMU test set contains several low-resolution images which do not provide fine details like the training set.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

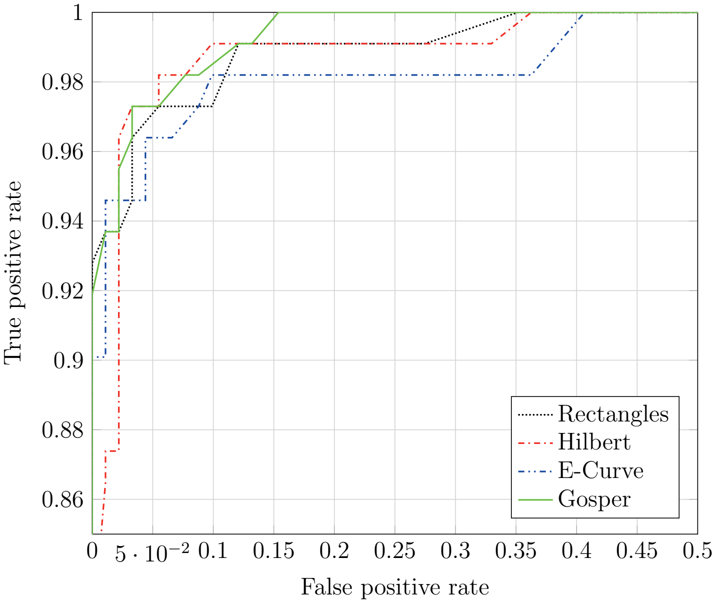


Figure 6.10: ROC curve showing the detection performance on the microscopic cell data set.

6.2.2 Microscopic Cell Detection

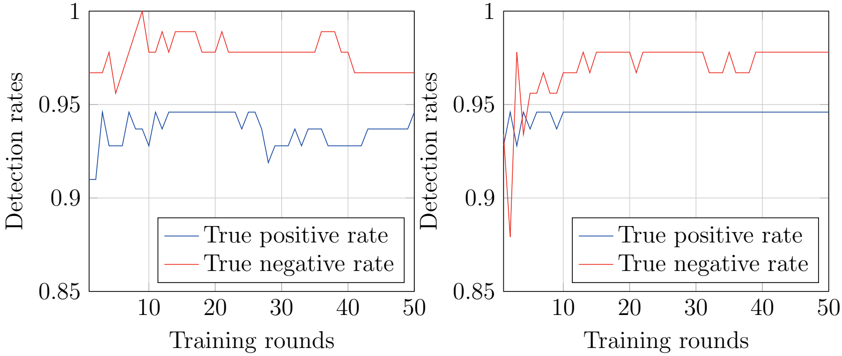
Additional experiments are conducted using microscopic cell images, see Figure 2.8. This data set is divided into a training and validation set using a 67/33 ratio following the suggestion of Crowther and Cox [33] for small data sets.

Classifiers are learned for Haar-like, Hilbert-, Gosper-, and E-Curve features. Table 6.1 presents the results of classifiers trained in 50 rounds. The results illustrate that the fractal features and above all the Hilbert curve increases the detection performance specified by the F-measure ($\mathbf{F_1}$) defined in Section 3.4.

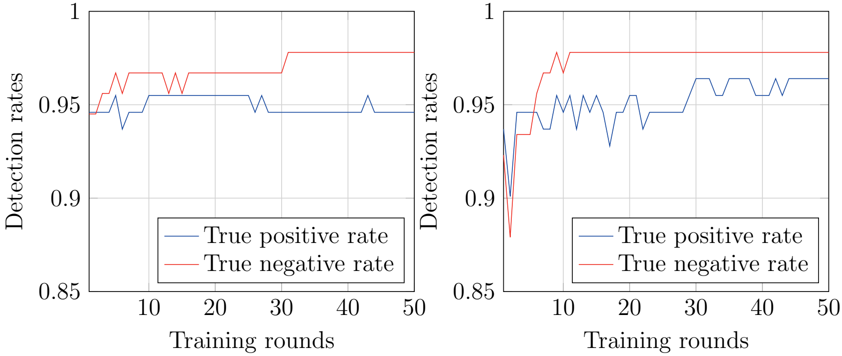
Table 6.1: Results of microscopic cell data detection. Hilbert curve achieves the best result.

Application	Feature class	$\mathbf{F_1}$ score
Microscopic cell data	Rectangles	95.89 %
Microscopic cell data	Hilbert curve	97.27 %
Microscopic cell data	Gosper curve	96.33 %
Microscopic cell data	E-Curve	96.33 %

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) Rectangles (b) E-Curve



(c) Gosper (d) Hilbert

Figure 6.11: Detection rates showing training success in microscopic cell detection vs. amount of training rounds.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

The ROC curves in Figure 6.10 illustrate that the fractal features, except for the E-Curve, slightly outperform the Haar-like features. The Gosper curve clearly reaches first a TPR of 100 %. The dependency of the detection rate from the number of training rounds is illustrated in Figure 6.11. In earlier rounds of the training, the detection rates of the fractals (see Figures 6.11b, 6.11c, 6.11d) are more stable having less fluctuation. Figure 6.11a shows that the rectangles reach with less classifiers 100 % True Negative Rate (TNR) but the TPR decreases to 93.69 %. But in the following rounds, the detection rates of the rectangle features highly fluctuate.

6.2.3 Training and Computing Time

Due to precalculated fractal paths, the training time between conventional and fractal features does not differ. Furthermore, the classical and fractal integral image computation is an initial process at the beginning of the algorithm. But there are some differences in the validation process using a sliding window. For the classical Haar-like features, it is sufficient to compute the integral image once for each scale and cut out sub-windows at arbitrary positions. In contrast using fractal features, the fractal integral image needs to be computed for every sub-window. Despite precalculated fractal paths the validation time, applying a sliding window, is slower. But this disadvantage could be overcome for example by programmable hardware like Field-Programmable Gate Arrays (FPGAs).

6.3 Discussion

This chapter introduces a new type of features for object detection which describe fractal structures that enable to better adapt to curved-shaped objects. Experiments in the domains of face detection and the detection of cells during cryo-conservation showed the improved detection performance of the fractal feature class.

Indeed, the usefulness of the fractal integral paths highly depend on the object classes to be detected. E.g. artificial objects, such as cars or manufactured parts might be better detected with rectangular features. But especially for high-resolution images of natural object classes, the fractal curves lead to a noticeable improvement with only minor algorithmic modifications.

By doing this, the proposed method can be easily incorporated in several boosting frameworks using integral image representations.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 7

Multi-Feature Mining for Detector Learning

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 7 deals with feature mining methods to generate feature sets for object detection that are customized to the object class in an automatic process that identifies characteristic structures. The motivation is to provide features to a machine learning algorithm which are more suitable for distinguishing the object. An additional advantage of such customized feature sets is their reduced size because sets of higher order features can swiftly reach sizes that cannot be processed during classifier learning in reasonable time if any possible feature manifestation is considered. This consequently leads to constraints reducing the variability of a feature type.

In case of Haar-like features (see Section 3.1.1), only a small set of feature templates that represent coherent rectangular regions is commonly defined and then scaled and translated to generate the feature set. Without that restriction to variations of feature templates the number of possible pairs of rectangular regions in a training image would be excessive. But such a feature set build from templates still contains a very large number of irrelevant and redundant features. It completely neglects the available domain knowledge given by characteristic structures of the positive training set. Feature mining addresses the task of extracting this knowledge in constructing a customized feature set.

This chapter is partly based on a published conference article [44].

Contributions

Section 7.1 presents a feature mining method to build a set of generalized Haar-like features based on information given by the positive training set. By exploiting domain knowledge, the portion of redundant and irrelevant information in the created feature set is significantly smaller compared to a conventional overcomplete set of Haar-like as described in Section 3.1.1. In return, this allows to loosen constraints on the variability of the Haar-like features. The size of the derived new feature set is drastically decreased but it is shown to be more discriminative, allowing a higher training success when learning an object detector using AdaBoost.

As a generalized type of Haar-like features, computation can be further on performed efficiently using integral images, essentially calculating intensity differences between local areas. Hence, this feature type utilizes coarse object characteristics represented by image areas but ignores fine object structures and shapes that seem to be especially vital in the domain of pedestrian detection since intensity differences between areas suffer e.g. from the variability induced by different clothing.

In order to observe as many different object characteristics as possible it is conducive to employ sets of multiple feature types that are complementary in their analyzed object structures. But this approach consequently holds the drawback of increasing the computational costs of classifier learning. Feature mining weakens this drawback by constructing manageable and efficient feature sets and thus promotes to enrich the overall pool of features by a complementary feature set exploiting fine structures. Therefore, a new type of customized features is introduced in Section

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

7.2 that bases on HOG descriptors (see Section 3.1.2) and is extracted as well from the positive object class.

Several experiments demonstrate a strong benefit in the number of required features and detection performance if two complementary feature types are enabled to strengthen each other. For this, the machine learning algorithm is allowed in a mixed training to freely select and combine features of different types in contrast to a separate training where a set consisting of only one feature type is provided. Figure 7.1 presents the first learned stage in each case for mixed pedestrian, face, and car detectors.

In summary, the contributions of this chapter are as follows:

- Exploiting domain knowledge given by the object class to construct customized feature sets in boosting-based object detection.
- Introduction of two complementary feature sets utilizing coarse object characteristics and fine object structures.
- The approach leads to a much smaller but more distinguishing mixed feature set.

Prior Work

Dollár et al. [40] propose several strategies to explore a very large feature space and identify meaningful features that are used to model the mined feature space incorporating a metric to measure the feature quality. A method for scene text recognition has been recently developed by Lee et al. [86] that extracts a feature space more directly from the image data of the object domain. In order to build a more efficient feature space from HOG features that commonly utilize the complete image plane by a grid of overlapping regions, Lee et al. develop a mid-level feature pooling algorithm that automatically learns a discriminative feature space from a set of randomly generated image sub-regions.

Instead of randomly sampling image regions, Section 7.1 proposes to extract domain knowledge by image segmentation in order to construct a pool of generalized Haar-like features.

Many methods e.g. [41, 89, 78] demonstrate that it is beneficial to utilize not only one but multiple feature types that represent different object characteristics. Benenson et al. present in a recent survey [21] a good overview on the mix of feature types in various pedestrian detection methods. One of the most applied feature types are HOG features and adapted variants [160, 76, 85, 86].

In the following sections, the two complementary feature types are described and the feature mining methods to create pools customized to the object class are illustrated.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

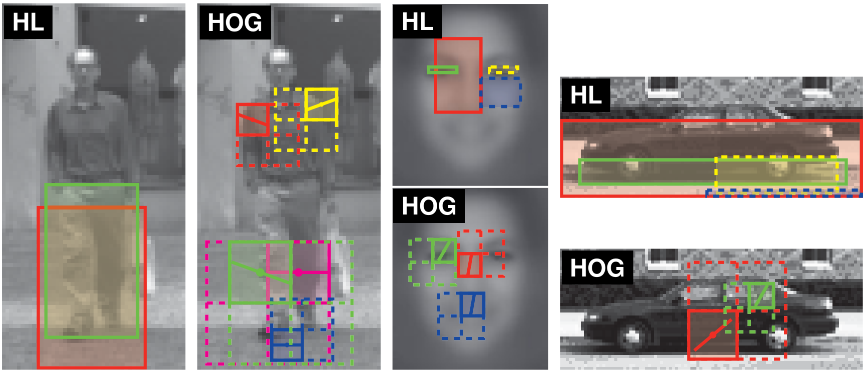


Figure 7.1: Learned features in the first stage of mixed pedestrian, face, and car detectors. The utilized feature types are generalized Haar-like (HL) features and HOG based features exploiting gradient information. Both feature types are proposed in this chapter along with the feature mining methods to generate feature pools customized to the object class. The two left-most images present the learned features of the first cascade stage of the mixed pedestrian detector. It consists of one generalized Haar-like feature and five HOG based features. The features are shown in separate images for more visual clarity. The remaining images present in the same way the first learned stage of the mixed frontal face and lateral car detector.

7.1 2Rec Features

The feature mining approach to create the first feature type aims on identifying rectangular image regions that are distinctive for the object class. The response computed by the feature is essentially the difference of the mean pixel intensities of such two regions so that the feature is called 2Rec feature. More specifically, the detection window is in addition variance normalized to achieve partial invariance to changing lighting conditions as proposed by Viola and Jones [146]. In contrast to conventional Haar-like features these rectangular regions are not required to be coherent but can be spatially separated. 2Rec features are therefore generalized Haar-like features that allow for a higher variability of represented shapes. The acquisition of domain knowledge in the feature mining process heavily restrains the amount of irrelevant and redundant data in the feature pool. By this, a feature pool is created that has a smaller size compared to conventional Haar-like features but is more distinguishing [44].

The mean image of the positive training set is analyzed for frontal face and lateral car detection in [44] to acquire domain knowledge for the pool of 2Rec features. Multiple watershed segmentations of the mean image are created using different parameterizations to generate a combined set richer in segments of varying size and position. These segmented areas are subsequently approximated by rectangu-

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

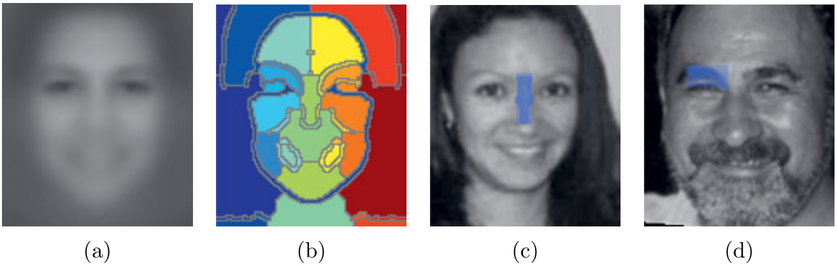


Figure 7.2: Example images for face detection illustrating the construction of rectangular regions for 2Rec features. (a) Mean image of the positive face training set. (b) Exemplary watershed segmentation of the mean face. For the creation of the complete 2Rec feature pool, multiple watershed segmentations are applied varying the number of segments. (c) and (d) Examples of rectangular regions approximated to segments taken from the watershed segmentation. Images from [1]

lar regions to preserve the fast and efficient computation using the integral image representation. In order to create rectangles that approach the watershed segments in terms of covered region, the rectangles' center positions and dimensions are computed, respectively, as the mean and twice the standard deviations of the coordinates of all pixels belonging to a segment.

Figure 7.2 presents examples of watershed segments and fitted rectangular regions for the face data set.

The complete pool of 2Rec features is then assembled from all 2-combinations of the approximated rectangles.

In the domain of pedestrian detection, the inner class variability is very high due to different body poses and viewing angles. For this reason, the mean image of the positive training set is not an appropriate source for characteristic structures of pedestrians since their variability cannot be represented. Furthermore, the distinctive regions in pedestrian images like limb segments are comparatively small. Superpixel methods, as described in Section 3.3.1, that aim on local, structure preserving over-segmentations are thus more appropriate for pedestrian images. The superpixel segmentation of Schick et al. [132] additionally regulates the compactness of the superpixels and allows for a better separation of individual body parts. Hence, a different method is proposed to create the pool of 2Rec features customized for the pedestrian data set. Instead of segmenting the mean image, distinctive regions are identified in 20 images randomly selected from the positive training set. The superpixel algorithm is then parameterized to create 18 superpixels in each image in order to derive a reasonable set of characteristic regions. Due to the compactness property of the superpixel algorithm of Schick et al. [132], the rectangular regions approximated in the same way as for the watershed segmentation cover only small center regions of the superpixels. Therefore, an additional larger rectangle is fitted

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

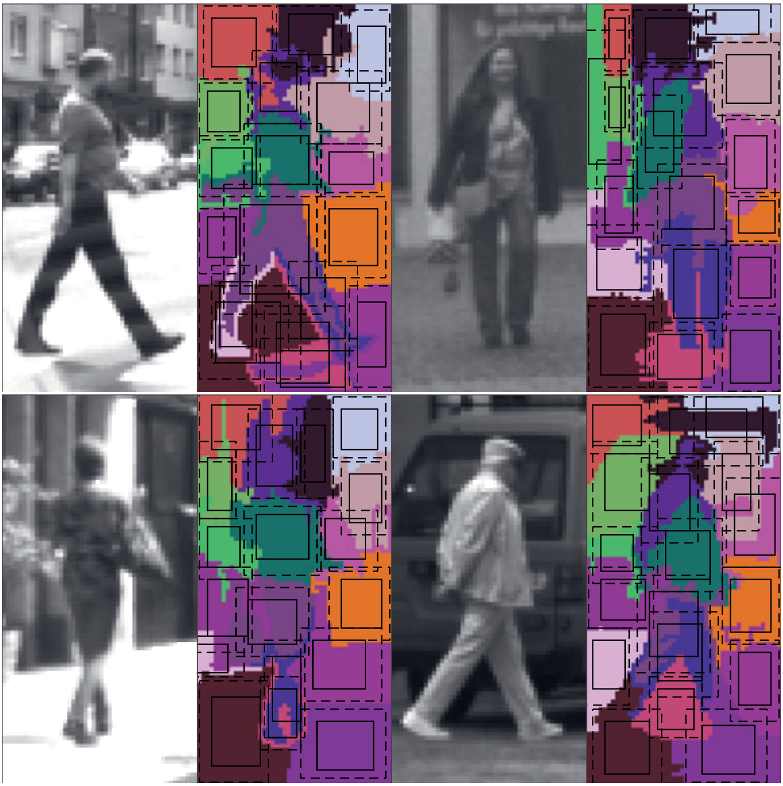


Figure 7.3: Superpixel label masks and approximated rectangular regions. Two rectangles of different size are fitted to each superpixel printed as solid and dashed lines. Pedestrian images from [47].

to each superpixel. The dimensions of the larger rectangle are set to be three times the standard deviation of the x-coordinates and y-coordinates of all pixels belonging to the superpixel. Figure 7.3 shows some examples of pedestrian images, the created superpixel label masks, and the approximated rectangular regions. The smaller fitted rectangles are printed as solid and the larger rectangles as dashed lines.

7.2 Keypoint HOG Features

The second feature type is constructed to supplement 2Rec features that represent coarse structures of the object class. Finer object details should be utilized such that a derivation based approach is selected. The proposed Keypoint HOG (KPHOG)

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

feature and the feature mining approach to generate it are heavily inspired by HOG descriptors [35] and SIFT features [99, 100]. But it is also considered that the developed feature has to be efficient with computational costs similar to Haar-like and 2Rec features.

Instead of utilizing a dense overlapping grid of HOG descriptors as proposed by Dalal and Triggs [35], a feature mining method is applied to locate and evaluate single HOG descriptors at distinctive image positions. This significantly decreases the number of computed HOG descriptors. Moreover in contrast to a grid structure, HOG descriptors can be evaluated as well on different scales without raising the computational costs too much.

The feature mining approach to create Keypoint HOG (KPHOG) features aims on identifying keypoints that are distinctive for the object class similar to the localization of SIFT descriptors. But with respect to the application of KPHOG features in a sliding window based object detector, a search for scale-space extrema as in SIFT is not necessary. The sliding window of a detector is applied on different image scales such that all learned features are scaled as well and the associated keypoints have not to be invariant to multiple scales.

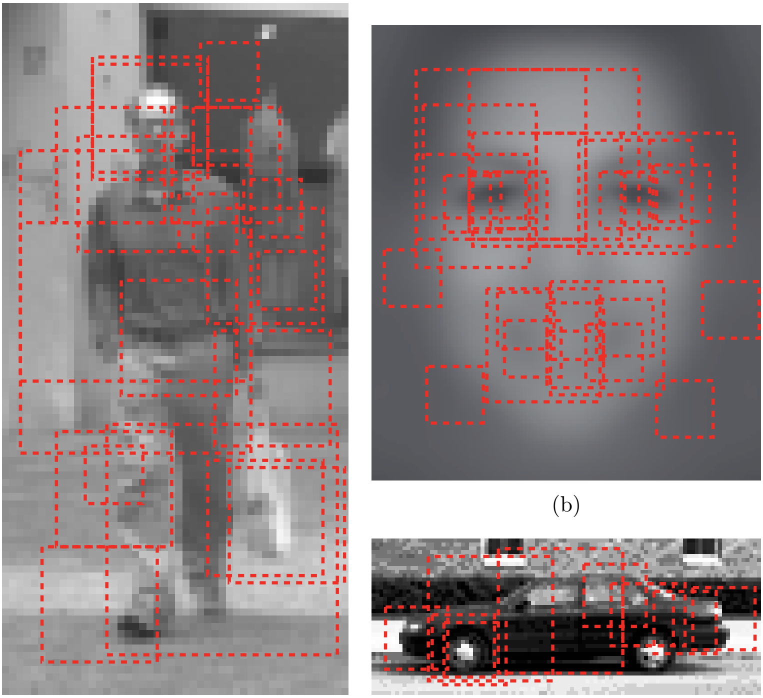
In the proposed method, keypoints that are distinctive for the object class are localized on multiple scales by a Harris corner detector [68] that is applied to a Gaussian image pyramid. A collection of 25 images is randomly selected from the positive training set and a Gaussian image pyramid is created for each image by reducing it as long as a basis HOG region of 16×16 pixels fits into the down-scaled pyramid image. The pyramid images of every image in the collection is then searched for 20 corner points using the Harris corner detector. All corner points belonging to images of the same pyramid level are clustered by the k-means algorithm [98] to gain 50 clusters for each pyramid scale. The cluster centers are taken as potential keypoints and are stored together with the down-scaling factor of their associated pyramid level and the number of elements in the cluster. This list of keypoints on multiple scales is then sorted by the cluster sizes and the keypoints belonging to the largest clusters are taken as center positions of the KPHOG features.

The number of added keypoints has been set as a parameter to 25 for frontal faces and lateral cars and to 50 for the pedestrian data set. In addition, the above procedure to extract keypoints from random subsets of the positive training data is repeated once for cars and four times for pedestrians to further enrich the feature set.

A KPHOG feature is specified on a quadratic image region around a keypoint such that its size is given by the dimension of the basis HOG region multiplied by the down-scaling factor of the associated pyramid level. Figure 7.4 presents KPHOG regions on multiple scales identified for pedestrian, face, and car data sets.

In order to create a very efficient feature, several simplifications have been incorporated compared to other HOG based features. Thus, the KPHOG feature represents a single bin in a histogram of oriented gradients over the quadratic regions around

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



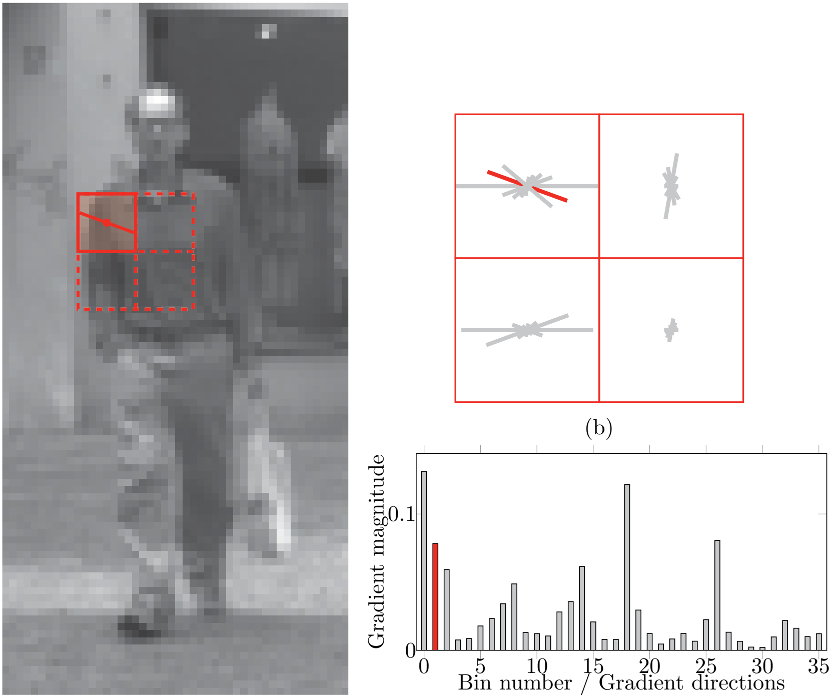
(b)

(a)

(c)

Figure 7.4: Multi-scale KPHOG regions identified in the feature mining procedure. (a) Regions extracted from the pedestrian data set. The complete set of regions utilized in the detector learning is considerably larger such that only every 25th keypoint region is visualized. (b) The complete set of keypoint regions used in face detector training. (c) Every 5th keypoint region extracted from the UIUC car data set.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a)

(b)

(c)

Figure 7.5: KPHOG feature learned in the first stage of the mixed pedestrian detector shown on an example image. (a) The keypoint location identified in the feature mining procedure and its surrounding quadratic region is plotted as dashed red lines. The learned KPHOG feature represents the depicted gradient orientation bin in the upper left quadrant. (b) Oriented gradients computed on the keypoint region in the example image. The bin selected by the machine learning algorithm is marked red. (c) Presentation as concatenated histogram of all quadrants.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

the keypoints. In contrast to a SIFT descriptor, the orientation assignment and transformation relative to that orientation is omitted. The computed gradient histogram is inspired by the HOG descriptor of Dalal and Triggs such that the region around the keypoints is divided into 2×2 cells. A separate histogram having 9 bins is calculated for each area covered by a cell. These bins represent unsigned gradient orientations evenly spaced over 0° to 180°. The magnitudes of gradients orientated in the corresponding directions are added up in the bins. Using the integral image representation, these histograms over rectangular regions can be efficiently computed using integral histograms [89, 120]. The construction of 9 integral images for the values of the histogram bins enables the efficient calculation of a single bin for any rectangular image region by only four memory accesses. The histogram of all four cells in a keypoint region are concatenated to a histogram of 36 bins. Dalal and Triggs suggest to normalize the HOG descriptor over such a block of cells. A L1 normalization is used in the proposed KPHOG feature that can be as well efficiently computed on an integral image. Therefore, a tenth integral image is created adding up all histogram bins such that the KPHOG feature can be computed by only eight image access operations.

The complete pool of KPHOG features consists of all histogram bins on all keypoint regions identified in the feature mining procedure and the feature response is the normalized value of such a single histogram bin.

Figure 7.5 illustrates a KPHOG feature learned in the first stage of the mixed pedestrian detector.

7.3 Experimental Results

This section evaluates the performance of each feature type and the impact of a mixed application of both feature types. Object detectors are learned in the domains of frontal face, lateral car, and pedestrian detection.

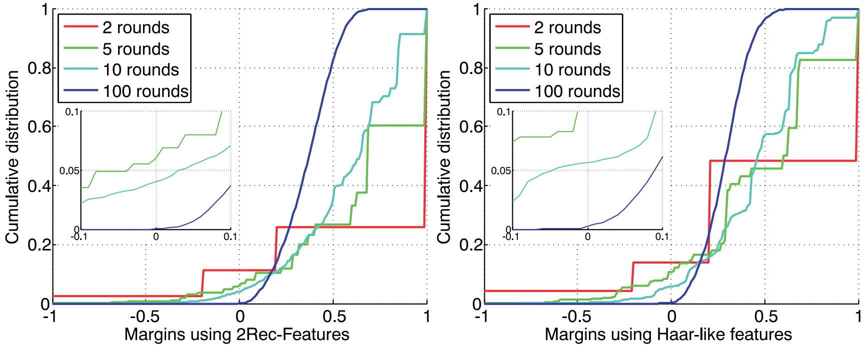
7.3.1 Face Detection

The face detectors are learned on the MPLab GENKI Database, GENKI-4K Subset [1] from the Machine Perception Laboratory in California. Figure 2.5 presents some example images of the data set.

Training Comparison of 2Rec and Haar-like Features

This section analyzes the training processes when learning from a customized set of 2Rec features, obtained from the feature mining method proposed in Section 7.1, in comparison to conventional Haar-like features.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b)

Figure 7.6: Cumulative margin distributions with detail view of face training set after 2, 5, 10, and 100 rounds in case of (a): 2Rec features, (b): Haar-like features.

Training success In the first experiment, statistics of the training process are collected to analyze the training success of classifiers employing 2Rec features compared to conventional Haar-like features. Single-stage classifiers are trained to measure the progress of the margin distribution (see Section 3.2.3) on the training set.

Figure 7.6a and 7.6b show the cumulative margin distribution of a 2Rec classifier and a Viola and Jones classifier after different rounds in training. In case of the 2Rec classifier the best feature is selected from a pool of only 44732 features while the Haar-like feature pool supplies $4.61 \cdot 10^6$ features. Nevertheless, the training algorithm is able to select more discriminative features out of the smaller pool of 2Rec features. This can in particular be observed on the red margin curves in Figure 7.6. These curves present the margin distribution after two rounds in training using solely 2Rec features or Haar-like features. The values of the cumulative distributions at a margin of -1 shows that the amount of training examples that have been misclassified by both learned classifiers derived after two training rounds is significantly higher in case of Haar-like features. Similarly, both learned Haar-like classifiers classify approximately 52 % of the training examples correctly, whereas both 2Rec classifiers decide correctly for roughly 74 % of the training set.

In other words, the more shallow progress for low margins of the 2Rec feature curves compared to the Haar-like feature curves after equal training rounds demonstrates that the 2Rec classifiers are not only classifying a bigger part of the training set correctly but also that these decisions are more clearly. This advantage persists during training. After 100 training rounds the combined 2Rec classifier shows at the zero margin position, in contrast to the Haar-like classifier, no misclassified training examples anymore.

Hence, the margin analysis of the training process demonstrates that in comparison to conventional Haar-like features the feature mining approach proposed in

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Section 7.1 creates a set of clearly more distinguishing features leading to a superior boosted classifier. Furthermore, the customized feature set is considerably smaller and thus allows for a much faster training.

Training time The benefit of the proposed method in processing time in the feature selection of the training phase is obvious. Referring to the previously reported sizes of the feature pools, the Haar-like feature space is 103-times bigger than the 2Rec feature space. The feature selection process scales nearly optimal using feature pools of different sizes. Due to the constant computation overhead, a slight reduction in efficiency per feature of 3.84 % is measured on a single multi-core workstation when processing the smaller pool of 2Rec features. This yields a total training speed-up of 99.04-times using 2Rec features. The sizes of the training sets and the measured speed-up are listed in Table 7.1.

Table 7.1: Properties of the evaluated sets of conventional Haar-like features and 2Rec features. The sizes of the feature sets and the measured training speed-up are presented.

Feature set	Size of set	Size ratio	Training speed-up
Haar-like features	4.610000	103	1
2Rec features	44732	1	99.04

Detectors Learned from Customized Features

For the following experiments, the object detectors have been learned by the distributed machine learning framework introduced in Chapter 4. Despite the massively parallelized computation in the developed framework, the training of capable objector detectors is still very time-consuming as large quantities of training data have to be processed. Since the customized 2Rec features demonstrated superior training success along with a significant speed-up in training (see Table 7.1), the conventional Haar-like features have been omitted in favor of 2Rec features for all detectors presented in the remainder of this chapter.

Detection performance For the face detectors learned on customized features by the distributed framework, the feature mining methods described in Section 7.1 and 7.2 extract from the positive set two feature pools consisting of 27495 approximated rectangle features (2Rec) and 936 KPHOG features, respectively. The initial negative training set contains 2916 non-face images and an additional set of 19774 scene images showing no faces is provided for sampling new negative training images in the bootstrap phase. Three cascaded face detectors are learned employing solely 2Rec or KPHOG features as well as a mixed detector utilizing both feature classes. Each stage of the cascaded detector is trained until it achieves a target true positive

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 7.2: Training statistic for face detection. The number of features are presented that are required to achieve the target rates of each cascade stage.

Cascade stage	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	mixed cont.
No. of 2Rec	5	16	19	31	47	69	85	123	145	186	248	353	411	538	621	852	942	11 (0 %)
No. of KPHOG	8	8	10	13	15	17	21	26	32	37	50	62	78	100	145	179	≥4106	10 (80 %)
No. of Mixed	5	7	8	10	12	15	16	19	24	27	32	41	46	56	59	72	83	-
Percentage of 2Rec	40 %	29 %	13 %	20 %	17 %	7 %	19 %	16 %	17 %	26 %	16 %	24 %	22 %	29 %	34 %	31 %	36 %	-

rate of 99.5 % while maintaining a true negative rate of at least 60 %. All correctly classified negative training examples are removed from the training set after each completed stage. The negative training set is then supplemented with 10000 new images that have been found as false positives by applying the detector of the so far completed cascade stages to the bootstrap set. In that way, the negative set assigned to the machine learning algorithm consists in later stages of increasingly harder training examples.

Table 7.2 presents the number of required features in the cascade stages of each face detector. For the mixed detector, the percentages of selected 2Rec features are listed as well. Figure 7.7 shows the cumulated sums of required features after each stage. The KPHOG features demonstrate to require clearly less features in each stage compared to 2Rec features to fulfill the target rates. In total 801 KPHOG features are selected in 16 stages compared to 3749 2Rec features. But the machine learning algorithm is not able to achieve the target rates in the 17th stage solely using KPHOG features. The learning of the KPHOG based detector has been aborted in the 4106th training round of the 17th stage while the 2Rec based detector completes this stage employing 942 features. Please note, that the negative training sets after the first stage are not identical for different detectors due to the application of the in each case currently learned detector during bootstrapping. An interesting additional experiment is to continue the training of both detectors that contain only a single feature class with the mixed feature set after the 16th cascade stage. The continued mixed training clearly demonstrates how well 2Rec features, exploiting coarse object characteristics, and KPHOG features, utilizing fine object structures, supplement each other. The 17th stage that the 2Rec detector completes using 942 features can be fulfilled by only eleven KPHOG features. Similarly the non-fulfilled 17th stage of the KPHOG detector can be completed in only ten mixed training rounds selecting eight 2Rec and two KPHOG features. Consequently, an entirely mixed training massively reduces the number of required features to in total 449 after 16 stages.

First, the FDDB face detection benchmark [74] is evaluated. Table 7.3 compares, relative to the 2Rec detector, the reduction factor of the required features in the

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

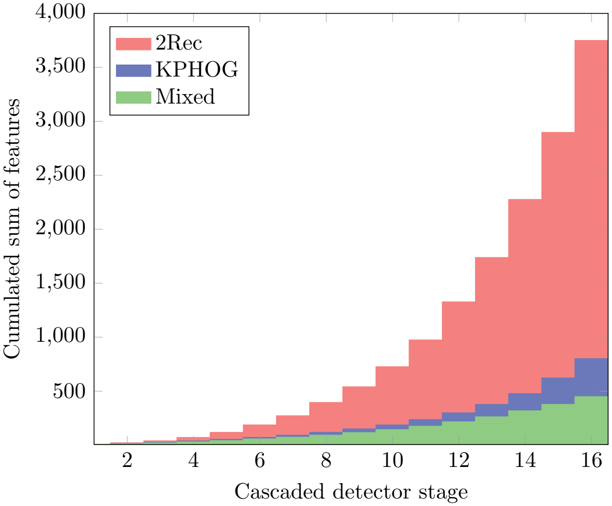


Figure 7.7: Required features in the face detectors. Cumulated sum of features contained in face detectors after each detector cascade stage. The KPHOG based detector requires considerably less features compared to the 2Rec based detector. But a mixed detector further on significantly reduces the number of features.

learned detector cascades with the resulting speed-up on the FDDB test set when computing the detector cascades from the integral images. The measured speed-ups are significantly below the reduction factors and indicate that the 2Rec detector is nevertheless very efficiently rejecting candidate sub-windows in early cascade stages. Figure 7.8 presents some example images of detected faces.

Figure 7.9 presents a performance comparison of the differently learned cascaded detectors on the FDDB test set. The result of the OpenCV Viola&Jones detector is additionally given as a baseline and the performance of a more extensively trained mixed detector is shown. The performance curves of the three different 16-stage detectors demonstrate that the KPHOG features clearly outperform the 2Rec features

Table 7.3: Improvement of learned detectors relative to the 2Rec detector for face detection. The reduction factor of the number of features required in the detector cascade are presented and the measured speed-ups in computing the detector cascade from the integral images are given.

Improvement relative to 2Rec detector	2Rec	KPHOG	Mixed
Reduction factor of required features	1	4.68	8.35
Speed-up of cascade computation	1	1.97	2.55

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

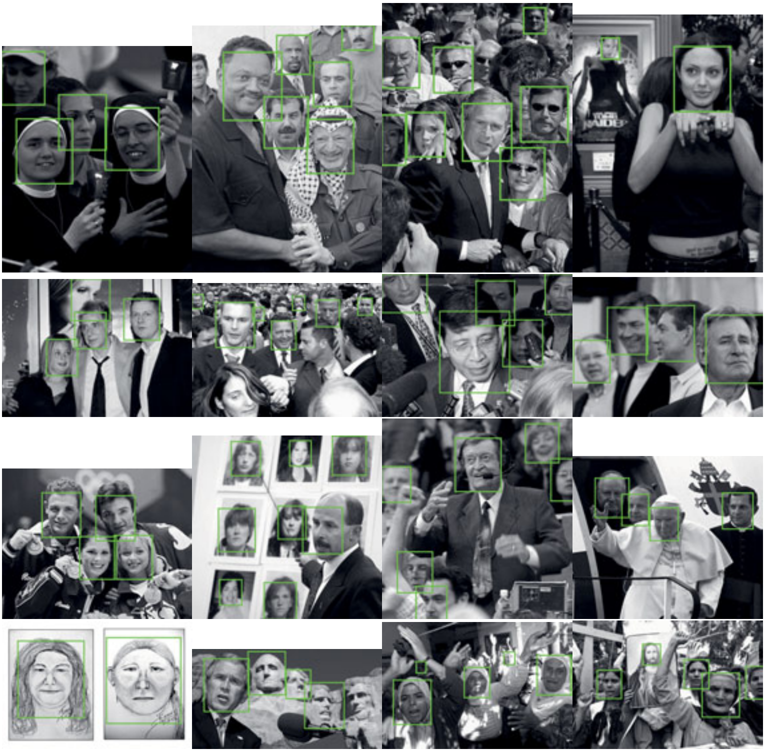


Figure 7.8: Example images of detected faces in the FDDB data set [74]. It contains many crowded scenes that are challenging because of very small or partly occluded faces as well as faces in multiple poses.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

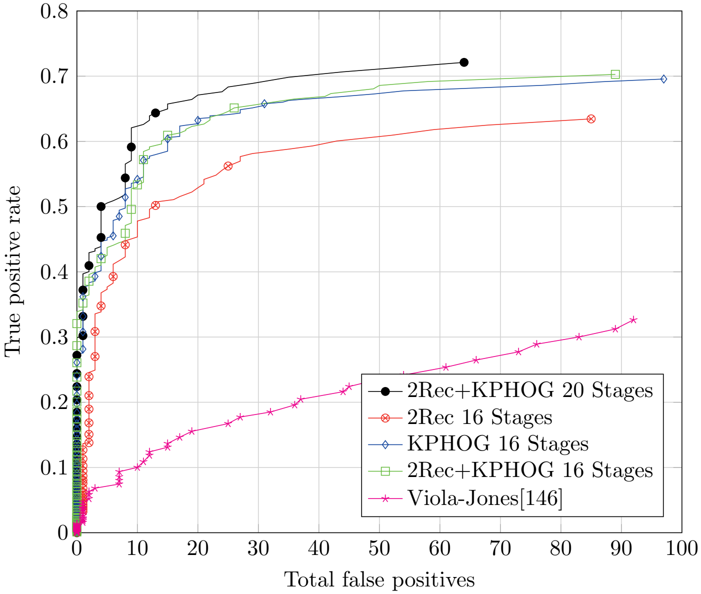


Figure 7.9: Performance curves showing the detector quality on the FDDB data set. A comparison of the three differently learned cascaded detectors is given. The performance of the OpenCV Viola and Jones detector is shown as a baseline and the results of a more extensively trained mixed detector is presented as well. The detector solely containing KPHOG features demonstrates on this data set better results than the 2Rec based detector. But the mixed detector presents comparable performance utilizing significantly less features.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

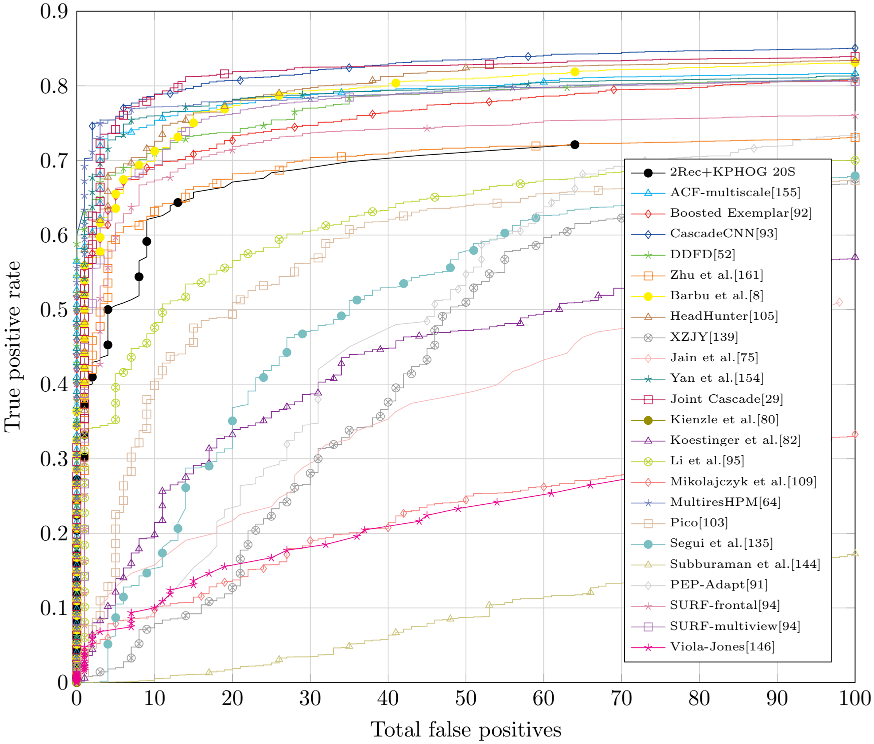


Figure 7.10: Performance curves showing the detection quality of the mixed detector in comparison to state-of-the-art methods on the FDDB data set. Legend titles are in accord with the performance curves presented on the FDDB web page. The rank of the mixed detector is below the top tier of face detection methods but still remarkable for an efficient, light-weight detector that is learned on frontal faces and only requires gray-scale images. The competing methods are in part considerably more complex, e.g applying CNNs or incorporating multiple views, 3D models, face alignment, or occlusion handling.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

in this benchmark. The mixed detector shows a detection quality that is similar to the KPHOG features. But as previously described, the mixed detector can achieve this performance by utilizing significantly less features.

A comparison to state-of-the-art methods is given in Figure 7.10. The more extensively trained mixed detector presents a performance below the top tier of face detection methods. But its performance is still remarkable as the detector is very efficient and light-weight. The mixed detector is learned on frontal faces and requires only gray-scale images. Many competing methods are in contrast considerably more complex, e.g applying CNNs or incorporating multiple views, three dimensional (3D) models, face alignment, or occlusion handling.

The second face detection benchmark is the MIT+CMU frontal face dataset [141]. Some example images of detected faces are shown in Figure 7.11 and Figure 7.12 presents the results on this benchmark.

The ROC curves are generated by adjusting the classification thresholds of the cascaded detectors. The comparison of the three 16-stage detectors shows a different result. Here, the 2Rec detector outperforms the KPHOG detector. The reasons for the lower performance of the KPHOG detector are likely the contained low-resolution and blurred images that do not provide sufficient fine structures. But the mixed detector even so achieves results equal to the leading 2Rec detector by utilizing massively less features.

Hence, the evaluation of both face detection benchmarks demonstrates that the mixed training of both complementary feature types enables a more robust detector. In addition, the mixed detector is more efficient as it utilizes significantly less features.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

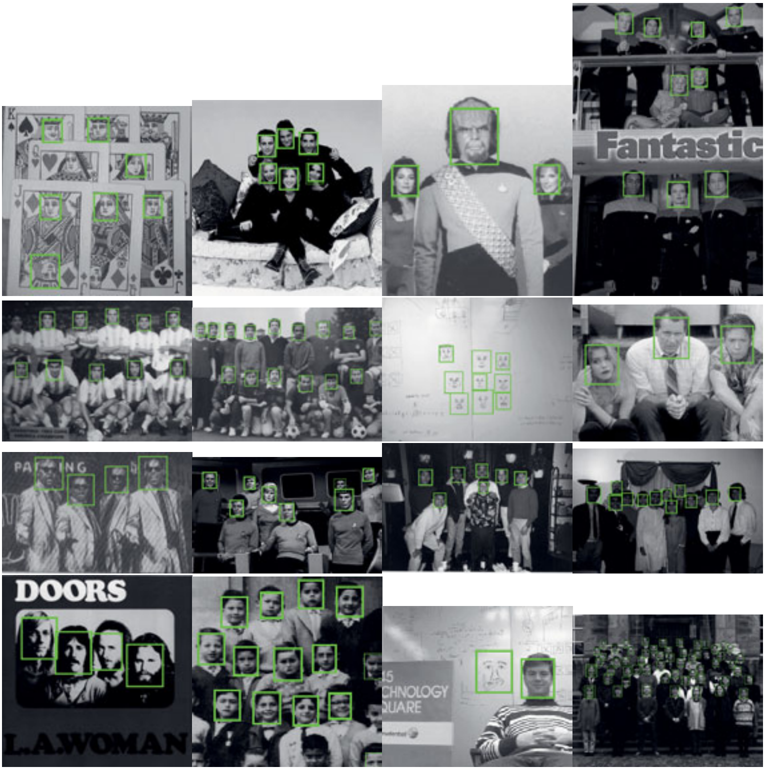


Figure 7.11: Examples images of detected faces in the MIT+CMU benchmark [141]. This data set contains several difficult test images that are noisy, blurred, or show faces illustrated by comics and line drawings.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

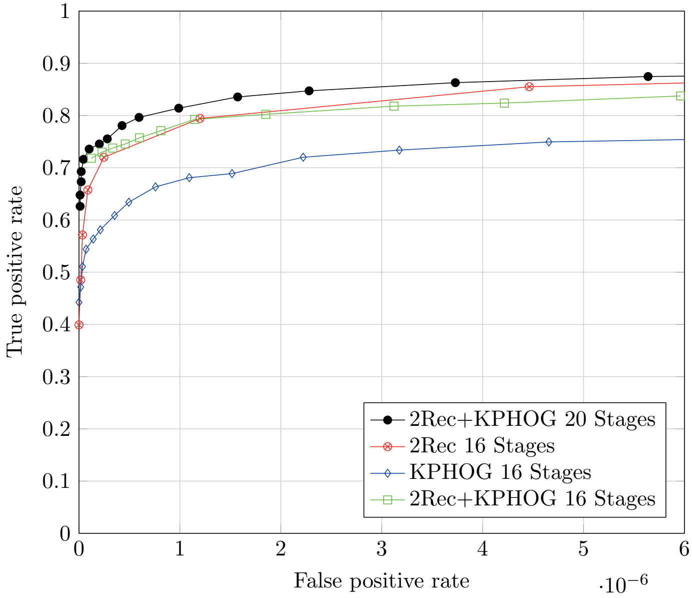


Figure 7.12: ROC curves showing the detection performance on the MIT+CMU data set. A comparison of the three differently learned cascaded detectors is given. The results of a more extensively trained mixed detector is presented as well. The detector solely containing 2Rec features demonstrates on this data set better results than the KPHOG based detector. But the mixed detector presents comparable performance utilizing significantly less features.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 7.4: Training statistic for car detection. The number of features are presented that are required to achieve the target rates of each cascade stage.

Cascade stage	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	mixed cont.
No. of 2Rec	9	5	4	8	11	19	26	31	43	50	66	83	109	139	131	170	197	-
No. of KPHOG	5	5	7	7	9	11	12	15	18	19	19	24	27	32	31	42	48	-
No. of Mixed	4	5	7	7	9	9	9	10	11	10	14	16	15	15	20	21	25	-
Percentage of 2Rec	50 %	0 %	0 %	0 %	11 %	22 %	0 %	40 %	9 %	30 %	36 %	13 %	13 %	20 %	30 %	19 %	24 %	-

7.3.2 Lateral Car Detection

Three lateral car detectors solely utilizing 2Rec or KPHOG features as well as a mixed detector are learned on the UIUC image database for car detection [3]. Figure 2.9 presents example images of the car database. Each stage of the learned cascaded detector is completed when the target true positive rate of 99.5 % and the target true negative rate of 60 % are fulfilled. Similar to the face detector training, all correctly classified negative training examples are removed from the negative training set when completing a detector stage. Afterwards, 5000 new negative training images are extracted from a set of 19774 scene images showing no cars.

The number of required features for car detectors solely employing 2Rec and KPHOG features and for a mixed car detector are listed in Table 7.4. The car database is easier to learn compared to faces such that significantly less features are required in all detectors. But again it can be observed that clearly fewer KPHOG features than 2Rec features are required. A mixed learning utilizing both features types in addition strongly reduces the overall number of features. In total after 17 stages, the 2Rec based detector requires 1101 features while the KPHOG based detector requires 331 features. A mixed training further reduces the total number of features to 207. Table 7.5 compares, relative to the 2Rec detector, the reduction factor of the required features in the learned detector cascades with the resulting speed-up on the car test set when computing the detector cascades from the integral images. The measured speed-ups for car detection are also below the reduction factors but not as clearly as for face detection. This indicates that the early cascade stages of the 2Rec detector are in comparison to face detection less efficient in rejecting candidate sub-windows. Some example images of detected cars in the test set are shown in Figure 7.13.

Figure 7.14 presents the results on the UIUC car database. The three 17-stage detectors achieve similar detection performances in this benchmark. The KPHOG based detector reaches a slightly higher recall but the mixed detector shows similar results using significantly less features. Hence, the mixed detector proves to be the most efficient detector in the car database in accordance to the other experiments.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 7.5: Improvement of learned detectors relative to the 2Rec detector for car detection. The reduction factors of the number of features required in the detector cascade are presented and the measured speed-ups in computing the detector cascade from the integral images are given.

Improvement relative to 2Rec detector	2Rec	KPHOG	Mixed
Reduction factor of required features	1	3.33	5.32
Speed-up of cascade computation	1	2.80	3.11

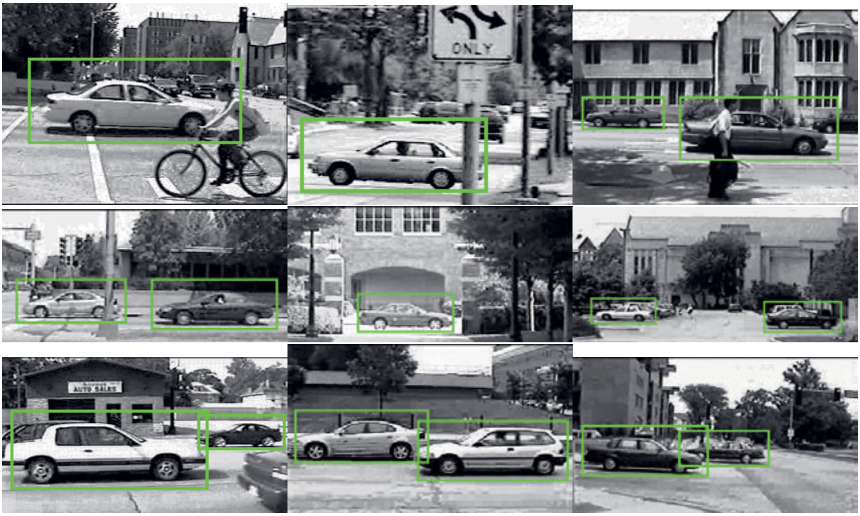


Figure 7.13: Example detections in the UIUC lateral car dataset [3].

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

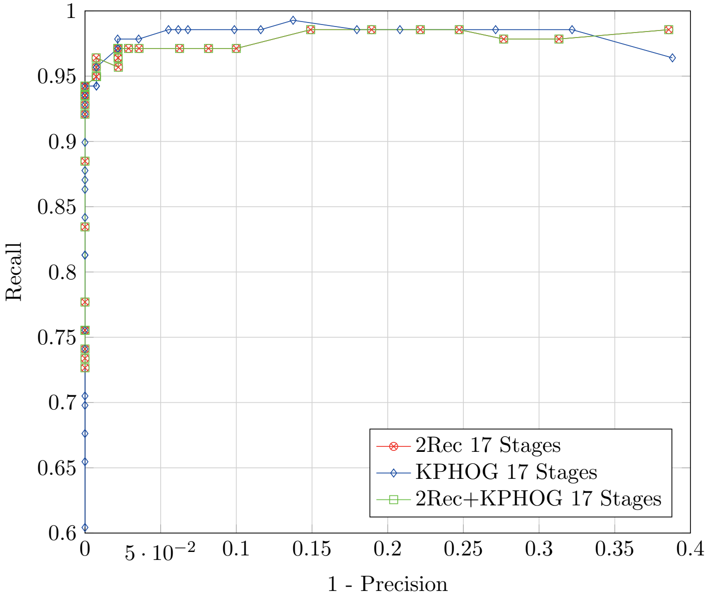


Figure 7.14: Performance curves showing the detection quality on the UIUC cars database. A comparison of the three differently learned cascaded detectors is given. All detectors demonstrate a comparable high performance on this easier benchmark. But the mixed detector utilizes significantly less features, similar to the experimental results on face and pedestrian detection.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 7.6: Training statistic for pedestrian detection. The number of features are presented that are required to achieve the target rates of each cascade stage.

Cascade stage	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	mixed cont.
No. of 2Rec	16	44	96	182	364	579	932	1030	1752	1805	1940	1885	1958	1740	1926	823	15 (0 %)
No. of KPHOG	9	17	26	39	59	93	133	199	268	367	503	687	816	1142	1260	1339	153 (75 %)
No. of Mixed	6	11	21	30	47	68	88	111	150	224	286	377	405	437	534	543	-
Percentage of 2Rec	17 %	18 %	14 %	27 %	19 %	19 %	23 %	22 %	25 %	29 %	33 %	37 %	42 %	39 %	45 %	45 %	-

7.3.3 Pedestrian Detection

Similar to the face detector training described in Section 7.3.1, pedestrian detectors are learned that (i) solely utilize approximated rectangle features (2Rec), (ii) solely utilize KPHOG features (KPHOG), and (iii) combine both feature types in a mixed detector. The Daimler mono pedestrian benchmark dataset [47] offers the training data required by the machine learning algorithm. Each cascade stage is learned to achieve a target true positive rate of 99.5 % and a target true negative rate of 60 %. The bootstrap phase works as described in Section 7.3.1 with the difference that 15000 new negative training images are added. Table 7.6 presents the number of selected features in the learning of the pedestrian detectors.

In common with the face and lateral car detectors requires the KPHOG based detector clearly less features than the 2Rec based detector. A mixed training of both feature types furthermore strongly reduces the number of required features. For 16 completed stages, the 2Rec detector requires 17072 features, the KPHOG detector 6957 features and the mixed detector 3338 features. In compliance with the experiments of Enzweiler and Gavrila [47] in their paper on the Daimler mono pedestrian benchmark, a saturation of the detector performance is observed after learning 16 cascade stages. The given bootstrap set of non-pedestrian images is most likely exhausted after that learning progress causing as well the decline of required features in the last learned cascade stage of the 2Rec based detector.

The experiment of repeating the last stage of both detectors, that contain only a single feature class, by a mixed learning demonstrates again the ability of both feature classes to complement each other. The last stage of the 2Rec detector consisting of 823 features can be completed as well by only 15 KPHOG features. Instead of applying exclusively 1339 KPHOG features, the last stage of the KPHOG detector can be learned using 153 mixed features containing 114 2Rec and 39 KPHOG features.

Table 7.7 compares, relative to the 2Rec detector, the reduction factor of the required features in the learned detector cascades with the resulting speed-up on the Daimler test set when computing the detector cascades from the integral images. The mixed detector cascade is able to perform this computation on average in less than 40 milliseconds on a standard desktop computer when using a detector setting

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Table 7.7: Improvement of learned detectors relative to the 2Rec detector for pedestrian detection. The reduction factors of the number of features required in the detector cascade are presented and the measured speed-ups in computing the detector cascade from the integral images are given.

Improvement relative to 2Rec detector	2Rec	KPHOG	Mixed
Reduction factor of required features	1	2.45	5.11
Speed-up of cascade computation	1	4.58	5.75

that applies, compared to the setting utilized in the benchmark, a slightly coarser sampling and stricter classification threshold. In contrast to face and car detection, the measured speed-ups are even higher than the reduction factors. This clearly indicates that 2Rec features on its own are not well suited for pedestrian detection. Figure 7.15 presents some example images of detected pedestrians in the Daimler test set.

Figure 7.16 presents a performance comparison of the differently learned cascaded detectors. Additionally, results of the `VJ` [146] and `HOG` [35] detectors are given as baselines and the performance of a more extensively trained mixed detector is shown. The performance curves of the three different 16-stage detectors demonstrate that the KPHOG features clearly outperform the 2Rec features in pedestrian detection. But the mixed detector shows a detection quality that is superior to the KPHOG based detector although it utilizes significantly less features.

Figure 7.17 compares the mixed detector (`2Rec+KPHOG 20S`) of 2Rec and KPHOG features to the best methods on the Daimler test set listed by the Caltech benchmark. Since the Daimler test set is gray-scale, only the subset of detection algorithms that do not rely on color information can be taken into consideration. The mixed 2Rec and KPHOG detector shows the best results together with the `MLS` [111] detector and outperforms methods using a variety of different feature types and learning algorithms. `MLS` learns mid-level features based on low-level HOG features by a boosting framework. The other methods in comparison are `LatSvm-V2` [53] applying deformable part-based models, `ConvNet` [136] based on a convolutional neural network, `RandForest` [102] that additionally exploits context information and MultiFtr+Motion [148] that incorporates optical flow information using previous frames.

Benenson et al. give in [21] a categorization of all detectors that are present in the Caltech benchmark at the time of July 2014.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

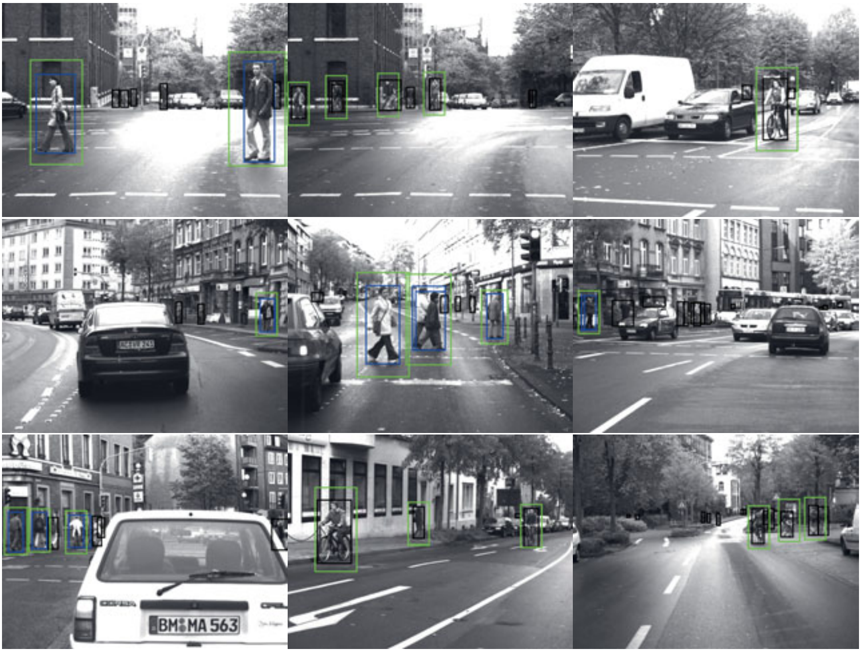


Figure 7.15: Example images of detected pedestrians in the Daimler Mono Pedestrian Detection benchmark [47]. The detections are evaluated and marked according to the criterion defined by the Daimler benchmark. Black bounding boxes visualize the ground-truth of optional pedestrians, that are to small to be a mandatory detection, or mark cyclist instead of pedestrians. Blue bounding boxes present the ground-truth of mandatory pedestrians. Detections that have been successfully matched to a ground-truth pedestrian are marked in green. Please note that the detector has been parametrized to detect only pedestrians big enough to be mandatory in order to avoid unnecessary false positive detections. For this reason, part of the smaller pedestrians might be detected as well when searched for.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

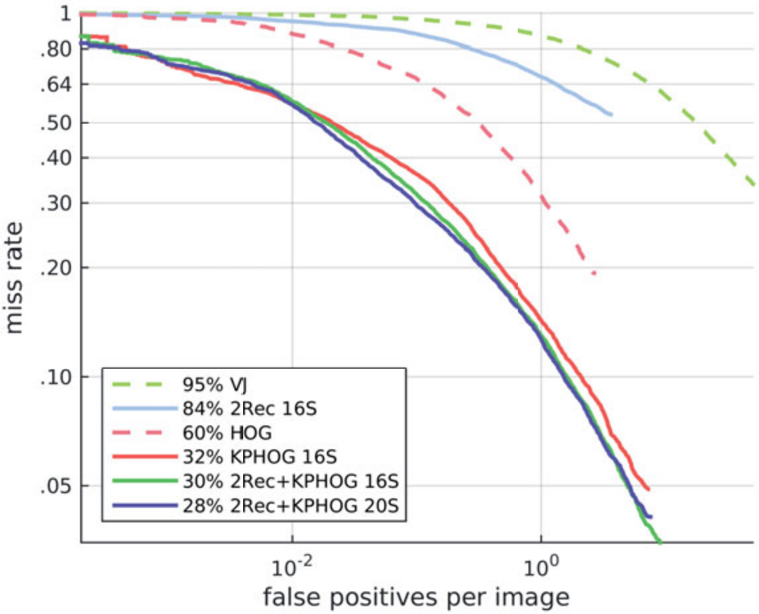


Figure 7.16: Performance curves showing the detector quality on the Daimler Mono Pedestrian data set. A comparison of the three differently learned cascaded detectors is given. The performance of the `VJ` [146] and `HOG` [35] detectors are shown as baselines and the results of a more extensively trained mixed detector (`2Rec+KPHOG 20S`) is presented as well. The detector solely containing KPHOG features demonstrates on this data set significantly better results than the 2Rec based detector. But the mixed detector presents superior performance utilizing significantly less features.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

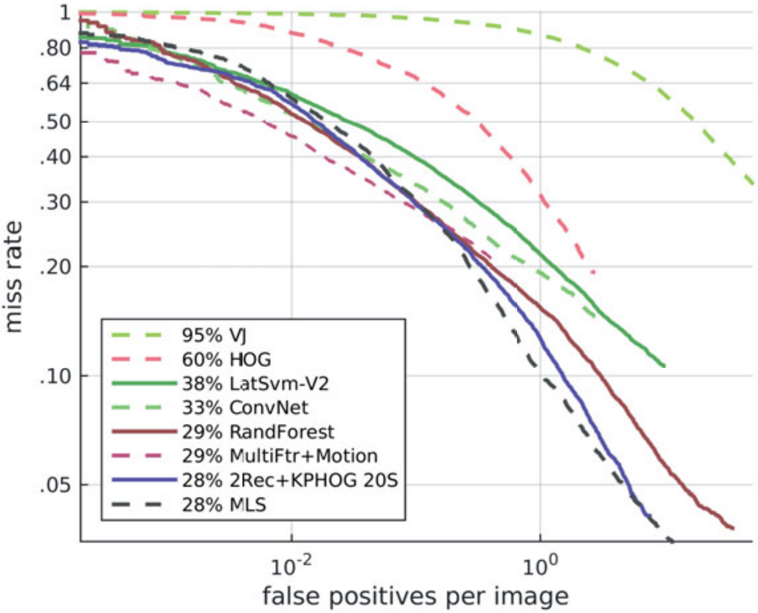


Figure 7.17: Performance curves showing the detector quality on the Daimler Mono Pedestrian data set. The mixed `2Rec+KPHOG 20S` detector shows the best results together with the `MLS` [111] detector and outperforms methods using a variety of different feature types and learning algorithms.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

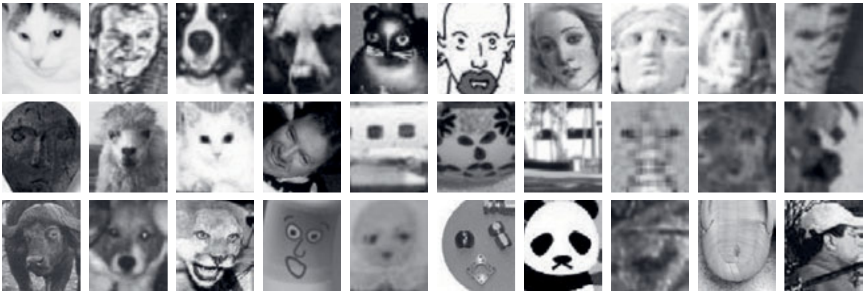


Figure 7.18: Example images taken from the negative training sets of the face detectors learned in Section 7.3.1. Top row: Images from the negative set of the mixed detector after 19 learned cascade stages. Middle row: Images of the 2Rec detector after 16 learned stages. Bottom row: Images of the KPHOG detector after 16 learned stages.

7.3.4 Insights into the Training Process

This section gives an insight into the process of learning the face, car, and pedestrian detectors. For this, example images are presented that have been extracted from the negative training sets that are utilized for learning the last stage of the respective cascaded detectors. The negative sets for later cascade stage are updated during training by the bootstrap strategy described in Section 3.2.2. As a result, they contain only training images that are classified by all preceding stages as the object class to be detected. Since the negative training set is visualized, the examples are supposed to be increasingly similar to the positive examples for later stages but should not contain any true instance of the object class. Examples of the detector training are presented for faces in Figure 7.18, for cars in Figure 7.19 and for pedestrians in Figure 7.20. The negative sets from training the face detectors and in particular the car detectors reveal that the utilized sets of bootstrap images accidentally contain positive examples. These flaws are undesired and should have a negative effect on the performance of the learned detectors.

It is important to note that the presented examples are selected with respect to human perception such that the positive classification by the preceding stages is comprehensible for some of the examples. But the negative sets contain as well image that show no similarity to the positive examples for a human observer. The reason for their occurrence is the strongly different perception provided by features. Vondrick et al. give in [147] a very nice insight into the perception of HOG features. They illustrate that in scenes that are too dark for the perception of human eyes, HOG features are still able to observe structures due to their invariance to illumination changes and their amplification of gradients. But this amplification has also the effect that in some cases very homogeneous or noisy image regions wrongly have an appearance similar to the object class in the perception of features and thus are

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

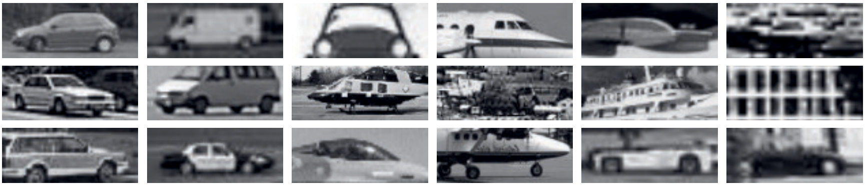


Figure 7.19: Example images taken from the negative training sets of the car detectors learned in Section 7.3.2 after 16 learned cascade stages. Top row: Images from the negative set of the mixed detector. Middle row: Images of the 2Rec detector. Bottom row: Images of the KPHOG detector.

collected by the bootstrap process in the negative training set.

Figure 7.18 presents negative examples from the mixed face detector training in the top row, examples from training the 2Rec detector in the middle row and examples from training the KPHOG detector in the bottom row. The examples demonstrate that the mixed detector fails on animal faces. Faces of human statues and a line drawing of a face are positively classified as well but it might be not clear to which class these examples should belong. The 2Rec detector shows a similar attitude but seems to be more focused on the eye and mouth region so that similar structures at the corresponding positions can deceive it. In contrast, the KPHOG detector seems to place more importance on the shape of faces.

The layouts of Figure 7.19 and 7.20 are the same as for the face detector examples. All car detectors suffer from the flaws of the bootstrap set so that true car images are very prominent in the negative set. Apart from that, the detectors fail on other vehicles like planes and helicopters that naturally have a similar aerodynamic shape. In addition, the 2Rec detector is irritated by different kinds of grid structures. The bootstrap set for the training of pedestrian detectors is flawless. The negative sets contain some examples that might be recognized as pedestrians at first glance. Other examples include structures that are similar to human legs or contain objects that roughly have the shape of a pedestrian. But a clear difference in the precedence of structures cannot be observed for the different detectors.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

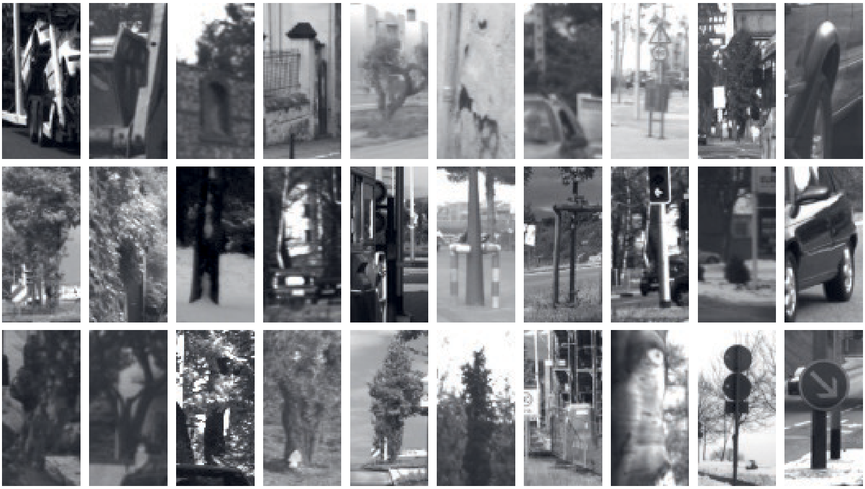


Figure 7.20: Example images taken from the negative training sets of the face detectors learned in Section 7.3.3. Top row: Images from the negative set of the mixed detector after 19 learned cascade stages. Middle row: Images of the 2Rec detector after 15 learned stages. Bottom row: Images of the KPHOG detector after 15 learned stages.

7.4 Discussion

This chapter proposes a generic approach to create feature pools customized to various object classes for learning boosting-based object detectors. The introduced feature mining strategies enable the combined utilization of two complementary feature classes that provide a high inner-class variability. But their customization to the object class renders the feature pools manageable and allows for a fast detector learning. A parallelized training in a distributed machine learning framework furthermore strongly reduces the time to learn a detector.

In-depth experiments demonstrate that a mixed learning of complementary features not only results in a more robust detector but also increases its efficiency as significantly less features are required. The learned detectors show competitive or state-of-the-art results in the domains of frontal face, lateral car, and pedestrian detection.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 8

Non-Maximum Suppression using Dempster's Theory of Evidence

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

This chapter addresses the post-processing in a object detection framework. In cascaded detectors, an image is passed through a cascade in which all stage detectors have to classify a found object positively. For this purpose, detectors commonly apply a sliding window which scans the scene image on shifted positions and varied scales. This frequently results in multiple detections of an object at slightly shifted and scaled positions. In a post-processing step, these multiple detections have to be combined to determine the final object position and scale.

Prior Work

Often only little effort is spend on detection merging and simple methods are applied. Although this subtask has a strong impact on the overall accuracy of the detection framework and the results achieved in benchmarks. E.g., Viola and Jones in [146] merge all overlapping detection windows to one detection. But this approach easily leads to worse results in case of increasing numbers of detections, in particular if detections on large scales are involved. Everingham et al. [50] thus reported in the Pascal VOC Challenge that the measured average precision steeply dropped for all participating methods when they tightened the tolerances for correct detections on the "car" class.

In this chapter, a novel method for merging multiple detections is proposed. The Dempster-Shafer Theory of Evidence (DS) is applied to combine confidence values similar to Real AdaBoost [129] and uncertainty information that is available in a cascaded detector. In this way, intra-cascade information is exploited in an improved merging of multiple detections during post-processing. Huang et al.[73] introduced a nested classifier to inherit classification confidences in detection cascades. But their approach is confined to the classification step and requires a retraining.

Contribution

The contribution of this chapter can be summarized as follows:

- First, a novel method is proposed for merging multiple detections that exploits intra-cascade confidences using Dempster's Theory of Evidence. The evidence theory allows in the process to model confidence and uncertainty information to compute the overall confidence measure for a detection. It is shown that the proposed confidence gives an appropriate measure to distinguish the reliability of detections.
- Second, this confidence measure is applied to improve the accuracy of the determined object position by refining the position and scale of merged detections. Figure 8.1 illustrates this process of Non-Maximum Suppression (NMS) that finds a more accurate object position.
- The proposed method is evaluated on public object detection benchmarks and is shown to improve the detection performance.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

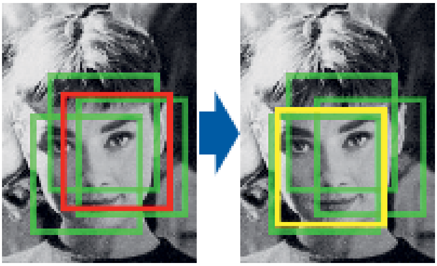


Figure 8.1: Synthetic example illustrating the position refinement in Non-Maximum Suppression (NMS). The red bounding box represents the average position of all detections while the yellow bounding box marks a more accurate position that could be obtained from assigned detection confidences. Image from [141].

- As a post-processing step, the proposed method is easily applicable in other object detection frameworks without the need of retraining the object classifiers. Hence, other object detectors could benefit from the proposed detection merging.

8.1 Merging Multiple Detections based on Dempster's Theory

In this section, the proposed strategies on merging detections are described in detail. The required methods of machine learning, object detection and evidence theory are briefly discussed in advance.

8.1.1 Cascaded Classifier

The object detection framework used in this work utilizes a cascaded classifier as introduced by Viola and Jones, see Section 3.2.2 and Figure 3.5. Each stage of this cascaded classifier consists of a strong classifier that is created using the AdaBoost machine learning algorithm. Hence in a cascade of N_S stages, N_S strong classifiers have to decide positively for a scanned sub-window $\mathbf{x}$ to be classified as an object. Any of these candidate sub-windows is then further processed in the post-processing step in which the merging of multiple detections is done.

Each strong classifier $H_s(\mathbf{x}) = \sum_{r=1}^{N_{R_s}} \alpha_{rs} h_{r_s}(\mathbf{x}), s \in 1 \dots N_S$ is composed of an ensemble of N_{R_s} weak classifiers h_{r_s} which have been selected in the training phase of the AdaBoost algorithm (see Algorithm 3 and Equation (3.17)). Each weak classifier returns 0 or 1 in case of a negative or positive classification, respectively. These ensembles decide in a weighted majority vote in which each weak classifier h_{r_s} supports its decision by an assigned weight α_{rs} that represents the classification

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

error of that weak classifier in training. Thus, the maximum positive classification of a strong classifier is given by $H_{s,\max} = \sum_{r=1}^{N_{R_s}} \alpha_{rs}$ and the decision threshold of AdaBoost is the weighted majority $\tau_s = \frac{1}{2} \sum_{r=1}^{N_{R_s}} \alpha_{rs}$.

AdaBoost's decision threshold aims at a low error rate on the training set without differentiating between positive and negative training examples. But due to the rejection opportunity of each cascade stage, a very high TPR is primarily desired. Hence according to Algorithm 3, a subsequently adjusted threshold τ_s is used to maintain a very high TPR accepting an also high FPR.

8.1.2 Dempster-Shafer Theory of Evidence

This section briefly describes Dempster's theory of evidence. It is utilized in the proposed method to model intra-cascade decision confidences and uncertainties. The Dempster-Shafer Theory of Evidence was introduced in 1968 by A. P. Dempster [38] and later in 1976 expanded by G. Shafer [137]. The Evidence theory can be interpreted as a generalization of Bayesian theory that directly allows the representation of uncertainty and inaccuracy information. The key element of the evidence theory is the definition of a mass function on a hypotheses set Ω. Let a hypotheses set be denoted by Ω and composed of n single mutually exclusive subsets Ω_i written as $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$. For each element A of the power set $\wp(\Omega)$ a mass function $m(A)$ is defined that expresses the proportion of all evidence assigned to this hypothesis. Hence, the mass function m represents a degree of confidence and is defined as $m : \wp(\Omega) \rightarrow [0,1]$. Furthermore, the following conditions have to be fulfilled by the mass function:

$$(i) \quad m(\emptyset) = 0 \quad (ii) \quad \sum_{A_n \subseteq \Omega} m(A_n) = 1\,. \tag{8.1}$$

Mass functions in evidence theory describe the totality of belief as opposed to Bayesian probability functions. This belief can be associated with single and composed sets of hypotheses allowing for a higher level of abstraction. The so-called additivity rule $p(A) + p(\overline{A}) = 1$ is in contrast to Bayesian theory not generally valid in Dempster-Shafer Theory of Evidence. This means that if $m(A) < 1$, the remaining evidence $1 - m(A)$ does not necessarily claim its negation $\overline{A}$.

Dempster's Rule of Combination

In order to combine information from different stages of the detection cascade, *Dempster's rule of combination* is applied. Dempster's rule combines two mass functions that are defined within the same frame of discernment but belong to independent bodies of evidence. Let m_1 and m_2 be two mass functions associated to such independent bodies of evidence. Then Dempster's rule defines the new body of evidence

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

by the mass function

$$m(A) = m_1(A) \otimes m_2(A) = \frac{\sum\limits_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum\limits_{B \cap C = \emptyset} m_1(B) m_2(C)}\,. \tag{8.2}$$

The denominator in Equation (8.2) works as a normalization factor that ignores the conflicting evidence. Hence, Dempster's rule of combination focuses on the measure of agreement between two bodies of evidence. Dempster's rule is associative and thus can be used to iteratively combine evidences obtained from arbitrary number of classifiers.

8.1.3 Joint Confidence based on Dempster-Shafer

In the proposed application of joining intra-cascade confidences, the frame of discernment is defined as $\Omega = \{TP{,}FP\}$ containing the set of hypotheses supporting a True Positive (TP) or a False Positive (FP) decision. The uncertainty of each cascade stage s is modeled by $m_s(\Omega)$ with respect to its size:

$$m_s(\Omega) = 1 - \frac{N_{Rs}}{\sum_{s=1}^{N_S} N_{Rs}} \tag{8.3}$$

This leads to a higher belief into stages that consist of larger number of weak classifiers.

The mass functions, expressing the proportion of evidence of a stage s, for TP or FP decisions are defined by:

$$m_s(TP) = \frac{H_s(\mathbf{x}) - \tau_s}{H_{s,\max} - \tau_s}(1 - m_s(\Omega)), \tag{8.4}$$

$$m_s(FP) = \left(1 - \frac{H_s(\mathbf{x}) - \tau_s}{H_{s,\max} - \tau_s}\right)(1 - m_s(\Omega)) \tag{8.5}$$

This results in higher stage confidence when the difference between the response of the strong classifier and the decision threshold grows. Using Dempster's rule of combination the stage confidences for a detection D_i are joined by

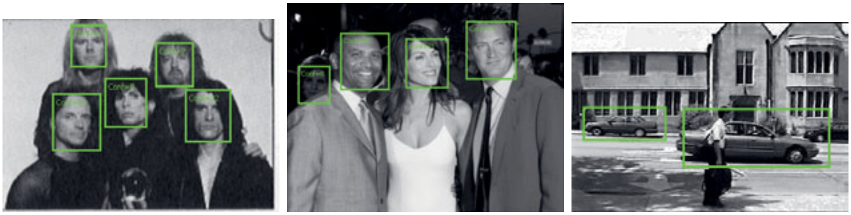
$$m_{D_i}(TP) = m_1(TP) \otimes m_2(TP) \otimes \ldots \otimes m_{N_S}(TP) \tag{8.6}$$

to gain an overall detection confidence.

8.1.4 Confidence-based Detection Merging

Merging of multiple detection commonly takes place in the post-processing step of an object detection framework. The position and scale information of the candidate sub-windows has to be processed to determine the true object location.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b) (c)

Figure 8.2: Example images showing detections of the proposed method on the three evaluated data sets: (a) MIT+CMU [141], (b) FDDB [74], and (c) UIUC lateral car database [3].

In this work, the candidate detections are first clustered using the Meanshift algorithm [30, 31] as the number of true objects and thus desired clusters is unknown in advance. The i-th candidate detection is hereby defined as a four-dimensional vector $D_i = (u_i,v_i,\iota_i,\kappa_i)^\top$ which represents the combined position $(u_i,v_i)^\top$ and scale $(\iota_i,\kappa_i)^\top$ in u and v-dimension. The set of n candidate detections is partitioned by the Meanshift algorithm in four-dimensional space into a family of subsets $\mathcal{L} = \{L_1,L_2,\ldots,L_j\}$ of $j \leq n$ candidate detection clusters L that are pairwise disjoint. The merged detections are then set as the cluster centers of the j clusters in $\mathcal{L}$ and a simple confidence of the j-th cluster is given by its cluster size $|L_j|$.

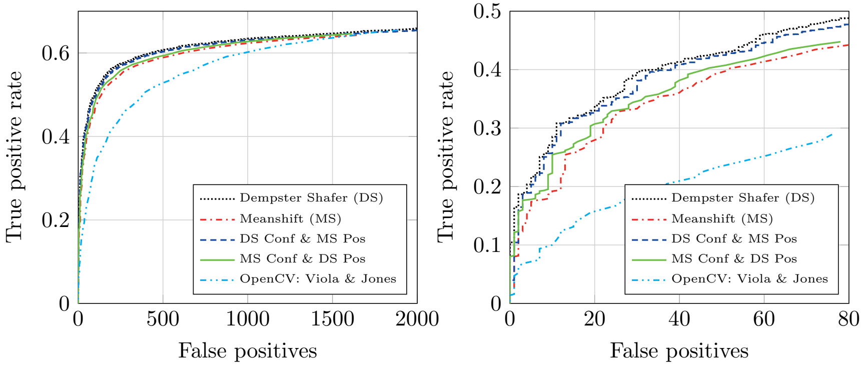
To improve the performance of the object detector, this chapter proposes two enhancements to the detection merging. First, the detection confidences given by Equation (8.6) are exploited to define the Dempster-Shafer based confidence of the j-th cluster as $\Gamma_j = \sum_{D_i \in L_j} m_{D_i}(TP)$. Second, these confidences of detections associated to one cluster are utilized to refine the position and scale of the cluster center. In this way, the Dempster-Shafer refined position/scale of the j-th cluster is defined by:

$$D'_j = \frac{1}{\Gamma_j} \sum_{D_i \in L_j} D_i m_{D_i}(TP) \tag{8.7}$$

8.2 Experimental Results

In this section, cascaded classifiers are applied by a sliding window to data sets for face and lateral car detection. The acquired multiple detections are post-processed using different merging strategies and results are presented for the Face Detection Data Set and Benchmark (FDDB), the MIT+CMU face data set and the UIUC car data set described in Section 2.2. Figure 8.2 exemplary shows detections found by the proposed method in the evaluated data sets.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a) (b)

Figure 8.3: Performance curves presenting the detector quality on FDDB [74] for different approaches on merging multiple detections. Confidence calculation and position/scale refinement based on Dempster-Shafer (DS) is compared to Meanshift-based confidence and position/scale (MS) as well as mixed approaches using Dempster-Shafer only for confidence and position/scale. The performance of the Viola and Jones implementation in OpenCV is presented as a baseline result. The shown range is (a) up to saturation and (b) a detailed view.

8.2.1 Face Detection

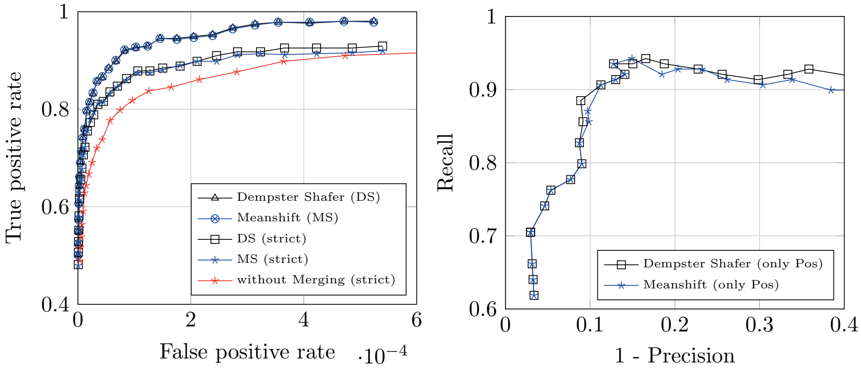
For the detection of faces, a classifier is trained on the MPLAB GENKI database, GENKI-4K subset introduced in Section 2.2. The obtained strong cascaded classifier consists of 10 stages and 593 weak classifiers in total.

Experiments Incorporating Confidence

The first experiments are conducted using the Face Detection Data Set and Benchmark. The inspection of the detection confidence enables the separate evaluation of two contributions in the proposed approach: The confidence computation based on Dempster-Shafer theory of evidence and the position and scale refinement using these confidences.

Figure 8.3 presents the detection results for different strategies on merging multiple detections. The performance of the Viola and Jones detector in OpenCV, supplied by the FDDB project page, is presented as a baseline result. But the primary topic of this work is the impact of the pre-processing step of multiple detection merging and not the comparison to different object detection methods. The proposed method (DS) is compared to an approach that only exploits the preceding Meanshift clustering (MS). For this, the number of detections forming each cluster is utilized as the

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.



(a)

(b)

Figure 8.4: (a) ROC curve presenting the detection performance on the MIT+CMU frontal face database. The effect of the position/scale refinement using Dempster-Shafer is compared to Meanshift clustering in the case of loosened and stricter ground truth tolerances. Additionally results when omitting multiple detection merging are presented. (b) Performance curve presenting the detector quality on the UIUC lateral car database. The effect of the additional position/scale refinement using Dempster-Shafer is compared to merging multiple detections by Meanshift clustering.

confidence value. In addition, the results of two mixed approaches are presented that use Dempster-Shafer solely for confidence calculation or position/scale refinement. The detailed view in Figure 8.3b demonstrates that, although the same detector is used, the performance can be significantly improved by about 5 % in terms of True Positive Rate. It can be also observed from the blue curve in Figure 8.3b that the proposed confidence computation causes the biggest part of the improvement. This demonstrates that the Dempster-Shafer confidence gives an appropriate measure to distinguish the reliability of detections. The position/scale refinement slightly improves the detection performance, indicating that the trained classifier is not detecting symmetrically around the true object location. The proposed refinement can correct that bias presenting improved results in the green curve of Figure 8.3b.

Experiments on Position/Scale Refinement

Additional experiments are performed on the MIT+CMU frontal face database. This test set gives ground truth information on the position and scale of the faces but no evaluation tool is provided. Hence, the evaluation against ground truth is done by a built-in function of the detection framework that governs the ROC curve by a threshold multiplier in the detection process instead of exploiting confidence values.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

For this reason, Figure 8.4a shows only the impact of the position/scale refinement. In addition, results for completely omitting the post-processing are presented as the built-in evaluation does not require the merging of multiple detections. The general benefit of the post-processing can be observed from the improved results compared to the approach without merging multiple detections. During the merging process detection outliers are suppressed that are outside the ground truth tolerances. The detector performance only slightly benefits from the position/scale refinement. This is partly a consequence of the properties of the MIT+CMU frontal face database that contains many very small faces but provides no subpixel accuracy in the ground truth data. As the accuracy of the detections position and scale has no influence on the ROC curve as long as they are inside the tolerances, additional results for stricter tolerances are presented by the curves labeled as strict. These curves reveal a slight improvement due to the proposed position/scale refinement even on this unfavourable test set.

8.2.2 Lateral Car Detection

To evaluate an additional object class, experiments are conducted on the UIUC lateral car database. Figure 8.4b compares the detection results achieved when merging multiple detections by Meanshift clustering and the proposed position/scale refinement using Dempster-Shafer confidences. The evaluation tool does not consider detection confidences but requires multiple detections to be merged to a single detection in advance. Hence, a concentration on only the impact of the position/scale refinement is predetermined. In this experiment, that utilizes a different object class, an improvement of the detection performance can be observed due to the position/scale refinement. This indicates that the car classifier as well does not detect symmetrically around the true object location but introduces a bias that can be rectified by the proposed method.

8.3 Discussion

This chapter presents a novel method for merging multiple detections which exploits classification information available in cascaded detectors. Two enhancements are proposed. First, Dempster-Shafer theory of evidence is applied to model a confidence measure which incorporates intra-cascade decision confidences and uncertainties. Second, a method is presented to refine the position and scale of merged detections based on these confidence measures. These methods can be easily integrated in existing detection frameworks to improve performance without retraining of typical cascaded detectors. Results are presented for a recent benchmark on unconstrained face detection (FDDB), the MIT+CMU face and the UIUC car database. The refinement of position and scale solely results in a slight improvement in detection performance. In addition, the proposed confidence measure shows an improvement of

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

5 % in TPR for applications that consider detection confidences. This demonstrates that Dempster-Shafer Theory of Evidence is a powerful technique to model and exploit intra-cascade confidences.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Chapter 9

Conclusion

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Summary

This thesis addresses the problem of visual object detection based on machine-learned classifiers. A distributed machine learning framework has been developed to train detectors for several object classes creating cascaded ensemble classifiers by the Adaptive Boosting (AdaBoost) algorithm.

Methods are proposed that enhance several components of an object detection framework to improve its performance. An approach for augmenting the training data is introduced in order to increase the performance of the learned detector. Improvements to the utilized features are proposed by feature mining methods that create feature sets customized to the object class. Additionally, a novel class of fractal feature is introduced. The detector post-processing is enhanced by a novel method that performs the Non-Maximum Suppression.

The proposed methods have been published at CAIP, SCIA, ISVC and CIARP.

Contributions

The major contributions of this dissertation are summarized in the following.

Learning from sparse training data: Detectors learned from training sets that contain only a small amount of positive samples often show only poor classification performance. In Chapter 5, a method is proposed that addresses this problem by augmenting the sparse training data. The key idea is to narrow the boundary between the positive and negative object class in order to constrain the AdaBoost algorithm to learn a more selective classifier. Hence, the training set is augmented based on a statistical model obtained from a Principal Component Analysis of the positive object class.

Fractal features: Chapter 6 proposes a novel type of features that comprise a wide variety of shapes by utilizing fractal structures. A special class of fractals, space filling curves, allows a smooth incorporation in a machine learning framework that processes Haar-like features since the fractal curves are stored in special integral image representations. Similar to Haar-like features, the fractal features are efficiently computed from the integral images.

Feature mining: In Chapter 7, generic methods are introduced to customize the feature pool to the specific object class that should be detected. Results are presented for frontal face, lateral car, and pedestrian detection. This approach addresses the problem that learning an object detector is in general computational expensive and very time-consuming. During detector learning, the AdaBoost algorithm evaluates the complete feature pool on the set of training images so that the training

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

complexity is greatly influenced by the size of the feature pool. The proposed methods allow to construct feature sets that are smaller but contain more distinctive features.

Two types of customized features are introduced in Chapter 7 that are complementary in exploiting image intensities of coarse structures as opposed to local gradient information:

2Rec features are generalized Haar-like features that are constructed to exploit characteristic image regions of the object class based on image segmentation. They represent a much wider variety of shapes compared to conventional Haar-like features since associated image regions are not constraint to be coherent but can be arbitrarily located. Due to the customization to the object class, the utilized pool of 2Rec features is significantly smaller compared to Haar-like features enabling a much faster and more successful detector training.

Keypoint HOG (KPHOG) features are customized to the object class by locating distinctive keypoints on multiple scales by a Harris corner detector. Gradient directions and magnitudes are exploited in local histograms based on regions around the the keypoints. In this way, KPHOG features represent fine object structures and are well suited to supplement 2Rec features.

A mixed learning that utilizes a feature pool consisting of both feature types is shown to create a superior detector that achieves higher detection performance and is more robust to noisy or low resolution image data. Additionally, the number of required features in the detector is strongly reduced by exploiting complementary features.

Dempster-Shafer based Non-Maximum Suppression: Commonly, object detectors perform a Non-Maximum Suppression (NMS) during post-processing to merge multiple detections of a single object to one final detection. Chapter 8 introduces a NMS method that models detection confidences and uncertainty using Dempster-Shafer Theory of Evidence. Several internal variables obtained from the detector cascade are combined by the evidence theory and incorporated to improve the position accuracy and overall confidence of merged detections.

Distributed machine learning framework: In order to utilize the huge amount of computing power that is required for learning high-performance object detectors, a distributed machine learning framework has been developed that parallelizes the detector learning on a computer cluster. Chapter 4 gives a brief introduction in the architecture of the framework that has been designed with a strong emphasis on generic programming allowing to transparently process different feature types.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Future Work

This thesis concludes in the following with a short discussion on future research.

A distributed machine learning framework has been developed as part of this work with a strong focus on generic programming in its architecture. This allows the framework to embed other machine learning algorithms as well as to transparently process different feature classes.

A field for future research is to apply concepts that are effective in other machine learning algorithms. In own works, Baumann et al. [13, 15, 17, 18] propose to improve the performance of Random Forests classifiers by incorporating concepts from Boosting algorithms. The cascade structure of the detector and the corresponding bootstrap strategy in learning is adapted to Random Forests in [13]. A weighting scheme inspired by AdaBoost is incorporated in [15], assigning individual weights to leaf nodes based on the tree topology and implementing a weighted majority decision in the forest. A different weighted majority voting with respect to class and tree specific error rates is proposed in [17].

The experiments in this thesis showed that a mixed detector learning that jointly utilizes complementary feature classes can significantly improve the detector's performance, robustness, and efficiency, meaning that the number of required features in the detector is reduced.

Recently, deep learning approaches, mostly performed by Convolutional Neural Networks, demonstrate very good results in many computer vision tasks [138, 77, 133]. Instead of learning from the feature responses to the training data, deep learning operates directly on the input data. The first layers of a trained CNN then represent structures comparable to conventional features. In the sense of transfer learning, these layers are frequently assumed to be universally usable and kept when learning for other visual recognition tasks [113]. Furthermore in recent research, these structures are extracted from CNNs and provide as features to other machine learning algorithms [138, 28].

Hence, a promising approach is to incorporate such CNN features in the developed framework for a joint detector learning. This strategy might be especially interesting for automotive applications that enable only very limited computational resources. In this way, capable features obtained from deep learning can be utilized in the efficient environment of a cascaded detector.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Appendix

A

Lindenmayer Systems Defining Fractal Integral Paths

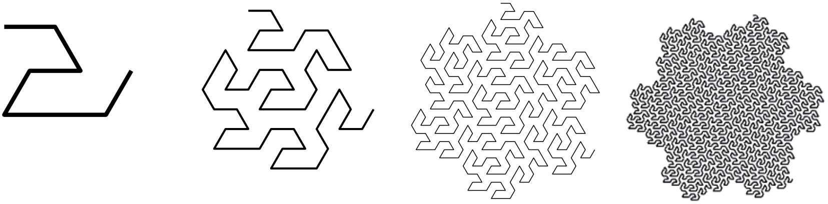
https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

A.1 L-System Defining Gosper Curve

- $\mathcal{V} = \{X{,}Y, + ,-\}$,
- $\omega = FX$,
- $\mathcal{R} : \begin{cases} X \rightarrow X + YF + +YF - FX - -FXFX - YF+ \\ Y \rightarrow -FX + YFYF + +YF + FX - -FX - Y \end{cases}$,

where F instructs the turtle bot to draw a line of length 1, + to rotate anticlockwise by 60° and − to rotate clockwise by 60°. The nonterminal symbols X and Y are ignored by the turtle bot.

Figure A.1 presents the first iterations of the Gosper curve drawn by the turtle bot using the instructions of the previously defined L-System.



(a) 1. iteration (b) 2. iteration (c) 3. iteration (d) 4. iteration

Figure A.1: First four iterations of the Gosper curve.

A.2 L-System Defining E-Curve

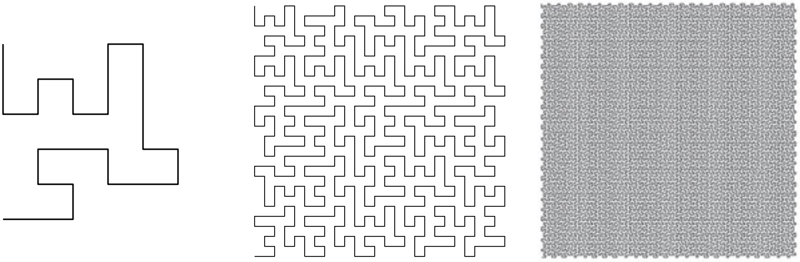
- $\mathcal{V} = \{X{,}Y, + ,-\}$,
- $\omega = X$,
- $\mathcal{R} : \begin{cases} X \rightarrow XXY + YX - XY + Y + X - -YXX - Y + +X - \\ \qquad -YXX - -Y + +X - Y + Y + +X - X - -YY + + \\ \\ Y \rightarrow XXY + YX - X - YXYY + +XYX - YY + +XY + \\ \qquad X - X - -Y + Y + +X - X - -YY \end{cases}$,
- $\mathcal{R}_{\text{final}} : \begin{cases} X \rightarrow FF + F + F - F - F + F + FF - F - FFF + F - \\ \qquad F - FF - F + FF + F + F - F - FF+ \\ \\ Y \rightarrow FF + F + F - F - FF - F + FF + F + F - FFF + \\ \qquad F + FF - F - F + F + F - F - FF- \end{cases}$,

where F instructs the turtle bot to draw a line of length 1, + to rotate anticlockwise by 90° and − to rotate clockwise by 90°. The set $\mathcal{R}$ of production rules is applied

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

iteratively to construct the symbol string that represents the fractal structure. In one final replacement step, the set $\mathcal{R}_{\text{final}}$ is used to substitute the nonterminal symbols by command strings for the turtle bot.

The output of the turtle bot is presented in Figure A.2 for the first iterations of the E-Curve defined in the preceding L-System. Since the production rules of the E-Curve are more complex, it grows much faster than the Gosper curve.



(a) 1. iteration (b) 2. iteration (c) 3. iteration

Figure A.2: First three iterations of the E-Curve.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Bibliography

[1] The MPLab GENKI Database, GENKI-4K Subset. URL `http://mplab.ucsd.edu`.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. *SLIC Superpixels.* Technical Report 149300, EPFL Technical Report, 2010.

[3] S. Agarwal, A. Awan, and D. Roth. *UIUC Image Database for Car Detection.* 2002.

[4] S. Ali and M. Shah. *An integrated approach for generic object detection using kernel PCA and boosting.* In IEEE International Conference on Multimedia and Expo (ICME), 2005.

[5] Y. Amit and D. Geman. *Shape Quantization And Recognition With Randomized Trees.* Neural Computation, 9(7):1545–1588, 1997.

[6] A. Ansari and A. Fineberg. *Image Data Ordering And Compression Using Peano Scan And LOT.* In IEEE International Conference on Consumer Electronics (ICCE), 1992.

[7] R. Appel, T. J. Fuchs, P. Dollar, and P. Perona. *Quickly Boosting Decision Trees - Pruning Underachieving Features Early.* In Proceedings of the 30th International Conference on Machine Learning (ICML), 2013.

[8] A. Barbu, N. Lay, and G. Gramajo. *Face Detection with a 3D Model.* CoRR, abs/1404.3596, 2014.

[9] M. F. Barnsley. *Fractals everywhere (2. ed.).* Academic Press, 1993.

[10] M. Bartlett, J. R. Movellan, and T. Sejnowski. *Face recognition by independent component analysis.* IEEE Transactions on Neural Networks, 13(6):1450–1464, 2002.

[11] L. E. Baum and T. Petrie. *Statistical inference for probabilistic functions of finite state Markov chains.* The annals of mathematical statistics, pages 1554–1563, 1966.

[12] F. Baumann, K. Ernst, A. Ehlers, and B. Rosenhahn. *Symmetry Enhanced Adaboost.* In 6th International Symposium on Advances in Visual Computing (ISVC), 2010.

[13] F. Baumann, A. Ehlers, K. Vogt, and B. Rosenhahn. *Cascaded Random Forest for Fast Object Detection.* In 18th Scandinavian Conference on Image Analysis (SCIA), 2013.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[14] F. Baumann, A. Ehlers, B. Rosenhahn, and J. Liao. *Computation strategies for volume local binary patterns applied to action recognition.* In 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2014.

[15] F. Baumann, F. Li, A. Ehlers, and B. Rosenhahn. *Thresholding a Random Forest Classifier.* In 10th International Symposium on Advances in Visual Computing (ISVC), 2014.

[16] F. Baumann, J. Liao, A. Ehlers, and B. Rosenhahn. *Motion Binary Patterns for Action Recognition.* In Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2014.

[17] F. Baumann, A. Ehlers, B. Rosenhahn, and W. Liu. *Sequential Boosting for Learning a Random Forest Classifier.* In IEEE Winter Conference on Applications of Computer Vision (WACV), 2015.

[18] F. Baumann, K. Vogt, A. Ehlers, and B. Rosenhahn. *Probabilistic nodes for modelling classification uncertainty for random forest.* In 14th IAPR International Conference on Machine Vision Applications (MVA), 2015.

[19] F. Baumann, A. Ehlers, B. Rosenhahn, and J. Liao. *Recognizing Human Actions using novel Space-time Volume Binary Patterns.* Neurocomputing, 173, Part 1:54–63, 2016.

[20] S. Belongie, J. Malik, and J. Puzicha. *Matching shapes.* In Eighth IEEE International Conference on Computer Vision (ICCV), 2001.

[21] R. Benenson, M. Omran, J. H. Hosang, and B. Schiele. *Ten Years of Pedestrian Detection, What Have We Learned?* In European Conference on Computer Vision (ECCV) Workshops, 2014.

[22] BioID-AG. *BioID face database.* BioID Switzerland, 2010.

[23] A. Blum and P. Langley. *Selection of Relevant Features and Examples in Machine Learning.* Artificial Intelligence, 97(1-2):245–271, 1997.

[24] G. Bradski. *The OpenCV library.* Doctor Dobbs Journal, 25(11):120–126, 2000.

[25] L. Breiman. *Bagging Predictors.* Machine Learning, 24(2):123–140, 1996.

[26] L. Breiman. *Random Forests.* Machine Learning, 45(1):5–32, 2001.

[27] T. Büschenfeld. *Klassifikation von Satellitenbildern unter Ausnutzung von Klassifikationsunsicherheiten.* Fortschritt-Berichte VDI.: Informatik/Kommunikationstechnik. VDI-Verlag, 2014.

[28] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. *Return of the Devil in the Details: Delving Deep into Convolutional Nets.* In British Machine Vision Conference (BMVC), 2014.

[29] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. *Joint Cascade Face Detection and Alignment.* In 13th European Conference on Computer Vision (ECCV), 2014.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[30] Y. Cheng. *Mean shift, mode seeking, and clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 17(8):790–799, 1995.

[31] D. Comaniciu and P. Meer. *Mean Shift: A Robust Approach Toward Feature Space Analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 24(5):603–619, 2002.

[32] C. Cortes and V. Vapnik. *Support-vector networks.* Machine learning, 20(3): 273–297, 1995.

[33] P. S. Crowther and R. J. Cox. *A Method for Optimal Division of Data Sets for Use in Neural Networks.* In 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES), 2005.

[34] R. Dafner, D. Cohen-Or, and Y. Matias. *Context-based Space Filling Curves.* Computer Graphics Forum, 19(3):209–218, 2000.

[35] N. Dalal and B. Triggs. *Histograms of oriented gradients for human detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[36] S. Dasgupta. *Learning mixtures of Gaussians.* In 40th Annual Symposium on Foundations of Computer Science. IEEE, 1999.

[37] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. *Linear Programming Boosting via Column Generation.* Machine Learning, 46(1-3):225–254, 2002.

[38] A. P. Dempster. *A generalization of Bayesian inference.* Journal of the Royal Statistical Society. Series B (Methodological), pages 205–247, 1968.

[39] H. Digabel and C. Lantuéjoul. *Iterative algorithms.* In Proceedings of the 2nd European Symposium Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, 1978.

[40] P. Dollár, Z. Tu, H. Tao, and S. Belongie. *Feature Mining for Image Classification.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[41] P. Dollár, Z. Tu, P. Perona, and S. Belongie. *Integral Channel Features.* In British Machine Vision Conference (BMVC), 2009.

[42] P. Dollár, C. Wojek, B. Schiele, and P. Perona. *Pedestrian Detection: An Evaluation of the State of the Art.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 34(4):743–761, 2012.

[43] A. Ehlers, F. Baumann, R. Spindler, B. Glasmacher, and B. Rosenhahn. *PCA Enhanced Training Data for Adaboost.* In 14th International Conference on Computer Analysis of Images and Patterns (CAIP), 2011.

[44] A. Ehlers, F. Baumann, and B. Rosenhahn. *Exploiting Object Characteristics Using Custom Features for Boosting-Based Classification.* In 18th Scandinavian Conference on Image Analysis (SCIA), 2013.

[45] A. Ehlers, B. Scheuermann, F. Baumann, and B. Rosenhahn. *Cleaning Up Multiple Detections Caused by Sliding Window Based Object Detectors.* In

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

18th Iberoamerican Congress on Pattern Recognition (CIARP), 2013.

[46] A. Ehlers, F. Baumann, and B. Rosenhahn. *Boosted Fractal Integral Paths for Object Detection.* In 10th International Symposium on Advances in Visual Computing (ISVC), 2014.

[47] M. Enzweiler and D. M. Gavrila. *Monocular Pedestrian Detection: Survey and Experiments.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 31(12):2179–2195, 2009.

[48] A. Ess, T. Mueller, H. Grabner, and L. J. Van Gool. *Segmentation-Based Urban Traffic Scene Understanding.* In British Machine Vision Conference (BMVC), 2009.

[49] European New Car Assessment Programme (Euro NCAP). Euro NCAP Puts Autonomous Pedestrian Detection to the Test, November 2015. URL `http://www.euroncap.com/en/press-media/press-releases/euro-ncap-puts-autonomous-pedestrian-detection-to-the-test/`. [Online; accessed 8-April-2016].

[50] M. Everingham, L. J. Van Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. *The Pascal Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, 88(2):303–338, 2010.

[51] M. Everingham, S. M. A. Eslami, L. J. Van Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. *The Pascal Visual Object Classes Challenge: A Retrospective.* International Journal of Computer Vision, 111(1):98–136, 2015.

[52] S. S. Farfade, M. J. Saberian, and L. Li. *Multi-view Face Detection Using Deep Convolutional Neural Networks.* In Proceedings of the 5th ACM International Conference on Multimedia (ICMR), 2015.

[53] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. *Object Detection with Discriminatively Trained Part-Based Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 32(9): 1627–1645, 2010.

[54] W. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. *Computer vision for computer games.* In Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996.

[55] Y. Freund. *Boosting a Weak Learning Algorithm by Majority.* Information and Computation, 121(2):256–285, 1995.

[56] Y. Freund and R. E. Schapire. *Experiments with a New Boosting Algorithm.* In Proceedings of the Thirteenth International Conference on Machine Learning (ICML), 1996.

[57] Y. Freund and R. E. Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* Journal of Computer and System Sciences, 55(1):119–139, 1997.

[58] E. Friedman and I. Maman. Boost C++ libraries: Boost.Variant. URL `ht`

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

`tp://www.boost.org/doc/libs/release/libs/variant/`. [Online; accessed 2-November-2015].

[59] J. Friedman, T. Hastie, and R. Tibshirani. *Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors).* The Annals of Statistics, 28(2):337–407, 2000.

[60] J. Fritsch, T. Kuehnl, and A. Geiger. *A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms.* In International Conference on Intelligent Transportation Systems (ITSC), 2013.

[61] K. Fukunaga and L. Hostetler. *The estimation of the gradient of a density function, with applications in pattern recognition.* IEEE Transactions on Information Theory, 21(1):32–40, 1975.

[62] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. *Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation.* In Proceedings of the 11th European PVM/MPI Users' Group Meeting, 2004.

[63] A. Geiger, P. Lenz, and R. Urtasun. *Are we ready for autonomous driving? The KITTI vision benchmark suite.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[64] G. Ghiasi and C. C. Fowlkes. *Occlusion Coherence: Detecting and Localizing Occluded Faces.* CoRR, abs/1506.08347, 2015.

[65] D. Gregor and M. Troyer. Boost C++ libraries: Boost.MPI. URL `http://www.boost.org/doc/libs/release/libs/mpi/`. [Online; accessed 2-November-2015].

[66] I. Guyon and A. Elisseeff. *An Introduction to Variable and Feature Selection.* Journal of Machine Learning Research, 3:1157–1182, 2003.

[67] I. Haritaoglu, D. Harwood, and L. S. Davis. *W4: real-time surveillance of people and their activities.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 22(8):809–830, 2000.

[68] C. Harris and M. Stephens. *A Combined Corner and Edge Detector.* In Proceedings of the Alvey Vision Conference (AVC), 1988.

[69] J. A. Hartigan and M. A. Wong. *Algorithm AS 136: A k-means clustering algorithm.* Applied statistics, pages 100–108, 1979.

[70] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., 2001.

[71] T. K. Ho. *Random decision forests.* In Third International Conference on Document Analysis and Recognition (ICDAR), 1995.

[72] T. K. Ho. *The Random Subspace Method for Constructing Decision Forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 20(8):832–844, 1998.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[73] C. Huang, H. Al, B. Wu, and S. Lao. *Boosting nested cascade detector for multi-view face detection.* In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004.

[74] V. Jain and E. Learned-Miller. *FDDB: A Benchmark for Face Detection in Unconstrained Settings.* Technical report, University of Massachusetts, Amherst, 2010. URL `http://vis-www.cs.umass.edu/fddb/results.html`. [Online; accessed 1-October-2015].

[75] V. Jain and E. Learned-Miller. *Online domain adaptation of a pre-trained cascade of classifiers.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[76] H.-X. Jia and Y.-J. Zhang. *Fast Human Detection by Boosting Histograms of Oriented Gradients.* In Fourth International Conference on Image and Graphics (ICIG), 2007.

[77] M. I. Jordan and T. M. Mitchell. *Machine learning: Trends, perspectives, and prospects.* Science, 349(6245):255–260, 2015.

[78] B. Jun, I. Choi, and D. Kim. *Local Transform Features and Hybridization for Accurate Face and Human Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 35(6):1423–1436, 2013.

[79] M. Kearns. *Thoughts on hypothesis boosting.* Unpublished manuscript, 1988.

[80] W. Kienzle, G. H. Bakir, M. O. Franz, and B. Sch&quot;olkopf. *Face Detection - Efficient and Rank Deficient.* In Advances in Neural Information Processing Systems (NIPS), 2004.

[81] D. Koller and M. Sahami. *Toward Optimal Feature Selection.* In Proceedings of the Thirteenth International Conference on Machine Learning (ICML), 1996.

[82] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. *Robust face detection by simple means.* In DAGM CVAW workshop, 2012.

[83] W. Lam and J. Shapiro. *A class of fast algorithms for the Peano-Hilbert space-filling curve.* In IEEE International Conference on Image Processing (ICIP), 1994.

[84] C. Lantuéjoul. *La squelettisation et son application aux mesures topologiques des mosaïques polycristallines.* PhD thesis, École nationale supérieure des mines de Paris, 1978.

[85] I. Laptev. *Improvements of Object Detection Using Boosted Histograms.* In British Machine Vision Conference (BMVC), 2006.

[86] C.-Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu. *Region-Based Discriminative Feature Pooling for Scene Text Recognition.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[87] S. H. Lee, M. K. Sohn, D. J. Kim, B. Kim, and H. Kim. *Smart TV interaction system using face and hand gesture recognition.* In IEEE International Conference on Consumer Electronics (ICCE), 2013.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[88] C. Leistner, H. Grabner, and H. Bischof. *Semi-supervised boosting using visual similarity learning.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[89] K. Levi and Y. Weiss. *Learning object detection from a small number of examples: the importance of good features.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[90] H. Li and C. Shen. *Boosting the Minimum Margin: LPBoost vs. AdaBoost.* In Digital Image Computing: Techniques and Applications (DICTA), 2008.

[91] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. *Probabilistic Elastic Part Model for Unsupervised Face Detector Adaptation.* In IEEE International Conference on Computer Vision (ICCV), 2013.

[92] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. *Efficient Boosted Exemplar-Based Face Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[93] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. *A Convolutional Neural Network Cascade for Face Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[94] J. Li and Y. Zhang. *Learning SURF Cascade for Fast and Accurate Object Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[95] J. Li, T. Wang, and Y. Zhang. *Face detection using SURF cascade.* In IEEE International Conference on Computer Vision (ICCV) Workshops, 2011.

[96] R. Lienhart and J. Maydt. *An extended set of Haar-like features for rapid object detection.* In IEEE International Conference on Image Processing (ICIP), 2002.

[97] A. Lindenmayer. *Mathematical models for cellular interaction in development: Parts I and II.* Journal of Theoretical Biology, 18:280–315, 1968.

[98] S. P. Lloyd. *Least squares quantization in PCM.* IEEE Transactions on Information Theory, 28(2):129–137, 1982.

[99] D. Lowe. *Object recognition from local scale-invariant features.* In Seventh IEEE International Conference on Computer Vision (ICCV), 1999.

[100] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, 60(2):91–110, 2004.

[101] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat. *A survey on industrial vision systems, applications and tools.* Image and Vision Computing, 21(2):171–188, 2003.

[102] J. Marin, D. Vazquez, A. Lopez, J. Amores, and B. Leibe. *Random Forests of Local Experts for Pedestrian Detection.* In IEEE International Conference on Computer Vision (ICCV), 2013.

[103] N. Markus, M. Frljak, I. S. Pandzic, J. Ahlberg, and R. Forchheimer. *A method for object detection based on pixel intensity comparisons.* CoRR,

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

abs/1305.4537, 2013.

[104] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean. *Boosting Algorithms as Gradient Descent.* In Conference on Advances in Neural Information Processing Systems (NIPS), 1999.

[105] M. Mathias, R. Benenson, M. Pedersoli, and L. J. Van Gool. *Face Detection without Bells and Whistles.* In 13th European Conference on Computer Vision (ECCV), 2014.

[106] W. S. McCulloch and W. Pitts. *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 5(4):115–133, 1943.

[107] E. Meijer and P. Drayton. *Static Typing Where Possible, Dynamic Typing When Needed: The End of theCold War Between Programming Languages.* In OOPSLA Workshop On The Revival Of Dynamic Languages, 2004.

[108] F. Meyer. *Topographic distance and watershed lines.* Signal processing, 38(1): 113–125, 1994.

[109] K. Mikolajczyk, C. Schmid, and A. Zisserman. *Human Detection Based on a Probabilistic Assembly of Robust Part Detectors.* In 8th European Conference on Computer Vision (ECCV), 2004.

[110] S. Milborrow, J. Morkel, and F. Nicolls. *The MUCT Landmarked Face Database.* Pattern Recognition Association of South Africa, 2010. URL `http://www.milbo.org/muct`.

[111] W. Nam, B. Han, and J. H. Han. *Improving object localization using macrofeature layout selection.* In IEEE International Conference on Computer Vision (ICCV) Workshops, 2011.

[112] OpenMP Architecture Review Board. The OpenMP API, 2015. URL `http://openmp.org`. [Online; accessed 2-November-2015].

[113] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. *Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[114] T. Ouni, A. Lassoued, and M. Abid. *Gradient-based Space Filling Curves: Application to lossless image compression.* In IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 2011.

[115] C. Papageorgiou, M. Oren, and T. Poggio. *A general framework for object detection.* In Sixth IEEE International Conference on Computer Vision (ICCV), 1998.

[116] K. Pearson. *On lines and planes of closest fit to systems of points in space.* Philosophical Magazine Series 6, 2(11):559–572, 1901.

[117] C. Peck and J. Mackay. Eye gaze control of dynamic information presentation, 2005. US Patent 6,886,137.

[118] M.-T. Pham, Y. Gao, V.-D. Hoang, and T.-J. Cham. *Fast polygonal integration and its application in extending haar-like features to improve object detection.*

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[119] S. Phuvan, T. K. Oh, N. P. Caviris, Y. Li, and H. H. Szu. *Texture analysis by space-filling curves and one-dimensional Haar wavelets.* Optical Engineering 31(09), 31(9):1899–1906, 1992.

[120] F. Porikli. *Integral histogram: a fast way to extract histograms in Cartesian spaces.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[121] M. T. Rahman and N. Kehtarnavaz. *Real-time face-priority auto focus for digital and cell-phone cameras.* IEEE Transactions on Consumer Electronics, 54(4):1506–1513, 2008.

[122] R. Ramey. Boost C++ libraries: Boost.Serialization. URL `http://www.boost.org/doc/libs/release/libs/serialization/`. [Online; accessed 2-November-2015].

[123] C. Reinders, F. Baumann, B. Scheuermann, A. Ehlers, N. Mühlpforte, A. Effenberg, and B. Rosenhahn. *On-The-Fly Handwriting Recognition Using a High-Level Representation.* In 16th International Conference on Computer Analysis of Images and Patterns (CAIP), 2015.

[124] X. Ren and J. Malik. *Learning a classification model for segmentation.* In Ninth IEEE International Conference on Computer Vision (ICCV), 2003.

[125] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, second edition, 2003.

[126] P. Sabzmeydani and G. Mori. *Detecting Pedestrians by Learning Shapelet Features.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[127] R. E. Schapire. *The Strength of Weak Learnability.* Machine Learning, 5: 197–227, 1990.

[128] R. E. Schapire. Explaining adaboost. In Empirical inference, pages 37–52. Springer, 2013.

[129] R. E. Schapire and Y. Singer. *Improved Boosting Algorithms Using Confidence-rated Predictions.* Machine Learning, 37(3):297–336, 1999.

[130] R. E. Schapire, Y. Freund, P. Barlett, and W. S. Lee. *Boosting the margin: A new explanation for the effectiveness of voting methods.* In Proceedings of the Fourteenth International Conference on Machine Learning (ICML), 1997.

[131] B. Scheuermann, A. Ehlers, H. Riazy, F. Baumann, and B. Rosenhahn. *Ego-motion compensated face detection on a mobile device.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2011.

[132] A. Schick, M. Fischer, and R. Stiefelhagen. *Measuring and evaluating the compactness of superpixels.* In Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

[133] J. Schmidhuber. *Deep learning in neural networks: An overview.* Neural Networks, 61:85–117, 2015.

[134] B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. *Kernel PCA pattern reconstruction via approximate pre-images.* Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, Springer Verlag, pages 147–152, 1998.

[135] S. Segui, M. Drozdzal, P. Radeva, and J. Vitrià. *An Integrated Approach to Contextual Face Detection.* In Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2012.

[136] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. *Pedestrian Detection with Unsupervised Multi-stage Feature Learning.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[137] G. Shafer et al. *A mathematical theory of evidence*, volume 1. Princeton university press, 1976.

[138] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014.

[139] X. Shen, Z. Lin, J. Brandt, and Y. Wu. *Detecting and Aligning Faces by Image Retrieval.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[140] K.-K. Sung and T. A. Poggio. *Example-Based Learning for View-Based Human Face Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 20(1):39–51, 1998.

[141] K.-K. Sung, T. A. Poggio, H. A. Rowley, S. Baluja, and T. Kanade. *MIT+CMU Frontal face dataset A, B and C.* MIT+CMU, 1998.

[142] M. Turk and A. Pentland. *Face recognition using eigenfaces.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1991.

[143] L. G. Valiant. *A Theory of the Learnable.* Communications of the ACM, 27 (11):1134–1142, 1984.

[144] B. S. Venkatesh and S. Marcel. *Fast bounding box estimation based face detection.* In 11th European Conference on Computer Vision (ECCV) Workshop on Face Detection, 2010.

[145] P. A. Viola and M. J. Jones. *Robust real-time face detection.* In Eighth IEEE International Conference on Computer Vision (ICCV), 2001.

[146] P. A. Viola and M. J. Jones. *Robust Real-Time Face Detection.* International Journal of Computer Vision, 57(2):137–154, 2004.

[147] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. *HOGgles: Visualizing Object Detection Features.* In IEEE International Conference on Computer Vision (ICCV), 2013.

[148] S. Walk, N. Majer, K. Schindler, and B. Schiele. *New features and insights for*

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

pedestrian detection. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[149] X. Wang, T. X. Han, and S. Yan. *An HOG-LBP human detector with partial occlusion handling.* In IEEE 12th International Conference on Computer Vision (ICCV), 2009.

[150] M. K. Warmuth, K. A. Glocer, and S. V. N. Vishwanathan. *Entropy Regularized LPBoost.* In 19th International Conference on Algorithmic Learning Theory (ALT), 2008.

[151] A. R. Webb. *Statistical pattern recognition.* John Wiley & Sons, second edition, 2002.

[152] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. *Toward Practical Smile Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 31(11):2106–2111, 2009.

[153] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. *Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 35(4): 882–897, 2013.

[154] J. Yan, Z. Lei, L. Wen, and S. Li. *The Fastest Deformable Part Model for Object Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[155] B. Yang, J. Yan, Z. Lei, and S. Z. Li. *Aggregate channel features for multi-view face detection.* In IEEE International Joint Conference on Biometrics (IJCB), 2014.

[156] C. Zhang and Z. Zhang. *A survey of recent advances in face detection.* Technical report, Microsoft Research, 2010.

[157] D. Zhang, S. Li, and D. Gatica-Perez. *Real-time face detection using boosting in hierarchical feature spaces.* In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004.

[158] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. *Face Detection Based on Multi-Block LBP Representation.* In International Conference on Advances in Biometrics (ICB), 2007.

[159] S. Zhang, C. Bauckhage, and A. Cremers. *Informed Haar-Like Features Improve Pedestrian Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[160] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[161] X. Zhu and D. Ramanan. *Face detection, pose estimation, and landmark localization in the wild.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Curriculum Vitae

Arne Ehlers
geboren am 24.11.1976
in Itzehoe

Beruf

Seit 12/2016	Entwicklungsingenieur bei der WABCO GmbH, Hannover
2010 - 2016	Wissenschaftlicher Mitarbeiter am Institut für Informationsverarbeitung (TNT) der Gottfried Wilhelm Leibniz Universität Hannover

Ausbildung

1997 - 2008	Studium des Informatik-Ingenieurwesen an der Technischen Universität Hamburg-Harburg
1983 - 1996	Schulische Ausbildung mit Abschluss Abitur am Sophie-Scholl-Gymnasium Itzehoe

Zivildienst

1996 - 1997	Zivildienst im Kreiskrankenhaus Rendsburg

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Online-Buchshop für Ingenieure

■■ VDI nachrichten

BUCHSHOP

Online-Shops

Fachliteratur und mehr - jetzt bequem online recherchieren & bestellen unter: www.vdi-nachrichten.com/Der-Shop-im-Ueberblick

Fahrzeugkonzepte für das 2. Jahrhundert Automobiltechnik
Vehicle Concepts for the 2nd Century of Automotive Technology
VDI-Berichte 1653

Täglich aktualisiert: Neuerscheinungen VDI-Schriftenreihen

Fortschritt-Berichte VDI
Weiterentwicklung der Endfunktionsprüfung von Verbrennungskraftmaschinen in der Großserienproduktion

Im Buchshop von vdi-nachrichten.com finden Ingenieure und Techniker ein speziell auf sie zugeschnittenes, umfassendes Literaturangebot.

Mit der komfortablen Schnellsuche werden Sie in den VDI-Schriftenreihen und im Verzeichnis lieferbarer Bücher unter 1.000.000 Titeln garantiert fündig.

Im Buchshop stehen für Sie bereit:

VDI-Berichte und die Reihe **Kunststofftechnik:**

Berichte nationaler und internationaler technischer Fachtagungen der VDI-Fachgliederungen

Fortschritt-Berichte VDI:

Dissertationen, Habilitationen und Forschungsberichte aus sämtlichen ingenieurwissenschaftlichen Fachrichtungen

Newsletter „Neuerscheinungen":

Kostenfreie Infos zu aktuellen Titeln der VDI-Schriftenreihen bequem per E-Mail

Autoren-Service:

Umfassende Betreuung bei der Veröffentlichung Ihrer Arbeit in der Reihe Fortschritt-Berichte VDI

Buch- und Medien-Service:

Beschaffung aller am Markt verfügbaren Zeitschriften, Zeitungen, Fortsetzungsreihen, Handbücher, Technische Regelwerke, elektronische Medien und vieles mehr – einzeln oder im Abo und mit weltweitem Lieferservice

VDI nachrichten | **BUCHSHOP** **www.vdi-nachrichten.com/Der-Shop-im-Ueberblick**

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-3**85510**-0

https://doi.org/10.51202/9783186855107 - Generiert durch IP 216.73.216.60, am 24.01.2026, 01:37:28. © Urheberrechtlich geschützter Inhalt. Ohne gesonderte
Erlaubnis ist jede urheberrechtliche Nutzung untersagt, insbesondere die Nutzung des Inhalts im Zusammenhang mit, für oder in KI-Systemen, KI-Modellen oder Generativen Sprachmodellen.