Uncertainty-Based Image Segmentation with Unsupervised Mixture Models

Von der Fakultät für
Elektrotechnik und Informationstechnik
der Technischen Universität Dortmund
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

von

Thorsten Wilhelm

Dortmund, 2019

Tag der mündlichen Prüfung:	30.10.2019
Hauptreferent:	Prof. Dr. rer. nat. Christian Wöhler
Korreferent:	Prof. Dr.-Ing Franz Kummert

Arbeitsgebiet Bildsignalverarbeitung Technische Universität Dortmund

ACKNOWLEDGEMENTS

First of all, I would like thank my supervisor Prof. Dr. rer. net. Christian Wöhler for giving me the opportunity to pursue a PhD. Thank you for giving me the freedom to work on my own ideas and sharing your knowledge with me. Further, I would like to thank Prof. Dr.-Ing. Franz Kummert for agreeing to be my second reviewer.

I would like to thank Prof. Dr.-Ing. Gernot A. Fink and Dr.-Ing. Rene Grzeszick for providing a wonderful working atmosphere in the DFG-project[1] we have been working on sucessfully. In great parts this project enabled the creation of this thesis.

In the last four years of being a research assistant at TU Dortmund University, I was fortunate enough to work in a pleasant working atmosphere with skilful colleagues. Special thanks to Malte Lenoch and Kay Wohlfarth for doing all the administrative stuff at the Image Analysis Group. Thanks to all the current and former members of the group I was happy to work with. I'm also thankful to my colleagues from the Information Processing Lab for the many fun hours we shared at table football. I would also like to thank the staff of the Pattern Recognition in Embedded Systems Group for the occasional drinks we shared and the fruitful discussions at work and afterwards. I wish you all the best for your own projects.

I'm in-depth grateful to Marcel Hess, Dominik Koßmann, Leonard Rothacker, and Kay Wohlfarth for taking their precious time to read through earlier versions of this manuscript and the valuable feedback they provided. Thank you.

I want to thank my friends and family for providing the right distractions at the right times and I'm grateful to my parents who have always supported me unconditionally. Lastly, I want to thank my wonderful wife. Jenny, I thank you whole-heartedly for your patience and support, giving me the freedom to pursue my dreams. And thank you, Leah, for providing just the right motivation to actually finish this thesis. I'm looking forward to seeing you.

Dortmund, April 21, 2019

1 Project number 269661170

ABSTRACT

In this thesis, a contribution to explainable artificial intelligence is made. More specifically, the aspect of artificial intelligence which focusses on recreating the human perception is tackled from a previously neglected direction. A variant of human perception is building a mental model of the extents of semantic objects which appear in the field of view. If this task is performed by an algorithm, it is termed image segmentation. Recent methods in this area are mostly trained in a supervised fashion by exploiting an as extensive as possible data set of ground truth segmentations. Further, semantic segmentation is almost exclusively tackled by Deep Neural Networks (DNNs).

Both trends pose several issues. First, the annotations have to be acquired somehow. This is especially inconvenient if, for instance, a new sensor becomes available, new domains are explored, or different quantities become of interest. In each case, the cumbersome and potentially costly labelling of the raw data has to be redone. While annotating keywords to an image can be achieved in a reasonable amount of time, annotating every pixel of an image with its respective ground truth class is an order of magnitudes more time-consuming. Unfortunately, the quality of the labels is an issue as well because fine-grained structures like hair, grass, or the boundaries of biological cells have to be outlined exactly in image segmentation in order to derive meaningful conclusions. Second, DNNs are discriminative models. They simply learn to separate the features of the respective classes. While this works exceptionally well if enough data is provided, quantifying the uncertainty with which a prediction is made is then not directly possible. In order to allow this, the models have to be designed differently. This is achieved through generatively modelling the distribution of the features instead of learning the boundaries between classes. Hence, image segmentation is tackled from a generative perspective in this thesis. By utilizing mixture models which belong to the set of generative models, the quantification of uncertainty is an implicit property. Additionally, the dire need of annotations can be reduced because mixture models are conveniently estimated in the unsupervised setting.

Starting with the computation of the upper bounds of commonly used probability distributions, this knowledge is used to build a novel probability distribution. It is based on flexible marginal distributions and a copula which models the dependence structure of multiple features. This modular approach allows great flexibility and shows excellent performance at image segmentation. After deriving the upper bounds, different ways to reach them in an unsupervised fashion are presented. Including the probable locations of edges in the unsupervised model estimation greatly increases the performance. The proposed models surpass state-of-the-art accuracies in the generative and unsupervised setting and are on-par with many discriminative models. The analyses are conducted following the Bayesian paradigm which allows computing uncertainty estimates of the model parameters. Finally, a novel approach combining a discriminative DNN and a local appearance model in a weakly supervised setting is presented. This combination yields a generative semantic segmentation model with minimal annotation effort.

KURZFASSUNG

In dieser Arbeit wird ein Beitrag zur Interpretierbarkeit von künstlichen Intelligenzen vorgestellt. Der Teilaspekt der künstlichen Intelligenz, der sich der Nachbildung der menschlichen Wahrnehmung widmet, wird von einer bisher vernachlässigten Richtung aus neu betrachtet. Die menschliche Wahrnehmung erstellt ein mentales Modell der Größe von semantischen Objekten, die ein Mensch sieht. Wird diese Aufgabe von einer Maschine durchgeführt spricht man von einer Bildsegmentierung. Aktuelle Ansätze in diesem Bereich nutzen nahezu ausschließlich tiefe neuronale Netze, die mit riesigen Mengen an annotierten Daten überwacht trainiert werden.

Dies hat jedoch zur Folge, dass sobald neue Sensoren, neue Domänen oder schlicht andere Größen von Interesse sind, das mühsame Annotieren der Daten erneut durchgeführt werden muss. Eine Annotation durch Schlüsselwörter kann unter Umständen in einem zeitlich vertretbaren Rahmen durchgeführt werden. Allerdings werden zum Trainieren von Segmentierungsnetzen Annotationen für jedes Pixel eines Bildes benötigt. Außerdem ist die Qualität der Annotation im Auge zu behalten, da nur so feine Strukturen wie Haare, Gras oder die genauen Grenzen biologischer Zellen erhalten bleiben. Weiterhin sind tiefe neuronal Netze diskriminative Klassifikatoren und lernen die Daten der unterschiedlichen Klassen zu trennen. In der Praxis funktioniert dieser Ansatz sehr gut, wenn genug Trainingsmaterial vorhanden ist. Allerdings ist in diesem Szenario keine Angabe von Unsicherheiten bei der Klassifikation möglich. Um dies zu erreichen muss die Verteilung der Daten aller Klassen geschätzt werden. Eine Methode, die das ermöglicht sind Mischverteilungsmodelle, die generativ die Verteilung der Daten im Raum mit Wahrscheinlichkeitsverteilungen nachbilden. Zusätzlich kann in diesem Fall auf Annotation verzichtet werden.

In dieser Arbeit werden zunächst die oberen Schranken der Genauigkeiten beim Segmentieren von Bildern mit verschiedenen Wahrscheinlichkeitsverteilungen berechnet. Dieses Wissen wird anschließend genutzt um eine Wahrscheinlichkeitsverteilung einzuführen, die auf flexiblen Randverteilungen und einer Copula basiert, die die Abhängigkeiten von mehreren Merkmalen modelliert. Dieser modulare Ansatz ermöglicht es sehr flexibel Verteilungen zu kreieren und zeigt eine sehr hohe Genauigkeit beim Segmentieren von Bilddaten. Im Anschluss werden Ansätze präsentiert wie die überwacht gelernte obere Schrank im unüberwachten Fall erreicht werden kann. Hier hat sich vor allem das probabilistische Modellieren für das Auftreten von Kanten als äußerst wichtig herausgestellt. Die vorgestellten Modelle übertreffen den aktuellen Stand der Technik vergleichbarer Modelle und werden im Sinne der Bayes-Statistik geschätzt. Dieses Vorgehen ermöglicht die Angabe von Unsicherheiten in der Schätzung der Modellparameter. Abschließend wird in dieser Arbeit eine Methode vorgestellt, die es ermöglicht ein diskriminativ trainiertes tiefes neuronales Netz mit einem lokalen Model zur Beschreibung der visuellen Erscheinung zu kombinieren. Das tiefe Netz wird dabei nur mit schwacher Überwachung trainiert. Die Kombination beider Ansätze ermöglicht es Bilder mit minimalem Annotationsaufwand semantisch zu segmentieren und außerdem Unsicherheiten in der Segmentierung anzugeben.

PUBLICATIONS

This thesis is based on the following publications of the author. The publications are listed in ascending chronological order.

Peer-reviewed conference contributions

[WW16] T. Wilhelm and C. Wöhler. "Flexible Mixture Models for Colour Image Segmentation of Natural Images." In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2016, pp. 1–7.

[WW17c] T. Wilhelm and C. Wöhler. "Improving Bayesian Mixture Models for Colour Image Segmentation with Superpixels." In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, (VISIGRAPP 2017)*. INSTICC. SciTePress, 2017, pp. 443–450.

[WW17a] T. Wilhelm and C. Wöhler. "Boundary aware image segmentation with unsupervised mixture models." In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3325–3329.

[Wil+17] T. Wilhelm, R. Grzeszick, G. A. Fink, and C. Wöhler. "From Weakly Supervised Object Localization to Semantic Segmentation by Probabilistic Image Modeling." In: *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2017, pp. 1–7.

[WW17b] T. Wilhelm and C. Wöhler. "On the suitability of different probability distributions for the task of image segmentation." In: *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. 2017, pp. 1–6.

Other publications which are not part of this thesis and co-authored work is listed in ascending chronological order in the following.

Peer-reviewed conference contributions

[LWW16] M. Lenoch, T. Wilhelm, and C. Wöhler. "Simultaneous Surface Segmentation and BRDF Estimation via Bayesian Methods." In: *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, (VISIGRAPP 2016)*. INSTICC. SciTePress, 2016, pp. 39–48.

[Woh+18] K. Wohlfarth, C. Schröer, M. Klaß, S. Hakenes, M. Venhaus, S. Kauffmann, T. Wilhelm, and C. Wöhler. "Dense Cloud Classification on Multispectral Satellite Imagery." In: *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*. 2018, pp. 1–6.

[Wil+19] T. Wilhelm, R. Grzeszick, G. Fink, and C. Wöhler. "Unsupervised Learning of Scene Categories on the Lunar Surface." In: *Proceedings of the 14th International*

Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, INSTICC. SciTePress, 2019, pp. 614–621.

Invited Talks

Thorsten Wilhelm. "Restraining AI's hunger for annotated training data—weak supervision in autonomous driving." *Image Sensors Auto Europe 2019*, Berlin 2019

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AIC	Akaike Information Criterion
AEM	Active Edge Movement
BIC	Bayesian Information Criterion
BoF	Bag-of-Features
BSDS500	Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11]
BSDS300	Berkeley Segmentation Data Set and Benchmarks 300 [Mar+01]
cdf	cumulative distribution function
CorLoc	Correct Localization
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DPMM	Dirichlet Process Mixture Model
DRAM	Delayed Rejection Adaptive Sampling
EM	Expectation Maximisation
FCN	Fully Convolutional Network
GMM	Gaussian Mixture Model
GP	Gaussian Process
HoG	Histogram of Oriented Gradients
IoU	Intersection over Union
KDE	Kernel Density Estimation
MCMC	Markov chain Monte Carlo
MRF	Markov Random Field
MDS	Multidimensional Scaling
MTMM	Multiple Scaled t-distribution Mixture Model
OWT	Oriented Watershed Transform
PCA	Principal Component Analysis
PEM	Passive Edge Model
pdf	probability density function
pmf	probability mass function
PRI	Probabilistic Rand Index
ReLU	Rectified Linear Unit
RCNN	Regional Convolutional Neural Network
SCMM	Sinh-asinh Copula Mixture Model
SIFT	Scale-invariant Feature Transform
SLIC	Simple linear iterative clustering [Ach+12]
SVM	Support Vector Machine
UCM	Ultrametric Contour Map
VOC	the PASCAL Visual Object Classes Challenge 2012 [Eve+15]
VoI	Variation of Information

Mathematical Expressions

x	a scalar.
$\mathbf{x}$	a vector.
$\mathbf{X}$	a matrix.
$\mathcal{X}$	a set.
$f(x)$	a function.
$\log(x)$	the natural logarithm.
$p(x)$	a probability density function.
$P(x)$	a cumulative distribution function.
$\mathbf{x}^T, \mathbf{X}^T$	the transposed.
$n!$	the factorial of a positive integer n.
$\binom{n}{k}$	the binomial coefficient.
$\lvert x \rvert$	the absolute value of a scalar.
$\lVert \mathbf{x} \rVert_2$	the euclidean norm of a vector.
$\bar{\mathbf{x}}$	the mean of a vector.
$\tilde{\mathbf{x}}$	the median of a vector.
x_j	the j-th entry of the vector $\mathbf{x}$.
$x^{(t)}, \mathbf{x}^{(t)}, \mathbf{X}^{(t)}, \mathcal{X}^{(t)}$	a variable at time t.
$X_{i,j}$	the j-th entry of the i-th row of the matrix $\mathbf{X}$.
$\mathbf{X}_i$	the i-th row of the matrix $\mathbf{X}$.
$\mathbf{X}_j$	the j-th coloumn of the matrix $\mathbf{X}$.
$\mathcal{X}_i$	the i-th entry of the set $\mathcal{X}$.
$\lvert \mathcal{X} \rvert$	the cardinality of a set.
$\mathcal{X} \cup \mathcal{Y}$	the union of the sets $\mathcal{X}$ and $\mathcal{Y}$.
$\mathcal{X} \cap \mathcal{Y}$	the intersection of the sets $\mathcal{X}$ and $\mathcal{Y}$.

Greek Symbols

α	a shape parameter of a beta distribution.
β	a shape parameter of a beta distribution.
$\gamma, \boldsymbol{\gamma}$	the asymmetry parameter of a generalized hyperbolic distribution.
$\Gamma(x)$	the gamma function.
$\delta, \boldsymbol{\delta}$	the tail parameter of a sinh-asinh distribution.

$\epsilon, \boldsymbol{\epsilon}$	the asymmetry parameter of a sinh-asinh distribution.
ζ	the concentration parameter of a Dirichlet distribution.
η	a tail parameter of a generalized hyperbolic distribution.
θ	a parameter of a probabilistic model, the quantity of interest.
$\boldsymbol{\Theta}$	the set of all parameters of a probabilistic model.
ι	the dispersion parameter of a Dirichlet process.
κ	the concentration parameter of a beta distribution.
$\lambda, \boldsymbol{\lambda}$	an eigenvalue, the vector of all eigenvalues.
$\boldsymbol{\Lambda}$	the diagonal matrix of all eigenvalues.
$\mu, \boldsymbol{\mu}$	the mean value, the location parameter of a distribution.
$\nu, \boldsymbol{\nu}$	degrees of freedom of a t-distribution/ multiple scaled t-distribution.
ξ	the acceptance pobability in a Metropolis-Hastings step.
ρ	the mode of a beta distribution.
$\sigma, \boldsymbol{\sigma}$	the standard deviation, the scale parameter of a distribution.
σ^2	the variance.
$\boldsymbol{\Sigma}$	a covariance matrix.
$\tau, \boldsymbol{\tau}$	the probability of success in a binomial/multinomial experiment.
$\phi_k(x)$	the probability density function of the k-th mixture component in a mixture model.
$\varphi(x)$	the distribution function of a standard normal distribution.
$\varphi^{-1}(x)$	the quantile function of a standard normal distribution.
χ	a tail parameter of the generalized hyperbolic distribution.
$\psi, \boldsymbol{\psi}$	an angle, a vector of angles.
$\omega, \boldsymbol{\omega}$	the mixture weights of a mixture model.

Roman Symbols

$\mathbf{1}$	the identity matrix.
a	the shape parameter of a gamma distribution.
$\mathcal{A}$	the set of all class activation maps.

$\mathbf{A}$	a class activation map.
b	the scale parameter of a Gamma distribution.
b, B	the permutation index and the number of permuations used in a permutation test.
$\mathbf{Bg}$	the background map used in deriving a semantic segmentation.
$\mathbf{B}$	a boundary map derived from a segmentation.
c, C	the class index and the total number of classes.
$\mathbf{C}$	a correlation matrix.
$c(\mathbf{x})$	the probability density function of a copula.
$C(\mathbf{x})$	the distribution function of a copula.
$C(\mathcal{S} \to \mathcal{S}')$	the segmentation covering of two segmentations $\mathcal{S}$ and $\mathcal{S}'$.
i, D	the dimension index and the total number of dimensions.
$\mathbf{E}$	the edge map of an image.
$E[x]$	the expectation value of a random variable x.
F	the F-measure.
$\mathcal{FP}$	the set of all false positives in a binary classification.
$\mathcal{FN}$	the set of all false negatives in a binary classification.
g_x, g_y	the image gradient in x- and y-direction.
$\mathbf{gt}$	a ground truth vector.
$G(\cdot)$	a Dirichlet process.
G_0	the base distribution of a Dirichlet process.
$\mathcal{G}$	a ground truth segmentation.
h	the bandwidth of a kernel function.
$H(x)$	the entropy of a discrete random variable.
H_0	the null hypothesis in a significance test.
H_1	the alternative hypothesis in a significance test.
H	the height of an image.
$I_\nu(x)$	the modified Bessel function of the first kind.
$\mathcal{I}$	an image.
$I(x, y)$	the mutual information of two discrete random variables.
k, K	the cluster/mixture component index and the total number of clusters/mixture components.
$K_\nu(x)$	the modified Bessel function of the second kind.
$k(x)$	a kernel function.

l_{ce}	the cross-entropy loss.
$L(\theta)$	the likelihood of a model.
$\mathcal{L}$	a line segment of a boundary map.
$\mathcal{L}_{seg}$	the set of all line segments of a boundary map.
m	the mean function of a Gaussian process.
M	the number of all free model paramters.
j, N	the instance index and the number of instances.
$\mathbb{N}^+$	the natural numbers excluding 0.
N_c	the number of colour channels.
N_R	the number of rotation matrices.
N_{img}	the number of pixels in an image.
N_s	the number of successes in a binomial experiment.
N_{spx}	the number of superpixels.
N_t	the number of trials in a binomial experiment.
$\mathcal{P}$	a set of pairs of pixels.
$\mathbf{Q}$	the matrix of transition probabilities of a Markov chain.
$r_{j,k}$	a binary cluster indicator set to true if the j-th instance belongs to the k-th cluster.
$\mathcal{R}$	a region of a segmentation.
$\mathbf{R}$	a rotation matrix.
$\mathbb{R}$	the real numbers.
sd	an estimate of the error induced by not using all possible permutations in a permutation test.
$\mathcal{S}$	a segmentation.
$\mathcal{TP}$	the set of all true positives in binary classification.
$T(\cdot)$	a test statistic.
$\mathbf{t}$	the vector of all computed test statistics in a permutation test.
t, T	a time index and the total duration.
u	the lower boundary of the uniform distribution.
v	the upper boundary of the uniform distribution.
$\mathbf{V}$	the eigenvectors of a matrix.
$\mathbf{W}$	the affinity matrix of a graph.
y	a class index.
$\hat{y}$	the predicted class index.
z	a latent variable of a mixture model.

1 INTRODUCTION

Partitioning an image into coherent regions is an ongoing area of research in computer vision and the study of artificial intelligence in general. In contrast to scene or object recognition, not only the presence or absence of objects is predicted but also their full extent. In fact, every pixel in an image is classified. This enables us to know not only where an object appears but also to which extent an object occupies an image. This is, for instance, important when autonomous systems try to perceive their surroundings as in autonomous driving, when geologists build maps, or when physicists analyse medical images.

Nowadays, these problems are mostly solved with Deep Neural Networks (DNNs) like [RFB15; LSD15; BHC15]. However, such techniques impose some limitations as well. For instance, they usually require a large amount of labelled training data [BV16a]. Acquiring them is especially challenging in the case of image segmentation because every pixel has to be annotated by a ground truth class. According to [Bea+16] annotating every pixel of an image takes roughly ten times longer (240 seconds compared to 20 seconds on average per image) than simply annotating the presence or absence of a set of object classes. The high effort makes creating datasets at best cumbersome, but mostly expensive. Another limitation of the DNN-based approach is the absence of useful uncertainty estimates, because the data are not modelled generatively, but only the decision boundary between classes is learned. This is a major drawback since the prediction of autonomous systems should be explainable. In fact, this is currently a major topic in politics and one of the central targets for artificial intelligence made in Europe[1].

In this thesis, a contribution towards explainable artificial intelligence is made by focussing on generatively modelling the observed data. This is achieved by utilizing a statistical concept termed *mixture models*. They are chosen because they offer uncertainty estimates through generatively modelling the observed data in the feature space with probability distributions. Essentially, the model does not only learn to decide but to accompany a decision with a probability which can be interpreted as a measure of confidence. Additionally, mixture models can directly be used in an unsupervised setting where the algorithms are only equipped with the raw data without any ground truth information. Mixture models then proceed at estimating the probability density function (pdf) of the observed data by building a mixture of multiple probability distributions. In this setting, it is assumed that every region is well described by *one* probability distribution.

In general, generatively modelling the data is harder than learning the decision boundary because the classes' densities have to be learned as well. Naturally, this imposes several challenges, like the suitability of the data and the used probability distributions. Further, in an unsupervised setting, the number of different regions in an image is a-

1 The European Commission's high-level expert group on artificial intelligence, "Draft Ethics guidelines for trustworthy AI" `https://ec.europa.eu/digital-single-market/en/news/draft-ethics-guidelines-trustworthy-ai` Retrieved June 23, 2020

priori unknown and has to be estimated. All these challenges are tackled in this thesis from a probabilistic perspective.

1.1 CONTRIBUTION

The contributions depicted in this thesis consist of three major parts. In the first part, the suitability of various probability distributions and image features are analysed in a supervised way. This successfully establishes an upper bound of the reachable segmentation accuracy. Additionally, a novel probability distribution is presented which shows an excellent performance. Using different distributions and their impacts are rigorously tested on predefined data sets and the impacts are assessed with statistical significance tests. The results have been previously presented in [WW16] and [WW17b]. This thesis additionally includes more probability distributions, an analysis regarding the used features, and an analysis regarding the expressiveness of various probability distributions.

In the second part, different ways to reduce the computational burden are examined and the mixture estimation is tackled from an unsupervised perspective. The major goal is to reach the previously defined upper bound. It is shown that one of the most limiting factors in an unsupervised estimation is the maximization of the model's likelihood. By including an additional model of the probable locations of region boundaries into the estimation, the performance is significantly increased. As a result, the gap between the upper bound and the unsupervised estimation is reduced. Again, the experiments are conducted on predefined data sets and accompanied by significance testing. The results have been previously presented in [WW17a] and [WW17c]. This thesis additionally includes in-depth experiments regarding the estimation of the number of mixture components in the unsupervised setting. Further, previously unpublished results yielding the current state-of-the-art in generative image segmentation are presented.

Lastly, a probabilistic image model is combined with a DNN. The concept of class activation maps is leveraged to get a generative segmentation model from a discriminatively trained DNN. The results have been previously presented in [Wil+17].

1.2 OUTLINE

Following the introduction, the remainder of this thesis is structured as follows: Chapter 2 reviews image segmentation from a general perspective, important terms and concepts are introduced, a review of different approaches is given, the benefits of generative models are discussed, and the requirements of a *good* segmentation model are defined. Additionally, relevant data sets used throughout this thesis are presented. This is followed by an explanation of the segmentation metrics used in this thesis to quantitatively asses the quality of the segmentations.

Chapter 3 presents a concise summary of the relevant statistical concepts used in this thesis. Beginning with an explanation of the Bayesian paradigm, different probability distributions are introduced. Further, Markov chain Monte Carlo (MCMC) is explained and significance testing in the context of image segmentation is briefly discussed. Building on the previously introduced concepts, the suitability of various probability distributions in the context of image segmentation is analysed in Chapter 4. The novel Sinh-asinh Copula Mixture Model (SCMM) is introduced and the upper bound of the segmentation

metrics of various probability distributions are computed. While the estimation of the upper bound has been conducted in a supervised fashion, in Chapter 5 the best way to reach the upper bound in an unsupervised way are examined. Afterwards, the gained knowledge about generative image segmentation is used in Chapter 6 to build a generative segmentation model from a discriminatively trained DNN. The experiments are conducted in the challenging weakly supervised setting. In the last chapter the results are summarized and possible future research directions are sketched.

2 IMAGE SEGMENTATION

The goal of image segmentation is to divide an image into regions. Each region is a set of pixels which share a specific trait. Depending on the analysed images these traits can vary. Typical traits are, among others, colour (e. g., [CM02]), texture (e. g., [JF91]), or semantic meaning (e. g., [Eve+15]).

Image segmentation is, for instance, applied in autonomous driving (e. g., [Cor+16]). The car needs to perceive its surroundings accurately in order to drive safely. One part of this perception is to know the extents of all occurring objects in the car's field of view. Typical objects which appear in a street scene are, among others, cars, persons, or street signs. In terms of image segmentation, the goal is to automatically divide the pixels of an image into disjoint sets where each region includes only one type of object. For instance, all pixels which depict a car are grouped into one region.

Creating maps is also a similar task. Like geologists build maps from remotely sensed data, image segmentation algorithms build regions from analysing the provided data (see e. g., [Zhu+17]). Depending on the features these results may differ and depending on which model is used the interpretation changes. Another example of image segmentation in practice is assisting physicians in analysing medical image data (see e. g., [Lit+17]) where they are trying to detect anomalies in the sizes of cells or organs. In every case, every pixel of an image needs to be given a label which indicates the region membership. Despite the possible semantic interpretation of a region, the question arises what is a *good* segmentation and how can it be defined. This is especially relevant if a semantic interpretation is not directly available or desired. In the literature, several properties are listed which characterize a good segmentation. For instance, according to [HP17] a segmentation of an image should have:

1. uniform and homogeneous regions according to some criterion,
2. significantly different adjacent regions,
3. simple regions without holes,
4. simple and accurate boundaries.

In image segmentation, a boundary is the set of pixels which have more than one region in its direct vicinity. While all these properties are worth following, in this thesis, two additional goals are followed. A segmentation should additionally have:

5. sensible uncertainty estimates,
6. been computed with an as small amount of expert knowledge as possible.

One of the major goals of this thesis is to additionally provide meaningful uncertainty measures regarding the affinity with which pixels belong to the regions of an image. This is a helpful tool when communicating segmentation results to domain experts, like physicians in medical image analysis or geologists in remote sensing applications.

Another key aspect is to develop methods which solve the image segmentation problem by using as few human annotations as possible. This is important because acquiring human annotations is at best cumbersome but mostly expensive. For instance, according to [Bea+16] annotating every pixel of an image takes roughly ten times longer than simply stating if an object is present or not. When creating annotations of everyday scenes this work can be crowdsourced (e. g., [Rus+15]) or brought by citizen scientists (e. g., [Swa+15]). However, in other fields, like medical image analysis or when analysing remotely sensed surfaces of planets, domain experts are needed to annotate data.

This chapter starts with a short treatise of the fundamentals of probability theory in Section 2.1 because the remainder of this thesis heavily relies on it. Depending on the used data (see Section 2.2), image segmentation can be done with the help of different levels of supervision (see Section 2.3). In Section 2.4 model-based approaches are reviewed in the context of image segmentation. Distance-based approaches are reviewed in Section 2.5, and approaches based on detecting contours in Section 2.6. This is followed by a small treatise of superpixels (see Section 2.7), which can be considered as a special case of image segmentation. Semantic segmentation, which is another special case of image segmentation, and Deep Neural Networks (DNNs) are discussed in Section 2.8. Afterwards, scene and object detection tasks (see Section 2.9) and segmentation tasks (see Section 2.10) are presented. The chapter concludes with a discussion of common evaluation metrics used in image segmentation (see Section 2.11).

2.1 FUNDAMENTALS OF PROBABILITY THEORY

Probabilities are a key concept in machine learning in general and in image segmentation in particular. In this section, the elementary notations and concepts are briefly introduced. Detailed introductions can, among others, be found in [Bis06; Mur12; BH14].

Following [Ly+17], a statistical model is a function $f(x_i|\theta)$ which relates the potential outcomes x_i of a random variable x through a set of parameters θ. The potential outcomes are elements of an output space $\mathcal{X}$ which is the set of all possible events or values the random variable can take. If $\mathcal{X}$ is a finite set of discrete events, like the result of a coin toss, the random variable x is termed *discrete*. Further, the function $f(x_i|\theta)$ is known as the probability mass function (pmf). In Section 3.2.1, some commonly used pmfs are presented. If $\mathcal{X}$ is continuous, the random variable x is termed *continuous* and $f(x_i|\theta)$ is known as the probability density function (pdf). In Section 3.2, relevant pdfs used in this thesis are presented.

The probability of observing the outcomes of two random variables together is known as the *joint* probability of x and y denoted by $p(x, y)$. For instance, two dice may be thrown simultaneously. The joint probability of both dice is then the Cartesian product of the respective sample spaces [BH14, p. 545]. Commonly, this is visualized as a matrix representation of all possible outcomes. However, all combinations are equally probable, because the two dice are *unconditionally independent* [Mur12, p. 30], that is, one die does not influence the outcome of the other. Formally, this is expressed as

$$p(x, y) = p(x) \cdot p(y). \tag{2.1}$$

If, however, the two random variables are not independent, this simplification is not valid. The concept of *conditional* probability, denoted by $p(x|y)$, expresses the probability

of a random variable x *after* another random variable y has been observed first. It is defined through the joint probability of x and y occurring together and the probability of y alone and computed as [Mur12, p. 29]

$$p(x|y) = \frac{p(x,y)}{p(y)}. \tag{2.2}$$

In the case of continuous random variables, the probability that a value x lies in an interval (a, b) is computed as [Bis06, p. 17]

$$p(x \in (a,b)) = \int_a^b p(x)dx \tag{2.3}$$

with $p(x)$ as the pdf. Since probabilities are always non-negative, the following two conditions have to be met such that $p(x)$ is a valid probability distribution [Bis06, p. 18].

$$p(x) \geqslant 0, \tag{2.4}$$

$$\int_{-\infty}^{\infty} p(x)dx = 1. \tag{2.5}$$

The first condition simply enforces that probabilities cannot be negative, and the second condition enforces that the chance of observing x somewhere on the real line is always true. Actually, the relation in Equation 2.3 is used to define the cumulative distribution function (cdf) of a random variable and is defined as [Mur12, p. 32]

$$P(z) = \int_{-\infty}^{z} p(x)dx. \tag{2.6}$$

Further, $p(x)$ is the derivative of $P(x)$. The cdf is used in answering questions of how probable it is to observe an outcome which exceeds a certain value. This is, for instance, important when using the cdf in conjunction with a *copula* [Nel06]. In this thesis, copulas are used to derive a highly flexible class of multivariate distributions which show an excellent performance at image segmentation. See Section 3.6 and Chapter 4 for further details.

Another important aspect of this thesis is Bayes' law [Bay63] and the implications of interpreting it as a sensible way to tackle image segmentation in a fully probabilistic way. In general, Bayes' law is a simple result of the foundations of probability theory, also known as Kolmogorov axioms (see [Fah+07, p. 182]), and the rules of probability derived from them (see [Fah+07, p. 185]). However, the law which was credited to the reverend Thomas Bayes has been discovered before Kolmogorov set out the foundations of probability theory. Bayes' law is expressed in the continuous case through several probability distributions as [Gel+13, p. 7]:

$$p(\theta|x) = \frac{p(\theta,x)}{p(x)} = \frac{p(x|\theta)p(\theta)}{p(x)} = \frac{p(x|\theta)p(\theta)}{\int_\theta p(x|\theta)p(\theta)d\theta'}, \tag{2.7}$$

with θ as the model parameters, $p(\theta|x)$ as the *posterior*, $p(x|\theta)$ as the *likelihood*, $p(\theta)$ as the *prior*, and $p(x)$ as the *evidence* [Bis06, p. 22]. The posterior distribution is the ultimate result of the analysis. It is a probability distribution over the parameters of the model conditional on the observed data. The likelihood describes the model which is fitted to the data.

Note that the posterior is, in contrast to the likelihood, conditioned on the data. Therefore, it expresses the uncertainty of the parameters conditioned on the observed data. The likelihood is defined conversely and quantifies the uncertainty of the data conditional on the parameters. However, this is not the quantity we are interested in. We are interested in quantifying the uncertainty with which we have estimated the parameters of our model and this can only be assessed by consulting the posterior distribution. This fundamental change of perspective is a key feature of Bayesian statistics.

In this thesis, several probability distributions are used to describe the observed image as well as possible. Each model consists of of parameters θ and the prior belief of the model parameters is expressed through the prior. An in-depth review of prior distributions and the implications of using them in a Bayesian framework are discussed in Section 3.1. Finally, the evidence is the probability of observing the data x. Since there is usually no sensible way to compute this quantity, it is commonly rewritten as an integral over the parameters θ of the joint probability distribution. As a result, closed form expression for the posterior can only be given analytically if certain conditions are met (see Section 3.1). Otherwise, the computation of the evidence is intractable and samples of the posterior have to be drawn. One powerful class of methods for drawing samples from arbitrary probability distributions is Markov chain Monte Carlo (MCMC) which is explained in Section 3.4.

2.2 THE DATA

Image segmentation is a special case of pattern recognition. The goal is to find patterns in image data by looking at local similarities and grouping pixels into regions. The data themselves consist of *instances* [Zhu+09, p. 2]. In a general setting, an instance is, for example, one specific animal. It can be described by its characteristics, like height and size, or, as shown later, by a photograph. Formally, one instance $\mathbf{x}$ of the data is a representation of one specific object. It is represented by a D-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in \mathbb{R}^D$. Each element of this vector is termed a feature of the instance [Zhu+09, p. 2]. Depending on the problem domain these features are measured, like height or weight, or they are derived from measurements, like the body mass index.

In this work, the measurements are images which have been captured by a photographer. Therefore, the measurements are matrices instead of vectors. One instance of the data is an image $\mathcal{I}$ which consists of a set of colour channels. Formally this is expressed as $\mathcal{I} = \left\{\mathbf{I}^1, \ldots, \mathbf{I}^{N_c}\right\}$. Each channel has the same size as the others. One colour channel $\mathbf{I}^c$ of the size $H \times W$ equals

$$\mathbf{I}^c = \begin{bmatrix} x^c_{1,1} & x^c_{1,2} & \cdots & x^c_{1,W} \\ x^c_{2,1} & x^c_{2,2} & \cdots & x^c_{2,W} \\ \vdots & \vdots & \ddots & \vdots \\ x^c_{H,1} & x^c_{H,2} & \cdots & x^c_{H,W} \end{bmatrix}$$

with H as the height and W as the width of an image. Commonly, an image consists of three colour channels R, G, and B which represent the intensity of the captured light in the red, green, and blue wavelengths.

Figure 2.1: Image of a bear cub (left), the corresponding gradient magnitude (middle), and gradient orientation (right). Image taken from the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11].

The CIE 1976 L*a*b* Colour space (Lab) [Rob77] is an often used alternative to the RGB colour space. Colours in Lab are computed through a nonlinear transformation of the RGB colour space. The key idea behind Lab is to better resemble the human perception of colour, that is, colour differences perceived by humans behave similar to distances in the Euclidean space. Colour spaces which exhibit this property are termed *perceptually uniform*. See, for instance, [Pas01] for an overview of common colour spaces and a performance evaluation of images in different colour spaces in a supervised classification setting. While in [Pas01] the Hue Saturation Value (HSV) colour space (see, e. g., [GW02, pp. 295-301]) performs best, Lab and RGB are almost exclusively used in image segmentation. In Chapter 4 different colour spaces are analysed regarding their suitability in image segmentation tasks and the question if perceptually uniform spaces are better at image segmentation is tried to be answered. In a recent study, [GY19] conducted an analysis where the influence of choosing different colour spaces on DNNs is studied.

Besides colour alone, a plethora of image features exist in computer vision. The most famous variants are probably the Scale-invariant Feature Transform (SIFT) descriptor [Low99] and the Histogram of Oriented Gradients (HoG) descriptor [DT05]. Both methods are based on gradients [GW02, p. 577]. They look at a local area around an interest point inside the image, termed *image patch*, and examine the orientations of the gradients in this area. Computing the gradients of an image can be interpreted as a simple version of *edge detection*. In Chapter 5 an edge detector plays an important role when increasing the quality of the computed segmentations. Formally, the image gradient is the derivative of a real multivariate function f in x and y dimension.

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

In practice, the real multivariate function f is the image $\mathcal{I}$. Since an image commonly consists of multiple colour channels, it is converted to greyscale first. Afterwards, the gradient is approximated numerically by finite differences. The orientation and the magnitude of the gradient at a given pixel are then calculated as [GW02, p. 577]

$$\text{orientation} = \tan^{-1}\left(\frac{g_y}{g_x}\right), \quad \text{magnitude} = \sqrt{g_x^2 + g_y^2}.$$

HoG and SIFT then proceed by quantizing the gradient information inside an image patch around a query point into a histogram representation. The query point at which

Figure 2.2: Visualization of the 48 filters used in [LM01] for recognizing different textures. The filter bank consists of first and second order Gaussian derivatives at six orientations and three scales, eight Laplacian of Gaussian filters, and four Gaussian filters.

this descriptor is computed can be estimated from potentially relevant key points, as in the original SIFT paper [Low99], or they may be extracted on a dense grid, for instance at every n-th pixel [BZM06]. This approach is also known as *Dense* SIFT.

A single image is thus described by a possibly large set of descriptors. This representation can then be used to infer the scene of an image or to detect the objects which are present in an image. In the original SIFT paper, this amounts to a 128-dimensional feature vector per key point. A possible extension of this is the Bag-of-Features (BoF) approach, where a visual *codebook* is generated by clustering a large set of descriptors from a large set of images. The key idea behind the BoF approach is to find a codebook which consists of discriminative descriptors. An image can then be described by a histogram of the occurrences of the codebook entries. While this approach originates from the Bag-of-Words model used in document analysis, it is successfully applied in image and scene classification tasks [FFP05; LSP06; OD11].

While the Bag-of-Words model can be considered as a precursor of the BoF approach, the idea to create a histogram representation through clustering was previously introduced in [Sch01] as an approach to image retrieval and in [LM01] as a way to discriminate between images with different textures. The latter name their approach *textons*. The main difference between BoF and the texton approach are the features which are clustered to form the codebook. Textons are computed by convolving the image with a filter bank and clustering the computed responses [LM01]. The used filter bank consists of a set of *Gabor* filters [FS89], which have a long tradition in texture recognition. These filters respond to different spatial frequencies like edges or stripe patterns. Commonly, filter banks not only consist of Gabor filters, but also of Gaussian filters. Additionally, their first and second derivatives are used. The primary goal of the Gaussian filters is having high responses at image structures which are not edges. Figure 2.2 depicts a typical filter bank used for recognizing images with different textures [LM01]. Due to their successful application in texture recognition, the use of textons has been extensively studied in image segmentation tasks as well [JF91; Mal+99; SJC08; Arb+11].

2.3 LEVELS OF SUPERVISION

Image segmentation can be formulated as a machine learning problem. In machine learning, different levels of supervision are distinguished in order to describe how much information the human expert provides such that algorithms learn to solve a problem. Conversely, different levels of supervision are necessary depending on the chosen algo-

Figure 2.3: Image of a bear cub (top left) and the corresponding ground truth annotations. The ground truth images are colour coded by the mean RGB value of the corresponding regions. While the similarities across different annotations are evident, they vary strongly in the level of abstraction, especially in the background. Images taken from the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11].

rithm. Estimating the parameters of a model is also known as *learning* [Bis06, p. 2]. In this thesis, four levels of supervision are distinguished: supervised learning, unsupervised learning, semi-supervised learning, and weakly-supervised learning.

The image segmentation problem can be learned in a fully supervised framework, where each instance of a set of images is accompanied by a labelled ground truth image. It is of the same size as the training image and each pixel is annotated with the desired prediction of the algorithm. They are the associated classes or objects visible in an image. Figure 2.3 provides a sample from the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11] (BSDS500) and depicts how different annotators perceive the regions of an image differently. This illustrates that there is sometimes not one single correct solution, but often a set of probable solutions. In this thesis, we tackle this issue by modelling the number of regions probabilistically (see Section 3.5.1 and Chapter 5).

2.3.1 *Unsupervised Learning*

This thesis deals mainly with approaches to *unsupervised* learning and how these can be used to solve the image segmentation problem. In general, unsupervised learning is concerned with describing the structure of the data and how different features or instances are grouped in the feature space *without* using any expert knowledge.

Formally, we are working with a set of instances $\mathcal{X} = \{x_1, \ldots, x_N\}$ and try to build models which resemble the unknown generating distribution as closely as possible. Therefore, the models are of the form $p(x_j|\theta)$, with θ as the set of model parameters. Each x_j is assumed to be drawn i. i. d. (independently and identically distributed) [Bis06, p. 26] from an unknown generating distribution. Assuming i. i. d. instances is of great importance because this enables us to independently evaluate the likelihood of all instances. This means that we can neglect possible correlations among the instances and

evaluate the likelihood of multiple instances as a product of the likelihoods of the single instances. Formally, the likelihood of N instances conditioned on a set of model parameters can be expressed as [Bis06, p. 26]:

$$p\left(\mathcal{X}|\theta\right) = \prod_{j=1}^{N} p\left(x_j|\theta\right). \tag{2.8}$$

The independence assumption is of central importance for many machine learning algorithms. For instance, predictions about previously unseen instances would not be possible without it [GBC16, p. 109]. However, the independence assumption is rather limiting in practice, because one can easily assume cases where instances are correlated.

Algorithms which are especially suited for unsupervised image segmentation are presented in Section 2.4. They include, among others, the well-known Gaussian Mixture Model (GMM). Further details about mixture models are presented in Section 3.2.4.

2.3.2 *Supervised Learning*

Unlike unsupervised learning, *supervised* learning uses additional information. Every instance x_j of the data is accompanied by a desired prediction $y_j \in \mathbb{R}$ which is also known as a *label* or *ground truth*. In the case of a continuous desired prediction, the problem is termed a *regression*. In the case of a discrete desired prediction $y_j \in \mathbb{N}^+$, the problem is termed a *classification*. In both cases, the set of instances $\mathcal{X}$ is accompanied by a set of desired predictions $\mathcal{Y} = \{y_1, \ldots, y_N\}$ which are assumed to be drawn from a joint probability distribution $p(\mathbf{x}, y)$.

The goal of parameter inference is to learn a model which predicts the correct label given an instance. Therefore, the model is not only conditioned on the parameters of the model, as in Equation 2.8, but also on the instances x_j. In a *generative* framework for classification, Bayes' theorem (see Section 2.1) is used to condition the joint probability distribution $p(x_j, y_j)$ on the C labels and estimate the respective density [Bar12, p. 300]:

$$p(y_j|x_j, \theta) = \frac{p(x_j, y_j|\theta)}{\sum_{c=1}^{C} p(x_j, c|\theta)} = \frac{p(x_j|y_j, \theta)p(y_j|\theta)}{\sum_{c=1}^{C} p(x_j|c, \theta)p(c|\theta)}. \tag{2.9}$$

This approach explicitly models the generation of an instance through $p(x_j|y_j, \theta)$. In Chapter 6 this relation will be used to compute a semantic segmentation. Again, i. i. d. instances are assumed such that the relation for all instances can be expressed as

$$p(\mathcal{Y}|\mathcal{X}, \theta) = \prod_{j=1}^{N} p(y_j|x_j, \theta). \tag{2.10}$$

Equation 2.10 expresses supervised learning from a probabilistic perspective. However, it is sufficient to learn a function f which maps the observed instances to the desired predictions. Formally, this relation is expressed as $f : \mathcal{X} \mapsto \mathcal{Y}$. Often, this mapping is done by dividing the feature space into cells. The boundaries of the cells are described by classifiers, like the Support Vector Machine [CV95] or a neural network (see Section 2.8.1). As a result, discriminative approaches model the boundaries between classes

and generative approaches model the distribution of the different classes in the feature space. While a generative model is more complex than a discriminative model, it allows the computation of uncertainty estimates.

Unlike unsupervised learning, supervised learning is more about predicting than explaining the data generation process. Further, supervised learning makes use of labelled instances explicitly. If those labels are available, supervised algorithms are supposed to yield better results in terms of prediction accuracy. However, the labels need to be acquired somehow in practice. Depending on the type of problem, this comes with exceptional human effort. For instance, in the case of image segmentation every single pixel has to be annotated by an expert. Depending on the problem domain even finding suitable experts is challenging.

2.3.3 *Semi-Supervised Learning*

Semi-supervised learning is the bridge between supervised and unsupervised learning. Instead of having a label for every instance, as in supervised learning, only a subset of all instances is annotated. Methods from the semi-supervised learning literature use the unlabelled data to improve supervised learning on the available subset of labelled instances.

Formally, the set of all instances $\mathcal{X}$ consists of a set of unlabelled instances $\mathcal{X}_u = \{x_1, \ldots, x_U\}$ and a set of labelled instances $\mathcal{X}_l = \{x_1, \ldots, x_L\}$. By assuming that knowledge about $p(\mathcal{X}_u)$ can be used to improve the model of the labelled instances $p(\mathcal{Y}|\mathcal{X}_l)$, both sets are used during training. For instance, in a generative framework (cf. Equation 2.9) this knowledge can be directly incorporated into the estimation of a GMM [Zhu+09, pp. 25-28]. Semi-supervised learning is especially well suited if numerous instances are available, but labelling is tedious or expensive.

In terms of image segmentation, semi-supervised learning can be done by partially annotating an image by an expert. For instance by painting brush strokes, called *scribbles* (see e. g., [Lin+16; HKH17]), or by annotating single points in an image [Bea+16]. Then, the algorithm does not only know how many regions the expert expects but is also equipped with a first set of labelled instances $\mathcal{X}_l$.

2.3.4 *Weakly-Supervised Learning*

In computer vision, weakly supervised methods are concerned with recovering the extent of an object by only knowing the presence or absence of an object in an image [HKH17]. Especially in image segmentation, acquiring annotated ground truth data is cumbersome [Bea+16; Cor+16], because every single pixel needs to be annotated by a human expert. One way of reducing the annotation effort is by asking experts to annotate only parts of an image, as in semi-supervised learning. Another approach is to increase the annotation's level of abstraction. Instead of annotating at pixel-level, the images are annotated at image-level. For instance, the image is then only annotated by keywords. The exact extent of an object inside an image, as in supervised learning, is not part of the annotation and thus remains unknown to the algorithm. This significantly increases the complexity of the learning problem, because the algorithm needs to derive the look and shape of objects on its own. Further, it needs to learn common features of, for ex-

ample, persons without knowing how persons look like or where to find them in an image. Commonly, DNNs are used for this kind of tasks [HKH17]. They learn these relations from a large collection of training images. An overview of current approaches in weakly-supervised image segmentation is presented, for instance, in [HKH17]. In Chapter 6 a generative approach towards weakly supervised object detection and semantic segmentation is presented.

2.4 MODEL-BASED APPROACHES

Assuming a model for the data is one approach to image segmentation. In the context of colour images, this means that each region of an image is associated with a specific colour or distribution of colours. However, this neglects the spatial arrangement of the pixels inside an image. Pixels in close vicinity should have a higher probability to belong to the same class. Therefore, the x- and y-position of the pixels can be included as additional features [CM02] or, as suggested in the related work, modelled as well [SNG08]. In the following, several model-based approaches are presented and reviewed in the context of their use of labelled data.

2.4.1 *Thresholding*

One of the simplest approaches to segment a greyscale image into a set of regions is thresholding [Sze10, p. 284]. A user defines the levels at which the regions are separated by observing how adjusting the thresholds change the segmentation until a satisfactory result is achieved. In terms of the previous section, this is a supervised approach. However, a rather simple model is used.

Since determining suitable thresholds for a collection of images is tedious, algorithms exist which return suitable thresholds based on some criteria. Among others, Otsu's method [Ots79] is one of the commonly used methods. It determines suitable thresholds by looking at the variance of the values inside the regions. If this *intra-class variance* is minimal the optimal thresholds are reached. This follows the assumption that each region is associated with a unique mode of the histogram. In principle, Otsu's method computes the parameters of a one-dimensional GMM (see Section 3.2.4) by exhaustively searching for the combination which minimizes the intra-class variance of the components.

However, Otsu's method was developed for thresholding greyscale images. Although thresholds for every colour channel can be computed, this neglects the common appearance of data points in the feature space, because each dimension is treated independently of the others. An approach to respect common appearances in the feature space, is partitioning the feature space by solving the K-means problem.

2.4.2 *K-Means Clustering*

Originating from the clustering literature, solving the K-means problem has a long tradition in many scientific fields [Jai10]. Lloyd's algorithm [Llo82], better known as *K-means clustering*, is the typical approach for solving the K-means problem. While K-means clustering is an unsupervised algorithm, it is limited to finding spherical groups in the

feature space. Note that K-means clustering is able to group non-spherical data to some extent but the underlying model assumes spherical groups.

Starting from an initial solution with K means, the data points are iteratively assigned to the region where the increase in variance is minimal, that is, the points are assigned to the regions where the reconstruction error (see Equation 2.11) is minimal. Commonly, the Euclidean distance is chosen [Mur12, p. 354]. Afterwards, the mean value of every region is updated. This procedure is repeated until convergence is reached. The grouping can only be performed flawlessly if the groups in the data are arranged in circular shapes, because the points are assigned to the nearest cluster centre. For different shapes, a different assignment strategy would be required. Since Lloyd's algorithm greedily searches for a partition of the input image, the final solution heavily depends on the initial mean values. Further, K, the number of clusters, has to be set by a user or further estimated, for instance, through cluster validation criteria (see Section 3.3). Setting K strongly influences the outcome of the clustering and is one of the major challenges in clustering with K-means [TK09, p. 744].

The main benefit of K-means compared to a simple thresholding approach is that spherical distributions at arbitrary positions in the features space can be handled. Therefore, pixels which share similar features across all three colour channels can be grouped. Further, K-means can be used in conjunction with an arbitrary number of features. Common choices include the incorporation of positional features, for example, x- and y-position of the pixels or texture features, for example, derived from filter responses [Mal+99]. Further, the computational simplicity is low and many data points and clusters can be processed in a reasonable amount of time [TK09, p. 743]. Formally, the reconstruction error [Bis06, p.430]

$$f(\mathbf{X}) = \sum_{j=1}^{N} \sum_{k=1}^{K} r_{j,k} \left|\left|\mathbf{X}_j - \boldsymbol{\mu}_k\right|\right|_2^2 \tag{2.11}$$

is minimized with N as the number of data points, K as the number of clusters, $\boldsymbol{\mu}_k$ as the cluster means, and $r_{j,k}$ as a binary cluster indicator. It is set to true if the j-th data point belongs to the k-th cluster.

The main drawbacks of K-means originate from the assumption of spherical clusters in the feature space which all have the same size and variance [TK09, p. 743]. Therefore, K-means can only cluster data flawlessly if the clusters are spherical. But even then the clustering heavily depends on the initial cluster centres [TK09, p. 743]. Further, a degree of association or a measure of uncertainty is not used in Lloyd's algorithm. The points are either in or out. Besides other approaches to quantify uncertainty, like the *fuzzy* variants of K-means (see [Bez81, pp. 65-79]), K-means can be considered as a mixture of K isotropic normal distributions. An isotropic normal distribution has a diagonal covariance matrix where the diagonal elements are equal in every dimension. Further, the variance is shared among all components of the mixture model. As a result, correlation among features cannot be modelled. The resulting GMM is then of the following form [Bar12, p. 415]:

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \omega_k \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma^2 \mathbf{1}). \tag{2.12}$$

Note that σ^2 and ω_k are not subject to optimization (cf. Equation 2.11), but simply a result of the cluster assignments $r_{j,k}$. Through interpreting the result of Lloyd's algorithm as a GMM, uncertainty estimates can be given and a probabilistic perspective can be derived.

In practice, K-means is commonly used in image segmentation. In [Mig08; KM17] images are clustered with K-means in various colour spaces. The resulting segmentations are then fused into one final segmentation. K-means is also used to compute a segmentation from the eigenvalues of a graph in normalized cuts (see Section 2.5.2) [SM00]. Further, it is used to find representative feature vectors in a regression framework for image segmentation [YWL12] or as an intermediate step inside a larger framework [Li+18]. Additionally, K-means is widely applied in other fields of computer vision. For instance when computing textons [Mal+99] or Bag-of-Features models [FFP05].

2.4.3 *Gaussian Mixture Models*

Gaussian Mixture Models (see also Section 3.2.4) solve nearly all problems of the aforementioned K-means approach at the cost of a more complex parameter estimation. In fact, K-means and Otsus' method both estimate a GMM, but with different restrictions and assumptions. In the general case, the pdf of a multivariate GMM equals

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \omega_k \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

The assumption of spherical clusters from K-means clustering which is formally expressed by the diagonal covariance matrix $\sigma^2\mathbf{I}$ (cf. Equation 2.12) is relaxed in the GMM setting. The covariance matrices $\boldsymbol{\Sigma}_k$ are now neither tied to a single value σ^2 nor restricted to the diagonal elements. Therefore, they can model correlations among different features and, as a result, the set of shapes the distribution can represent now includes ellipses. However, the parameter estimation is different from K-means. Commonly, the parameters of a GMM are estimated by Expectation Maximisation (EM) [DLR77]. In fact, EM is generalization of Llyod's algorithm [Bar12, p. 415].

In a first step, the cluster assignments are computed. However, the decision is not binary, as in Lloyd's algorithm, but probabilistic. Therefore, each data point contributes to the parameter estimation of every cluster. The probabilities of each data point belonging to every cluster are termed responsibilities and are computed by evaluating [Bis06, p. 432]

$$p(k|\mathbf{x}) = \frac{\omega_k \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \omega_{k'} \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}. \tag{2.13}$$

In the EM framework, this assignment is termed the expectation step or short E-step. Afterwards, the parameters are updated according to the computed responsibilities (see [Bar12, pp. 410-411]). This is termed the maximization step or short M-step. Again, the E- and M-step are alternated until convergence is reached. EM is often chosen for estimating the parameters of a GMM because it directly enforces certain constraints on the parameters. For instance, the covariance matrix needs to be symmetric and positive definite in order to be a valid covariance matrix. Simple gradient-based optimization or

MCMC can still be used, but require a re-parametrisation of the problem or an adequate error handling such that invalid covariance matrices are not estimated.

However, Gaussian Mixture Models share one of K-means' drawbacks. The number of clusters K needs to be set by a user or further estimated. Besides cluster validation criteria (see Section 3.3.2), more sophisticated statistical approaches can be followed, like the infinite mixture model (see Section 3.5). In Chapter 5 different approaches for estimating the number of mixture components in the image segmentation setting are reviewed.

In practice, GMMs are frequently used in image segmentation. For instance, for the computation of superpixels [BLC18] (see Section 2.7), in conjunction with a random walk on a graph to improve the spatial extent of the regions [Des14], in the context of spatially varying mixture weights [SGH98; SNG08; Sfi+10], as a feature to asses the similarity of regions in a graph cut framework [ZZW13], as a model for texture [Yan+08], or as a baseline to measure the influence of improvements [Sfi+10; Des14].

In the remaining parts of this work, the GMM is chosen as the baseline method as well. In Chapter 4 the influence of various probability distributions on their suitability at accurately recovering a ground truth segmentation is evaluated. In Chapter 6 GMMs are used as a generative classifier in weakly supervised object detection and semantic segmentation.

In summary, at the cost of more complex parameter estimation, the GMM generalizes K-means through relaxing the assumption of equal sized spherical cluster shapes to elliptical shapes with varying variances and correlations. Further, by explicitly estimating the mixture weights ω_k unevenly dense clusters can be handled better. Additionally, by solving image segmentation in a generative framework, uncertainty estimates regarding the affinity with which a pixel belongs to a region can easily be quantified and visualized.

2.4.4 *Mixture of t-Distributions*

Although Gaussian Mixture Models can model a variety of shapes, outliers can easily fool a GMM [Pri12, p. 115]. This property is inherited from the normal distribution (see Section 3.2.2). One way to deal with outliers is a preprocessing of the data where possibly occurring outliers are deleted. However, in a Bayesian framework, it is more natural to include such behaviour in the modelling process. Therefore, the need for an extensive preprocessing can be circumvented by using a *robust* probability distribution instead. Probability distributions are termed robust if the tails of the distribution are able to place probability mass on infrequently occurring instances—the outliers. Further details about the t-distribution and the properties of other distributions can be found in Section 3.2.

The multivariate t-distribution has been used in image segmentation tasks as well. For instance as a substitute for a GMM [SNG07], in medical image segmentation [NW12], in conjunction with an edge preserving smoothness prior [SNG08], or as a region descriptor [Yan+13].

The parameters of a mixture of t-distributions can be estimated with EM. However, there is no closed form solution for the degrees-of-freedom ν of every component. Therefore, a root-finding problem has to be solved in every iteration to update them. Detailed

steps can be found, among others, in [Pri12, pp. 117-120] or [PM00]. Again, gradient-based optimization or MCMC can be used as well, but they require special treatments of the covariance matrices and the mixture weights.

Through introducing an additional parameter to the model and at the cost of a more complex estimation the model is made robust against outliers. Similar to GMMs the individual components of the mixture can model elliptic shapes only. If more complex shapes are desired, different distributions need to be chosen. Conversely, the question arises if more complex models are advantageous at modelling image data in practice. Chapter 4 provides an in-depth analysis of various probability distributions for the task of image segmentation.

2.5 DISTANCE-BASED APPROACHES

While all previously mentioned approaches to image segmentation assume some type of model, methods exist which do not assume a specific type of distribution. Where GMMs assume that each region is associated with a specific behaviour of the features—characterized by a mean vector and a covariance matrix—distance-based methods instead build their clusters through local proximities in the features space—the nearest neighbours.

2.5.1 *Mean-shift*

Mean-shift [FH75; Che95; CM02] is an iterative procedure which aims at finding modes in an unknown probability distribution. In image segmentation, each mode is then assumed to correspond to a specific region in an image. The assignment of the pixels to the regions depends on the mode each pixel is attracted to. In contrast to model-based approaches, like K-means or GMMs, where the unknown probability distribution is assumed to belong to a specific parametric form, mean-shift is not making such an assumption. Instead, the gradient of the underlying pdf is used to traverse the surface of the pdf to the nearest mode. Since the pdf of the underlying distribution is unknown, it is approximated by the gradient of a local neighbourhood. The local neighbourhood at every data point is defined through a kernel function K.

The width and the shape of the kernel function define the size of the local neighbourhood. Common choices include the Epanechnikov kernel, which has a parabolic shape, and the Gaussian kernel [Sze10, p. 294]. Formally, the gradient of the underlying pdf is given by [Sze10, p. 293]

$$\nabla f(\mathbf{x}) = -\sum_{j=1}^{N} \left(\mathbf{X}_j - \mathbf{x}\right) k'\left(\frac{\|\mathbf{x} - \mathbf{X}_j\|_2^2}{h^2}\right), \tag{2.14}$$

with $\mathbf{x}$ as the current instance, $\mathbf{X}$ as the points of the local neighbourhood, h as the bandwidth of the kernel, and $k'(\cdot)$ as the first derivative of the kernel function. The kernel bandwidth h has an indirect influence on the number of modes because it controls the volume of the local neighbourhood which is used to compute the gradient. The larger the width of the kernel the smaller the number of modes, because the computed mean is more consistent. However, if the kernel size is too large nearby modes may be merged unintentionally [Che95].

Mean-shift is an iterative procedure which starts from a single point in the feature space and converges towards the nearest mode in the feature space. In order to find *all* local modes of the underlying probability distribution, the algorithm has to be started from every point in the feature space. While this is computationally demanding, it has the advantage that the number of modes only depends on the kernel size and does not have to be known a priori. However, determining the form of the kernel and the size of the kernel is rather heuristic, although some principled approaches exist [Sze10, p. 293].

Mean-shift is used in [Li+18] as a baseline algorithm on which subsequent groupings are performed and in [Liu+14] as a drop-in replacement for a hierarchical segmentation technique (see Section 2.6) for saliency detection (see e. g., [IKN98; Zha+15]). Saliency detection can be considered as a special case of image segmentation in which the goal is to find the most salient object in an image and recover its outlines.

2.5.2 *Normalized Cuts*

While all previously mentioned approaches directly operated on the provided feature spaces, graph-based methods like normalized cuts [SM00] operate on a graph-based representation of an image. The graph captures the similarity of pairs-of-pixels in an image. Therefore, high dimensional representations of the pixels can be used to compute a similarity without the need to generatively model the probability of the observed data. However, the benefits of a generative model—the interpretability and the uncertainty estimates—are lost as well.

Contrary to the previously mentioned approaches, normalized cuts is a discriminative method. Instead of modelling the generation of an image through a probabilistic distribution, the image is simply subject to a partition. The division into regions is mathematically modelled by a graph partition. Which results, as shown in [RR04], in finding a hyperplane which minimizes a weighted squared distance between the hyperplane and the data points. In [Zha+18] and [TK09, p. 779] normalized cuts is linked to manifold learning [Bar12, p. 311]. Further, normalized cuts itself is a hyper-parameter free method, because the resulting segmentation only depends on the computed affinities. However, computing the affinities is usually accompanied by a set of hyperparameters (cf. [Mal+99; SM00; Arb+11; Zha+18]), and the resulting segmentation heavily depends on the affinity matrix. Contrary to all previously mentioned methods, it is not based on a probabilistic interpretation of the observed data. It is an engineered approach where the probabilistic viewpoint is set aside in favour of a discriminative view, which is usually easier to model.

In practice, normalized cuts are widely used in the field of image segmentation. The major differences among various approaches originate from different ways of building the affinity matrix $\mathbf{W}$. In the original work [SM00] different features are used depending on the characteristics of the input image. In the case of greyscale images, the intensity value of each pixel is used, in the case of colour images, the HSV colour space is used, and in the case of textured images, the activations of a filter bank are used to compute the affinity matrix. In [Zha+18] the mean-values of a local neighbourhood in the Lab colour space and the diagonal elements of the covariance matrix of the local neighbourhood in the RGB space are used to compute the entries of $\mathbf{W}$. In [Mal+99], textons are used in conjunction with boundary information derived from the image brightness.

One of the major difficulties is to find a suitable combination of the used cues. For instance, if texture and contour features are used to build the affinity matrix, the features have to be weighted differently in different areas. Texture features are less important at boundary regions and contour features are less important inside homogeneous regions. In [BM98] texture and intensity cues are merged through concatenating histogram representations of greyscale intensities and texture features. Further, the histograms are approximately scaled to the same dynamic range before concatenation and a custom weighting function is proposed. In subsequent work the cue combination is extended with further cues and the weighting is learned from human annotations [MFM04; Arb+11] (see also Section 2.6).

From an image segmentation perspective, normalized cuts is reported to have issues with large uniform regions [Arb+11] and to suffer from a weak alignment at contrast boundaries [Tan+18b]. From a clustering perspective, normalized cuts suffer from a sensibility to outliers and a tendency of splitting elongated clusters [RR04]. In [Tan+18b] normalized cuts are combined with regularization techniques such as a Markov Random Fields (MRFs) [Sze10, p. 180-192] to solve some aforementioned issues. Recently, a *soft* version of normalized cuts has been used as a part of a DNN's loss function for image segmentation [XK17] and for weakly supervised semantic segmentation [Tan+18a].

2.6 CONTOUR DETECTION

Another approach to image segmentation is detecting contours which separate the regions of an image. From this, the regions can be recovered. One of the earliest approaches is active contours, also known as *snakes* [KWT88]. Starting from an initial solution set by a user, the algorithm minimizes an energy function to move a contour in an image. Common energies include the internal energy of the snake, that is how strong it is deformed, and an external energy which attracts the snake towards edges in an image. Many modifications of this basic approach exist and are, for instance, summarized in [Sze10, pp. 270-284].

The segmentation method presented in [Arb+11], which is based on contour detection, is among the best performing *supervised* methods and can be considered as a central step to advance image segmentation. Details of this approach are presented in the following paragraphs. The algorithm consists of three major blocks, that are, feature computation and cue combination, contour detection and region building, and hierarchical segmentation.

2.6.1 *Feature Computation and Cue Combination*

In [MFM04] a contour detection algorithm is presented which uses multiple local image features, like brightness, colour, and texture. The key idea is to combine multiple cues to accurately detect contours in natural images. Using, for example, only brightness to compute contours will produce a high number of false positives in textured regions. Hence, texture is a central feature to discriminate different regions in an image. Then again, texture alone is not able to distinguish regions with different colours or intensities. Therefore, all cues need to be combined. However, all features have different scales, dimensionality, or types, for example, ordinal and continuous. Combing these cues is

not straightforward. Some approaches have been discussed in the context of normalized cuts. However, in [MFM04] the cues are combined in a supervised way, that is, the used image features are linked through human annotations. The model learns to predict contours at the same places as humans based on the provided features.

As a first step, a circular neighbourhood around a pixel—a patch—is split in half along an angle. Afterwards, both halves are compared in terms of their similarity using different cues, that are, brightness, colour, and texture. Brightness is modelled by the L-channel of the Lab colour space, colour is modelled by channels a and b, and texture is modelled by textons (see Section 2.2). Further details about the specific choices can be found in [MFM04].

In order to measure the similarity of all cues, histogram representations are computed of every half disc in every cue. Afterwards, the histograms are compared by the χ^2 distance [PHB97]. Varying the radius is equal to changing the scale of the detector and varying the angle of the separation is equal to detecting edges along different orientations. Although various combinations are possible, [MFM04] use eight orientations and three scales. However, in [MFM04] no benefit of including multiple scales is observed. Successful integration of multiple scales was later presented in [Arb+11] and [Arb+14].

While the gPB approach presented in [Arb+11] shows excellent performance in detecting contours in natural images, the computed contours are not always closed. However, a meaningful segmentation can only be computed from *closed* contours. Therefore, an additional step is required.

2.6.2 *Enforcing Closed Contours*

A general method to compute a segmentation with closed contours hierarchically is presented in [Arb+11]. It builds on combing a variant of the watershed transform [NS96]—the Oriented Watershed Transform (OWT) [Arb+11]—with a hierarchical representation of the contours—the Ultrametric Contour Map (UCM)[1] [Arb06]. Details of the algorithms can be found in the corresponding papers. In short, OWT computes a boundary mask which consists of small line segments—the arcs—and is essentially an over-segmentation of the initial image. Each arc is associated with a specific edge strength which matches the orientation of the underlying edge signal. This reduces artefacts which would otherwise occur at arcs which are in the vicinity of strong edges but do not share the same orientation as the arc. Afterwards, UCM is used to build a hierarchy of the over-segmentation computed by the OWT. It is based on a graph representation which is then iteratively reduced by merging neighbouring regions based on a similarity measure until all regions are merged. The resulting hierarchy can then be thresholded at a specific level to generate a segmentation. Again, these thresholds need to be learned from human annotations. The combination of all methods, that are, gPB, OWT, and UCM, is termed gPB-owt-ucm. Alternatively, they are termed *-owt-ucm, where the asterisk is a place holder for the name of the contour detection method.

Improvements to the owt-ucm approach are presented in [KLL13] where the computation of the affinity matrix $\mathbf{W}$ is improved by considering *full* pairwise affinities. A

1 The term ultrametric stems from modifying the triangle inequality which takes an essential part in defining a valid metric between two sets. In fact, the triangle inequality is not relaxed but strengthened. Further details on ultrametricity can be found, for example, in [RTV86].

related approach is presented in [SWW17] which focuses on the merging part and uses additional features to build the affinity matrix. In [XK17] the UCM is combined with a DNN for contour detection and in [YFL15] an embedding is proposed to improve the contour generation. Recent methods to improve contour detection on the BSDS500 include, among others, structured forests [DZ15], combinatorial grouping [Arb+14], and various approaches based on DNNs [KWH14; She+15; Kok15].

However, all methods based on contour detection are fully supervised, because the algorithms are trained on ground truth boundary images. In contrast, the methods presented in this thesis refrain from using supervised information as good as possible. See Chapter 5 for further details.

2.7 SUPERPIXELS

Commonly, pixels form the building blocks of an image. However, close-by pixels are often very similar in terms of their appearance, like colour or brightness. Often, it is desirable to group sets of very similar pixels into superpixels [RM03] during a preprocessing step. For instance, in order to reduce the computational burden of an algorithm. Superpixels can be considered as a special case of image segmentation where the focus does not reside on segmenting *object*-like instances in an image, but to group pixels together which share the same vicinity and are similar in appearance. The main objective is to compute an over-segmentation where the true boundaries of the regions are kept and the true regions itself are split into multiple groups of pixels. The groups should have roughly the same size, resemble a grid-like structure, and should be fast to compute [Ach+12].

Simple linear iterative clustering [Ach+12] (SLIC) is a typical representative of the superpixel methods and is widely used in various applications. As shown in [SHL18] it is among the top performing methods in terms of boundary recall and produces a predefined number of clusters. As the name suggests, simple linear iterative clustering is an iterative procedure which is closely related to K-means (see Section 2.4.2). SLIC uses the coordinates of the Lab colour space (see Section 2.2) and the (x, y) position of the pixels inside an image as features. They are compared by using a custom distance which includes a hyperparameter. It is used to control the importance of colour versus positional features. As the positional features become more and more important, the superpixels become more and more uniform in size and shape. In contrast to K-means, SLIC does not use the whole image for distance computations, but only a local neighbourhood. The size of the local neighbourhood is defined through the number of superpixels SLIC is set to find. As in K-means, each pixel is then assigned to the nearest cluster centre and all cluster centres are updated in every iteration. The algorithm terminates if the change in the cluster centres falls below a threshold.

In the case of SLIC, the user is able to directly set a desired number of superpixels. In contrast, other approaches like the previously mentioned GMM superpixels [BLC18] only allow rough steering through hyperparamters. Further details about superpixels itself, a performance evaluation, and an overview of recent advances can be found in [SHL18]. How superpixels, in this case, SLIC can be used to improve the segmentation models presented in this thesis at unsupervised image segmentation is shown in Chapter 5.

2.8 SEMANTIC SEGMENTATION

Besides the contour-based approaches to image segmentation which are trained on human annotations, another big branch of image segmentation exists. In *semantic* segmentation, the regions in an image belong to a specific class. Therefore, algorithms not only need to predict the outlines of every region in an image but also predict the class every region belongs to. This is far more challenging because algorithms now need to learn how given objects like cars look like in various images. Therefore, similarities across *several* images can be exploited. This type of problems is seldom solved by generative or unsupervised methods, but by DNNs. However, weakly supervised approaches exist. A contribution to this field using the combination of a DNN and a generative image model is presented in Chapter 6.

2.8.1 *History of Deep Neural Networks*

As of now, DNNs are commonly used to solve semantic segmentation problems, because they show the best performance given enough training data. For a better understanding, a brief overview of some trends in DNN research is given. Starting with a short historical excursion, the development of modern architectures, including architectures especially designed for image segmentation, is sketched. The reader may refer to the rich deep learning literature, for example, [GBC16], for further information and in-depth discussion of architectures and applications of DNNs.

Commonly, an artificial neural network consists of an input layer, an output layer, and in the case of multilayer perceptrons or deep architectures of one or many more hidden layers. Every layer consists of a specific number of neurons and every neuron is connected via a weight to all neurons of the previous layer. The output of a single neuron is the weighted sum of all incoming connections, followed by applying a non-linear activation function (see e. g., [Bis06, Ch. 5]).

Starting in the late fifties, the *perceptron* [Ros58] can be considered as the first artificial neural network which was able to learn from labelled examples. However, only problems which are linearly separable could be solved [MP69]. Extending the perceptron to multiple layers solved this problem. Actually, it can be shown that a neural network with one hidden layer and non-polynomial activation functions can approximate any function [HSW89; Les+93].

In the early sixties, the first work regarding convolutional neural networks was made by observing cells in the visual cortex of cats [HW62]. These cells correspond to orientation selective receptive fields. The *neocognitron* [Fuk80] can be considered as one of the first models employing a hierarchical structure similar to current Convolutional Neural Networks, albeit trained differently than nowadays [LKF+10; Sch15]. Currently, Deep Neural Networks are trained through back-propagation [GBC16, p. 203]. Convolutional architectures [GBC16, Ch. 9] have been rediscovered in the late nineties [LeC+98] and celebrated their most recognized breakthrough in 2012 by the invention of AlexNet [KSH12]. Another important milestone was the use of Rectified Linear Units (ReLUs) [GB10], which are considered as today's standard activation function in deep learning [GBC16, p. 173]. ReLU solved the vanishing gradient problem in a DNN (see [Nie15, Ch. 5]). During back-propagation, the magnitudes of the gradients become smaller and

smaller and eventually vanish. Therefore, no efficient training of the network is possible. Prior to ReLU this issue was circumvented by using pre-trained networks. More about the historical developments, along with further references, can be found in [LKF+10; Sch15; GBC16].

2.8.2 *Deep Neural Networks in Semantic Segmentation*

Parallel to the development and the success of the aforementioned networks in classification tasks, networks have been developed which did not only predict one class per image but made a prediction for every pixel in an image to get a semantic segmentation as an output.

One of the major issues in bridging from pure classification, that is, predicting *one* label per image, to semantic segmentation, that is, predicting *one* label for *every* pixel, is the intrinsic reduction of resolution in a Convolutional Neural Network (CNN) through layer-wise pooling. Starting with patch-based approaches [MH10], architectures especially suited for pixel-wise predictions have been developed. Among the first was the *U-Net* [RFB15] developed for biomedical applications. The key idea was to build a U-shaped architecture which has a contracting path—an encoder—used for discrimination and an expanding path—a decoder—for accurate localization [RFB15]. Intermediate results of the same resolution are shared such that the network can learn accurate and correct segmentations.

A similar approach is followed in [KWH14] by using Fully Convolutional Networks (FCNs). By transferring the internal representations at different layers to the output layer, localizing information at the beginning of the network can be combined with the discriminative power at the end of the network. This is achieved by scaling the feature maps at intermediate layers to the resolution of the input image and combining them for predicting the segmentation. Further, common network architectures, like AlexNet [KSH12] or VGG [SZ14], which are pre-trained on a classification task can easily be adapted to the new semantic segmentation setting by this approach.

In [BHC15] the *SegNet* is proposed. The key idea is to use a network which consists of an encoding and a decoding part. In order to reduce the amount of learnable weights, the pooling indices of the encoder part are reverted in the decoder part. Therefore, the network can learn discriminative representations in the encoder part which are then upsampled in the decoder part. Further, the upsampling does not need to be learned but is simply a result of the operations in the encoder part.

However, DNNs have one drawback and that is their need for vast amounts of annotated training material. Acquiring images is not difficult but labelling every pixel in an image, like it is needed in image segmentation, is at best tedious but mostly expensive because human labour needs to be done [Cor+16]. In some fields, for instance, when creating annotations of everyday scenes this work can be crowdsourced [Rus+15]. However, in some fields, like medical image analysis or when analysing remotely sensed surfaces of planets, domain experts are needed to annotate data. Further details about semantic segmentation using deep architectures can, for instance, be found in [HKH17; LDY18].

2.8.3 *Class Activation Maps*

Class activation maps are one way to visualize the predictions of a CNN. However, the network has to be designed in a special way to achieve this. The aim of class activation maps is to learn one filter for every occurring class [Zho+16]. This is achieved by introducing a global pooling layer which replaces the fully connected layers at the end of common DNN architectures. This approach was first introduced in [LCY13] as a form of regularization in order to prevent overfitting in the fully connected layer and forcing the network to learn confidence maps. This concept was then applied to weakly supervised object detection in [Oqu+15] and [Zho+16]. In [Oqu+15] a global *max* pooling is used and in [Zho+16] and [LCY13] a global *average* pooling. Global average pooling is better at localizing the full extent of objects than global max pooling, because global max pooling often only learns to detect the most discriminative parts [Zho+16]. This claim is verified by experimental results on the ImageNet database in [Zho+16].

In both cases, the basic idea is to learn a filter that detects which parts of an image are discriminative for every class. Due to the specific design of the convolutional layers before the global pooling layer, a class activation map is computed which indicates at which parts in an image the respective classes are likely to be present. Details on the computation of class activation maps can be found in [Zho+16]. Note that, by modelling each class by its own filter, multiple objects can be detected at once. Therefore, the DNN is not trained on one hot encoded vectors, but on vectors in which the presence of multiple classes can be encoded.

However, class activation maps only indicate the presence or absence of image parts at areas where a DNN *thinks* class relevant information is present. Since there is no supervisory signal, like bounding boxes which guide the DNN, errors can easily occur. One can easily assume that the class *train* can be learned by recognizing rails only because the visual variability of rails is smaller. Additionally, trains seldom appear without rails. Further examples in a general setting of explaining the predictions of DNNs are presented, among others, in [Lap+19].

Therefore, class activation maps do not necessarily coincide with the true objects. Further, it can be observed that class activation maps often have high activations at specific parts of an object. For instance, in the case of a person, especially at the head and the hands. Some examples are presented in Chapter 6. While these two cues might be enough to discriminate between persons and other objects, it makes it difficult to reconstruct the full extent of persons in an image. However, the results in this challenging task are encouraging. The network is able to localize objects in images, although it has never been told how they look like or even where to find them in an image. Based on this reasoning, in Chapter 6 a method is presented which combines class activation mapping with a generative image model to improve the detection accuracy and further present a way to learn a weakly supervised segmentation model.

2.9 SCENE AND OBJECT DETECTION TASKS

In this section, a short overview of commonly used benchmark tasks to asses the quality of vision algorithms is given. In Chapter 6 two tasks of the PASCAL Visual Object

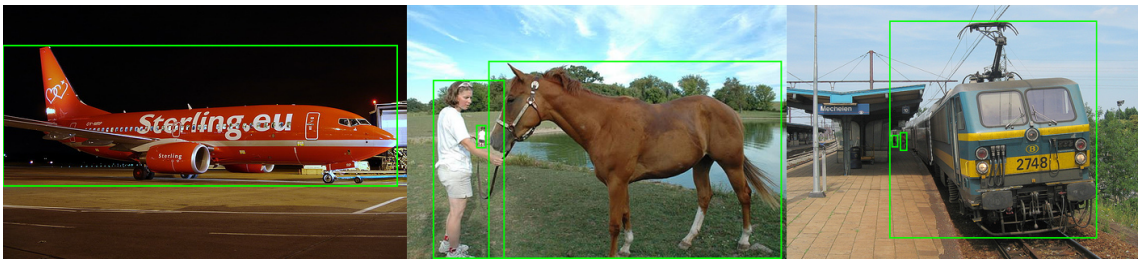


Figure 2.4: Three sample images of the detection task in the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC). The ground truth annotations are depicted by green bounding boxes. Multiple objects per image are possible.

Classes Challenge 2012 [Eve+15] (VOC) are used to measure the performance of the proposed weakly supervised object detection and semantic segmentation framework.

VOC is considered as one of the standard benchmarks [Eve+15; Rus+15] and is commonly used to asses the quality of computer vision algorithms in an *everyday* setting. The 19 737 images of the dataset are not designed for a specific case, like autonomous driving, but resemble everyday objects commonly occurring in natural images. Starting in 2005 with a classification and a detection task, VOC developed into a set of vision tasks, ranging from the already mentioned tasks to semantic segmentation and action recognition. Especially relevant for this thesis are the object detection and semantic segmentation task (see Chapter 6).

The object detection task contains twenty different object classes which are annotated by bounding boxes. Sample images along with ground truth bounding boxes are presented in Figure 2.4. Nowadays, *The ImageNet Large Scale Visual Recognition Challenge* [Rus+15] and the corresponding data set with 1 000 categories and 1 461 406 images is used to asses the quality of object classification. Further details about ImageNet and a comparison to VOC can be found in [Rus+15].

2.10 IMAGE SEGMENTATION TASKS

In this section, various benchmarks used in assessing the quality of segmentation algorithms are presented. Two cases can be distinguished. The first case is segmenting an image into a set of regions where no information about the semantic meaning of regions is used or available. A typical example is the BSDS500 which is presented in the following section. In this scenario, the images of the benchmark do not share semantic classes among images. Instead, each image is treated independently of the others.

The second case is semantic segmentation. Here, the semantic classes are shared among images and frequently reoccur. As a result, only the fixed semantic classes, like bikes, cars, or cows, are annotated in the data set. An example is presented in Figure 2.5. Semantic segmentation is nowadays mainly tackled by approaches related to deep learning, which can efficiently make use of rich annotations. Tasks which need to be solved without labelled regions are mostly tackled by non-deep learning methods since the number of annotations is usually not sufficient to train deep networks. In this thesis, the BSDS500 (see Section 2.10.1) is the standard benchmark used to asses the performance of the proposed segmentation methods. The approach towards weakly supervised semantic segmentation presented in Chapter 6 is evaluated on the semantic segmentation task

Figure 2.5: Three sample images (top row) of the segmentation task in the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC). The ground truth annotations (bottom row) are depicted by colour coded regions. Multiple objects per image are possible and the boundaries are surrounded by a five pixel margin which is not considered during evaluation.

of VOC. Further, additional data sets used in the related work are presented to provide an overview over current trends and briefly sketch possible future research directions.

2.10.1 *BSDS300 and BSDS500*

Prior to 2001, segmentation algorithms have usually been evaluated by *looking good* [Mar+01]. The Berkeley Segmentation Data Set and Benchmarks 300 [Mar+01] (BSDS300) was the first publicly available data set for image segmentation and consists of 300 images with at least four human segmentations per image. Further details on the construction of the database can be found in [Mar+01]. In [Arb+11] the BSDS300 has been extended with 200 additional images, resulting in a new test set. The images of the BSDS300 form the training and validation set of the extended version—the BSDS500.

Image segmentation suffers from ambiguity because different annotators might interpret regions differently. This might be resolved in an object detection framework—a car is a car—but it is challenging in the non semantic setting—a car consists of multiple parts. In the BSDS300 and BSDS500 this ambiguity is tackled by providing multiple human annotations per image. Figure 2.3 provides an example of a sample image along with ground truth segmentations by humans. In practice, the average over all ground truth segmentations is computed to account for variability.

The BSDS500 is almost always used to asses the quality of superpixels and non semantic image segmentation, see, for instance, [Arb+14; SWW17; SHL18; Tan+18b; Li+18; Zha+18]. Therefore, the BSDS500 is chosen as the standard benchmark in this thesis as well.

2.10.2 *VOC2012*

The semantic segmentation task of *The PASCAL Visual Object Classes Challenge 2012* [Eve+15] contains twenty different classes of everyday objects. Sample images along with ground truth segmentations are presented in Figure 2.5. The annotations are not perfect, but by design coarse. A margin of five pixels is introduced at every region border to account for an easier annotation. Therefore, accuracy was traded off for the speed of annotating samples. The boundaries are marked as *void* and are not considered during evaluation. The segmentation task of VOC is considered as the standard benchmark in (weakly supervised) semantic segmentation [HKH17].

2.10.3 *LeafSnap Field Data Set*

The LeafSnap field data set [Kum+12] is a semantic segmentation data set consisting of two classes, leaf and background. The aim is to segment each image as accurately as possible. Since the outline of a leaf contains valuable information about the leaf genera [Kum+12], special care should be taken to accurately reconstruct the border between leaf and background. However, especially at the border regions shading makes an accurate separation difficult. Further, leaves are subject to specularities because their reflectance behaviour is not uniform. It is therefore of great interest to study the influence of different generative models at their capability of modelling these influences. As there are only two classes, leaf and background, in the data set, there is no need to select the number of regions on a per image basis, but it can be fixed at two. Therefore, computing a segmentation is independent of the number of classes and the resulting segmentation mainly depends on the chosen generative model. Sample images along with ground truth annotations are depicted in Figure 2.6. Note that stems are excluded from the ground truth as they are not providing useful information for plant phenotyping.

2.10.4 *Others*

Semantic segmentation is an active area of research. Specialised data sets for autonomous driving [Gei+13; Cor+16; Hua+18; Yu+18], plant phenotyping [Kum+12; Min+16], medical image analysis [Maš+14; Set+17], or object segmentation in different scenes [LYT09; Sil+12; Zho+17; CUF18] exist. Other special cases of image segmentation are foreground-background segmentation [Alp+07] or salient object detection [Bor+15], that is, detecting the most salient object in image.

2.11 MEASURING THE PERFORMANCE OF IMAGE SEGMENTATION

The chapter concludes with a summary of current approaches to asses the quality of computed segmentations and discusses the advantages and disadvantages of the measures. Note that all the following interpretations are for a supervised evaluation, that is, a ground truth is available. However, there exist unsupervised evaluation metrics as well. See, for instance, [ZFG08] for a survey. In summary, unsupervised evaluation metrics focus on computing measures which summarize the quality of the segmentation by computing (weighted) distances between the segmentation and the input im-

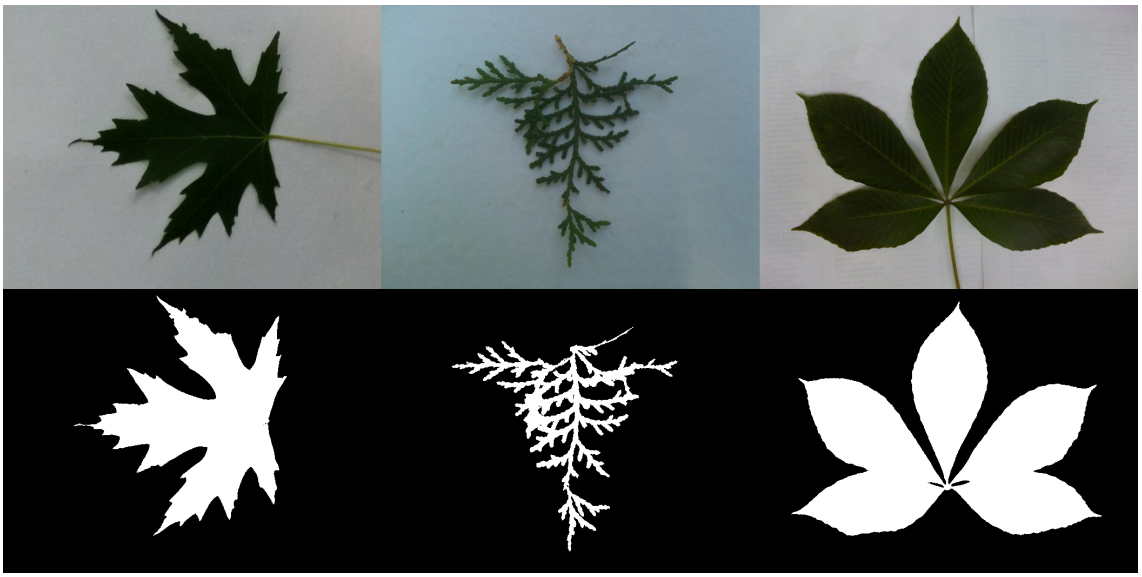


Figure 2.6: Three sample images (top row) of the LeafSnap Field data set [Kum+12]. The ground truth annotations (bottom row) are binary and the stems are intentionally excluded from the ground truth, because they do not contribute strongly to distinguishing different genera.

age. This includes, among others, the mean squared error between the data and the cluster centres or more sophisticated criteria which punish non-homogeneous regions or over-segmentations. While this can be considered as a more objective measure to segmentation accuracy [HP17], the resulting "optimal" segmentations seldom coincide with an object-based interpretation. Especially, if regions in an image show shading, texture, specularities, or a gradient. Therefore, a supervised evaluation of segmentations is performed in this thesis, because this is more in line with an object-based interpretation followed throughout this thesis.

Following [PTM16], two general types of supervised measures can be distinguished. There are *object-based measures*, which assume a ground truth with fore and background, that is, a two-class problem, and *partition-based* methods, which partition an image into an arbitrary number of regions, that is, a multiclass problem. For this thesis, the partition-based measures are most relevant because the algorithms developed in the remaining part of this thesis can handle an arbitrary number of regions. One exception is the study of the LeafSnap field data set (see Section 2.10.3 and Chapter 4) which only consists of two regions.

2.11.1 *Object-based Measures*

The object-based measures can be subdivided into two cases. In the first case, the focus resides on all pixels associated with both classes, and in the second case, the focus resides on the boundary between foreground and background. The boundary-based measures are discussed in Section 2.11.3.

In the first case, three sets are used to compute a measure in image segmentation. The true positives ($\mathcal{TP}$) are those pixels that are labelled in the ground truth *and* the segmentation as foreground. False positives ($\mathcal{FP}$) are those pixels which are detected

as foreground in the segmentation, but are labelled as background in the ground truth. False negatives ($\mathcal{FN}$) are those pixels which are labelled as foreground but are classified as background by the segmentation.

The F measure is the harmonic mean between precision and recall. It is commonly used and realizes a trade-off between precision, that is, the number of items in $\mathcal{TP}$ over the total number of foreground pixels in the segmentation and the recall, that is, the number of items in $\mathcal{TP}$ over the total number of foreground pixels in the ground truth. Trading off these measures is essential because both metrics alone can be fooled easily [PTM16]. Formally, the F measure is computed as [PTM16]:

$$F = \frac{2 \cdot |\mathcal{TP}|}{2 \cdot |\mathcal{TP}| + |\mathcal{FP}| + |\mathcal{FN}|} \tag{2.15}$$

with $|\cdot|$ as the cardinality of a set, that is, the number of elements inside it. The F measure is used as a metric in conjunction with the LeafSnap field data set (see Chapter 4). The F measure is also known as *Dice* coefficient, or *Spatial Overlap Index* [PTM16].

Another commonly used metric is the segmentation accuracy [PTM16] or *Intersection over union* (IoU):

$$IoU = \frac{|\mathcal{TP}|}{|\mathcal{TP}| + |\mathcal{FP}| + |\mathcal{FN}|}. \tag{2.16}$$

The IoU is used as a metric in conjunction with the segmentation task of VOC (see Chapter 6). It is also known as the Jaccard Similarity Coefficient, overlap score, spatial support score, ratio of intersection, or simply overlap [PTM16]. In fact, it can be shown that the F measure and IoU are closely related and if used to derive a ranking, equivalent [PTM16].

2.11.2 *Partition-based Measures*

The partition-based measures can be subdivided into three interpretations [PTM16], that are, *region-based*, *pairs-of-pixels-based*, and *boundary-based*.

2.11.2.1 *Region-based Measures*

The partition $\mathcal{S} = \{\mathcal{R}_1, ..., \mathcal{R}_K\}$ of an image $\mathcal{I}$ with N pixels into K regions $\mathcal{R}$ is compared to another partition $\mathcal{S}'$ into K' regions $\mathcal{R}'$. Based on this, several measures can be defined. Among others, the *Segmentation covering* [PTM16]:

$$C(\mathcal{S} \to \mathcal{S}') = \frac{1}{N} \sum_{\mathcal{R} \in \mathcal{S}} |\mathcal{R}| \cdot \max_{\mathcal{R}' \in \mathcal{S}'} \frac{|\mathcal{R} \cap \mathcal{R}'|}{|\mathcal{R} \cup \mathcal{R}'|}, \tag{2.17}$$

with $\mathcal{A} \cup \mathcal{B}$ as the union of two sets, that is, the set that includes all elements which are either in $\mathcal{A}$ or $\mathcal{B}$, and $\mathcal{A} \cap \mathcal{B}$ as the intersection of two sets, that is, the set that includes all elements which are members of both A and B Segmentation covering is one of the three metrics introduced along with the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11] and is therefore commonly used when assessing the segmentation performance on the data set.

While all previously mentioned measures originate from a set perspective, the Variation of Information (VoI) measure looks from an information theoretic perspective. The VoI is defined as [Mei05]

$$\mathrm{VoI}(\mathcal{S}, \mathcal{S}') = \mathrm{H}(\mathcal{S}) + \mathrm{H}(\mathcal{S}') - 2 \cdot \mathrm{I}(\mathcal{S}, \mathcal{S}'). \tag{2.18}$$

with entropy $\mathrm{H}(\cdot)$ and mutual information $\mathrm{I}(\cdot, \cdot)$. VoI is the second of the standard measures when evaluating the BSDS500. The smaller the value the better the agreement.

2.11.2.2 *Pairs-of-Pixels Measures*

Pairs-of-pixels measures are based, as the name suggests, on looking at all possible pairs of pixels in a partition $\mathcal{S}$ and comparing this to another partition $\mathcal{S}'$ [PTM16]. The set of all pairs of pixels $\mathcal{P}$ is divided into four sets based on the partitions $\mathcal{S}$ and $\mathcal{S}'$. The first set $\mathcal{P}_{11}$ are all those pairs which are in the same region in $\mathcal{S}$ and $\mathcal{S}'$, the second set P_{10} are all those pairs which are in the same region in $\mathcal{S}$ but different in $\mathcal{S}'$, the third set $\mathcal{P}_{01}$ are all those pairs which are in the same region in $\mathcal{S}'$ but different in $\mathcal{S}$, and the last set $\mathcal{P}_{00}$ are all those pairs which are in different regions in $\mathcal{S}$ and $\mathcal{S}'$ [PTM16]. These sets are then summarized in the *Rand Index* [Ran71]

$$\mathrm{RI}(\mathcal{S}, \mathcal{S}') = \frac{|\mathcal{P}_{00}| + |\mathcal{P}_{11}|}{|\mathcal{P}|} \tag{2.19}$$

as a measure of how coherent the two partitions are. The key idea behind the Rand index is to define a measure which is independent of the region labels since they can be arbitrarily exchanged and have no semantic meaning in non-semantic segmentation. Depending on the values of the sets, the RI is one, if all pairs agree, and zero, if no pairs agree.

In [UH05] the Probabilistic Rand Index (PRI) is introduced which can be used to compare a segmentation $\mathcal{S}$ to a set of G ground truth segmentations $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_G\}$. According to [Arb+11] it is computed as:

$$\mathrm{PRI}(\mathcal{S}, \mathcal{G}) = \frac{1}{|\mathcal{P}|} \sum_{i<j} \left[c_{ij} \cdot p_{ij} + (1 - c_{ij})(1 - p_{ij}) \right] \tag{2.20}$$

where c_{ij} indicates if pixels i and j are from the same region, and p_{ij} is the corresponding probability of observing this. If the p_{ij} are estimated empirically, that is, by computing how often pixels i and j share the same region in all ground truth segmentations, the PRI simplifies to [UPH07]:

$$\mathrm{PRI}(\mathcal{S}, \mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{i} \mathrm{RI}(\mathcal{S}, \mathcal{G}_i) \tag{2.21}$$

and is thus simply the average of the Rand indices over all ground truth segmentations. The PRI is said to have a low dynamic range [Arb+11]. Therefore, in [UPH07] a normalized version is introduced which normalizes the PRI by an expected value. However, this measure is seldom used to asses the quality of images in practice and the simpler variant—the PRI—is used instead. The PRI is the last of the three standard measures when evaluating the BSDS500.

2.11.3 *Boundary-based Measures*

All the aforementioned measures neglect spatial dependencies among pixels. However, they are quite common in segmentations, because neighbouring pixels have a high probability of being from the same region. Following [PTM16] the set of all neighbouring pixels is clustered to define a set of boundary pairs, that is, a boundary segment can be constructed by connecting them. Therefore, the problem is transformed into a two-class problem and this way a precision-recall framework can be constructed. However, ground truth and segmentation seldom coincide exactly. In order to be robust against small deviations, a threshold is introduced which is used to tolerate small differences. Further details can be found in [PTM16].

The boundary-based measure is today's standard evaluation scheme for contour detection algorithms on the BSDS500 and often used to asses the quality of segmentation algorithms as well. The major benefit of the boundary-based measures is its easy integration of multiple ground truth segmentations and the precision-recall framework itself, which is indicative of how the algorithm can be improved.

2.11.4 *Summary*

In [PTM16] the boundary-based precision-recall accuracy assessment in conjunction with the object-and-parts-based precision-recall framework (see [PTM16]) is suggested as the best way to compare different segmentation algorithms.

However, as reported in [Arb+11], segmentation frameworks which are supervised, for example, because they are based on contour detection, have an advantage over clustering-based approaches, like the mixture models used in this thesis, when evaluated in a boundary-based framework. Simply because contour detection methods not necessarily produce closed contours. However, in this thesis, the focus resides on generatively modelling the regions in an image. Therefore, the region-based criteria are weighted stronger than the boundary-based criteria. Further, one of the major goals of this thesis is to reliably quantify the uncertainty of the probability that a specific pixel belongs to a region, something not really possible in a boundary-based evaluation. Following this, the partition-based measures are used to make inferences.

In general, VoI and PRI are mostly used to asses segmentation quality when comparing different approaches, and are therefore the primary measures in the following chapters. Design choices can, however, be evaluated on all the aforementioned measures or any other segmentation metric. An extensive list of other metrics is presented, among others, in [PTM16].

3

FUNDAMENTALS OF BAYESIAN INFERENCE

The content of this chapter has been adapted and/or adopted from [WW16], [WW17c], [WW17b], [WW17a], [Wil+17].

The methods developed in the following chapters heavily rely on concepts and methods from the statistics literature. Therefore, common statistical concepts are introduced in this chapter. Starting with the key concepts of Bayesian statistics in Section 3.1, several probability distributions which are important in the remaining parts of the thesis are discussed in Section 3.2. This is followed by an explanation of model selection in Section 3.3, and Markov chain Monte Carlo (MCMC) in Section 3.4. Afterwards, non-parametric approaches to density estimation are presented in Section 3.5, and the concept of copulas is introduced in Section 3.6. The chapter concludes with a short treatise of significance testing in Section 3.7.

3.1 THE PRIOR - SUBJECTIVISM IN DATA ANALYSIS

In this thesis, one of the major goals is to derive uncertainty estimates of the computed models. This is easily possible in the Bayesian setting (see e. g., [Gel+13, Ch. 1]). Therefore, the focus does not reside on including prior information itself, but on the probabilistic interpretation of the parameter estimation.

As already mentioned, the prior distribution (see Section 2.1) is the central element which separates Bayesian statistics from classical statistics. In [Gel+13, p. 34], the prior is interpreted as a population of reasonable values. What is reasonable depends on the available knowledge prior to the experiment. Following [Gel+13, p. 34], two extreme types of prior distributions can be distinguished. They are either informative or uninformative. Naturally, everything between the two extremes is possible as well.

An informative prior is a prior distribution which conveys information about a parameter [Gel+13, p. 34]. For instance, the knowledge that the sky is commonly blue can be used in estimating the parameters of an image segmentation model. In this case, by using the prior for modelling a strong relationship between the colour blue and the regions which depict sky in an image. An uninformative prior would instead refrain from including such strong information into the estimation and would instead favour a vague relation.

Another benefit of prior distributions is their ability to impose constraints on the values of parameters [Gel+13, p. 84]. For instance, if a parameter is physically restricted to positive values, this is easily realized by using a probability distribution which is only defined on the positive real line. Depending on the available prior information, the hyperparameters, that are, the parameters of the prior distribution, are adjusted accordingly.

Conjugate prior [Gel+13, p. 35] play an important role in Bayesian statistics, because they enable the practitioner to get a closed form solution. In this case, the posterior is of a known parametric distribution. Therefore, the quantities like the maximum a-posteriori estimate and the confidence regions can directly be derived from the known parametric form. In fact, there are lists summarizing known conjugate prior likelihood relations (see, e. g., [Lun+12, pp. 46-47]). Historically, the use of conjugate distributions stems from the work presented in [RS61]. At that time, it enabled a broader use of Bayesian statistics, because the computationally demanding sampling can be circumvented. This is comparable to the rise of potent graphical processing units (GPUs) and the success of Deep Neural Networks (DNNs). Without the help of massively parallel computation on GPUs, DNNs would need way more time for training and testing. However, through the general rise of computational power, the demanding computations which have been limiting Bayesian statistics in the sixties, are possible today. Further information about prior distributions can be found in [Gel+13].

In this thesis, mainly non-informative prior distributions or weakly informative prior distributions are used. A notable exception is the inclusion of prior information about the probable location of edges. They are integrated into a probabilistic segmentation model to improve the performance. Further details are presented in Chapter 5.

3.2 PROBABILITY DISTRIBUTIONS

In the following chapters, the image segmentation problem will be tackled from a probabilistic perspective. Therefore, each region of an image will be associated with a probability distribution which describes how probable specific characteristics of the regions are. Starting with discrete distributions which only constitute a minor but important aspect in this thesis, a variety of continuous univariate and multivariate distributions are introduced.

3.2.1 *Discrete Probability Distributions*

Discrete probability distributions are used to model discrete events, that is, the outcomes are from a finite set of events, for instance, the throw of a die which falls on one of its six sides or a coin toss. In the following, four discrete distributions are shortly introduced. While the fourth distribution, the categorical distribution, is used in this thesis, the first three help to understand it.

3.2.1.1 *Binomial and Bernoulli Distribution*

A typical example of a Binomial distribution is modelling the outcome of a sequence of coin tosses. In this case, the set of discrete events are the two possible events that a coin either lands on heads or tails. If tossing the coin is repeated N_t times a random sequence of the same length is observed indicating if a coin landed heads or tails. The probability mass function (pmf) is then given by [Mur12, p. 34]:

$$\mathrm{Bin}\left(N_t | N_s, \tau\right) = \binom{N_t}{N_s} \tau^{N_s} \left(1-\tau\right)^{N_t - N_s} \tag{3.1}$$

with τ as the probability of observing heads, and N_s as the number of successes, that is, how often heads is observed. A special case emerges if the number of trials N_t equals one. In this case, the pmf simplifies to [Mur12, p. 34]:

$$\text{Ber}(x|\tau) = \begin{cases} \tau & \text{if } x = 1 \\ 1-\tau & \text{if } x = 0 \end{cases} \tag{3.2}$$

and the random variable is termed to follow a Bernoulli distribution. In the context of computer vision and machine learning the Bernoulli distribution is for instance used to build a probabilistic model for classifying handwritten digits [Bis06, pp. 444-448].

3.2.1.2 *Multinomial and Categorical Distribution*

In the case of a set with more than two discrete events, for instance, when rolling a D-sided die N_t times, the multinomial distribution is used for modelling. The pmf of the multinomial distribution is given by [Mur12, p. 35]:

$$\text{Multin}(\mathbf{x}|N_t, \boldsymbol{\tau}) = \binom{N_t}{x_1, \ldots, x_D} \prod_{i=1}^{D} \tau_i^{x_i}, \tag{3.3}$$

with $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_D)$ as the vector of probabilities of every possible outcome, $\mathbf{x} = (x_1, \ldots, x_D)$ as a vector of counts how often an outcome is observed, and

$$\binom{N_t}{x_1, \ldots, x_D} = \frac{N_t!}{\prod_{i=1}^{D} x_i!} \tag{3.4}$$

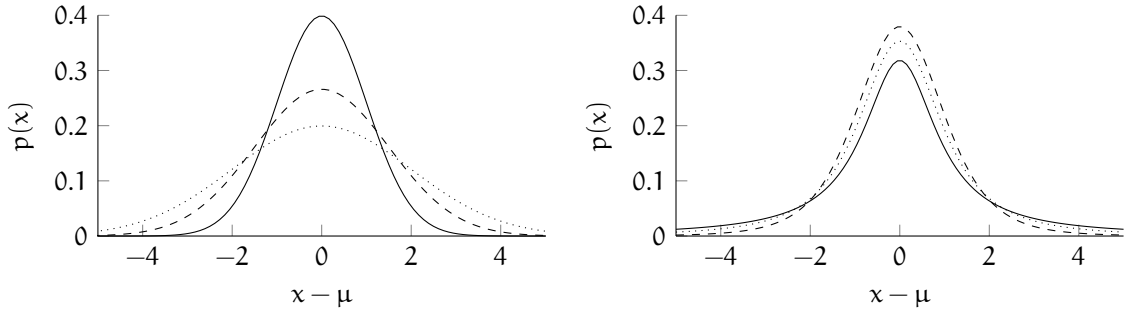
as the multinomial coefficient. Again, a special case emerges if the number of trials N_t equals one. In this case, the pmf simplifies to [Mur12, p. 34]:

$$\text{Cat}(x|\boldsymbol{\tau}) = \prod_{i=1}^{D} \tau_i^{x_i}, \tag{3.5}$$

and the resulting distribution is termed categorical distribution. The categorical distribution plays a small but important role in this thesis and is used to model the latent component memberships in Bayesian mixture models (see Section 3.2.4).

3.2.2 *Continuous Distributions*

Starting with distributions which can be used to model the regional distribution of grey values, special distributions commonly used to define prior believe in Bayesian inference are presented in the following. The distributions are distinguished by their flexibility which is itself assessed by how many moments [Was13, Ch. 3] of a random variable a distribution can model compared to the normal distribution. For instance, in the case of the normal distribution only the first and second moment—mean and variance—can be modelled. The third and fourth moment—skew and kurtosis—are a result of the variance and can therefore not be controlled actively [AS64, p. 930]. However, other distributions are able to model different behaviours with reference to the normal distribution by introducing additional parameters.



(a) Normal distribution

(b) t-distribution

Figure 3.1: Probability density function of a normal distribution (left) and Student's t-distribution (right). In both cases, various shapes are depicted. In the case of a normal distribution, by varying the standard deviation σ, and, in the case of Student's t-distribution, by varying the degrees-of-freedom ν. By comparing the distributions depicted in solid lines, which are both computed with $\sigma = 1$ the modified tail behaviour of Student's t-distribution is visible.

3.2.2.1 *Normal Distribution*

The normal distribution, also known as Gaussian distribution [Bar12, p. 166], is a ubiquitous choice in nearly all scientific domains when modelling the density of data. It is a symmetric distribution which is defined on the whole real line. It has a bell-shaped curve and a unique mode (see Figure 3.1a). The mode of a distribution is the point with the highest probability. The probability density function (pdf) of a normal distribution, see Equation 3.6, is described by two parameters, the mean μ and the standard deviation σ.

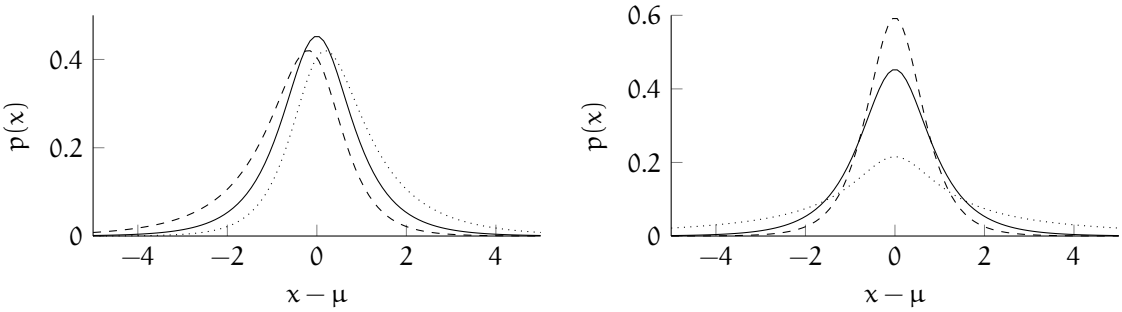
$$\text{Normal}(x|\mu, \sigma) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{3.6}$$

Additionally, the square of the standard deviation is known as the *variance* and the inverse of the variance is known as the *precision* [Bis06, p. 24]. In this thesis, the normal distribution is expressed through the standard deviation, because it has the advantage of having the same unit as the mean. This can be important when interpreting the results. The precision parametrization of the normal distribution is commonly used in Bayesian statistics (see, e. g., [Gel+13, p. 40]). The normal distribution is able to control the first and second moment of a random variable.

3.2.2.2 *Student's t-Distribution*

Real data are often corrupted by noise and outliers. While the former is often assumed to behave like a normal distribution, the latter seldom does. One reason is that outliers are not frequent enough to truly behave like a normal distribution. In order to model such deviations William Sealy Gosset developed the Student's t-distribution [Stu08] under his synonym *Student*[1]. In this thesis, the parametrization presented in [Mur12, p. 39]

1 William Sealy Gosset has been working for the Guinness brewery at that time. The company prohibited his employees to publish in scientific journals, because secret parts of the brewing procedure had been published unintentionally in the past [Sal01, p. 27].



(a) Modifying skew.

(b) Modifying kurtosis.

Figure 3.2: Probability density functions of a generalized hyperbolic distribution with modified skew (left) and modified tail behaviour (right) with respect to a normal distribution (solid line).

is used. In contrast to the normal distribution, Student's t-distribution has an additional parameter ν, termed the degrees-of-freedom. In the limiting case $\nu \to \infty$ the normal distribution is recovered. Therefore, Student's t-distribution (see Equation 3.7) is a generalization of the normal distribution. Again, μ and σ model the mean value and the standard deviation. Through ν the tail behaviour of the distribution can be adjusted. This is illustrated in Figure 3.1b. By lifting the tails of the distribution, infrequent values far from the mean receive more probability mass. Therefore, infrequent data points—the outliers—can be modelled better as they would otherwise shift the mean unintentionally.

$$t(x|\mu,\sigma,\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\nu\pi)^{1/2}\sigma}\left(1+\frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}} \tag{3.7}$$

$\Gamma(x)$ denotes the Gamma function. Student's t-distribution is able to control the first, second, and fourth moment of a random variable.

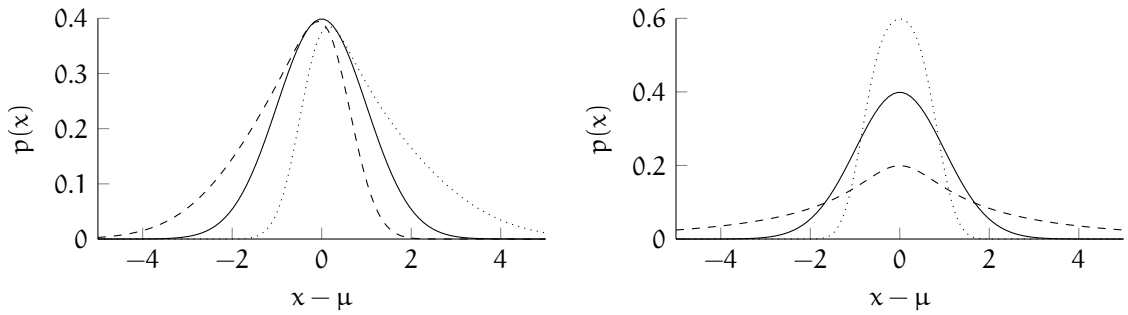
3.2.2.3 *Generalized Hyperbolic Distribution*

The two aforementioned distributions are both symmetric. Therefore, skewed data cannot be modelled adequately. One way to model skewness is to use the generalized hyperbolic distribution introduced in [BN77]. The pdf of a generalized hyperbolic distribution is presented in Equation 3.8.

$$\mathrm{Ghd}(x|\mu,\sigma,\chi,\gamma,\eta) = \frac{(\varpi/\sigma)^\eta}{(2\pi)^{1/2}K_\eta(\sigma\varpi)}\frac{K_{\eta-1/2}\left(\chi\sqrt{\sigma^2+(x-\mu)^2}\right)}{\left(\sqrt{\sigma^2+(x-\mu)^2}/\chi\right)^{1/2-\eta}}\exp\big(\gamma(x-\mu)\big) \tag{3.8}$$

$$\varpi = \sqrt{\chi^2-\gamma^2}$$

By introducing three additional parameters compared to the normal distribution, that are, χ, γ, and η, the flexibility of the distribution is greatly increased. The parameter γ is used to steer the skew of the distribution, χ and η are used to steer the tail behaviour



(a) Modifying skew.

(b) Modifying kurtosis.

Figure 3.3: Probability density functions of a sinh-asinh distribution with modified skew (left) and modified tail behaviour (right) with respect to a normal distribution (solid line).

of the distribution. The parameter σ is closely related to the variance of the distribution and $K_n(\cdot)$ is the modified Bessel function of the second kind which is defined as [AS64, p. 375]:

$$K_\nu(z) = \frac{\pi/2}{\sin(\nu\pi)}\left(I_{-\nu}(z) - I_\nu(z)\right), \tag{3.9}$$

with $I_\nu(z)$ as the modified Bessel function of the first kind [AS64, p. 375]:

$$I_\nu(z) = (z/2)^\nu \sum_{k=0}^{\infty} \frac{\left(z^2/4\right)^k}{k!\Gamma(\nu + k + 1)} \tag{3.10}$$

The generalized hyperbolic distribution is a favourable choice compared to other distributions with the ability to model the third moment of a random variable because it has added the ability to control the fourth moment as well. Further, the generalized hyperbolic distribution includes the normal distribution and Student's t-distribution as special cases.

Albeit its flexibility, the generalized hyperbolic distribution has one major drawback. It heavily relies on the use of Bessel functions. Since the Bessel function of the second kind is defined by a definite integral, a closed form expression of the derivative is not available. Therefore, only derivative-free optimization schemes or MCMC (see Section 3.4) are possible for parameter estimation. Further, advanced MCMC schemes are not possible, because they often rely on derivatives as well. The generalized hyperbolic distribution is able to independently control the first, second, third, and fourth moment of a random variable. Possible shapes of the distribution are depicted in Figure 3.2.

3.2.2.4 *Sinh-asinh Distribution*

As mentioned in the previous paragraph, the generalized hyperbolic distribution suffers from extensive use of Bessel functions, which are defined by definite integrals. An alternative distribution which is as flexible as the generalized hyperbolic distribution is

the sinh-asinh distribution introduced in [JP09]. The probability density function of the sinh-asinh distribution is presented in Equation 3.11.

$$\begin{aligned} \mathrm{Sha}(x|\mu,\sigma,\epsilon,\delta) &= \frac{1}{(2\pi)^{1/2}\sigma}\frac{\delta C_{\epsilon,\delta}\left(\frac{x-\mu}{\sigma}\right)}{\left(1+\left(\frac{x-\mu}{\sigma}\right)^2\right)^{1/2}}\exp\left(-\frac{1}{2}S^2_{\epsilon,\delta}\left(\frac{x-\mu}{\sigma}\right)\right) \\ C_{\epsilon,\delta}(x) &= \cosh\left(\delta\sinh^{-1}(x)-\epsilon\right) \\ S_{\epsilon,\delta}(x) &= \sinh\left(\delta\sinh^{-1}(x)-\epsilon\right) \end{aligned} \tag{3.11}$$

The pdf is defined by the mean μ, standard deviation σ, skewness parameter ϵ, and tail parameter δ. Instead of Bessel functions, the pdf is defined by hyperbolic functions and their inverses which have closed form derivatives.

Similar to the generalized hyperbolic distribution, the sinh-asinh distribution contains several distributions as special cases. Among others, the normal distribution is recovered if $\epsilon = 0$ and $\delta = 1$ [JP09]. Additionally, the sinh-asinh distribution not only models heavier tails than the normal distribution but also lighter tails. A range of possible shapes is depicted in Figure 3.3. Further, the sinh-asinh distribution pairs nicely with a Gaussian copula (see Section 3.6) when constructing a multivariate distribution. The sinh-asinh distributions is able to control the first, second, third, and fourth moment of a random variable. Further details and an application in image segmentation is presented in Chapter 4 and Chapter 5.

3.2.2.5 *Gamma Distribution*

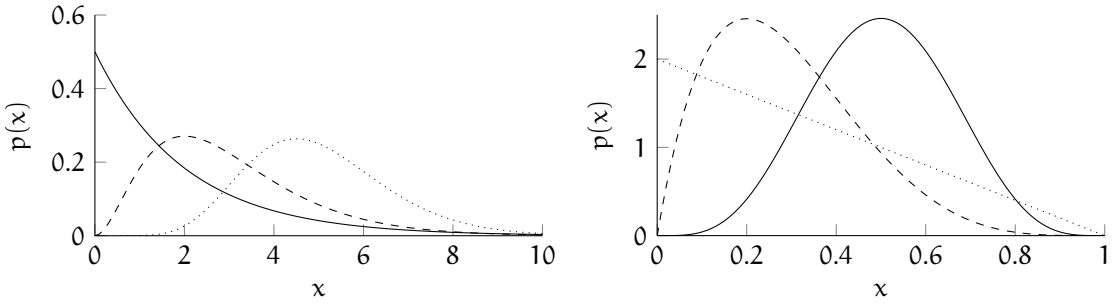
While all previously mentioned distributions have been presented as possible distributions for modelling the observed pixel values in an image region, special distributions exist which are commonly used to model the prior belief of parameters in a Bayesian framework. A typical representative is the gamma distribution [GH06, p. 44]. The pdf is presented in Equation 3.12.

$$\mathrm{Gamma}(x|a,b) = \frac{b^a}{\Gamma(a)}x^{-(a+1)}\exp(-bx) \tag{3.12}$$

Two hyperparameters a and b are used to control the shape of the distribution. Both hyperparameters have to be positive [Bar12, p. 164]. A range of possible shapes is depicted in Figure 3.4a. In contrast to the previously mentioned distributions, the gamma distribution is only defined on the positive real line. Therefore, it is suitable for modelling parameters which are supposed to be positive by definition and is commonly used as a prior distribution for the precision of a normal distribution. The gamma distribution is able to model the first and second moment of a random variable defined on the positive line.

3.2.2.6 *Beta Distribution*

While the gamma distribution is defined on a half-open interval, that is, the positive real line, the beta distribution is defined on the closed interval $[0, 1]$ [Bar12, p. 165]. Therefore, the beta distribution is especially well suited for parameters which are restricted to this



(a) Gamma distribution (b) Beta distribution

Figure 3.4: Probability density function of a gamma distribution (left) which is only defined on the positive real line and the pdf of a beta distribution which is only defined on the interval $[0, 1]$.

range. This includes, among others, proportions and probabilities. The pdf of the beta distribution is presented in Equation 3.13.

$$\text{Beta}(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \tag{3.13}$$

The beta distribution is defined by two hyperparameters α and β. Both have to be positive. A range of possible shapes is depicted in Figure 3.4b. The beta distribution is able to control the first, second, third, and fourth moment of a random variable defined on the interval $[0, 1]$.

3.2.2.7 *Uniform Distribution*

If a parameter is known to lie in an interval and all values in it are equally probable, the uniform distribution [Mur12, p. 32] is used as a model . For instance, to express ignorance, that is, assuming all values from an interval are equally likely observed. The pdf of a uniform distribution is presented in Equation 3.14.

$$\text{Uniform}(x|u, v) = \begin{cases} \frac{1}{v-u} & \text{for } u \leqslant x \leqslant v, \\ 0 & \text{for } x < u \text{ or } x > v \end{cases} \tag{3.14}$$

The distribution is characterized by two hyperparameters u and v which define the start and the end of an interval. The height of the resulting rectangle is computed such that it has an area of one and is thus a valid probability distribution. The uniform distribution controls the first and second moment of a random variable from a closed interval.

3.2.2.8 *Others*

Besides the already mentioned distributions, a plethora of other distributions exist. While the previously mentioned distributions are relevant in this thesis, that is, they are used in the experiments presented in the following chapters. Other distributions may be alternative choices. For instance, the normal distribution and Student's t-distribution

can be skewed by the relation introduced in [Azz85]. By expressing a probability distribution $p(x)$ through a symmetric density $f(x)$ and its cumulative distribution function $F(x)$ with additional parameter s, a valid distribution is expressed by:

$$p(x) = 2\, f(x)\, F(s \cdot x). \tag{3.15}$$

As a result, a skewed variant of the distribution $f(x)$ is recovered by modifying s. However, in the case of the normal distribution and Student's t-distribution the respective cumulative distribution functions do not have closed form expressions, but are defined through definite integrals or infinite sums similar to the Bessel functions (cf. Section 3.2.2.3).

Besides, generalized variants of the normal distribution exist which either allow to model the kurtosis of a random variable, see, for instance, [Nad05], or skew, see, for instance, [HW05, p. 197]. Another distribution which is based on the normal distribution is the log-normal distribution. Similar to the gamma distribution it is defined on the positive real line and has two hyperparameters, the mean and the variance. A variable is said to follow a log-normal distribution if it is normally distributed on a logarithmic scale.

Another class of distributions which exhibit great flexibility and include many distributions as special cases are *stable* distributions (see, e. g., [Nol03]). However, stable distributions are defined through characteristic functions and the Fourier transformation which makes them computationally expensive when evaluated.

Lastly, the skewed generalized t-distribution [The98] is another five parameter distribution which offers great flexibility. It is similar to the generalized hyperbolic distribution and the sinh-asinh distribution and includes many parametric distributions as special cases. However, only a univariate variant is presented and no cumulative distribution function (cdf) is presented. Therefore, it is not possible to integrate the distribution in the proposed copula framework presented in Chapter 4 where multivariate distributions are generated from flexible marginal distributions.

3.2.3 *Multivariate Distributions*

In the previous section, univariate distributions have been discussed. However, data of colour images is multivariate, because colour is commonly described by three colour channels. Therefore, the distributions in the previous section are only useful for modelling regions in grey scale images. If the regions of a colour image are to be modelled, multivariate distributions need to be considered. Of course, every colour channel can be treated independently of the others such that every colour is modelled by a univariate distribution. However, this neglects possible correlations among the colour channels. These correlations are very important in image segmentation. In Chapter 4 experiments regarding this aspect are presented.

In the following, the multivariate extensions of the normal distribution, Student's t-distribution, and generalized hyperbolic distribution are discussed. Additionally, a more flexible variant of the multivariate Student's t-distribution is reviewed and an important prior distribution in the context of mixture models, the Dirichlet distribution, is presented.

3.2.3.1 *Normal Distribution*

The multivariate formulation of a D-dimensional normal distribution is presented in Equation 3.16. The distribution is parametrized by a vector of means $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$.

$$\text{Normal}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{3.16}$$

The diagonal elements of the covariance matrix $\boldsymbol{\Sigma}$ are the variances in the respective dimension and the off-diagonal elements are used to describe correlations among different dimensions. The multivariate normal distribution is symmetric and has elliptic contours. Further properties of the multivariate normal can, for instance, be found in [Bis06, pp. 78-102].

3.2.3.2 *Student's t-Distribution*

Following the same reasoning as in the univariate case, the multivariate equivalent of Student's t-distribution is used if the observed multivariate data are corrupted by outliers. Again, robustness is achieved by modifying the tail behaviour of the distribution with the help of an additional parameter. By increasing the probability in the tails of the distribution it is able to assign higher probabilities to points which are far from the mode of the distribution. Further, by increasing the tail weights, the weight of the mode is flattened, which may help to better model near uniform distributions. The pdf of a multivariate t-distribution can be written as [Bis06, pp. 105]:

$$t(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma\left[\frac{\nu+D}{2}\right]}{(\nu\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{1/2}\Gamma\left[\frac{\nu}{2}\right]} \left(1 + \frac{(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}-\boldsymbol{\mu})}{\nu}\right)^{-\frac{\nu+D}{2}} \tag{3.17}$$

with $\boldsymbol{\mu}$ as the mean vector, $\boldsymbol{\Sigma}$ as a description of the covariance structure, and ν as the degrees-of-freedom used to steer the tail behaviour.

3.2.3.3 *Multiple Scaled t-Distribution*

One drawback of the previously described t-distribution is that the tail behaviour is modified equally in every dimension because ν is a scalar. However, one may easily assume a situation where this behaviour may not be desired because not every dimension might be corrupted equally. In those cases, a multiple scaled variant of the t-distribution [Tor+14] can be used. The pdf of the multiple scaled t-distribution is presented in Equation 3.18.

$$t_{ms}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{V}, \boldsymbol{\nu}) = \prod_{i=1}^{D} \frac{\Gamma\left(\frac{\nu_i+1}{2}\right)}{(\lambda_i\nu_i\pi)^{1/2}\Gamma\left(\frac{\nu_i}{2}\right)} \left[1 + \frac{\left(\mathbf{V}^{\mathsf{T}}\left[\mathbf{x}-\boldsymbol{\mu}\right]_i\right)^2}{\lambda_i\nu_i}\right]^{-\frac{\nu_i+1}{2}} \tag{3.18}$$

The distribution is parametrized by a mean vector $\boldsymbol{\mu}$, the matrix of eigenvectors $\mathbf{V}$, the vector of eigenvalues $\boldsymbol{\lambda}$, and $\boldsymbol{\nu}$ as the number of degrees of freedom in every dimension. In comparison to the t-distribution (cf. Equation 3.17), the multiple scaled variant uses an eigenvector and eigenvalue decomposition of the covariance matrix $\boldsymbol{\Sigma}$ to define the distribution. The t-distribution is a special case of the multiple scaled variant if all $\boldsymbol{\nu}_i$ are equal and the normal distribution is recovered if all $\boldsymbol{\nu}_i \to \infty$.

3.2.3.4 *Generalized Hyperbolic Distribution*

None of the previously described distributions is able to model skewed data in the multivariate case. The generalized hyperbolic distribution is, as in the univariate case, a parametric distribution which is able to model skewness in every dimension independently. Further, the distribution is able to model different tail behaviours, which makes it a very flexible distribution in the multivariate case. The general form (see [MFE15, p. 78]) is not uniquely defined in the multivariate case, because it is identical under certain parameter combinations. Therefore, several reparametrizations exist [MFE15, p. 79]. In this thesis, the parametrization by [BM15] is used. The pdf of the generalized hyperbolic distribution then equals:

$$\begin{aligned}\mathrm{Ghd}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\gamma},\chi,\eta) = {} & \frac{\left(\chi+\boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}\right)^{(D/2-\eta)/2}}{(2\pi)^{D/2}\left|\boldsymbol{\Sigma}\right|^{1/2}K_{\eta}\left(\chi\right)}\times \\ & \frac{K_{\eta-(D/2)}\left(\left(\left(\chi+\delta(\mathbf{x})\right)\left(\chi+\boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}\right)\right)^{1/2}\right)}{\left(\chi+\delta\left(\mathbf{x}\right)\right)^{(D/2-\eta)/2}}\times \\ & \exp\left(\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}\right)\end{aligned} \tag{3.19}$$

with $\boldsymbol{\mu}$ as the mean vector, $\boldsymbol{\Sigma}$ as the covariance matrix, tail parameters χ and η, skewness vector $\boldsymbol{\gamma}$, D as the dimension of the model space, and

$$\delta\left(\mathbf{x}\right) = \left(\mathbf{x}-\boldsymbol{\mu}\right)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)$$

as the squared Mahalanobis distance. Again, $K_{n}\left(\cdot\right)$ is the modified Bessel function of the second kind (see Equation 3.9). Further details about the distribution and different parametrizations can, for instance, be found in [BM15; MFE15].

3.2.3.5 *Dirichlet Distribution*

The Dirichlet distribution is a multivariate extension of the beta distribution [Mur12, p. 47]. While the beta distribution can be used to model a random variable coming from the interval $[0,1]$, the Dirichlet distribution is used if the random variable comes from a simplex, that is, the single realizations of the random variable each come from the interval $[0,1]$ and additional sum to one. The pdf of the Dirichlet distribution can be expressed as [Biso6, p. 76]:

$$\mathrm{Dirich}(\mathbf{x}|\boldsymbol{\zeta}) = \frac{\Gamma\left(\zeta\right)}{\prod_{i=1}^{D}\Gamma\left(\zeta_{i}\right)}\prod_{i=1}^{D}x_{i}^{\zeta_{i}-1},\quad \zeta=\sum_{i=1}^{D}\zeta_{i}. \tag{3.20}$$

with $\boldsymbol{\zeta}$ as the distribution's hyperparameters. The Dirichlet distribution is commonly used as a prior distribution in Bayesian statistics [Mur12, p.79]. It is especially well suited in cases where a set of random variables is modelled which all come from the interval $[0,1]$ and additionally sum to one. This includes, among others, the mixture weights of a mixture model or proportions which are positive and sum to one.

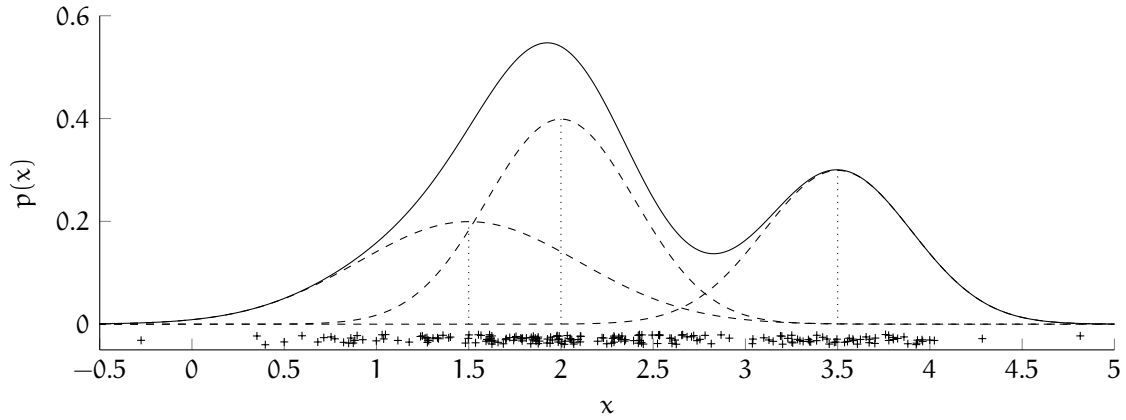


Figure 3.5: Probability density function (solid line) of a Gaussian mixture model with means $\boldsymbol{\mu} = [1.5, 2.0, 3.5]$, standard deviations $\boldsymbol{\sigma} = [0.6, 0.4, 0.4]$, and mixture weights $\boldsymbol{\omega} = [0.3, 0.4, 0.3]$. Although the mixture consists of three components, only two unique modes are visible. The first two components form a skewed shape. The pdfs of the individual components (dashed lines) and the component means (dotted lines) are additionally presented. Further, 500 random draws from the mixture model are illustrated by a "+"-sign. For a better visibility the samples are modified by uniform noise and a constant offset in the y-direction.

3.2.4 *Mixture Models*

Mixture models are a class of especially flexible probability distributions and are a central part of the approaches developed in this thesis. The pdf of an exemplary mixture model with three components is illustrated in Figure 3.5. In general, the probability distribution of a mixture model is built by a linear combination of two or more independent probability distributions. In order to enforce that the result is still a valid probability distribution, that is, it integrates to one and is strictly positive, the weights are taken from the interval $[0, 1]$ and have to sum to one (cf. Section 3.2.3.5), that is, a *convex* combination. Formally, this is expressed as [Gre17]:

$$p\left(\mathbf{x}|\boldsymbol{\Theta}\right) = \sum_{k=1}^{K} \omega_k \phi_k\left(\mathbf{x}|\boldsymbol{\Theta}_k\right), \quad \sum_{k=1}^{K} \omega_k = 1. \tag{3.21}$$

with $\boldsymbol{\omega}$ as the mixture weights, K as the number of mixture components, and $\phi_k(\mathbf{X}|\boldsymbol{\Theta})$ as any probability distribution. Note that, the distributions do not necessarily need to be of the same parametric form. A mixture of, for instance, two normal distributions and a gamma distribution is possible.

By interpreting the individual probability distributions in a mixture model differently, two application scenarios can be distinguished. In the first case, known as an *indirect* application of mixture models in [Tit+85] or as an *analytic* point of view in [Gre17], the K components are not assumed to convey any information about possible subgroups in the data. In this case, the mixture model is simply used to derive a probabilistic description of the data without interpreting the individual components. It is, therefore, more focused on the density estimation itself and not on the latent structure of the components.

In the second case, known as the *direct* application of mixture models in [Tit+85] or as the *synthetic* view in [Gre17], the opposite is true. Here, the components are interpreted as subgroups or subpopulations of the complete data. Therefore, it is assumed that each of the occurring K subgroups is appropriately described by *one* distribution. Formally, this is realized as a two-stage procedure. First, a group is selected with a probability which is proportional to the mixture weights $\boldsymbol{\omega}$. This latent group variable, commonly denoted by z, then indicates to which of the K components a sample "belongs". Afterwards, a realization of the distribution is drawn conditional on z, that is, based on the probability distribution z is associated with. Formally, the relation is expressed by [Gre17]:

$$x_j|z_j \sim p(\cdot|\theta_{z_j}), \quad \text{with} \quad p(z_j = k) = \omega_k. \tag{3.22}$$

Note that, this interpretation does not change the general equation of a mixture model (Equation 3.21), it is just a different interpretation. In fact, this interpretation is of utmost importance when deriving some methods to estimate the parameters of a mixture model. This includes, among others, the Expectation Maximisation (EM) [DLR77] algorithm which can be considered as today's standard approach for estimating the parameters of mixture models based on normal distributions. Further details about EM are provided in Section 2.4.3.

In a Bayesian framework, both interpretations are easily realized. In the second case—the direct application—the synthetic generation of samples is completely modelled by probability distributions. This explicitly includes the latent variables z_j which are not directly observed [MMR05]. The latent variables z_j are modelled through a categorical distribution, the mixture weights as a Dirichlet distribution, and the parameters of the components by appropriate prior distributions. For instance, a common choice for the mean and the variance of a normal distribution are the normal distribution and the Gamma distribution. During parameter inference through Markov chain Monte Carlo (MCMC) (see Section 3.4) the estimation of the latent variables and the parameters are alternated. First, the latent variables are drawn based on the current estimates of the parameters and afterwards the parameters of the components are updated with respect to the computed latent variables. Each component is updated one after each other using only those points where the latent variable equals the component index. Therefore, the estimation can be thought of as treating the different components separately and using only those points which probably belong to the respective components to update the parameters. Further details can, for instance, be found in [Gel+13, Ch. 22] or [MMR05]. A complete probabilistic description of the mixture models used in this thesis can be found in Chapter 5.

In the first case—the indirect application—the model is estimated as is in Equation 3.21 and the latent variables are not directly modelled. This is also known as a *marginalized* mixture model because the latent variables z which are used to indicate the component memberships are marginalized out of the estimation [Gel+13, p. 522]. In this case, this is possible by not separating the components from each other when modifying the parameters of the distribution. Therefore, the likelihood function [Gre17]

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \prod_{j=1}^{N} \sum_{k=1}^{K} \omega_k \phi_k \left(\mathbf{x}_j|\boldsymbol{\Theta}_k\right), \tag{3.23}$$

with N as the number of data points, is evaluated completely. The single components are not treated individually but as a whole. Therefore, in a fully Bayesian treatment, prior distributions and a model for the latent variables do not need to be assumed. As a result, however, no uncertainties regarding the latent variables can be computed. Commonly, marginalizing parameters out of a model leads to a better and faster sampling with MCMC, because the chain has to traverse a smaller parameter space.

In this thesis, the direct application is followed. For computing a segmentation, each region is assumed to correspond to *one* component of a mixture model. In order to model the various distributions occurring in natural images as good as possible, it is of utmost importance that the distribution of each component is as flexible as possible. An analysis regarding this aspect using different parametric probability distributions is presented in Chapter 4.

A typical representative of the class of mixture distributions is the Gaussian Mixture Model (GMM) which is a mixture of K normal distributions. The pdf of a GMM with K components is written as:

$$p(x|\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\omega}) = \sum_{k=1}^{K} \omega_k \text{Normal}\,(x|\mu_k, \sigma_k) \tag{3.24}$$

with $\boldsymbol{\mu}$ as the vector of means and $\boldsymbol{\sigma}$ as the vector of standard deviations. The pdf of a mixture model with three components is illustrated in Figure 3.5. Although the mixture consists of three components, only two modes can be distinguished. In fact, two of the components form a skewed shape. Following the direct interpretation, it would be beneficial to model these two components by one distribution which can also model skewed data. In the indirect interpretation, this would not matter because the density itself has the priority.

Further details about the GMM and how it is used in image segmentation has already been presented in Section 2.4.3. One of the major challenges in working with mixture models is how to decide on the number of mixture components K. Several approaches are presented in the following section. Another approach, where the number of mixture components is treated as a random variable as well is presented in Section 3.5.1. An evaluation of their suitability in image segmentation is presented in Chapter 5.

3.3 MODEL COMPARISON

Estimating the number of mixture components is a special case of model comparison [Gel+13, Ch. 7]. Therefore, model comparison is reviewed in light of selecting a suitable number of mixture components. Commonly, a mixture model is computed with a different number of mixture components and afterwards the *best* solution is selected. The major challenge is to determine what is best in a given context. In this thesis, two views are described. First, model comparison is reviewed from a statistical perspective (see Section 3.3.1) and, second, from a clustering perspective (see Section 3.3.2).

Additionally, model selection may be performed during parameter inference. Therefore, not only the space of parameters needs to be explored but also the space of models. In the case of mixture models, the number of components is then treated as an additional random variable and estimated during inference. This approach is also known as infinite mixture models [Mur12, p. 879] and has the advantage of being the statistically

most sound way to solve the number of component problem in a Bayesian framework. Additionally, an uncertainty estimate regarding the number of mixture components can be given. However, changing from one model to another model includes a change in dimension, because components are dropped or added. This makes the estimation more complicated such that additional criteria have to be met. Further details regarding this approach are presented in Section 3.5.1.

3.3.1 *Statistical Approaches*

The Akaike Information Criterion (AIC) is derived from the Kullback-Leibler divergence[2] between the density of the true model and an estimate of this density [MP00, p. 202]. Following [MP00, p. 203], the AIC [Aka74] is given after several simplifications as:

$$\mathrm{AIC} = -2 \log L(\theta) + 2M, \tag{3.25}$$

with $L(\theta)$ as the likelihood function (see Equation 3.23) and M as the number of estimated parameters. The model with the lowest AIC is chosen as the best fit. The AIC penalizes the likelihood of the data linearly with the number of estimated parameters. The formula stems from the asymptotic limit of the posterior distribution. In this limit, the posterior distribution is assumed to follow a normal distribution and subtracting the number of estimated parameters correctly adjusts for the possibility that an increase in prediction accuracy happened by chance. However, if this assumption is violated, for instance, if the models are non-linear or hierarchical the AIC struggles to give sensible estimates [Gel+13, p. 172]. In the case of mixture models, the AIC tends to overestimate the number of mixture components [MP00, p. 203].

A commonly used alternative to the AIC is the Bayesian Information Criterion (BIC). In contrast to the AIC, the BIC is derived from a concept termed *Bayes factors*[3]. A detailed derivation is presented in [MP00, pp. 208-209]. Following [MP00, p. 209], the BIC [Sch78] is given as:

$$\mathrm{BIC} = -2 \log L(\theta) + M \log N. \tag{3.26}$$

Again, the model with the lowest BIC is chosen as the best fit. The BIC penalizes the complexity of the model through the number of estimated parameters and, in contrast to the AIC, additionally through the number of data points which are used to compute the model. As a result, the BIC penalizes more complex models stronger if more data is considered.

AIC and BIC barely scratch the surface of a broad range of methods found in the statistical literature. However, these two are the most commonly used. Further details about model comparison can, among others, be found in [MP00; Lun+12; Gel+13]. According to [MP00, p. 220] the BIC performs better than the AIC when analysing synthetic data, that is, the model assumption is not violated.

2 The Kullback-Leibler divergence [KL51] measures the similarity of two distributions. In the continuous case, it is computed as [Bis06, p. 55]: $\mathrm{KL}(p||q) = -\int p(x) \cdot \log \frac{q(x)}{p(x)} \mathrm{d}x$.

3 Bayes factors are the ratio of two *marginal* posteriors of two different models, that is, the posterior distributions of two models are integrated over the parameters of the respective models and then divided by each other. Further details can be found, among others, in [Gel+13, Sec. 7.4] or [Lun+12, Sec. 8.7]

In Chapter 5 the experiments are repeated with real image data where the model assumption is violated. Empirically the best way to choose K in this setting is additionally presented.

3.3.2 *Cluster Validation Criteria*

While the statistical approaches utilize the likelihood function to derive criteria, cluster validation criteria directly look at the result of the estimation—the clustering. Therefore, they are not generally applicable in terms of model comparison, as the statistical approaches, but only useful when the data is divided into clusters. Since the direct application of mixture models (see Section 3.2.4) conveys a clustering interpretation, clustering criteria are a valid way to asses the quality of mixture models. However, this neglects the uncertainty in the affinity which is in the case of mixture models a membership probability.

3.3.2.1 *Silhouettes*

Silhouettes by [Rou87] is a clustering criterion where the cluster quality is graded by mimicking a visual inspection of the clustering. Silhouettes are computed for every datum. If a measure of the overall quality is desired, the average of the silhouettes is computed. This is termed the silhouette coefficient and denoted by SH in this thesis. A silhouette is defined as [Rou87]:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases} \tag{3.27}$$

with $a(i)$ as the average distance between the i-th point in a cluster $\mathcal{A}$ and all other points in $\mathcal{A}$, and $b(i)$ as the average distance between the i-th point to the points in the nearest cluster other than $\mathcal{A}$. In principle, every distance measure can be used to compute $a(i)$ and $b(i)$. In the original work, the Euclidean distance has been chosen. A clustering is supposed to be better if the silhouette coefficient is as close to one as possible.

3.3.2.2 *Calinski-Harabasz Criterion*

The Calinski-Harabasz criterion [CH74] is based on assuming that a good clustering has compact clusters which are well separated. This can be measured by computing the within-cluster variance W and the between-cluster variance B. The Calinski-Harabasz criterion is then computed as the ratio of those two quantities multiplied by a correction term which includes the number of clusters K and data points N:

$$\mathrm{CH} = \frac{W}{B}\frac{N-K}{K-1}. \tag{3.28}$$

The best clustering is achieved by the model with the highest Calinski-Harabasz criterion.

3.3.2.3 *Davies-Bouldin Index*

Another more sophisticated method to asses the quality of a clustering using variances is the Davies-Bouldin index [DB79]. The key idea is to have an index which measures the average dissimilarity of each cluster as:

$$\mathrm{DBI} = \frac{1}{N} \sum_{i=1}^{K} \max_{i \neq j} \frac{S_i + S_j}{M_{ij}} \tag{3.29}$$

with i and j as two cluster indices, S as the average distance within a cluster, and M_{ij} as the distance of the characteristic values of two clusters, for instance, the means. The former is a measure for the dispersion within a cluster and the latter is a measure for the dispersion of two clusters. If the dissimilarity is high, which is indicated by a low value of the DBI, a good clustering is achieved. Further details can be found in the original work.

3.3.2.4 *Others*

Besides the already mentioned criteria, a plethora of other criteria exist and can, among others, be found in [TK09, Ch. 16] or [AR13, Ch. 23]. In fact, the previously mentioned criteria are termed *internal* cluster validation measures, because they simply look at the distribution of the data in the feature space. They are solely based on internal information. If prior information about a clustering or even a ground truth clustering is available, other measures exist which are termed *external* cluster validation measures. They have been presented in Section 2.11 and are used to asses the quality of the derived image segmentations in the following chapters (see Chapter 4 and Chapter 5). External validation criteria are excluded here, because their sole purpose in this thesis is to asses the quality of a segmentation. Therefore, they are reviewed together with other approaches to grade the quality of a segmentation with respect to a ground truth annotation. The internal cluster validation criteria are used to estimate the number of mixture components in an unsupervised way, that is, without external knowledge.

3.4 MARKOV CHAIN MONTE CARLO

Until now, only *one* way—conjugate priors—has been presented as a method to perform an analysis in a Bayesian setting. However, the use of conjugate priors is limited to special cases where the used distributions for prior and model exhibit certain properties. In the general case, conjugate prior might not be known or not even be desired because the conjugate prior is too restrictive when modelling the prior belief. In those cases, Bayesian statistics is tackled with Markov chain Monte Carlo (MCMC). MCMC is a powerful class of methods which can draw samples from arbitrary probability distributions *without* needing a closed form expression of the distribution. Instead of having a closed form expression, the unknown distribution is approximated by drawing samples from it. The more samples are drawn and the better the samples are, the better the approximation of the unknown probability distribution. In the specific application scenario of drawing samples from the unknown posterior distribution, it is enough to know the distribution of the model and the prior which are both set by the practitioner.

3.4.1 *Monte Carlo Integration and Markov Chains*

MCMC is a combination of two separate techniques, that are, Monte Carlo integration and Markov chains. Both methods are briefly introduced. Monte Carlo integration is a numerical method to solve integrals. While other methods use a deterministic grid, Monte Carlo integration is non-deterministic, that is, it uses random samples to approximate an integral. The quantities of interest in the Bayesian setting are the moments of the probability distributions. The mean value of an unknown random variable can be computed as [Lun+12, p. 8]:

$$\mathrm{E}[x] = \int_x x\, p(x)\mathrm{d}x \approx \frac{1}{T}\sum_{t=1}^{T} x^{(t)}, \tag{3.30}$$

with $\mathrm{E}[x]$ as the expected value of the random variable x, $p(x)$ as the pdf, $x^{(t)}$ as the t-th sample of $p(x)$, and T as the number of samples used to approximate the expected value through Monte Carlo integration. In general, any function of x can be used. When estimating the expected value of a posterior distribution $\mathrm{E}[\theta|x]$ conditional on the observed data x Equation 3.30 becomes:

$$\mathrm{E}[\theta|x] = \int_\theta \theta\, p(\theta|x)\mathrm{d}\theta \approx \frac{1}{T}\sum_{t=1}^{T} \theta^{(t)}. \tag{3.31}$$

Still, the major challenge is how to compute the samples $\theta^{(1)}, \ldots, \theta^{(T)}$ of the posterior distribution. In those cases, Monte Carlo integration is combined with a Markov chain.

In general, a Markov chain creates a sequence of random variables $x^{(1)}, \ldots, x^{(T)}$ where the current random variable $x^{(t)}$ solely depends on the previous random variable $x^{(t-1)}$. Therefore, future and past are independent of each other [BH14, p. 460]. Note that, all derivations are presented for discrete time and discrete state space Markov chains. However, all concepts are valid for general state spaces but require a more technical description [GRS95, p. 46].

A Markov chain consists of several states, where the transition probabilities between two states i and j are described by a matrix $\mathbf{Q}$ of the individual transition probabilities [BH14, p. 461]:

$$Q_{ij} = p\left(x^{(t+1)} = j | x^{(t)} = i\right). \tag{3.32}$$

If the Markov chain meets specific properties, the long run behaviour of the Markov chain can be described by a *stationary* distribution and the Markov chain is guaranteed to reach the stationary distribution from any initial condition [BH14, p. 469]. In order to have a stationary distribution, a Markov chain has to be [GRS95, p. 46]:

1. irreducible,
2. aperiodic,
3. positive recurrent.

A chain is termed irreducible if every state can be reached from every other state in a finite number of steps with a positive probability. A chain is aperiodic if no oscillatory

behaviour is observed, that is, jumping from a specific set of states to another. And finally, a chain is termed positive recurrent, if once the chain reaches the stationary distribution all subsequent draws are from the stationary distribution.

The ultimate goal is to create a Markov chain where the stationary distribution approximates the posterior distribution as good as possible. As a result, samples from the unknown posterior $p(\theta|x)$ can be generated and the sought quantities of the posterior can be computed through Monte Carlo integration. How this is realized in detail is presented in Section 3.4.2 with the help of the *Metropolis-Hastings* algorithm. Further details about Markov chains in general and in the context of MCMC can be found in [BH14, Ch. 11] and [GRS95, Ch. 3].

3.4.2 *Metropolis-Hastings*

Metropolis-Hastings [Has70] is an algorithm which modifies a "normal" Markov chain into a Markov chain which has the unknown posterior distribution as the stationary distribution. This modification is implemented by introducing an accept/reject step within the algorithm. However, Metropolis-Hastings is not restricted to the application scenario of drawing random variates from a posterior distribution but it can also be used in an arbitrary setting to create samples from a distribution. In general, the algorithm is used to draw random variates from a distribution $p(x)$ by sampling from a proposal distribution $q(x \to x') = q(x'|x)$ which creates a new sample x' based on the current value x. Commonly, the normal distribution is chosen as a proposal distribution, because samples can easily be generated from it. The goal of the sampling is to generate a sequence $x^{(0)}, \ldots, x^{(T)}$ of T random draws from the distribution $p(x)$ starting from an initial value $x^{(0)}$. Metropolis-Hastings proceeds as follows [Ntz11, pp. 42-34]:

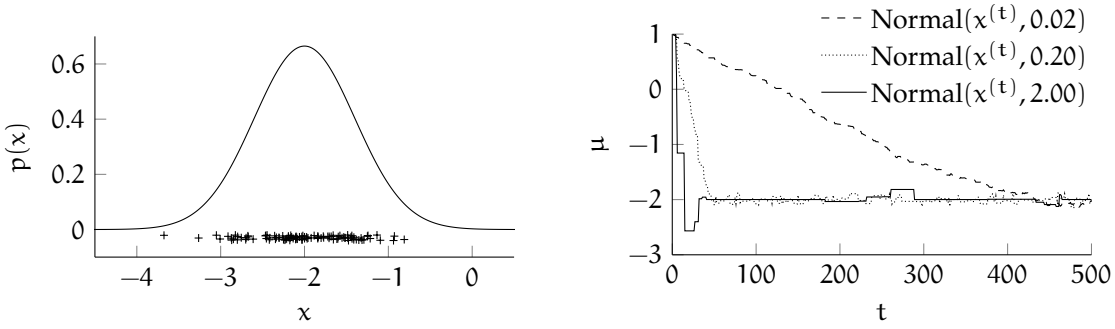
1. Set starting value $x^{(0)}$.
2. For $t = 1, \ldots, T$
   a) Propose a new value from $q(x'|x)$,
   b) Compute acceptance probability ξ,

$$\xi = \min\left(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right),$$

   c) Accept $x^{(t)} = x'$ with probability ξ. Otherwise, set $x^{(t)} = x = x^{(t-1)}$.

Regardless of the proposal distribution $q(\cdot)$, the algorithm presented above has $p(x)$ as its stationary distribution. In practice, the choice of the proposal distribution heavily influences the speed of MCMC. In terms of Markov chains, the chain generated by $q(\cdot)$ is a Markov chain which lacks the desired property of sampling from $p(x)$. Metropolis-Hastings now modifies this Markov chain through an accept/reject step such that the Markov chain generated with $q(\cdot)$ equals $p(x)$.

If the Metropolis-Hastings algorithm is used in the Bayesian setting, the unknown distribution from which samples should be generated is the posterior $p(\theta|x)$. Unfortunately, the parametric form is still unknown. In order to create a sequence $\theta^{(0)}, \ldots, \theta^{(T)}$ of T



(a) Probability density of Normal$(-2, 0.6)$ (solid line) and 100 samples (crosses).

(b) Trace plots of different proposal distributions.

Figure 3.6: Influence of the proposal distribution in Metropolis-Hastings. 100 Samples of Normal$(-2, 0.6)$ and the respective pdf (left) are used to estimate the mean value with different proposal distributions (right). Depending on the chosen proposal distribution, different behaviours can be observed. While all chains reach the stationary distribution eventually, the number of accepted samples varies strongly.

samples, Bayes' law is used to rewrite the computation of the acceptance probability ξ as:

$$\begin{aligned} \xi &= \min(1, \Xi) \\ \Xi &= \frac{p(\theta'|x)q(\theta|\theta')}{p(\theta|x)q(\theta'|\theta)} \overset{\text{Bayes}}{=} \frac{\frac{p(x|\theta')p(\theta')}{p(x)}q(\theta|\theta')}{\frac{p(x|\theta)p(\theta)}{p(x)}q(\theta'|\theta)} = \frac{p(x|\theta')p(\theta')q(\theta|\theta')}{p(x|\theta)p(\theta)q(\theta'|\theta)}. \end{aligned} \tag{3.33}$$

The intractable posterior $p(\theta|x)$ is expressed through Bayes' law as the product of likelihood and prior divided through the evidence. As a result, the evidence $p(x)$, which is difficult to compute, cancels from Equation 3.33 and the computation is solely based on the proposal, likelihood, and prior distributions, which are all set by the practitioner. Further, if $q(\cdot)$ is a symmetric distribution, like the normal distribution, it cancels from Equation 3.33 as well.

Choosing the proposal distribution has a major impact on the speed of the Markov chain. In the case of continuous parameters, the normal distribution is commonly chosen. Formally, one sets $q(x'|x) = \text{Normal}(x, \sigma_x)$ and the parameter σ_x can be adjusted accordingly. This is illustrated in Figure 3.6. On the one hand, if the standard deviation is too large, the chance of accepting a new value is small, because it is far from the current value. On the other hand, if the standard deviation is too small, this will result in a bad exploration behaviour, because the samples barely differ. Therefore, the proposal distribution need to be tuned to match the given problem. In the literature, this is often achieved by modifying σ_x until a "good" acceptance rate is reached. In [Ntz11, p. 44] the acceptance rate is targeted at twenty to forty per cent of the proposed samples as a good trade-off between exploration and sample generation.

In the multivariate case, Metropolis-Hastings is still possible. If the model has more than one parameter, the posterior is multivariate as well. Therefore, prior and proposal distribution become multivariate as well. However, the parameters can also be treated independently of each other and are thus updated one after another. In this case, prior and proposal distribution are univariate. However, the posterior is still a multivariate

distribution. Updating the parameters independently neglects the possibility to exploit correlations among parameters, but it makes it easier to tune the proposal distribution. Besides the already mentioned manual tuning of the acceptance rate, automatic approaches exist which tune the proposal distribution during inference.

3.4.3 *Delayed Rejection Adaptive Sampling*

Delayed Rejection Adaptive Sampling (DRAM) [Haa+06] is one method to tune the proposal distribution during parameter inference. Actually, DRAM is a combination of two methods, that are, delayed rejection [TM99] *and* adaptive Metropolis [HST01].

In contrast to the accept/reject step in Metropolis-Hastings, delayed rejection proposes a new value in the same iteration instead of rejecting the proposed value. Therefore, the new sample may come from a different proposal distribution or even be based on the rejected sample. The key idea is to use the information from the rejected sample to create a *better* proposal. In fact, this procedure can be repeated, for instance, in a fixed number of steps where the proposal distribution is iteratively scaled. Delayed rejection does not influence the properties of the Markov chain because past and future are still independent of each other. The dependence between the samples is only introduced while proposing the values.

Adaptive Metropolis does instead violate the assumptions of a Markov chain by updating the proposal distribution based on the already accepted samples. For instance, if the proposal distribution is a normal distribution, the covariance matrix is updated in every iteration based on the previously drawn samples. This helps the proposal distribution to automatically tune itself to a suitable scale. However, this uses past information of the Markov chain and therefore violates the assumption of an independent past and future. In practice, DRAM works [Haa+06] because a better exploration of the parameter space outweighs the violated assumption. It is especially beneficial in cases where the posterior distribution has highly correlated parameters. To circumvent the violation issue, adapting the covariance matrix can be limited to a fixed number of iterations. Afterwards, the tuned covariance matrix is used to generate the samples but not updated any longer. As a result, the assumptions of a Markov chain are no longer violated.

In this thesis, DRAM is used for sampling from the unknown posterior distributions. Although the assumptions of a Markov chain are violated, the adoption is not stopped during sampling. As already observed in [Haa+06], no negative implications have been noticed in practice.

3.4.4 *Others*

The literature on sampling algorithms is as rich as the literature on univariate probability distributions. Other commonly used methods for sampling in the context of MCMC are, among others, Gibbs sampling [GG84], slice sampling [Nea03], or Hamiltonian Monte Carlo [Dua+87]. Nowadays, a variant of Hamiltonian Monte Carlo—the No-U-turn sampler [HG14]—is the standard technique. For computationally demanding approaches, the posterior distribution is approximated. For instance, automatic differentiation variational inference [Kuc+17] is a commonly used variant to turn the iterative sampling into an optimization by fitting known distributions to the posterior. However, these samplers

are not considered in this thesis, because the models developed in the following chapters are too complex or too uncommon to be easily integrated into the current software frameworks, like Stan [Car+17], since essential functions are missing. Being an approach that can always be applied, the Metropolis-Hastings algorithm and the DRAM extension are used in this thesis as a trade-off between speed and ease of use.

3.5 NON-PARAMETRIC APPROACHES

Besides the already mentioned parametric approaches at modelling the density of data, non-parametric approaches exist as well. For instance, the concept of mixture models (see Section 3.2.4) can be extended to get a non-parametric mixture model. By letting the number of mixture components K reach infinity, an infinite mixture model is recovered which can set the number of mixture components autonomously [Mur12, p. 879]. Further details are presented in Section 3.5.1. While the infinite mixture model increases the possible number of components to infinity, Kernel Density Estimation (KDE) places a fixed number of kernel functions in the range of the data. By combining these kernels, a non-parametric estimate of the density is recovered. See Section 3.5.2 for further details. Both approaches are important in this thesis. While the infinite mixture model is used to estimate the unknown number of mixture components (see Section 5.2), the KDE is utilized as a flexible distribution to model multi-modal data in semantic segmentation (see Section 6.3).

3.5.1 *Infinite Mixture Models*

Dirichlet processes [Fer73] are used when the parameters of the model are allowed to change its dimensionality. In the case of mixture models, the Dirichlet process is used to model the number of mixture components. Hence, it does not need to be known a-priori, but is estimated in conjunction with the parameters of the mixture model. In the following, key concepts of an infinite mixture model are briefly introduced.

In general, a Dirichlet process can be viewed as a distribution over distributions. A draw from a Dirichlet process is a distribution itself. Formally, a realization G of a Dirichlet process is a discrete distribution. From such a realization G weights are drawn. Since the Dirichlet process is unbounded, the number of drawn weights $\omega_1, \ldots, \omega_\infty$ is infinite as well. Since these weights are drawn from a Dirichlet distribution (cf. Section 3.2.3.5), they additionally sum to one. Depending on the chosen parameters, more or less probability mass is concentrated on a few weights. Formally, a Dirichlet process is expressed as [Lun+12, p. 292]:

$$G \sim \mathrm{DP}(G_0, \iota) \tag{3.34}$$

with G_0 as the base distribution, and ι as the dispersion parameter. The base distribution G_0 represents the mean distribution and ι steers how close realizations from G are to G_0. Often, the standard normal distribution $\mathrm{Normal}(0, 1)$ is chosen as base distribution. The greater the dispersion parameter is chosen, the greater is the resemblance of the base distribution and a draw from G. The smaller the dispersion parameter is chosen, the greater is the dissimilarity and the sampled G places more weight on a few values ω_i.

The Dirichlet process can be used to model an infinite mixture model by using the p_i from the Dirichlet process as the mixture weights of the mixture model. As a result, the Dirichlet process models the uncertainty regarding the number of mixture components.

In practice, realizing a Dirichlet process as a prior distribution of the mixture weights comes at a high computational cost. However, the Dirichlet process has the advantage of exploring the model and the parameter space at once. Under certain circumstances, this can even be faster than testing a different number of mixture components [Mur12, p. 881].

Commonly, a Dirichlet Process Mixture Model (DPMM) is realized as a mixture of normal distributions. According to [Mur12, p. 886], the simplest way to estimate the parameters of a DPMM is a collapsed Gibbs sampler. In some cases it is possible, through a clever choice of distributions, to marginalize some parameters out of the model, that is, integrating over them (cf. Section 3.2.4). This is termed a collapsed Gibbs sampler. Actually, in the case of a GMM the parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and ω_k can all be integrated from the posterior. As a result, only the latent variables have to be sampled. A detailed derivation is presented in [Mur12, pp. 842-843]. Therefore, the collapsed variant of the Gibbs sampler can easily be adapted to integrate a Dirichlet process as a prior of the mixture weights. Details can be found in [Mur12, pp. 886-887].

In this thesis, the DPMM framework is tested as one variant to estimate the number of mixture components in an unsupervised way, that is, without external knowledge by an expert. In theory, this has the following benefits. First, the DPMM allows modelling the uncertainty in the estimates of the number of mixture components. Second, the DPMM uses a coherent statistical framework with verified properties. Third, the DPMM does not rely on unjustified assumptions like common cluster validation criteria or any other heuristic to estimate the number of mixture components, and last, it solves the problem in a data-driven way. In practice, the results are presented in Chapter 5 and strongly favour the DPMM over other variants.

3.5.2 *Kernel Density Estimation*

While the DPMM is in accordance with the direct interpretation of mixture models (see Section 3.2.4), kernel density estimation is more in line with an indirect interpretation, that is, focussing more on the density estimation itself than on the latent interpretation. Therefore, it is not suitable for image segmentation unless the different regions of an image are known to some degree. Then, a non-parametric estimation of the density can be built. This is favourable because the range of possible shapes a KDE can model is supposed to be more variable than parametric distributions. In general, estimating the pdf of a distribution with KDE is written as [Mur12, p. 508]:

$$\hat{p}_{kde}(x) = \frac{1}{N h} \sum_{j=1}^{N} k\left(\frac{x - x_j}{h}\right) \tag{3.35}$$

with $k(\cdot)$ as a kernel function, and h as the bandwidth of the kernel. If the kernel is chosen to be a zero mean normal distribution, KDE is comparable to a mixture of normal distributions [Mur12, p. 508]. In the univariate case this is expressed as:

$$p(x) = \frac{1}{N} \sum_{j=1}^{N} \text{Normal}(x_j, \sigma) \tag{3.36}$$

where a normal distribution is placed on every datum x_i with a fixed standard deviation σ. Instead of placing a component on each datum, a fixed grid with equal distances is often used as an alternative. In both cases, the number of components does not depend on the density of the data but is simply a result of the desired resolution or the number of data points. Since all points are weighted equally, the mixture weights are equal as well and result in $1/N$.

In general, other kernels are possible as long as they are always positive, have a zero mean, and integrate to one. The bandwidth h is, among others, set empirically, by cross-validation [Mur12, p.508], or in the case of normally distributed samples optimally by using Scott's normal reference rule [Sco79; Sco15].

Further, it is possible to introduce a weighting of the samples by assuming that not every sample contributes equally to the unknown probability distribution. The KDE is then changed to

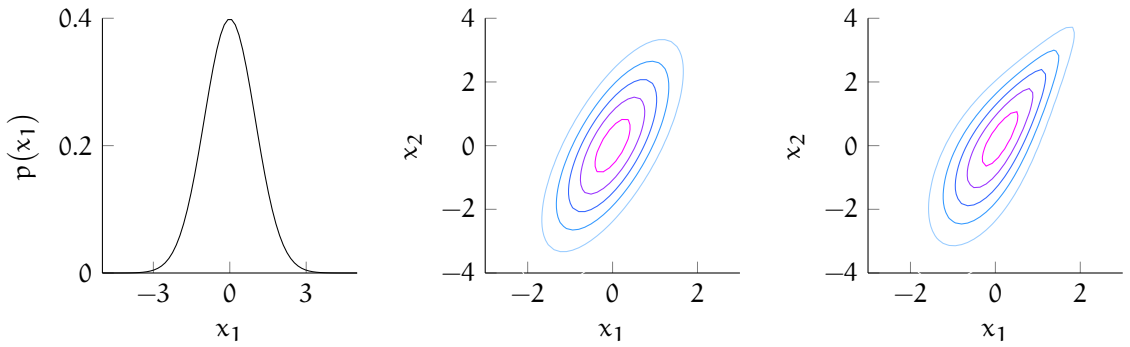
$$\hat{p}_{\text{wkde}}(x) = \frac{1}{h} \sum_{j=1}^{N} \omega_j k\left(\frac{x - x_j}{h}\right) \tag{3.37}$$

with ω_j as the weight for the j-th sample. All weights have to be positive and sum to one.

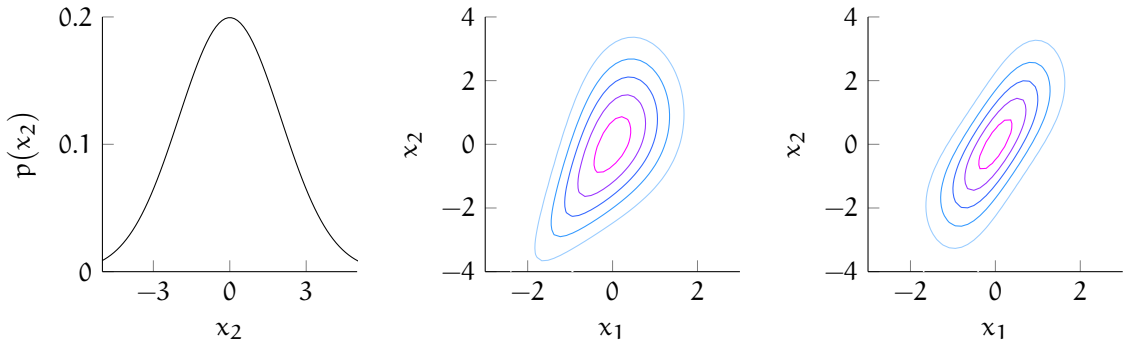
On the one hand, the approach seems favourable for creating flexible semi-parametric descriptions of the observed density, but, on the other hand, KDE is memory consuming and computationally demanding [Mur12, p. 508]. For analysing univariate data, KDE is a viable tool, but in the case of multivariate data, it becomes more and more memory consuming. In order to keep the same resolution in the multivariate case as in the univariate case, the number of sampling points grows exponentially with the number of dimensions. As a result, the KDE of multivariate data is either crude or infeasible. How a distribution can be separated into its marginal distribution and a distribution which solely describes the correlation among the marginals is subject of the following section.

3.6 COPULA

Copulas are functions which describe the link between the univariate marginal distributions of a multivariate random variable and its joint multivariate dependence structure [Joe14]. Essentially, a copula may describe how parameters behave when they are observed together without necessarily having been observed together. As mixture distributions have two interpretations, copulas can be used in two application scenarios. In the first scenario, a set of random variables is not observed together, for instance, when collecting data from separate sources or different features from different persons, and an artificial correlation among the features is imposed. This model can then be used



(a) Marginal distribution of x_1 (b) Gaussian copula (c) Gumbel copula



(d) Marginal distribution of x_2 (e) Clayton copula (f) Frank copula

Figure 3.7: Probability density functions of bivariate data with different copulas. While the marginal distributions $p(x_1)$ and $p(x_2)$ are equal in all cases, changing the copula alters the bivariate behaviour of the distribution.

to make correlated predictions about seemingly uncorrelated quantities. A similar approach was followed in [Li00], which was the central model eventually leading to the financial crisis in 2008 [Sal09].

The second approach is to have a flexible way to model multivariate data that are observed jointly and are correlated. By using copulas the correlation structure and the univariate marginals can be modelled separately. Unlike the parametric multivariate distributions, where marginal distributions and correlation are restricted to the current parametric form, copulas enable the practitioner to combine different univariate probability distributions with different correlation structures. This can be thought of as a modular approach to model building which greatly increases the flexibility.

The copula approach is theoretically justified by Sklar's theorem [Skl59] which states that every multivariate cumulative distribution function $P(x_1, \ldots, x_D)$ can be expressed through the cdfs of the marginal distributions and a copula C:

$$P(x_1, \ldots, x_D) = C\big(P_1(x_1), \ldots, P_D(x_D)\big). \tag{3.38}$$

As a result, the pdf of a multivariate distribution can be expressed as:

$$p(x_1, \ldots, x_D) = c\big(P_1(x_1), \ldots, P_D(x_D)\big) \cdot \prod_{i=1}^{D} p_i(x_i), \tag{3.39}$$

with $c(\mathbf{x})$ as the density of a copula. Now, the marginal distributions are independent of each other and $P_i(x_i)$ and $p_i(x_i)$ can be chosen freely, but need to be of the same parametric family. Mixing the pdf of a normal distribution with the cdf of a beta distribution for the i-th dimension is not possible. However, for different dimensions, different distributions can be chosen.

Depending on the used copula, different correlation structures can be realised. In Figure 3.7 two marginal distributions and the resulting bivariate densities with four different copulas are depicted. While the Gaussian copula recreates a bivariate normal density, the Gumbel [Nelo6, p. 96], Clayton [Nelo6, p. 116], and Frank copula [Nelo6, p. 116] show different behaviours. A throughout treatise of copula can, among others, be found in [Joe14] or [Nelo6].

3.6.1 *Gaussian Copula*

The Gaussian copula is closely related to the multivariate normal distribution (cf. Equation 3.16) and is in fact derived from it. In contrast to the multivariate normal which is recovered, if the marginals are chosen as univariate normal distributions, it can handle different marginal distributions. The pdf of the Gaussian copula corresponds to [Joe14, p. 163]

$$c\big(P_1(x_1),\dots,P_D(x_D)\big) = \frac{1}{|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{u}^T\left(\mathbf{C}^{-1}-\mathbf{1}\right)\mathbf{u}\right\}, \tag{3.40}$$

with $\mathbf{C}$ as a correlation matrix, and $\mathbf{u} = (u_1,\dots,u_D)$ as the quantile function of a standard normal distribution of the cumulative distribution function of every marginal distribution. The quantile function is the inverse of the cumulative distribution function. Formally, $\mathbf{u}$ is expressed as:

$$\mathbf{u}_i = \varphi^{-1}\big(P_i(x_i)\big), \tag{3.41}$$

with $\varphi^{-1}(\cdot)$ as the quantile function of a standard normal distribution, and $P_i(x_i)$ as the cumulative distribution function of the i-th marginal distribution.

In this thesis, the Gaussian copula is used in conjunction with the sinh-asinh distribution (see Section 3.2) as a very flexible model for image segmentation. See Chapter 4 for further details and a performance evaluation. The Gaussian copula has the advantage of borrowing the correlation structure from a normal distribution which has been proven to be suitable in many tasks. Other multivariate copulas often only use one parameter to model the dependence structure of multiple dimensions. This might be too restrictive in the context of generative image segmentation.

The Gaussian copula uses almost the same number of parameters as the multivariate normal distribution. Note that, a correlation matrix $\mathbf{C}$ which can be used in a Gaussian copula can be derived from the covariance matrix $\mathbf{\Sigma}$ of a multivariate normal distribution by computing

$$\mathbf{C} = \mathbf{D}^{-1}\mathbf{\Sigma}\mathbf{D}^{-1}, \quad \text{with} \quad \mathbf{D} = \big(\operatorname{diag}(\mathbf{\Sigma})\big)^{1/2}. \tag{3.42}$$

As a result, the diagonal elements of $\mathbf{C}$ are equal to one and $\mathbf{C}$ can be thought of as a scaled version of the covariance matrix, which only describes the dependences among

the variables. In the context of copulas, the scales itself—the variances—are modelled through the marginal distributions. A list of other multivariate copulas can, for instance, be found in [Ozd+18]. An extensive list of bivariate copulas is presented in [Nelo6, pp. 116-119].

3.7 SIGNIFICANCE TESTING

The chapter concludes with a small treatise of hypothesis testing, also known as significance testing [Was13, Ch. 10]. Significance testing is a tool from classical statistics which tries to answer the question if realizations of two random variables come from the same or from different distributions. This is also termed a two-sample test because two distributions—two groups—are compared. A one-sample test simply compares one distribution to a hypothesis, for instance, if the mean is greater than a threshold. Two-sample tests can be used to compare if the improvements of a model are significant with respect to another model or if the improvements only happened by chance. Since a significance test assumes a distribution for a random variable, different tests are used in different situations. In the simplest case, that is, testing for a difference in means, where the two random variables are assumed to be drawn from two normal distributions with known standard deviations, the significance level is computed by a Gauss test [SL15, p. 400], also known as z-test. If the standard deviations are unknown a t-test is used [BHH+78, p. 74]. Special tests exist which answer the question if a random variable is normally distributed. A common test for normality is the Anderson-Darling test [AD52].

If the normal assumption is no longer viable, non-parametric tests are used. These tests are termed *exact*, because they are not based on the central limit theorem, that is, assuming normality in the limit of $N \to \infty$ [Was13, p. 161]. A permutation test assumes that the samples in the sets $\mathcal{X}$ and $\mathcal{Y}$ are exchangeable. In the case of equal distributions, permuting the groups of the data does not change the distribution. Formally, two random sequences $x^{(1)}, \ldots, x^{(N)} \sim F_x$ and $y^{(1)}, \ldots, y^{(M)} \sim F_y$ are tested for the null hypothesis H_0 that the two sequences are from the same distribution by [Was13, p. 162]:

$$H_o \ : \ F_x = F_y \, , \quad H_1 \ : \ F_x \neq F_y . \tag{3.43}$$

The test statistic $T(x^{(1)}, \ldots, x^{(N)}, y^{(1)}, \ldots, y^{(M)})$ is, for instance, a difference in means:

$$T(x^{(1)}, \ldots, x^{(N)}, y^{(1)}, \ldots, y^{(M)}) = |\bar{x} - \bar{y}| = t_{\text{obs}} \tag{3.44}$$

with $\bar{x}$ and $\bar{y}$ as the empirical averages of the samples. Following [Was13, p. 163], a permutation test iterates the following steps:

1. Compute the test statistic $T(\cdot) = t_{\text{obs}}$ for the two groups.
2. Randomly permute the group memberships B times, recompute the test statistic T, and store it in a vector $\mathbf{t}$.
3. The p-value is then approximated as:

$$\hat{p} = \frac{1}{B} \sum_{b=1}^{B} t_b > t_{\text{obs}} \tag{3.45}$$

In the case $B = (N + M)!$, that is, B equals the total number of possible permutations, the computed p-value is exact. However, even for low sample sizes B quickly grows very large. In practice, only a subset of the possible permutations is used. An estimate of the error induced through the reduced set of permutations is computed as [OG10]:

$$sd = \sqrt{\frac{p(1-p)}{B}} \tag{3.46}$$

with p as the true p-value and B as the number of permutations. In practice, the upper bound of the variance $1/(4B)$ is used to define the number of necessary permutations [OG10]. For a desired error of $sd = \pm 0.001$, 250 000 permutations are required.

While the number of permutations is still high, they can be evaluated in a reasonable amount of time. In practice, a paired t-test is often a viable alternative, but a permutation test is preferable in general. The Wilcoxon signed rank test [Wil45], which is the historical predecessor of the permutation test, should not be used any longer [SAC07].

In this thesis, two-sample tests are used to answer the question if the proposed improvements are a significant contribution to the field of image segmentation. Note that, in order to answer this question, it is not sufficient to test for the equality of two distributions with, for instance, a two-sample Kolmogorov-Smirnov test [Smi39]. With the help of such tests only the question is answered if two distributions are equal or not. However, distributions can be different although their means are equal. In those cases, this would lead to false conclusions when the question is tried to be answered if a model performs better than another. In order to measure if a model is an improvement over the other, we need to compare the average performances of the models and answer the question if the observed differences are significant or happened by chance. The used metrics are summarized in Section 2.11 and results are presented in Chapter 4.

4

SUITABILITY OF VARIOUS PROBABILITY DISTRIBUTIONS INSIDE A GENERATIVE MIXTURE MODEL APPROACH FOR IMAGE SEGMENTATION

The content of this chapter has been adapted and/or adopted from [WW16], [WW17c], and [WW17b]. Using the generalized hyperbolic distribution in the context of image segmentation has been presented at DICTA-2016 [WW16]. A mixture model based on the multiple scaled t-distribution has been presented at VISAPP-2017 [WW17c]. An in-depth analysis of multiple distributions at image segmentation and a model based on the novel combination of the sinh-asinh distribution and a Gaussian copula have been presented at IVCNZ-2017 [WW17b].

While the previous chapters laid out the foundations of image segmentation and Bayesian inference in general, this chapter explores the limits of mixture models in generative segmentation. In this thesis, mixture models are chosen to segment images, because they can be used in the unsupervised setting where no ground truth annotations are required or available. Additionally, a generative model quantifies the uncertainty with which a prediction is made in a coherent statistical framework. This helps to better understand the limits of a model and is especially helpful in the setting of exploring a new data set together with a domain expert, for instance, a geologist or a physician. Uncertainties are then an important tool to communicate and discuss the results.

In the following, the assumptions of different probability distributions are presented and their implications in image segmentation tested. The impacts of using different distributions are presented visually and with the help of the segmentation metrics presented in Section 2.11. In Section 4.1 the model assumptions of a Gaussian Mixture Model (GMM) are discussed in the light of two common data sets—the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11] (BSDS500) and the LeafSnap Field data set [Kum+12]. Afterwards, different colour spaces are tested and the question is answered if perceptually uniform colour spaces are advantageous in image segmentation. This is followed by an analysis of different ways to include positional data into the segmentation model in Section 4.3. Additionally, the importance of keeping the dependence structure between multiple features is discussed in Section 4.4. Building on these results, a novel probability distribution is introduced in Section 4.5 and the suitability of this model is tested against other distributions. The results are summarized in Section 4.7 and future research directions are briefly sketched in Section 4.8.

4.1 MODEL ASSUMPTIONS

The simplest approach in order to improve the performance of a generative image segmentation method is to take a well-known basis model and build upon it. In this thesis, the Gaussian Mixture Model (GMM) (see Section 2.4.3) is chosen as the baseline. Accord-

ing to the direct interpretation of mixture models (see Section 3.2.4), the GMM assumes a normal distribution (see Section 3.2.2) for every region. For a high agreement between the real data and the model assumption, the data of every region has to be as normally distributed as possible.

In the following sections, two directions are examined to improve the segmentation model. In the first case, the baseline model is kept and the input data is varied. In the other case, the model is adapted to better fit the data. The first case is easier, and is, therefore, conducted first. If a modification of the input data helps a normal distribution, it helps the other distributions as well, because they include the normal distribution as a special case. In a first step, however, the model assumptions of a GMM are tested by taking a segmentation benchmark (see Section 2.10) and evaluating how often the model assumption is valid. In this thesis, the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11] (BSDS500) is the standard benchmark and is used to answer most of the questions in this chapter.

From a statistical perspective, we can simply test how many of the annotated ground truth regions in the BSDS500 are normally distributed. In order to ease the testing, only the marginal distributions of the multivariate data are tested. An Anderson-Darling test (see Section 3.7) is used for this. Note that, marginal normality does not imply multivariate normality, because the dependence structure has to be of the same form as a multivariate normal distribution as well. Therefore, marginal normal distributions are a necessary but not a sufficient condition. Essentially, the dependence structure of a multivariate normal is a matrix of correlations. See [Joe14, Sec. 2.6] for further details on the dependence structure of multivariate normal distributions.

In Figure 4.1 the percentage of marginally normally distributed regions are presented in various colour spaces. Different spaces are analysed to test if one of them has a theoretical advantage over others when using a GMM. In case the percentage of normally distributed regions is higher, the model should perform better because the assumptions of it are violated less often. In Section 4.2 this is tested empirically. The analysed colour spaces are the RGB, YUV, Lab, HSV, and HSV V2 colour space. The coordinates of the respective colour spaces are used as input data for the normality test. The HSV V2 colour space is a transformed variant of the HSV colour space and computed by [SM00]:

$$\begin{bmatrix} h' \\ s' \\ v' \end{bmatrix} = \begin{bmatrix} v \\ v \cdot s \cdot \sin(h) \\ v \cdot s \cdot \cos(h) \end{bmatrix}. \tag{4.1}$$

It is evident that in comparison to the standard RGB colour space the Lab colour space has a significantly higher percentage of marginally normally distributed regions. This is especially visible in the colour channels a and b. The remaining colour spaces are comparable to the RGB colour space and do not deviate strongly. Therefore, the Lab colour space is—at least from a theoretical perspective—favourable. In the following section, this claim is tested experimentally.

The same experiments are repeated on the LeafSnap Field data set. In comparison to the BSDS500, the LeafSnap field data set has not a single marginally normally distributed region in all considered colour spaces according to an Anderson-Darling test. This can be explained, by acknowledging that the chance of a normally distributed marginal distribution rises if there are more than two regions per image. However, in

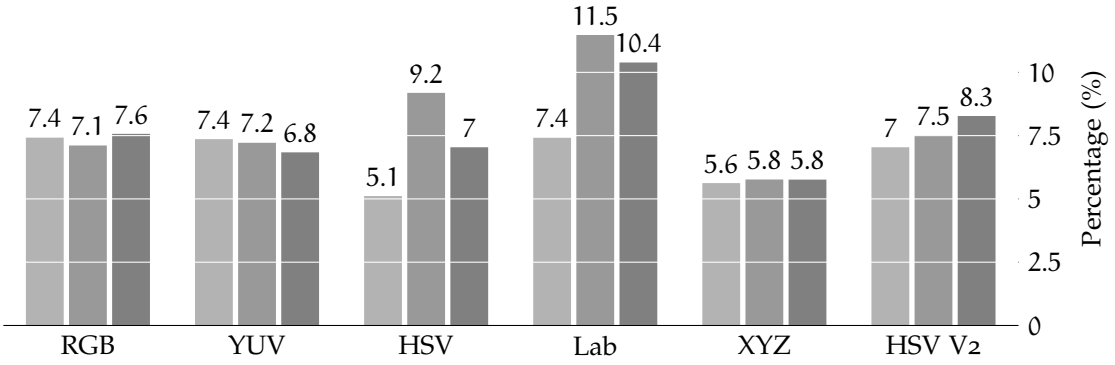


Figure 4.1: Percentage of normally distributed marginal regions on the BSDS500. Different colour spaces are analysed to test if the coordinates of the respective colour spaces behave like normal distributions. Each bar represents one dimension of the respective colour space.

the case of the LeafSnap field data set the images consist of two regions and the depicted leaves are highly corrupted by shading and other real-world capturing artefacts.

4.2 COLOUR SPACES

In this section, the impacts of using different colour spaces for image segmentation with a GMM are analysed. On the one hand, the Lab colour space is favourable since the amount of normally distributed regions is significantly higher, but, on the other hand, experimental results do not verify this claim (see Table 4.1).

The parameters of a GMM are estimated in a supervised fashion based on the provided ground truth segmentations. The ground truth information is exploited in order to compute an upper bound of the maximal achievable segmentation accuracy. The derived conclusions are therefore robust against issues in estimating the models due to local minima.

The BSDS500 and the LeafSnap Field data set are used. In both sets ground truth segmentations are available. In the BSDS500 multiple ground truth annotations are provided per image. Each ground truth is treated independently of the others and is used to estimate the parameters of GMM. The resulting segmentation is then compared to the respective ground truth. As a result, the 200 images of the test set become a set of 1 063 pairs of ground truth and image. In the LeafSnap Field data set one ground truth per image is provided. This results in 300 pairs of ground truth and image. The standard measures of the data sets are used for evaluation. In the case of the BSDS500 these are the Probabilistic Rand Index (PRI), Variation of Information (VoI), and segmentation covering. In the case of the LeafSnap Field data set precision, recall, and F-measure are used. To asses if the observed differences in the metrics are significant, a permutation test with 250 000 random samples is performed. The results of the RGB colour space are used as reference distributions. The analysis is summarized in Table 4.1.

Unsurprisingly, the results of the RGB and YUV colour space are identical, because the conversion is a simple linear transformation. It only shifts the means and scales the features spaces differently. Both changes can easily be modelled by a GMM. It is, however, considered as a proof of concept.

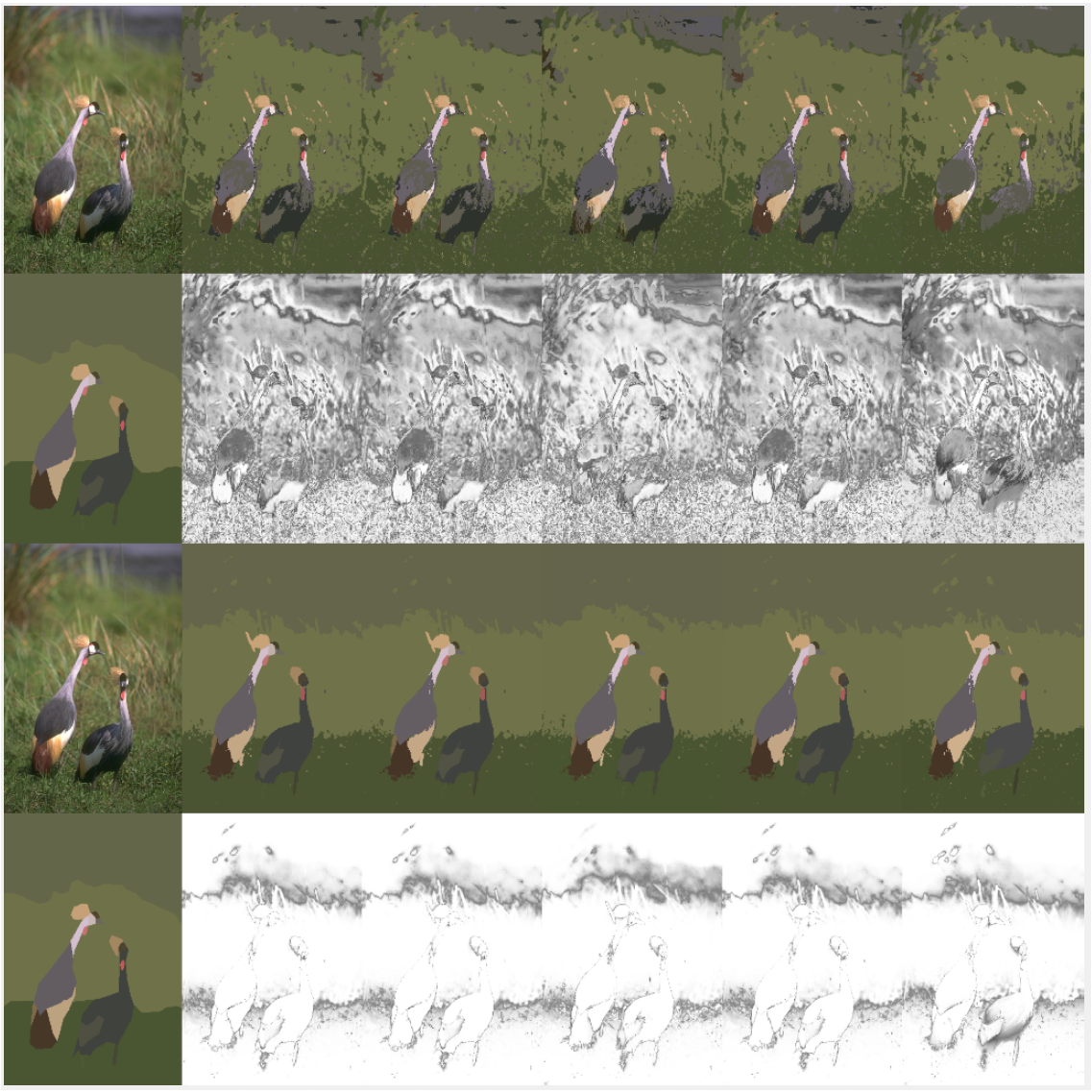


Figure 4.2: Qualitative results of the analysed mixture models on the BSDS500 for different colour spaces with and without positional features. First row: image, segmentations using the coordinates of the RGB, HSV, LAB, XYZ, and HSV V2 colour spaces without positional features. Second row: ground truth, corresponding uncertainty maps. Third row: image, segmentations using the coordinates of the RGB, HSV, LAB, XYZ, and HSV V2 colour spaces and positional features. Second row: ground truth, corresponding uncertainty maps. All colour spaces appear to be equally well suited and the differences between colour spaces are barley visible. However, including positional features has a major impact on the quality and the uncertainties of the segmentations.

Surprisingly, using perceptually uniform colour spaces like Lab is neither increasing nor decreasing the performance of the GMM significantly. In contrast, the differences of both HSV spaces are significantly worse on the BSDS500 and HSV is significantly worse on the LeafSnap field data. This is in direct contrast to the results presented in [Pas01] which favoured HSV in a retrieval setting. In the case of image segmentation with a GMM, this claim needs to be objected. The reason for this might lie in the Euclidean dis-

	BSDS500						LeafSnap					
	no position			position			no position			position		
colour	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓	P ↑	R ↑	F ↓	P ↑	R ↑	F ↓
RGB	0.842	0.605	2.231	0.952	0.890	0.757	0.942	0.994	0.965	0.917	0.998	0.951
YUV	0.842	0.605	2.231	0.952	0.890	0.757	0.942	0.994	0.965	0.917	0.998	0.951
HSV	*0.823*	*0.574*	*2.351*	*0.947*	*0.879*	*0.819*	0.924	0.995	*0.955*	*0.910*	*0.997*	*0.947*
Lab	0.842	0.604	2.234	0.952	0.890	0.759	0.939	0.994	0.963	0.916	0.998	0.950
XYZ	*0.811*	*0.549*	*2.462*	*0.933*	*0.852*	*0.929*	*0.973*	0.981	*0.975*	*0.958*	*0.992*	*0.973*
HSV V2	0.832	*0.585*	*2.312*	*0.948*	*0.881*	*0.803*	0.939	0.994	0.963	*0.916*	*0.998*	*0.950*

Table 4.1: Segmentation accuracies of a GMM in different colour spaces with and without positional features. Significant differences with respect to the RGB colour space—the baseline—are marked in italic font. The arrows indicate the direction of better results. Changing the colour space does not significantly alter the accuracies, except in the XYZ and the HSV colour spaces. While XYZ is significantly better on the LeafSnap Field data, it is significantly worse on the BSDS500. The HSV colour spaces are on both data sets significantly worse. Including positional features has a major impact on the segmentation accuracies.

tance commonly used in retrieval systems where scaling the features has a high impact. The GMM used to segment the images is robust to scale changes, because it can model this with the help of the covariance matrices.

The XYZ colour space performs significantly worse on the BSDS500 but significantly better on the leave images. The reason for this might be a better separation of green colours which dominate the foreground class. In general, the performance on the LeafSnap Field data set is remarkable. If correctly initialized, a simple GMM allows a near perfect separation of foreground and background, whereas the results on the BSDS500 are not encouraging. Although the model is estimated from the true regions, the resulting segmentation show on average heavy clutter and high uncertainties (see Figure 4.2).

One of the major drawbacks of the current model is the lack of positional information. The model cannot distinguish between pixels with similar colours which appear in different regions of an image. This does not matter on the LeafSnap data where a clear distinction between fore and background is possible by colour alone. However, on more complex tasks like the BSDS500 with more than two regions, a unique assignment of colour and region is not always as easy. Analysing different ways of including this information is the subject of the following section.

4.3 POSITIONAL DATA

In this thesis, we propose to include the position of a pixel inside the image as an additional feature. This idea is taken from the mean-shift [CM02] and the original work on normalized cuts [SM00], where a similar approach is followed. However, when using generative models for image segmentation, this concept has not been explored exten-

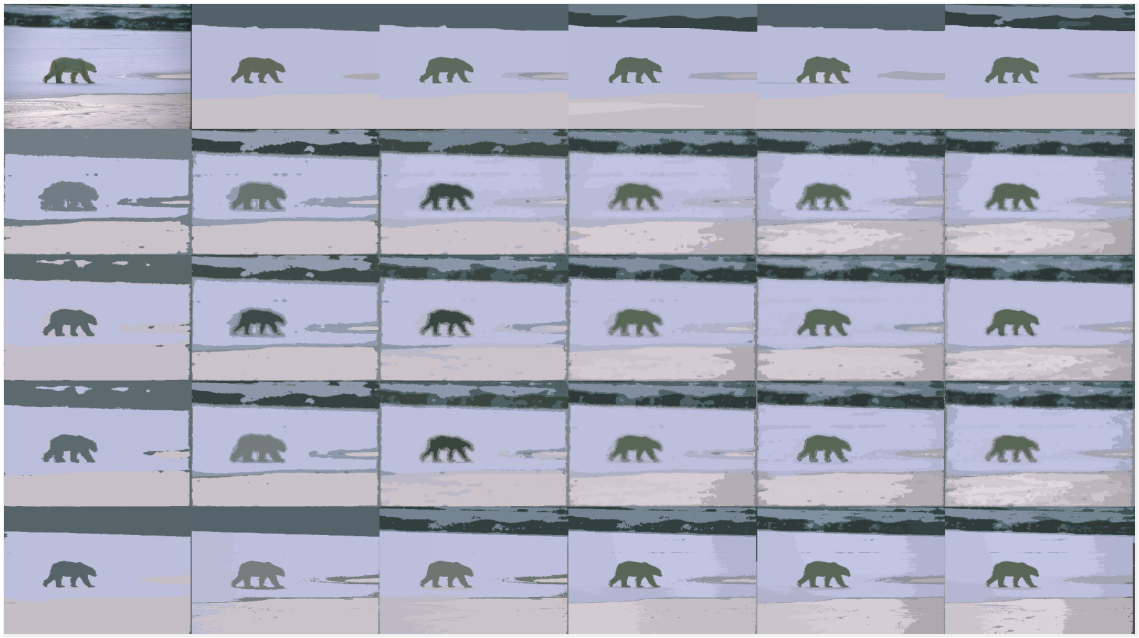


Figure 4.3: Qualitative results of the analysed mixture models on the BSDS500 with a varying number of mixture components. Top row: image, ground truth segmentations. Following rows: segmentations using the algorithms presented in Table 4.2 with $K = \{3, 5, 7, 15, 20\}$ components. From top to bottom: base, SVGMM, CLP, position.

sively. Instead, spatial relations are modelled by an Markov Random Field (MRF) as in [SNG08; Sfi+10; Sfi+11; NW13] or included with a post-processing step [Des14]. In this section, both approaches are compared. Starting with a repetition of the analysis of the previous section, the inclusion of pixel positions is compared to more sophisticated methods from the literature.

The results of the previous section are repeated with positional features included and are also presented in Table 4.1. Note that, the coordinates of the pixel positions do not really cluster. The distribution is uniform. Therefore, the GMM is not a suitable model. However, in the case of the BSDS500, the results are far better if the positional features are included. Again, using different colour spaces does not provide significantly better results. In the case of the XYZ and HSV colour spaces, the results are significantly worse.

In the case of the LeafSnap field data set, using positional information is not beneficial and degrades the segmentation accuracy. The reason for this is the central position of the leave which is entirely surrounded by the background class. This configuration is not easily modelled by a two-component GMM, because both distributions will probably share the same mean in the positional features. Separation can only be achieved by the covariance matrices. In fact, the covariance matrix of the background class has to span the total image. To improve the results in this setting, the distribution of the background class should have the ability to spatially model a hole, or be modelled by a mixture distribution to achieve the same effect. Again, the coordinates of the XYZ colour space perform significantly better than the RGB coordinates, while the remaining colour spaces do not provide significantly different results.

Since the inclusion of pixel coordinates provides a significant improvement in all considered metrics on the BSDS500, the inclusion is compared to the MRF-based approaches

	BSDS500											
	base [Sfi+10]			SVGMM [Sfi+10]			CLP [Sfi+10]			position		
K	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓
3	0.684	0.460	2.449	0.695	0.467	2.456	0.700	0.475	2.448	**0.706**	**0.505**	**2.209**
5	0.735	0.414	2.634	0.740	0.419	2.648	0.741	0.418	2.683	**0.763**	**0.480**	**2.223**
7	0.744	0.369	2.811	0.749	0.372	2.852	0.750	0.370	2.893	**0.770**	**0.443**	**2.319**
10	0.746	0.316	3.075	0.748	0.315	3.110	0.749	0.311	3.163	**0.769**	**0.396**	**2.449**
15	0.743	0.261	3.388	0.743	0.253	3.452	0.742	0.248	3.530	**0.761**	**0.337**	**2.671**
20	0.740	0.224	3.632	0.741	0.254	3.489	0.737	0.247	3.765	**0.755**	**0.301**	**2.847**

Table 4.2: Comparison of different approaches for including positional information in a generative mixture model. The best values are marked in bold font. All differences are significant with respect to the best performing model from the literature. A simple inclusion of the positional features beats MRF-based approaches from the literature in all considered cases.

presented in the literature. Unlike many other methods, the work of [SNG07] is an exception and uses positional features as well. However, the reported results appear to be too good to be true and outperform all other methods on the Berkeley Segmentation Data Set and Benchmarks 300 [Mar+01] (BSDS300) although the method does not estimate the number of mixture components on a per image basis, but is fixed for every image. One possible reason for the excellent accuracies computed by the boundary displacement error can be a reduction of the image resolution on which the analysis is performed. If there are fewer borders, the error is reduced conversely. However, a change of resolution is not reported. For a better comparison with the related work and the methods developed in this thesis, the results of [SNG07; SNG08; Sfi+10; Sfi+11] on the BSDS500 are computed by using the authors' own implementation[1] and evaluated with the same metrics which are used throughout this thesis to asses the segmentation quality (see Section 2.11).

The baseline experiment of [SNG07] is repeated using our own implementation of a GMM with the feature set reported in [Sfi+10], which are reported in [Sfi+10] to yield the same accuracies. The feature set consists of the coordinates of the Lab colour space smoothed by the blobworld scale selection [Car+02] and the blobworld texture contrast which is one of three texture descriptors defined in [Car+02]. The number of clusters K is not estimated on a per image basis but kept constant across all images [Sfi+10]. Afterwards, the metrics are averaged for all segmentations with the same number of clusters. In contrast to the original work presented in [Sfi+10], the three standard evaluation metrics of the BSDS500—PRI, segmentation covering, and VoI—are used to asses the quality of the segmentations.

The results are summarized in Table 4.2. Including positional features is compared to the baseline model of [Sfi+10] denoted by base, a spatially varying GMM (SVGMM) [SNG08], and a GMM with a continuous line process [Sfi+10]. Note that, the baseline

1 https://github.com/sfikas/duguepes-matroutines Retrieved June 23, 2020.

	BSDS500			LeafSnap		
distribution	PRI ↑	C ↑	VoI ↓	P ↑	R ↑	F ↑
Normal (full covariance)	0.952	0.890	0.757	0.917	0.998	0.951
Normal (diagonal covariance)	0.897	0.780	1.344	0.935	0.994	0.960
KDE (diagonal covariance)	0.924	0.831	1.086	0.956	0.996	0.975

Table 4.3: Segmentation accuracies of different types of probability distributions using the coordinates of the RGB colour space and the pixel positions as features. The impact of including correlation is studied. The models are trained in a supervised fashion to mimic the ground truth segmentation as good as possible.

model does not include positional information in any form. It is included to asses the improvements by using MRFs.

While using an MRF increases the segmentation accuracy with reference to the base model, simply including positional features significantly outperforms the other approaches in all considered metrics. Therefore, the more complex MRF approaches can be replaced by including the in terms of model complexity simpler pixel positions as a feature. As a result, the considered feature space becomes greater. On the one hand, this increases the number of free parameters which have to be estimated. But, on the other hand, this may additionally help to estimate a better model with an increased separation of the regions in the feature space.

4.4 UNIVARIATE VS. MULTIVARIATE DISTRIBUTIONS

In the previous section, multivariate distributions have been used to segment images. However, when using multivariate data the number parameters grows faster than the additional dimensions. In the case of a multivariate normal distribution, the number of free parameters for a model of dimension D is computed by $D + D(D+1)/2$. The first term of the sum corresponds to the number of free parameters of the mean vector and the second term of the sum corresponds to the number of free parameters of the covariance matrix. For an increasing dimension of the feature space, the number of free parameters of the covariance matrix quickly grows. Recall, the covariance matrix models the probability of observing different features together, that is, it captures the correlation among the provided features. It is of desirable interest to test if this information is necessary for image segmentation or if the main information is captured by the marginal distributions alone.

Further, many flexible probability distributions are only defined in the univariate case, like the sinh-asinh distribution (see Section 3.2.2.4). It is therefore of great interest to test if the correlation can be neglected in favour of more flexible marginal distributions. In this section, the extent to which correlation is of importance when modelling image data is quantified. A GMM is estimated in a supervised fashion with full covariance matrices and compared to a GMM where only the diagonal elements of the covariance matrix are estimated. This corresponds to a mixture model which only uses the marginal distributions of the multivariate data. As an upper bound to quantify what is possible

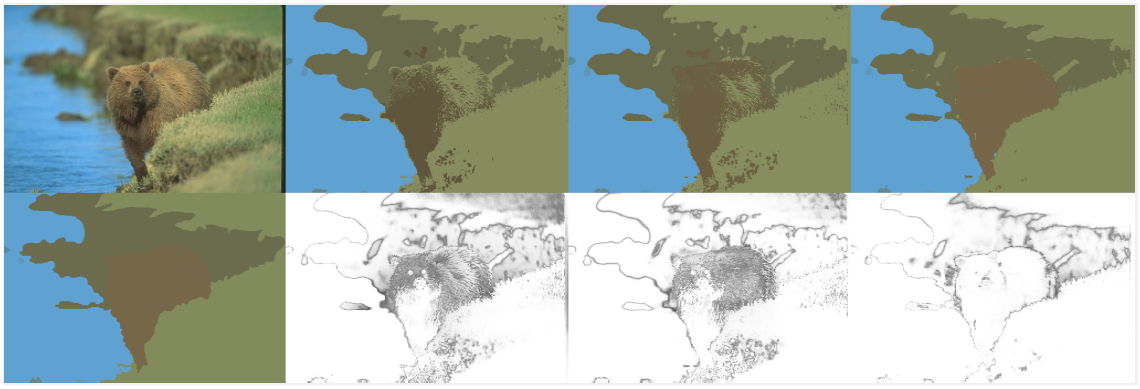


Figure 4.4: Qualitative results of the analysed mixture models on the BSDS500 with and without correlation included. Top row: image, segmentations by normal mixture model with diagonal covariance, mixture of univariate kernel density estimates with diagonal covariance matrices, and normal mixture model with full covariance matrix. Bottom row: ground truth, uncertainties of the corresponding mixture models. The models with diagonal covariance matrices struggle to separate the bear from the background.

by using as flexible as possible distributions, a Kernel Density Estimation (KDE) is used. Note that, the KDE can only be used meaningfully if a ground truth is available or as presented in Chapter 6 another source of information which aids the KDE is included. Otherwise, only an indirect interpretation of the mixture model is possible, because the latent variables have no specific meaning in KDE. Further, a KDE is not conveniently used in feature spaces with two or more dimensions, because the number of necessary kernels grows exponentially if the same resolution is kept as in the univariate case.

The results of the analysis are summarized in Table 4.3. By restricting the covariance matrices of the normal distributions to the diagonal elements—a univariate view on multivariate data—the segmentation accuracies are significantly degraded. If, however, a more flexible distribution is used—a KDE in this case—the drop in performance can be softened. Therefore, more flexible marginal distributions are desirable in general, but the dependence structure of the features is essential and should not be omitted during model building. In the next section, different ways to keep the important correlations and increase the flexibility of the mixture models are discussed. Ultimately, this leads to a representation where the correlation structure is separated from the marginal distributions. Both can then be modelled independently of each other while keeping both cues for building the segmentation model.

4.5 DISTRIBUTIONS

Recall, roughly eight per cent of the marginal distributions of the BSDS500 are normally distributed (see Figure 4.1). Conversely, assuming a normal distribution for every region is insufficient in the vast majority of times. Therefore, exploring different ways to generatively model the observed image data are presented. The key idea behind this approach is—in contrast to modifying the observed data—to increase the expressive capability of the model. The major benefit compared to modifying the feature set is that the new model better copes with real data and is, therefore, more flexible in different applications.

In the literature, Student's t-distribution (see Section 3.2.2.2) is a commonly used alternative (see, e. g., [SNG07] or [NW12]). In this thesis, we additionally explore the suitability of the multiple scaled t-distribution (see Section 3.2.3.3), and the generalized hyperbolic distribution (see Section 3.2.2.3). Further, a novel probability distribution is introduced which is built on combining the univariate sinh-asinh distribution (see Section 3.2.2.4) and a Gaussian copula (see Section 3.6) for modelling the dependence structure.

4.5.1 *Sinh-asinh Copula*

In Chapter 3 the sinh-asinh distribution is described as a very flexible univariate distribution with pleasing analytical properties. It is created by modifying the probability density function (pdf) of the univariate normal distribution to more complex shapes by applying a parametric transformation. In contrast to the univariate normal, two additional parameters are introduced which can control the skew and the kurtosis of the distribution.

As already described in Section 3.6, a copula can be used to dismantle a multivariate probability distribution into a probability distribution which solely describes the dependence structure among the features and the pdfs of the univariate features. This follows a modular concept and greatly increases the flexibility of the distribution. In this thesis, we propose to use a Gaussian copula (see Section 3.6) for modelling the dependence structure and the sinh-asinh distribution for modelling the univariate densities. This distribution is then used to build a mixture model and will be termed Sinh-asinh Copula Mixture Model (SCMM).

In general, three steps are taken to build a mixture model in the copula framework. The first one is choosing a copula. In this case, a Gaussian copula is chosen, because it closely resembles the dependence structure of a multivariate normal. Further, it has been shown to greatly improve the segmentation metrics when the dependence structure is included (see Table 4.3). The second step is choosing the pdfs of the marginal distributions. In this thesis, all features—position and colour—are modelled by the sinh-asinh distribution, because it is a very flexible distribution with a convenient parametric form. How to exploit the parametric form to omit complicated calculations is shown in the following. Lastly, the newly created probability distribution is used within a mixture model (see Section 3.2.4) as a model for all regions of an image.

Recall, the pdf of a Gaussian copula equals (see Equation 3.40):

$$c\big(P_1(x_1),\dots,P_D(x_D)\big) = \frac{1}{|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{u}^T\left(\mathbf{C}^{-1}-\mathbf{1}\right)\mathbf{u}\right\},$$

with $\mathbf{u}_i = \varphi^{-1}\big(P_i(x_i)\big)$. Since the cumulative distribution function (cdf) of the sinh-asinh distribution equals [JP09]

$$P(x|\mu,\sigma,\epsilon,\delta) = \varphi\left(S_{\epsilon,\delta}\left(\frac{x-\mu}{\sigma}\right)\right), \tag{4.2}$$

the computation of $\mathbf{u}$ can be simplified as follows:

$$\begin{aligned} u_i &= \varphi^{-1}\left(\varphi\left(S_{\epsilon,\delta}\left(\frac{x-\mu}{\sigma}\right)\right)\right) \\ &= S_{\epsilon,\delta}\left(\frac{x-\mu}{\sigma}\right). \end{aligned}$$

As a result of the simplification, the computation of the cdf and the quantile function is omitted entirely. Note that, this simplification is only possible, because of the specific combination of marginal distributions and copula. Finally, the pdf of the proposed SCMM equals:

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \omega_k \left[c_k(\mathbf{x}) \prod_{i=1}^{D} p(x_i|\mu_{ki}, \sigma_{ki}, \delta_{ki}, \epsilon_{ki}) \right] \tag{4.3}$$

$$= \sum_{k=1}^{K} \omega_k p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \boldsymbol{\delta}_k, \boldsymbol{\epsilon}_k, \mathbf{C}_k). \tag{4.4}$$

4.6 ACCURACY ASSESSMENT

In order to qualitatively and quantitatively asses the quality of the different mixture models, the BSDS500 and the LeafSnap field data sets are utilized. All parameters of the respective mixture models are estimated with conjugate gradient descent on a suitable manifold [TKW16] if possible. By using a suitable manifold for every parameter, the range of possible values is restricted. This is especially useful to tie the covariance matrix to the manifold of symmetric positive definite matrices. Further details about manifolds can, among others, be found in [TKW16]. A gradient-based estimation is chosen here to ensure that the computed set of model parameters is at least a local optimum of the achievable segmentation metrics.

In the case of the normal distribution, an optimal solution is obtained by empirically estimating the sample mean, the sample covariance, and the mixture weights from the ground truth annotations. In the case of the generalized hyperbolic distribution, gradients cannot be computed analytically because of the Bessel functions used to define the pdf (see Section 3.2.2.3). Therefore, the mixture model is optimized with a derivative-free Nelder-Mead method [NM65].

The function subject to minimization is the quadratic difference between a binary vector indicating the region membership and the responsibilities computed from the mixture model evaluated at every pixel j. This is a supervised evaluation of the mixture models and establishes an upper bound of the possible segmentation accuracies. Formally, this is expressed as:

$$f(\mathbf{X}) = \frac{1}{2} \sum_{j=1}^{N} \left\| \mathbf{gt}_j - \mathbf{z}_j \right\|_2^2 \tag{4.5}$$

with N as the number of pixels in an image, $\mathbf{gt}_j$ as a one hot encoded vector of the region membership at pixel j, and $\mathbf{z}_j$ as the vector of responsibilities at pixel j. The computation

	BSDS500			LeafSnap		
distribution	PRI ↑	C ↑	VoI ↓	P ↑	R ↑	F ↑
Normal	0.952	0.890	0.757	0.917	0.998	0.951
t	0.953	0.891	0.762	0.929	**0.996**	0.956
multiple scaled t	0.954	0.893	0.752	0.927	**0.996**	0.955
generalized hyperbolic	0.931	0.822	1.180	0.986	0.988	0.987
Sinh-asinh (kurtosis) Gauss Copula	**0.966**	**0.919**	**0.606**	**0.988**	**0.990**	**0.989**
Sinh-asinh (skew) Gauss Copula	**0.964**	**0.913**	**0.639**	**0.987**	**0.989**	**0.988**
Sinh-asinh Gauss Copula	**0.977**	**0.943**	**0.442**	**0.982**	**0.986**	**0.984**

Table 4.4: Segmentation accuracies of different types of probability distribution using the coordinates of the RGB colour space and the pixel positions as features. The model is trained in a supervised fashion to mimic the ground truth segmentation as good as possible. Significant differences with respect to the normal mixture model are marked in bold font ($p < 0.05$). While the sinh-asinh Gauss copula shows a significantly superior performance, using a t-distribution or a multiple-scaled t-distribution on the BSDS500 does not significantly alter the achievable segmentation accuracy. On the LeafSnap data, using a t-distribution or a multiple-scaled t-distribution significantly degrades the recall. The differences in precision and F-measure are, however, not significant. Again, using the sinh-asinh Gauss copula significantly alters the metrics.

of the $\mathbf{z}_j$ depends on the pdf of the chosen mixture model. In general, the computation of the k-th element of $\mathbf{z}_j$ can be formally expressed as:

$$z_{jk} = \frac{\omega_k p\left(\mathbf{X}_j|\theta_k\right)}{\sum_{k'=1}^{K} \omega_{k'} p\left(\mathbf{X}_j|\theta_{k'}\right)}. \tag{4.6}$$

Note that, Equation 4.6 is equal to the E-step of the EM algorithm (see Section 2.4.3) if a GMM is used.

All models are initialized with the empirical estimates of the mean value and the covariance matrix of a region. Additional parameters with respect to the normal distribution are initialized such that the distribution resembles the normal distribution as close as possible. For instance, in the case of the t-distribution the degrees of freedom ν is initialized with the value 10. Afterwards, the influence of the newly gained flexibility is analysed by starting the supervised optimization of Equation 4.5.

Qualitative results of a challenging sample image of the BSDS500 are presented in Figure 4.5. All models except the proposed SCMM fail to compute a sensible segmentation although the ground truth is known. However, the uncertainty estimates directly show where the models are uncertain. This is a great benefit of mixture models in general.

Quantitative results of using different distributions are summarized in Table 4.4. To asses the significance of the changes, the normal mixture model is the reference model and the null hypothesis is the assumption of equals means, that is, the average performance is not affected by changing the model distribution. Significant differences are marked in bold font. Surprisingly, using a t-distribution or the multiple scaled variant does not significantly alter the achievable accuracies on the BSDS500. In fact, the null hypothesis of equal means cannot be rejected at the 5 per cent significance level, although a

small improvement is measurable. The proposed combination of a Gaussian copula and the univariate sinh-asinh distribution offers the desired flexibility and significantly outperforms all other models in all considered metrics on the BSDS500. The null hypothesis of equal means is rejected at $p = 4 \times 10^{-6}$.

In the case of the LeafSnap data, using a t-distribution or the multiple-scaled variant slightly improves precision and F-measure. The differences are not significant at the five per cent level. However, the sinh-asinh Gauss copula provides a significant improvement over the reference model in terms of precision and F-measure. The null hypothesis of equal means is rejected at $p = 4 \times 10^{-6}$. The best accuracy on the LeafSnap data is achieved by the generalized hyperbolic distribution, which appears to be especially well suited for this task. However, in the multi-class setting of the BSDS500 the performance is significantly worse. This can be ascribed to the more complex estimation of the parameters and is not necessarily a sign of an inability to model the data.

The proposed SCMM yields the current state-of-the-art in generative image segmentation if evaluated in the given setting. The proposed model contributes a significant improvement over the commonly used normal distribution. Additionally, less common distributions like the t-distribution and the multiple scaled t-distribution are significantly outperformed by the proposed model.

On the one hand, the provided analysis constitutes an upper bound of the maximal reachable segmentation accuracies. In order to achieve this, the ground truth has to be known. In this setting, it can be proved that distributions have to be more than robust to improve the segmentation accuracies. Additionally, the proposed framework is a modular approach. In future work, different combinations of copula and marginal distribution can be examined for further improvements. Further, the proposed model is generative. Sensible uncertainty estimates are a direct result of the analysis.

On the other hand, it is unclear how the results transfer to the unsupervised setting. The reached improvements might not be achievable if the ground truth information cannot be exploited. In the following chapter, this aspect is taken up in further detail.

4.7 SUMMARY

In this chapter, the foundations of a successful image segmentation in a generative framework are lain out. Starting with the question if a normal mixture model is a suitable choice for images of natural scenes, different colour spaces are analysed with respect to the same question. Including positional information in the segmentation model has been verified as a significant improvement over more sophisticated methods like MRFs. Further, the correlation of the analysed features is another important cue for a successful segmentation model. Afterwards, a method for independently modelling the correlation and the pdf of the marginal distributions is presented and a novel mixture model is introduced which significantly outperforms all other distributions regarding the used segmentation metrics.

Using robust distributions like the t-distribution as proposed in [SNG07; NW12] does not provide a significant improvement over the ubiquitous GMM. Instead, skew and a *lower* kurtosis appear to be more beneficial. In this thesis, the Sinh-asinh Copula Mixture Model is proposed which allows both, modifying skew and kurtosis in both directions.

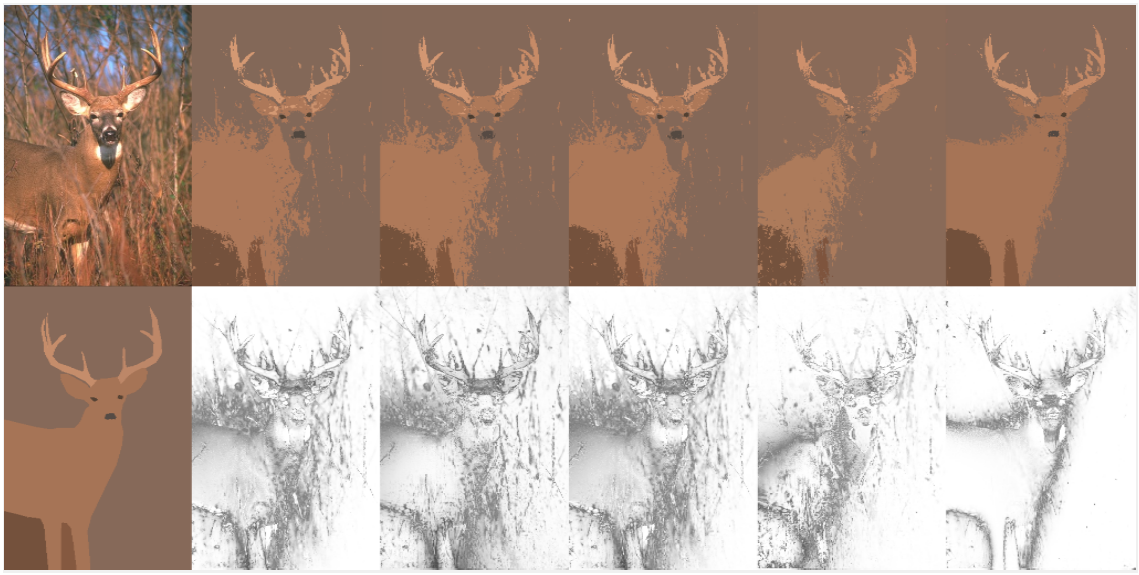


Figure 4.5: Qualitative results of the analysed mixture models on the BSDS500. Top row: image, segmentations by normal mixture model, t mixture model, multiple scaled t mixture model, generalized hyperbolic mixture model, and sinh-asinh copula mixture model. Bottom row: ground truth, uncertainties by normal mixture model, t mixture model, multiple scaled t mixture model, generalized hyperbolic mixture model, and sinh-asinh copula mixture model.

Still, the major issues of the proposed model are the spatial uncertainty regarding the region boundaries and the noisiness of the results in challenging image parts. Additionally, the proposed model suffers from high processing times due to numerous parameters which need to be estimated. In the next chapter, both issues are tackled by using superpixels (see Section 2.7).

4.8 FUTURE RESEARCH DIRECTIONS

Although the expressive capability of the proposed mixture model is already high and a significant improvement over different approaches towards image segmentation, some limitations can be tackled in future work. As shown in this chapter, including correlation is important in reaching the highest segmentation accuracies. However, only the dependence structure of a normal distribution has been considered yet. The statistics literature (see, e. g., [Joe14]) offers a broad range of possible copula which can be tested on their suitability at improving the segmentation quality.

Currently, the model does not include textural features. While this can be considered limiting, generative approaches which include texture features like [SNG07; SNG08; Sfi+10; Sfi+11] perform significantly worse. Discriminative approaches like [MFM04; Arb+11] showed, however, that texture is an important cue in image segmentation. Therefore, the main challenge in generative frameworks appears to be a suitable model for texture features. Simply including it as another feature, like colour and position, to be generatively modelled by a probability distribution appears to be insufficient.

Further, maximizing the likelihood of the model might not be the best way to estimate the parameters in terms of a good segmentation. In the limit of a maximal flexible

distribution—KDE can be considered as a representative of this class of distributions—a ground truth is required to estimate a meaningful segmentation. By using a maximal flexible distribution, one component of the mixture model may be able to model the probability distribution of the whole image and thus violates the direct interpretation of a mixture model which is essential when mixture models are used to compute a segmentation.

In the following chapter, this issue is taken up by generatively including the probability of boundaries in the segmentation model. However, other approaches can be explored in the future. To some extent, including texture features or a different set of features or another model for the positional data might already suffice.

5 SUPERPIXEL BASED IMAGE SEGMENATION

The content of this chapter has been adapted and/or adopted from [WW16], [WW17c], and [WW17a]. The idea to use superpixels within a generative framework and learning a relation between the segmentation metrics and the likelihood of the model has been published at VISAPP-2017 [WW17c]. Learning a relation between the number of mixture components and the image characteristics has been published at DICTA-2016 [WW16]. Integrating the edges of an image in order to improve the unsupervised estimation has been presented at ICIP-2017 [WW17a].

In the previous chapter, the foundations of a successful generative image segmentation model have been lain out. However, in image segmentation a large number of data points have to be considered, because one image usually consists of millions to billions of data points depending on the resolution of the image. Therefore, the processing times of the previously presented estimation are high. Additionally, parameter estimation has been conducted in a supervised fashion. While this successfully establishes an upper bound on the achievable segmentation metrics, it is unclear how the results transfer to the unsupervised case. The following sections are engaged in answering this question.

Starting, with an analysis of different ways to reduce the computational burden (see Section 5.1), different ways to determine the number of mixture components are presented in Section 5.2. Afterwards, the knowledge gained from all previous experiments is gathered into defining the final model (see Section 5.3). In order to improve the unsupervised estimation of the model parameters different ways to include edges as part of the probabilistic model are presented in Section 5.4. The models are compared to the current state-of-the-art in Section 5.5, the results are summarized in Section 5.6, and future research directions are presented in Section 5.7.

5.1 SUPERPIXELS IN GENERATIVE MODELLING

Superpixels (see Section 2.7) are a commonly used preprocessing step in computer vision. In the special case of image segmentation, they have been successfully applied, among others, in [Yan+08; YQG17; LWC12; Zha+18]. Yet, none of the aforementioned approaches performs image segmentation in a generative way. They are either based on iteratively merging adjacent superpixels or on building a graph representation which is then cut to get a segmentation.

In this thesis, we want to benefit from superpixels thrice. First, superpixels shall reduce the number of freckled region assignments at border regions. Second, superpixels shall reduce the computational burden of the proposed models as a domain-specific subsampling. And third, the boundaries of each superpixel shall be integrated into the generative segmentation as an additional cue to enforce that the estimated region bound-

aries lie at probable places. Further details of the latter approach are presented in Section 5.4. However, the primary goal is to reduce the computational demand.

An alternative to superpixels are local windows. They are, among others, used in [Mig08; YWL12; YWC15; Cho+17]. While they are good at defining a local neighbourhood, which can then, for instance, be used to define a measure of texture, the window boundaries seldom adhere to the true region boundaries. They are, therefore, not suitable for integrating an edge model into the probabilistic framework.

An additional alternative to enforcing locality are the Markov Random Field (MRF) approaches (e. g., [Sfi+10]) mentioned in the previous chapter. While they can be successfully used to reduce the high frequency label noise at boundaries, they do not show a good performance in general (see Section 4.3). Further, they are not suitable for subsampling the input data to reduce the computational burden.

The statistics literature offers different ways to reduce the computational burden during model inference with Markov chain Monte Carlo (MCMC). This is for instance achieved by a subsampling. Then, not all data points are used for likelihood evaluation but only sub-set [KCW14; BDH14]. Further, it is proposed to compute lower bounds [MA15] to build a sub-set of the most relevant data. Alternatively, we propose to use a domain-specific approximation technique. Superpixels will be used to vastly reduce the computational demand.

The key idea is to use only one instance instead of all pixels which best reflects the properties of all values. In this thesis, the most probable value within every superpixel is chosen as a substitute for all pixels.

5.1.1 *Subsampling vs. Superpxixels*

The primary goal of using superpixels is to reduce the computational burden of the used mixture models. Hence, the number of data points used in estimating the parameters is kept minimal. As mentioned in the introductory part of this chapter, subsampling the data is an often used alternative to the proposed approach. However, a direct application of subsampling neglects the spatial dependences among neighbouring pixels and simply takes values from a fixed grid, for instance, every n-th point. In contrast, superpixels build a representation of neighbouring pixels which are likely very similar. These groups of pixels are therefore especially well suited to be summarized and excluded from the analysis.

In this section, both approaches—subsampling and superpixels—are compared and two variants are additionally tested. The results of the analysis are presented in Table 5.1. We follow the evaluation scheme presented in [SNG07] (cf. Table 4.2) and evaluate the proposed approaches in the unsupervised setting and vary the number of clusters K from three to fifteen. Afterwards, the standard metrics of the Berkeley Segmentation Data Set and Benchmarks 500 [Arb+11] (BSDS500) are used to asses the quality of the sampling scheme.

In the case of a superpixel representation, titled “Spx” in Table 5.1, Simple linear iterative clustering [Ach+12] (SLIC) (see Section 2.7) is used to compute 1 602 superpixels. SLIC is used, because it shows the best performance regarding speed, boundary recall, and robustness with respect to under-segmentation error [Ach+12]. Only the most probable value of the features within every superpixel is then used to estimate the parameters

	BSDS500											
	Spx			Spx Scaled			Sub			Sub scaled		
K	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓	PRI↑	C↑	VoI↓
3	0.709	0.497	2.247	*0.700*	*0.483*	2.312	*0.700*	*0.482*	2.299	*0.691*	*0.472*	*2.315*
5	0.755	0.457	2.321	0.746	*0.440*	*2.416*	0.745	*0.438*	*2.419*	*0.733*	*0.421*	*2.478*
7	0.765	0.423	2.404	0.757	*0.404*	*2.525*	0.755	*0.401*	*2.540*	*0.744*	*0.383*	*2.624*
10	0.765	0.380	2.549	0.758	*0.360*	*2.692*	0.754	*0.349*	*2.732*	*0.746*	*0.334*	*2.833*
15	0.757	0.320	2.804	0.751	*0.299*	*2.979*	0.749	*0.292*	*3.005*	*0.743*	*0.278*	*3.131*
20	0.751	0.282	3.000	0.746	*0.262*	*3.190*	0.744	*0.254*	*3.208*	*0.739*	*0.244*	*3.352*

Table 5.1: Comparison of four different strategies to reduce the computational burden on the BSDS500. The analysis is conducted in the unsupervised setting with a fixed number of mixture components using a GMM. As segmentation metrics the Probabilistic Rand Index (PRI), the segmentation covering C, and the Variation of Information (VoI) are used. Statistically significant differences are marked in italic font.

of the mixture model. This amounts to a large reduction of the computational demand depending on the chosen number of superpixels. For instance, a typical image of the BSDS500 has $321 \times 481 = 154\,401$ pixel. If only 1500 superpixels are used, this amounts to a reduction of the computational burden of roughly ninety-nine percent. In general, the reduction factor r is computed as:

$$r = 1 - \frac{N_{spx}}{N_{img}}$$

with N_{spx} as the number of superpixels and N_{img} as the number of image pixels. Figure 5.1 illustrates the six feature channels used in the following experiments.

In the case of subsampling, titled "Sub" in Table 5.1, the image resolution is reduced with a factor of 0.9 which results in $49 \times 39 = 1617$ pixels to roughly match the number of superpixels. The reduced image is then used to estimate the parameters of a mixture model. Additionally, two variants titled "scaled" are tested where the parameters of the mixture models are estimated as before, but the final segmentation is not computed from the reduced image and scaled accordingly, but from the data in full resolution.

In order to verify that the superpixels approach is best when reducing the computational demand, the approaches are tested quantitatively. As features, the pixels' positions in the image and the coordinates of the RGB colour spaces are used. As mixture model, a Gaussian Mixture Model (GMM) is chosen where the number of mixture components is not estimated on a per image basis but kept constant across all images. Further, a varying number of mixture components are tested. The BSDS500 is used and the results are summarized in Table 5.1.

In total, the superpixel sampling performs best across all considered metrics and number of mixture components. This becomes more and more evident if the number of clusters increases. However, estimating the final segmentation on the full resolution after the parameters have been estimated on the reduced set does more harm than good and degrades the segmentation metrics.

If the performance of the superpixel sampling is used as the reference and compared to the other three approaches, nearly all differences in means of segmentation covering and VoI are significant. A permutation test with 250 000 combinations has been conducted. Therefore, all remaining experiments in this chapter are carried out with the superpixel sampling scheme if not stated otherwise. The degradation of the segmentation metrics compared to using the full resolution for estimation and segmentation is measurable but not dramatic (cf. Table 4.2).

5.1.2 *Building a Texture Feature from Superpixels*

Since superpixels define a local neighbourhood, building a texture representation is a natural next step. Recall, texture can only be defined by looking at the local neighbourhood. Therefore, it is difficult to define a meaningful texture feature on a per pixel basis. Superpixels solve this problem by defining non-overlapping local entities which can be used to compute a texture feature. In this work, the primary goal of using superpixels is to reduce the computational burden. Hence, a texture representation with only one dimension is desirable. Otherwise, the benefit of a reduced set of data might be consumed by the necessity to estimate a large amount of additional model parameters.

In order to build the texture feature, SLIC is again used to compute a superpixel representation of an image in a first step. Histograms are then chosen to describe the content of the superpixels because they best fit the irregular shape. Further, histogram representations of image parts are rotationally invariant. In fact, they are invariant to any permutation of the input data. Other descriptors usually look at square regions, which would not match the varying shapes of the superpixels. Additionally, we tested Dense SURF [BTVG06], but the resulting texture maps did not look as meaningful as the ones obtained by a histogram representation.

The texture feature itself is built by creating a custom distance matrix, which covers the distance from every superpixel to each other, the distance between the colour histograms of the pixels within the superpixels, and the mean squared difference between the median colour values inside a superpixel.

$$D_{KL}(p\|q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}, \tag{5.1}$$

$$D_{KL}^{Sym}(p,q) = D_{KL}(p\|q) + D_{KL}(p\|q). \tag{5.2}$$

The RGB colour space is used because—as shown in the previous chapter—the influence of using different colour spaces is minimal. As a distance measure for the histogram representations we use the symmetric variant of the Kullback-Leibler divergence for discrete distributions (see Equation 5.2). In order to have a one-dimensional texture feature, the distance matrix is transformed to a one-dimensional subspace by multidimensional scaling [BG03]. As a result, an additional texture channel is integrated into the feature space which is comparable to the RGB colour channels. Further, the position of the pixels within an image in the Cartesian coordinate system is used as an additional feature. This has been proven as a successful way to model neighbourhood relations in a generative model (see Section 4.3). Figure 5.1 illustrates the six feature channels used for the following experiments in this chapter.

Figure 5.1: Overview of the used feature channels for an exemplar image of the training set of the BSDS500. Due to the proposed superpixel sampling the superpixel structure is visible in every channel. From left to right: image, x-position, y-position, three colour channels, and texture channel.

5.2 DETERMINING AN APPROPRIATE NUMBER OF MIXTURE COMPONENTS

After defining the used feature set, an appropriate way to estimate the number of mixture components has to be determined. Recall, in this chapter an algorithm shall be developed which uses as few expert knowledge as possible. Therefore, the estimation as proposed in the previous chapter, where ground truth information has been exploited, is not possible any longer. However, determining a suitable number of mixture components is challenging and heavily influences the results of the algorithm. Commonly, the number of regions in a segmentation is determined by thresholding a quantity of the algorithms. In the context of mixture models for image segmentation this topic is commonly omitted entirely (see e. g., [Sfi+10]).

In this section, we explore several ways of estimating the number of mixture components. First, the cluster validation criteria (see Section 3.3.2) are examined regarding their suitability in image segmentation. Note that in image segmentation, the model assumption of those criteria are almost always violated. Therefore, it is unclear how those approaches perform. Second, a method to learn a relationship between how textured an image is and the number of clusters is explored. Additionally, a modified variant of the Bayesian Information Criterion (BIC) is presented where the relation between a suitable number of mixture components is tried to be learned by introducing an additional penalty term. Lastly, an infinite mixture model is used as the representative of the statistical approaches.

5.2.1 *Cluster Validation Criteria*

The most straightforward approach to estimate the number of mixture components is to use the cluster validation criteria presented in Section 3.3.2. They have been developed for estimating the number of mixture components based on specific traits of the clustering. However, they originally assume that the model assumption is correct. In image segmentation, this assumption does not hold in the great majority of cases. Even in the setting of the LeafSnap data where the modes of the classes are well separated, a test for normality fails every time (cf. Section 4.1).

To test the suitability of cluster validation criteria in image segmentation, we use the GMM as a surrogate model for more complex models developed in the remaining parts of this thesis, because the estimation with Expectation Maximisation (EM) is faster than

computing the results of more flexible distributions with MCMC. Choosing a surrogate model is valid in this case, because if a GMM profits from a cluster validation criterion, then a more complex model which includes the GMM as a special case should profit as well.

Cluster validation criteria choose one segmentation among a set of alternatives with a varying number of mixture components. The best segmentation is chosen according to the cluster validation criterion. Here, we use the feature set described in the previous section including the superpixel approximation which allows computing multiple segmentations in a reasonable amount of time. The number of mixture components is varied from two to twenty and should reflect a suitable range of different regions. In order to asses the overall quality of the approaches, the standard evaluation protocol of the BSDS500 is followed. The three standard metrics—Probabilistic Rand Index, segmentation covering, and Variation of Information—are computed, averaged over all ground truth annotations per image, and finally averaged over all images. The results of all approaches are summarized in Table 5.2.

5.2.2 *Regression Based Approaches*

In contrast to cluster validation criteria, where the best segmentation among a set of possible candidates is chosen based on a criterion, we propose a way to learn the relation between the number of mixture components and the features of an image. Additionally, a second variant is explored where not the mixture components are predicted, but the relation between model characteristics and the segmentation metrics is learned.

5.2.2.1 *Predicting the Number of Mixture Components*

In this section, a way to estimate the unknown number of mixture components based on a Poisson regression (PR) [GH06, p. 110-116] is presented. In contrast to simple linear regression, where a linear relationship is assumed between the dependent and independent variables, Poisson regression assumes that the relationship is best described by a Poisson distribution. This is a common choice when dealing with count data. The regression model is of the following form [GH06, p. 111]:

$$y_j \sim \text{Poisson}(\lambda_j) \tag{5.3}$$

$$\lambda_j = \exp(\mathbf{X}_j \theta) \tag{5.4}$$

The regression is performed on a logarithmic scale, because λ_j is supposed to be positive [GH06, p. 111]. In statistical terms, the model is a generalized linear model with an exponential link function, it is linear in the predictors, and a Poisson distribution is assumed for the output. In order to perform a regression, independent variables need to be chosen for every image. We propose to divide the image $\mathcal{I}$ into a set $\mathcal{I}_{\text{sub}}$ of $S \times S$ sub-images of roughly the same size. For every sub-image summary statistics like the number of Canny edges [Can86] in every colour channel are computed in order to have a measure of how textured and how *object-like* a sub-image is. Additionally, the number of SURF [BTVG06] and BRISK [LCS11] key points, the number of FAST corners [RD05], and the number of MSER regions [Mat+04] are used as independent variables. The resulting values are concatenated into one vector to form the j-th row $\mathbf{X}_j$ of the matrix $\mathbf{X}$.

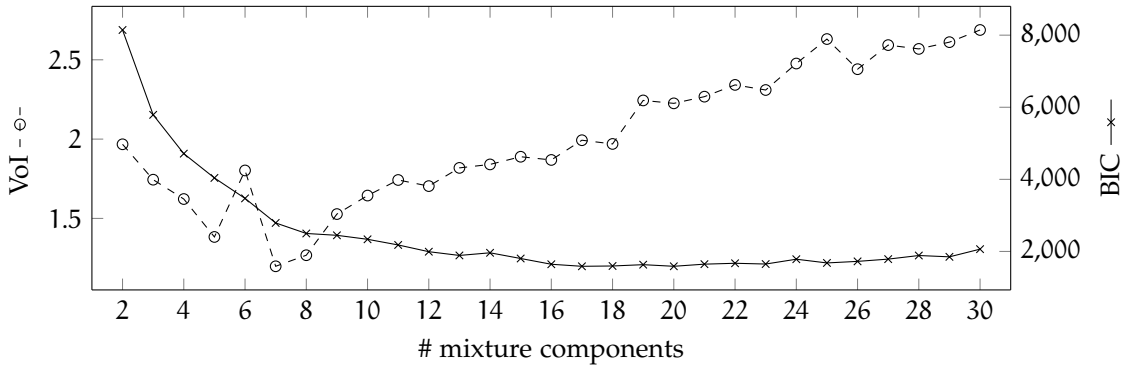


Figure 5.2: Relation between BIC and VoI for a sample image of the BSDS500. The segmentations are computed by a GMM with a varying number of mixture components. It is evident that the minimum of the supervised evaluated VoI, that is, the best segmentation according to the criterion, and the best segmentation according to the unsupervised evaluated BIC do not coincide and strongly deviate.

The model parameters are estimated by maximum likelihood and the training set of the BSDS500. The results of this approach are presented in Table 5.2 in the row titled "PR" (Poisson regression).

5.2.2.2 *Predicting the Segmentation Metrics*

Due to the speed-up of using superpixels it is possible to evaluate many clusterings for one image with different parameters in a reasonable amount of time. This includes, among others, testing a varying number of mixture components. However, in the given setting, the ground truth information is not available any longer. Therefore, it can not be directly assessed which segmentation is best given an image. To circumvent this issue, we propose to learn the relationship between the computed metrics and the model parameters from a set of training images. Specifically, we use the likelihood of the models with a varying number of mixture components and the number of used parameters as features. We restrict the possible range of mixture components from two to twenty. This approach can be thought of as an advanced version of the BIC (see Section 3.3) which uses the same features to predict the number of components. However, the BIC assumes that the model assumption is correct, but this assumption is not valid when dealing with images of everyday scenes (cf. Section 4.1). This leads to the result that the BIC favours models with many mixture components. This is visualized in Figure 5.2, where for an exemplary image the BIC and the VoI are computed for a varying number of mixture components. It is clearly evident that the optimal solution with respect to VoI and BIC widely differ.

In order to generate a mixture model with an appropriate number of classes we propose the following procedure. First, the parameters of a GMM with $\mathcal{K} = \{2, 3, \ldots, 20\}$ components are computed for every image of the training set. Afterwards, the resulting likelihood and the number of free parameters are stored in a feature matrix, and the segmentation metrics for every image are computed from the ground truth annotations and stored as dependent variables. Both quantities—the features and the dependent

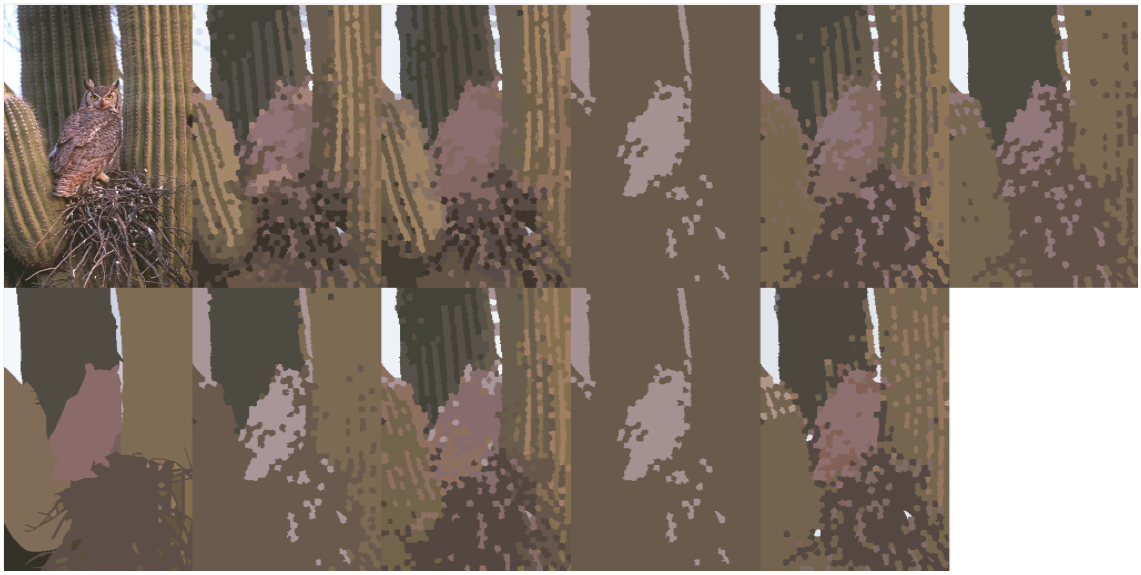


Figure 5.3: Qualitative results of the analysed methods to choose a number of mixture components, applied to an image of the BSDS500. From top left to bottom right: image, AIC, BIC, DBI, SH, CH, ground truth, PR, GP-PRI, GP-VoI, Dirichlet Process Mixture Model (DPMM).

variables—are then used to train a regression model in order to learn the relation between them.

A Gaussian Process (GP) [RW06] is chosen for this task. Broadly speaking, a GP is a distribution over functions and can be seen as a further generalization of a Gaussian distribution to the domain of continuous functions. Mathematically, the relation of a random function Y and a GP is expressed as

$$Y \sim \mathrm{GP}(m, A). \tag{5.5}$$

The parameters of the covariance function A and the mean function m are commonly learned from data. An overview and further details about GP are given in [RW06]. We use a squared exponential covariance function and estimate the parameters of the Gaussian process model with the training set of the BSDS500. We train two separate GPs, one with the PRI as the dependent variable and one with the VoI as the dependent variable, because the measures often deviate in selecting an optimal number of mixture components. Both measures are standard metrics on the BSDS500 (cf. Section 2.10.1 and Section 2.11). The results of this approach are presented in Table 5.2 in the rows titled "GP" (Gaussian process regression).

5.2.3 *Infinite Mixture Models*

In contrast to the previous approaches, which either originate from the clustering literature or are based on learning a functional relation between the images and the number of mixture components, we additionally review the Dirichlet process presented in Section 3.5.1. The resulting DPMM is classified as a statistical method, because it relies on a fully probabilistic formulation of the mixture estimation problem.

	BSDS500								
	metrics			region sizes [px]			# mixture components		
approach	PRI ↑	C ↑	VoI ↓	5%	$\tilde{x}$	95%	5%	$\tilde{x}$	95%
AIC	0.746	0.265	3.156	1 716	6 857	16 691	19	20	20
BIC	0.747	0.275	3.094	1 911	7 455	17 559	15	20	20
DBI	0.703	0.496	2.284	4 641	21 138	102 627	2	3	13
SH	0.699	0.464	2.387	4 040	20 625	89 097	2	3	13
CH	0.708	0.427	2.515	2 432	12 263	69 193	2	6	18
PR	0.747	0.376	2.665	2 571	11 685	40 062	3	9	20
GP-PRI	0.760	0.394	2.607	3 792	15 071	41 720	4	8	18
GP-VOI	0.709	0.500	2.293	16 305	45 618	97 173	2	3	4
DPMM	0.784	0.482	2.391	108	9 445	52 920	7	10	13
GT	-	-	-	1	729	39 236	3	12	60

Table 5.2: Results of selecting an appropriate number of mixture components. For a quantitative assessment of the results the three standard metrics of the BSDS500 are presented. Additionally, the average region size and the average number of mixture components are presented. For a better assessment of the variability of the approaches, the 5 and 95 percent quantile are also presented. The last row presents the quantities of the ground truth (GT).

In comparison to the regression based approaches, the DPMM does not use a labelled set of training images, but can directly be applied to an unseen image and is, therefore, more in line with the primary goal of this thesis to lessen the necessity of labelled data.

5.2.4 *Results*

The results of the different approaches are summarized in Table 5.2. Qualitative results are presented in Figure 5.3. The GMM is chosen as the mixture model for every experiment and the parameters of the mixture models are estimated with EM if not stated otherwise.

As previously mentioned, the Akaike Information Criterion (AIC) and the BIC both overestimate the number of mixture components and almost always choose the maximal number of clusters. This overestimation is reflected in the high values of the VoI metric. However, the AIC and BIC are among the top performing methods regarding the PRI which does not punish over segmentation as strongly as the VoI. The remaining cluster validation criteria—Davies-Bouldin index (DBI), silhouettes (SH), and Charlinksy-Harabsz (CH)—underestimate the number of mixture components on average. This leads in term to large regions on average and is especially visible for the DBI in Figure 5.3.

The regression based approaches perform on average very well. Surprisingly, the approach based on Poisson regression performs very well, even though it simply looks at image statistics. In contrast to the GP approaches, it is independent of the chosen mixture model and, therefore, has never seen a segmentation by the model. The advanced version of the BIC with the regressed relation between segmentation metrics and model properties performs better than the BIC. Therefore, the primary goal of having a more suitable metric is fulfilled. However, in the case of learning to predict the VoI, the number of mixture components is again underestimated. This leads to the conclusion that the VoI is easily fooled by a low number of mixture components, although it is considered as one of the most sound metrics. In general, however, it appears to be difficult to learn a relation between the model properties and the expected values of the metrics.

In comparison to the ground truth (GT) the DPMM appears to be well suited in terms of regions sizes and number of mixture components. Additionally, the achieved PRI is best among the considered approaches. Segmentation covering and VoI are among the best performing, too. While the competing methods all are estimated with EM, the DPMM estimates the number of mixture components in a fully Bayesian fashion. The model itself—a GMM—is equal for every method. In summary, the Bayesian interpretation appears to be beneficial in terms of segmentation metrics. In the following, the DPMM is chosen as the method to predict the number of mixture components, because it shows excellent performance and is in line with the initial requirements defined in Chapter 2 of using as few expert knowledge as possible.

5.3 MODEL DEFINITION AND PARAMETER ESTIMATION

After defining the used feature set, adequately reducing the computational burden, and analysing the best way to estimate the number of mixture components in the given setting, the final model can be defined and a way to estimate the parameters can be developed.

Since one of the primary goals of this thesis is to increase the flexibility of the used mixture models, we need to switch the parameter estimation scheme. Although an estimation with EM is possible, it requires a significant amount of analytical work when changing the mixture model. Instead, we estimate the parameters of the model with Markov chain Monte Carlo (MCMC).

One of the major issues of using MCMC in the setting of multivariate mixture models is ensuring that the covariance matrices of the distributions are positive definite. Recall, MCMC iteratively proposes new values based on the previous estimate. In the case of covariance matrices, it has to be ensured that the proposed covariance is positive definite. Therefore, it is not enough to simply modify the entries of the matrix. In the previous chapter, this issue has been solved by manifold optimization. Since we want to use MCMC in this chapter to benefit from the Bayesian framework, a different approach has to be used.

When dealing with a GMM, the Wishart distribution [Wis28], which is the conjugate prior of the covariance matrix of a normal distribution, can be used to ensure that only valid covariance matrices are proposed. If, however, a different distribution is used, like the multiple scaled t-distribution and sinh-asinh distribution in this thesis, conjugacy is not available. In such cases, statistical frameworks like Stan [Car+17] propose to separate

the covariance matrix into a matrix of standard deviations and a correlation matrix. This is similar to the copula approach presented in Section 3.6. However, this time it is not used to modify the flexibility of the distribution, but to place a suitable prior over the covariance matrix. Commonly, the LKJ-prior [LKJ09] is placed over the correlation matrix and the entries of the matrix of standard deviations are modelled by gamma distributions.

In this thesis, we explore another possibility. In general, every square matrix can be decomposed into an eigenvector and eigenvalue representation by:

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}, \tag{5.6}$$

with $\mathbf{V}$ as the matrix of eigenvectors $\mathbf{v}_i$ and $\boldsymbol{\Lambda}$ as a diagonal matrix of the corresponding eigenvalues λ_i. If $\mathbf{A}$ is a symmetric matrix, like a covariance matrix, Equation 5.6 simplifies to

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{T}. \tag{5.7}$$

In this case, the matrix of eigenvectors $\mathbf{V}$ is an orthogonal matrix, that is, it describes a rotation of the feature space. Following this line of thought, proposing a new covariance amounts to rotating the eigenvectors and scaling the eigenvalues. Note that in this setting the eigenvalues are always real and positive. Therefore, a gamma distribution is a suitable choice as prior distribution.

While proposing new values for the eigenvalues is straightforward, proposing new eigenvectors, that is, rotating them, is not. In general, an n-dimensional rotation matrix is defined as [DB94]

$$\mathbf{R}_{kl}(\psi) = \left[\begin{array}{c|lcll} & r_{ii} & = & 1, & i \neq k, j \neq l \\ & r_{ll} & = & \cos\psi, & \\ & r_{kk} & = & \cos\psi, & \\ r_{ij} & r_{kl} & = & -\sin\psi, & \\ & r_{lk} & = & \sin\psi, & \\ & r_{ij} & = & 0, & \text{elsewhere} \end{array}\right] \tag{5.8}$$

with k and l as the axes of the feature space which define the rotation. Note that in the case of four or more dimensions a rotation is defined with respect to a plane. In the simplest case, which is defined above, the plane is defined by the unit vectors of the space. A rotation of the whole space is achieved by multiplying the rotation matrices of all $N_R = \binom{D}{2}$ possible axis pair combinations.

$$\mathbf{R}(\boldsymbol{\psi}) = \prod_{i=1}^{N_R} \mathbf{R}_i(\psi_i). \tag{5.9}$$

During inference, $\mathbf{R}(\boldsymbol{\psi})$ can then be used to propose a change in the orientation of the eigenvectors. Either by rotating the whole space or independently as in componentwise Metropolis-Hastings by rotating around a single plane. The new eigenvectors $\mathbf{V}'$ are computed by

$$\mathbf{V}' = \mathbf{R}(\boldsymbol{\psi})\mathbf{V}, \tag{5.10}$$

with ψ as a vector of random rotation angles. In combination with a new matrix of eigenvalues Λ' the relation described in Equation 5.7 is exploited to propose a new covariance matrix Σ' by

$$\Sigma' = \mathbf{V}'\Lambda'\mathbf{V}'^{\mathsf{T}}. \tag{5.11}$$

As long as the entries of Λ' are real and positive, and $\mathbf{R}(\psi)$ is a valid rotation matrix, a positive definite matrix is created.

In the following, two models are tested—the multiple scaled t-distribution and the sinh-asinh distribution with a Gaussian copula. The multiple scaled t-distribution is tested because it has been used in the published work [WW17c; WW17a], albeit it does not increase the performance significantly (see Table 4.4). In the following, this model is termed Multiple Scaled t-distribution Mixture Model (MTMM). The combination of a Gaussian copula and the univariate sinh-asinh distribution is tested, because it has shown superior performance in the previous experiments (see Table 4.4). The resulting model is termed Sinh-asinh Copula Mixture Model (SCMM).

The following list depicts important relations, the mixture model itself, the used prior distributions, and the hyperparameters of the prior distribution if the SCMM is used.

$$\mathbf{X}_{j=1,\dots,N,\cdot} \sim \sum_{k=1}^{K} \omega_k \left[c_k\left(\mathbf{X}_{j,\cdot}|\mathbf{C}_k\right) \prod_{d=1}^{D} \mathrm{Sha}\left(X_{j,d}|\mu_{k,d}, \sigma_{k,d}, \delta_{k,d}, \epsilon_{k,d}\right) \right],$$

$$\begin{aligned}
\boldsymbol{\omega} &\sim \mathrm{Dirich}(\cdot|\boldsymbol{\beta}),\\
\mathbf{z}_{i=1,\dots,N} &\sim \mathrm{Cat}(\boldsymbol{\omega}),\\
\mathbf{C}_k &= \mathbf{V}_k\Lambda_k\mathbf{V}_k^{\mathsf{T}},\\
\Lambda_k &= \mathrm{diag}(\lambda_{k,1},\dots,\lambda_{k,D}),\\
\mathbf{V}_k &= [\mathbf{v}_{k,1}\ \mathbf{v}_{k,2}\ \dots\ \mathbf{v}_{k,D}],\\
\mathbf{v}_{k,d} &\sim \mathrm{Normal}(\cdot|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\\
\lambda_{k,d} &\sim \mathrm{Gamma}(\cdot|a_0, b_0),\\
\mu_{k,d} &\sim \mathrm{Normal}(\cdot|\mu_0, \sigma_0),\\
\sigma_{k,d} &\sim \mathrm{Gamma}(\cdot|a_0, b_0),\\
\delta_{k,d} &\sim \mathrm{Gamma}(\cdot|a_0, b_0),\\
\epsilon_{k,d} &\sim \mathrm{Normal}(\cdot|\mu_0, \sigma_0),\\
\boldsymbol{\beta} &= [1,\dots,1]^{\mathsf{T}},\\
\boldsymbol{\mu}_0 &= [\mu_0,\dots,\mu_0]^{\mathsf{T}},\\
\boldsymbol{\Sigma}_0 &= \mathrm{diag}(\sigma_0,\dots,\sigma_0).
\end{aligned}$$

$$\mu_0 = 0, \sigma_0 = 100, a_0 = 0.001, b_0 = 0.001,$$

As already mentioned, we choose a mixture of univariate sinh-asinh distributions and a Gaussian copula for modelling the dependence structure as the model for the observed image data. The prior on the mixture weights is a Dirichlet distribution with a non-informative hyperparameter choice. Additionally, the probabilistic formulation of the latent variables $\mathbf{z}$ is also included (see Section 3.2.4). The presented eigendecomposition of the covariance matrix is applied to the correlation matrix of the Gaussian copula. The eigenvectors itself are modelled by a multivariate normal distribution, and the eigenvalues, which have to be positive, by a gamma distribution. Again, the hyperparameters are chosen such that the priors are vague. The parameters of the sinh-asinh distribution are modelled by suitable priors which reflect the range of possible values.

When using the multiple scaled t-distribution, the model itself is changed to a MTMM. Therefore, the model is no longer able to model skewed data. Since the multiple scaled t-distribution is parametrized through eigenvalues and eigenvectors instead of a covariance matrix, the same priors are used for them. On the elements of the vectors $\boldsymbol{\nu}_{k,d}$ which model the degrees of freedom in every dimension a gamma distribution is placed as a prior distribution because the entries have to be positive.

As mentioned earlier, the parameters of the mixture models are estimated with MCMC. This enables us to compute uncertainty estimates of the parameters and monitor the convergence behaviour conveniently. Further, MCMC has—in theory—a guarantee to reach an optimal configuration and is able to bypass local minima during parameter inference. In this thesis, we use a special variant of a Markov chain, that is, Delayed Rejection Adaptive Sampling (see Section 3.4.3).

The main reason behind using Delayed Rejection Adaptive Sampling (DRAM) is to speed up the parameter estimation. One of the main challenges of using Metropolis-Hastings is to design a suitable proposal distribution. One of the easiest ways to lessen the issue is to sample component-wise (see Section 3.4.2) because then the chance of accepting a proposal solely depends on the change of a single parameter. However, this leads to a costly evaluation of the likelihood at every single parameter update. Over the course of iterations, this becomes expensive and slows down the inference. If, however, all parameters or groups of parameters are proposed to be updated at once, the chance of accepting such a proposal is lower. Mainly, because the values of all updated parameters need to be reasonable. This is for instance achieved by exploiting the correlation of different parameters in the sample space. However, manually designing such an engineered proposal distribution is tedious.

The main benefit of DRAM is its ability to automatically adapt the covariance matrix of the proposal distribution in common Metropolis-Hastings sampling. As a result, the chance of accepting multiple samples at once is increased, because the dependence structure among the parameters can be learned and exploited during inference. Otherwise, the dependence structure of the parameters has to be guessed or a diagonal matrix with small variances has to be chosen such that the proposed values do not deviate strongly from the current estimate. This in turn slows down parameter inference as well (cf. Figure 3.6).

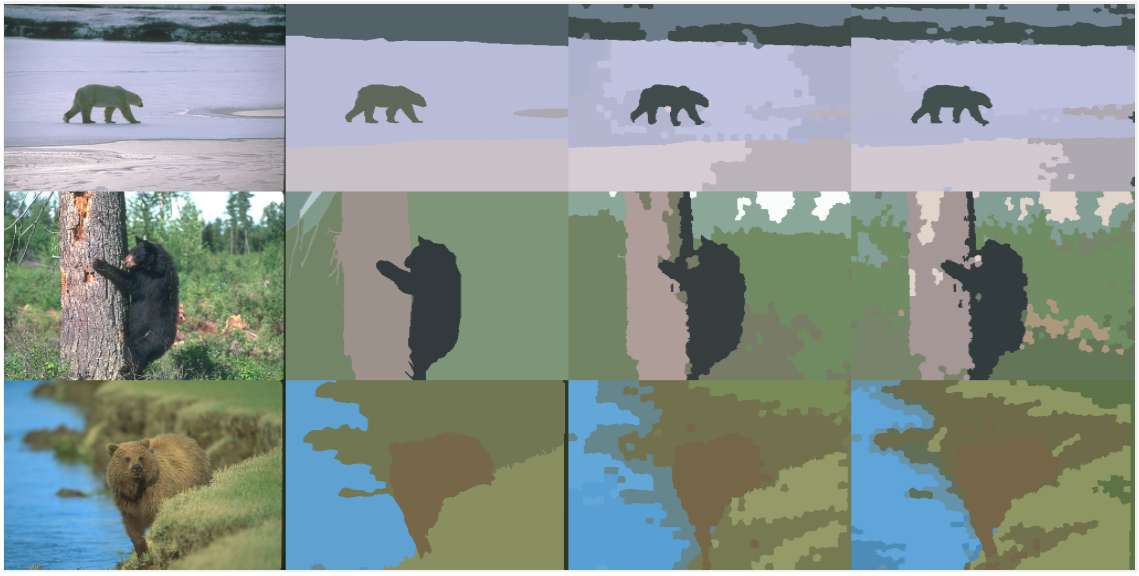


Figure 5.4: Exemplar segmentations computed from a superpixel representation, using images of the BSDS500. From left to right: image, ground truth segmentation, segmentation computed by an MTMM, and by an SCMM. In both cases, a DPMM is used to estimate the number of mixture components.

With the help of DRAM, the costly likelihood evaluations do not have to be performed after changing a single parameter, but after changing *all* parameters or at least a *group* of parameters and keeping the chance of accepting a proposal reasonably high while exploring the parameter space.

For instance, an eight-component mixture of the SCMM and a six-dimensional feature space yields $8 \cdot 46 = 368$ free parameters. Instead of evaluating the likelihood at least 368 times per iteration, it is only evaluated eight times, because the parameters of every mixture component are updated at once with the help of K independent proposal distributions—one for every mixture component. The proposal distributions are multivariate normal distributions from which random draws can be created conveniently. The covariance matrices of the proposal distributions are updated with the help of the already drawn samples. Therefore, a speed-up is achieved while maintaining a high chance of proposing reasonable samples.

5.3.1 *Accuracy Assessment*

Qualitative results of the proposed mixture models are presented in Figure 5.4 and quantitative results are summarized in Table 5.3. On a first glimpse the results of using a multiple scaled t-distribution and predicting the number of mixture components with a Gaussian process and the results of using the proposed SCMM and a DPMM are comparable and the observed errors equal. On a closer look the more flexible sinh-asinh model adheres better to the true boundaries of the segmentations. However, this comes at the cost of an oversegmentation of the true regions. The reason for this behaviour is the unsupervised estimation of the number of mixture components. Since the observed image data matches the model assumption only to a certain degree, the mismatch is

	BSDS500	
	PRI ↑	VoI ↓
GP + GMM	0.82	2.20
GP + MTMM	0.82	2.22
DPMM + GMM	0.80	2.35
DPMM + SCMM	0.80	2.22

Table 5.3: Evaluation of the segmentation accuracy of different algorithms on the BSDS500. Probabilistic Rand Index (PRI) and Variation of Information (VoI) are presented. The mixture models where the number of components is predicted by a trained Gaussian process (GP) are on average better than the DPMM based approaches. However, the use of the SCMM greatly improves the GMM baseline if the case of the VoI criterion.

compensated by an overestimation of the number of mixture components. As a result, boundaries are created although the transition of colour values is smooth, but probably not well described by the DPMM used to determine the number of mixture components.

The combination of the multiple scaled t-distribution and the Gaussian process produces fewer mixture components, which is closer to the true number of regions, but the visual inspection reveals that the region boundaries do not match the true boundaries well. The Gaussian process has learned to compensate the model's imperfections to a certain degree and predicts a lower number of regions than the unsupervised DPMM. However, the lower flexibility of the multiple scaled t-distribution compared to the sinh-asinh copula model limits the achievable segmentation accuracies.

In terms of quantitative evaluation, the supervised approach produces better scores than the unsupervised variants based on the DPMM, although the absolute differences are small. Surprisingly, the proposed SCMM performs better than the surrogate model—the GMM—used to determine the number of mixture components. Prior to the analysis, this could not be taken for granted, because the increased flexibility of the model might not be beneficial. If the model is solely estimated on the likelihood of the observed image data, the increased flexibility does not have to coincide with a better segmentation with respect to the ground truth. However, in the case of image segmentation, it appears to be true and the proposed SCMM performs better. As a result, we can conclude that the model assumption itself is one of the most limiting factors in generative image segmentation and the proposed SCMM is not only the best choice in the supervised setting but also in the unsupervised setting. However, the overestimation of the number of mixture components in the unsupervised setting and the low flexibility of the multiple scaled t-distribution hinder the segmentation accuracies strongly in both cases.

Note that the differences in the computed results presented in Table 5.2 and Table 5.3 regarding the Gaussian process for estimating the number of mixture components are ascribed to the change in distribution and different parameter estimation. After selecting the number of mixture components with the Gaussian process, the parameters of the mixture model are estimated with MCMC by using the predicted mixture model as an initialization. This approach appears to be beneficial because the segmentation accuracies are increased due to a better exploration of the parameter space. This in turn

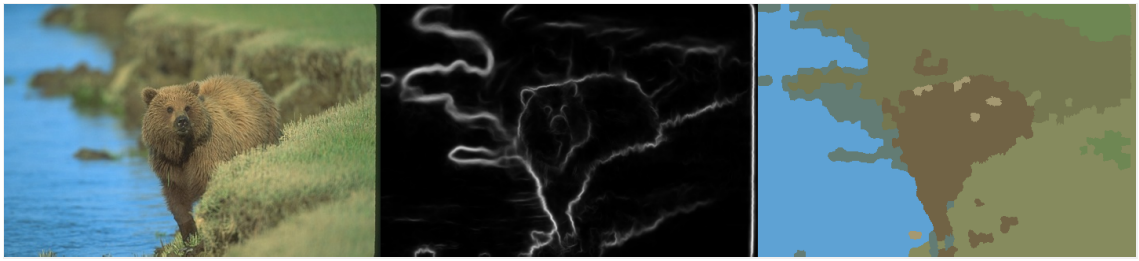


Figure 5.5: Visualization of the idea to include edges as a part of the segmentation model, using an image of the BSDS500. The goal is to reduce the occurring segmentation noise, and to improve the segmentation by shifting the boundaries to high edge values. From left to right: image, edges, and segmentation computed by a DPMM.

hints at a parameter space which has multiple local minima which can easily fool EM and are better explored by MCMC which allows solutions that are worse than the current estimate. In the given setting, the better adaptation to the parameter space is also ascribed to the use of DRAM.

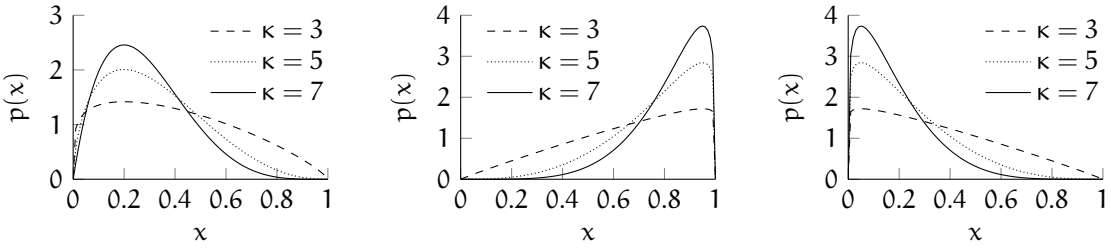
5.4 INCLUDING EDGES AS A BOUNDARY PRIOR

Edges in an image mark areas where the values of the first derivative exceed a certain threshold. Therefore, they hint at a change in intensity or any other type of feature from which the derivative is computed. This includes, among others, texture and depth. Note that, in computer vision, the derivatives are computed with respect to the spatial position of a pixel (see Section 2.2 for further details).

In general, computing the gradient follows the assumption, that regions are described by a smooth distribution of values, which abruptly changes when the regions change. Therefore, the magnitudes of the gradient are small within a region and large at region boundaries. If we look at the image presented in Figure 5.5, it is evident that the computed edges mark the borders of the regions reasonably well. Note that, the edges are computed by the method presented in [DZ15]. However, the borders of the segmentation computed by the DPMM do not match those boundaries completely (see Figure 5.5).

The key idea of the following paragraphs is to create a model where both cues are combined into one probabilistic model and exploit the boundary information to improve the quality of the segmentation model. The edges of an image are used as a weak condition for the mixture model to place its region boundaries at the same positions. In terms of mixture models, the model aims to place high uncertainties in those regions. Note that, when mixture models are used in image segmentation, the boundaries are only an indirect result after computing the responsibilities (cf. Equation 4.6).

In this section, two ways of including edges within a probabilistic image segmentation framework are proposed. One variant is a Passive Edge Model (PEM) which grades the segmentation according to a precomputed edge map. The second variant uses this method in conjunction with an Active Edge Movement (AEM) scheme, where the boundary between two segments is actively shifted. The former rates a change in the segmentation according to an edge model, whereas the latter actively proposes to modify the current boundaries of the segmentation beneficially. Note that, every pixel has its own edge model which is independent of the neighbouring pixels.



(a) $\rho = 0.2$ (b) $\rho = 0.95$ (c) $\rho = 0.05$

Figure 5.6: Influence of choosing an appropriate mode for the beta distribution of the edge model. If the mode ρ is chosen as a value of the edge map (e. g., $\rho = 0.2$), too much probability mass is concentrated near zero (a), which means a low probability of an edge. This misbehaviour can be circumvented if the mode is chosen by the current segmentation.

5.4.1 *Passive Edge Model*

The proposed PEM weights each resulting region border in the current segmentation of the Markov chain according to a pre-defined model distribution. The definition of this model is described in the following paragraphs.

In general, the edge model should fulfil three requirements. First, it should give a high probability to data points where an edge is probably present and this edge is captured by the current segmentation. Second, a high probability should be given to data points where an edge is unlikely and this absence is captured by the current segmentation. And last, a low probability is given to those parts where the current segmentation does not match the probable edge map.

In order to obtain an edge map $\mathbf{E}$, the method provided in [DZ15] is used. The values of the resulting edge map reside in the range of $x \in [0, 1]$. A natural choice in this domain is the beta distribution (see Section 3.2.2.6). Recall, the probability density function (pdf) equals

$$p(x|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}. \tag{5.12}$$

Since the parameters α and β of the beta distribution do not allow an easy interpretation, a different parametrization can be chosen where the distribution is parametrized by the mode, that is, the point with the highest density and a concentration parameter which reflects the spread of the distribution. The mode of the distribution ρ equals

$$\rho = \frac{\alpha-1}{\alpha+\beta-2},$$

and the concentration parameter equals

$$\kappa = \alpha + \beta.$$

It is then possible to use a different parametrization of Equation 5.12 where the parameters α and β are expressed in terms of the mode ρ and the concentration parameter κ:

$$\alpha = \rho(\kappa-2)+1,$$
$$\beta = (1-\rho)(\kappa-2)+1.$$

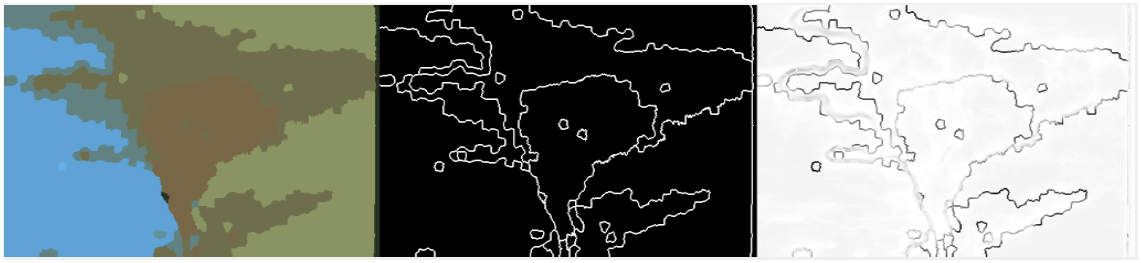


Figure 5.7: Visualization of the used quantities to build the PEM. From left to right: current segmentation $\mathcal{S}^{(t)}$, current boundary map $\mathbf{B}^{(t)}$, and likelihood of the PEM at every pixel.

With the help of this parametrization, it is easier to define an appropriate edge model because the values of α and β can now be chosen with respect to the mode of the distribution.

Due to the flexibility of the beta distribution the edge model can now be built according to the aforementioned requirements. In general, the beta distribution can depict a range of different shapes (cf. Figure 3.4b). The influence of the parameter choices in the context of this chapter is depicted in Figure 5.6. In contrast to common practice, the input of the edge model is not the current segmentation, but the edge map $\mathbf{E}$ itself. The reason for this is explained in the following.

With the help of the current segmentation $\mathcal{S}^{(t)}$ the resulting boundary mask $\mathbf{B}^{(t)}$ at iteration t is computed. Note that the entries of $\mathbf{B}^{(t)}$ are binary. Therefore, it is rather difficult to compare the edge map $\mathbf{E}$, where all values are from the unit interval, directly with $\mathbf{B}^{(t)}$. As can be seen in Figure 5.5 not all edges are equally probable and not all pixels around an edge are equally likely an edge.

If an edge has a low probability, say around 0.2, and a beta distribution with mode $\rho = 0.2$ and a reasonably high concentration is placed at this edge, the corresponding probabilities will be low (see Figure 5.6a) because $\mathbf{B}^{(t)}$ is binary and the resulting value is one if an edge is present at this position. Therefore, it would be highly improbable that an edge is accepted at this position, which contradicts the prior belief of 0.2. Since this behaviour is undesirable, $\mathbf{E}_{j,i}$ is not a good choice as the mode of the edge model at pixel coordinates (j, i). However, we define the edge model the other way round. To stick with our example of a weak edge with probability 0.2, we now choose the value of the current boundary $\mathbf{B}^{(t)}_{j,i}$ as the mode of the distribution, which results in the distributions depicted in Figure 5.6b and Figure 5.6c.

If $\mathbf{B}^{(t)}_{j,i} = 0$, a high probability score can be achieved. However, if $\mathbf{B}^{(t)}_{j,i} = 1$, a lower, but still, a reasonable amount of score can be gained by setting this pixel to an edge pixel. Therefore, $\mathbf{B}^{(t)}_{j,i}$ is chosen as the mode and the concentration parameter is set empirically to $\kappa = 2.5$. This model is termed passive edge model and is used in conjunction with a mixture model in order to compute the posterior distribution of the segmentation. While the proposed approach can be used in conjunction with any mixture model, in this thesis the MTMM and SCMM are studied. The results are summarized in Table 5.4.

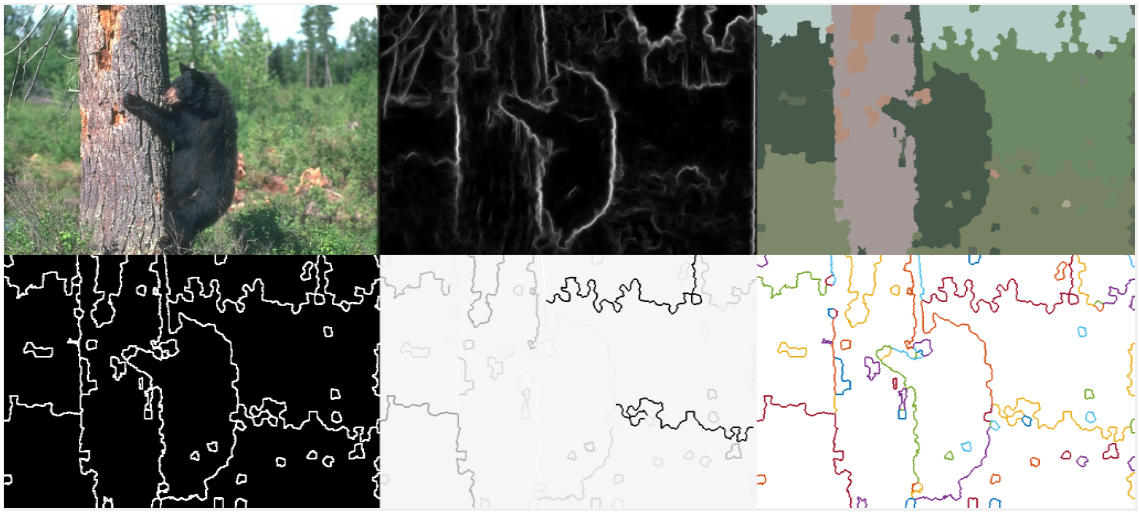


Figure 5.8: Visualization of the idea to include edges as a part of the segmentation model, using an image of the BSDS500. The goal is to reduce the occurring segmentation noise, and to improve the segmentation by shifting the boundaries to high edge values. From top left to bottom right: image, edge map $\mathbf{E}$, current segmentation $\mathcal{S}^{(t)}$, corresponding boundary map $\mathbf{B}^{(t)}$, likelihood of individual boundaries, and visualization of the set of line segments $\mathcal{L}_{\text{seg}}$.

5.4.2 *Active Edge Movement*

While the use of a passive grading is beneficial (see Section 5.4.3 and Table 5.4), one can easily assume cases where a simple grading leads to a segmentation where a model is stuck, because it can not move multiple parts of a boundary at once. Of course, this movement may happen by chance, but with an increasing length of the border, it becomes exponentially more unlikely. Therefore, we additionally propose a variant—the Active Edge Movement—which shifts parts of the boundaries at once. As a result, an AEM should speed up the convergence and allow the Markov chain to surpass local minima, where the passive edge model may get stuck because it is not probable that complete borders are moved by chance. Additionally, the convergence speed of the algorithm is supposed to rise, because whole boundaries can be moved at once.

In order to build such a movement scheme, it has to be defined what is meant as a boundary and how we propose to move it such that the proposed movement integrates into the current framework. As a first step, each appearing line in $\mathbf{B}^{(t)}$ gets an appropriate unique label for identification (see Figure 5.8 for a visualization). A line is termed unique if it is connected by a four-point neighbourhood and the adjacent labels of the line are equal. As a result, the boundary map $\mathbf{B}^{(t)}$ is divided into a set of line segments $\mathcal{L}_{\text{seg}} = \{\mathcal{L}_1, \ldots, \mathcal{L}_{N_L}\}$ which can be uniquely identified.

After labelling, a score for each line can be computed from the passive edge model by summing over the log likelihood of the associated pixels. The scores are normalized and transformed to the unit interval, in order to randomly select one boundary based on its inverse score. After selection, it will be used to propose a change in the current segmentation. Therefore, lines with the lowest scores are proposed to be updated more frequently in the Markov chain.

After a border is selected, two new segmentations are proposed. Since the current border has two adjacent labels A and B, the first variant sets the adjacent superpixels of the border to A and the second variant to B. Therefore, the border is either shrunk or expanded. Finally, the posterior distributions of the resulting segmentations are computed, the best variant is selected, and then accepted or declined in a common Metropolis-Hastings acceptance step based on the combined likelihoods of the mixture model and the edge model. Results of using this approach are summarized in Table 5.4.

5.4.3 *Accuracy Assessment*

In order to test, the potential benefit of including edges into the probabilistic model, the model is tested on the BSDS500. For all experiments, 1500 superpixels are computed by SLIC and the most probable point within every superpixel is used to describe this superpixel. This amounts to a reduction in the computational burden of roughly 99% on the BSDS500. Moreover, the position and the colour values in the Lab colour space are used as features. Two different mixture models are tested. First, a mixture of multiple scaled t-distributions as a trade-off between model flexibility and complexity, and second, the proposed Sinh-asinh Copula Mixture Model (SCMM) as a maximal flexible variant. In both cases, the parameters of the mixture model are estimated with DRAM and the number of mixture components is determined by a DPMM.

Qualitative and quantitative results of the proposed segmentation methods are presented in Figure 5.9 and Table 5.4. From a visual inspection, the computed segmentations are very good and closely resemble the ground truth segmentations. Note that, the estimation of the model parameters is performed in an unsupervised way. Depending on the characteristics of the image, occasional single superpixel regions can still be observed. However, the visual quality is greatly improved when compared to the results without an edge model in Figure 5.4.

Quantitative results are presented in Table 5.4. DPMM indicates the performance of the initialization without an edge model, and PEM and AEM are the results with the respective edge models included. Note that, by using the AEM the PEM is always included as well. Including the probable location of an edge is beneficial in every case and significantly improves the metrics in all cases. Simply including the passive grading of the computed boundaries in every iteration already leads to a significant improvement. However, as noted previously, the passive grading alone might converge slowly or get stuck faster in a configuration. Therefore, additionally including an active movement scheme again increases the performance significantly. By visually analysing the computed results of both approaches it can often be observed in the computed segmentation that regions with the same region label are disconnected. The model has then found a solution, where the colour appearance across different parts of the image is comparable. This happens for instance when background regions are occluded by objects. In the ground truth segmentations, these regions are then split into different regions because they are not connected any longer. In order to account for this behaviour, we additionally perform a post-processing where the region labels are generated anew. Connected component analysis is performed on every region label. If the regions are disconnected, they receive new unique labels. As a result, performance can be increased further. Note that the boundaries of the post processed variant remain the same and only the region labels

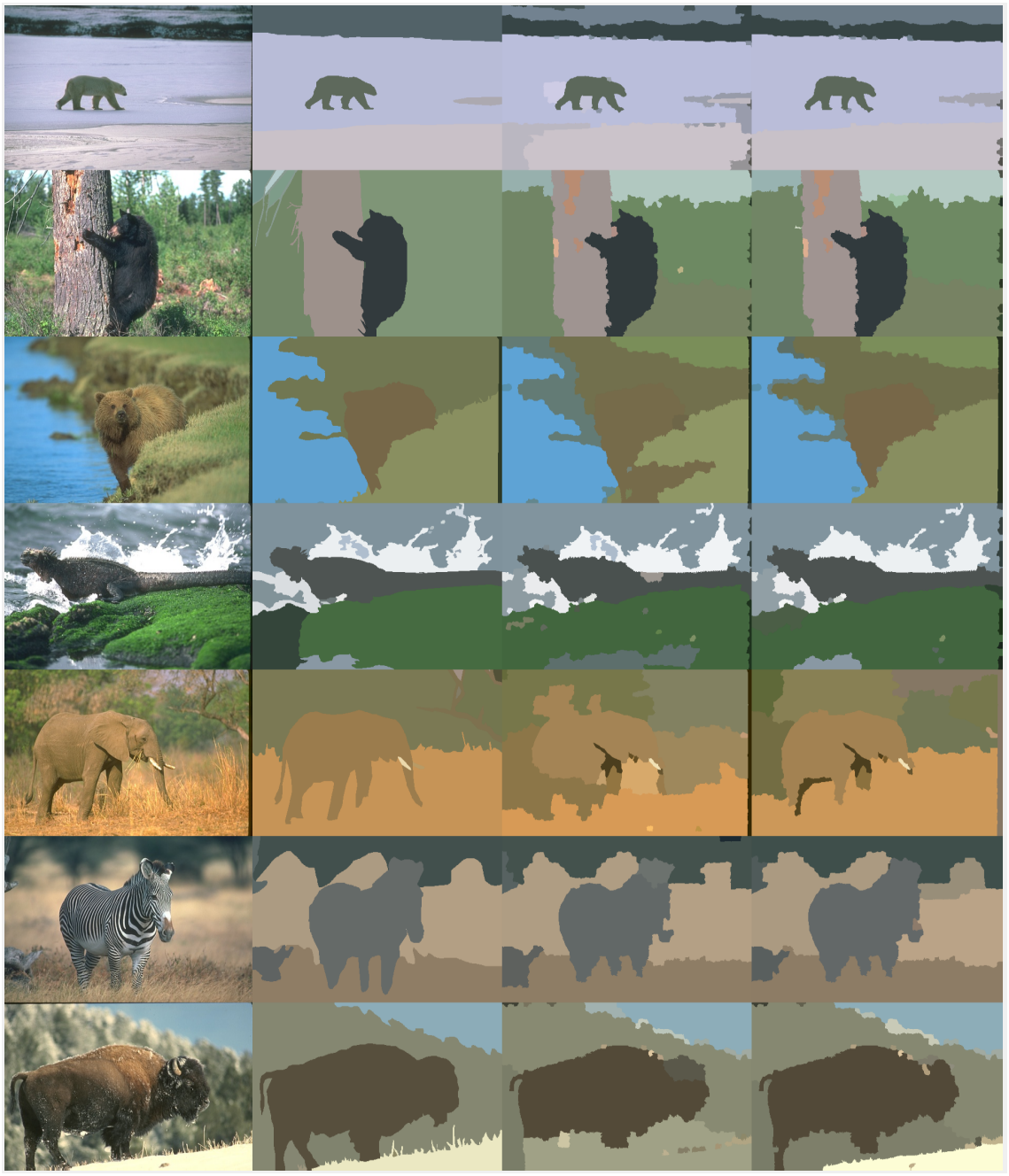


Figure 5.9: Qualitative results of the proposed models including the probable locations of edges on the BSDS500. From left to right: image, ground truth, SCMM + PEM, SCMM + AEM.

are changed. In the next section, the proposed model is compared to the state-of-the-art approaches in image segmentation.

5.5 COMPARISON TO OTHER APPROACHES

After introducing the proposed generative framework, we compare it to competing methods from the literature. The results are summarized in Table 5.5 where the ap-

	BSDS500		
	PRI ↑	C ↑	VoI ↓
DPMM	0.80	0.51	2.35
MTMM + PEM	0.83	0.54	1.94
MTMM + PEM + post processing	0.84	0.55	1.88
MTMM + AEM	0.83	0.55	1.90
MTMM + AEM + post processing	0.84	0.54	1.87
SCMM	0.80	0.51	2.22
SCMM + PEM	0.82	0.53	1.92
SCMM + PEM + post processing	0.83	0.54	1.83
SCMM + AEM	0.85	0.58	1.73
SCMM + AEM + post processing	0.85	0.59	1.67

Table 5.4: Evaluation of the proposed edge models on the BSDS500. Including the probable location of edges is beneficial and increases the performance. Again, the Sinh-asinh Copula Mixture Model shows the best performance. In all cases, the accuracies can be further increased by a simple post processing step. See text for additional details.

proaches are divided into different groups depending on the characteristics of the methods. The groups are separated by horizontal lines. While some methods are based on a region based interpretation, like the methods proposed in this thesis, the majority of approaches is based on detecting contours and building a hierarchy of segmentations from the detected contours. These hierarchies of closed contours are then thresholded at learned levels to derive a segmentation and are summarized in the first and second group. The approaches in the first group are based on OWT-UCM presented in [Arb+11] (see also Section 2.6). The second group is based on multiscale combinatorial grouping proposed in [Arb+14]. The latter is considered as the current state-of-the-art.

The third group are other approaches which are not based on contour detection, but on a region based analysis comparable to our approach. The last group contains the methods presented in this thesis which are based on generatively modelling the observed image. Unfortunately, none of the competing methods from the literature fit this category directly. Often, these methods do not evaluate their models comparably. For example, the work presented in [Sfi+10] proposes a generative model for image segmentation, but they only evaluate their approaches with a fixed number of mixture components for every image. Naturally, this drastically degrades the possible performance of the segmentation metrics. However, a comparison between this method and a simple GMM has been presented in Table 4.2 to evaluate the impact of including positional features.

The last row depicts the upper bound of the SCMM. Note that, the upper bound is different from the values presented in Table 4.4. In the previous case, a segmentation has been computed from all available ground truth annotations. In order to allow for

	BSDS500		
	PRI ↑	C ↑	VoI ↓
Canny-owt-ucm [Arb+11]	0.79	0.49	2.19
W-Net+ucm [XK17]	0.82	0.59	1.67
gPb-owt-ucm [Arb+11]	0.83	0.59	1.69
cPb-owt-ucm [KLL13]	0.83	0.59	1.65
PFE-owt-ucm [YFL15]	0.83	0.61	1.64
RGC-SE+ucm [Zha+18]	0.83	**0.62**	1.59
RGC+MCG [Zha+18]	0.84	**0.62**	1.57
PFE+MCG [YFL15]	0.84	**0.62**	**1.56**
Spectral Cut [Tan+18b]	0.78	0.42	2.34
FBTS [YWC15]	0.79	-	2.10
RFCL [Li+18]	0.79	-	1.89
HFCL [Li+18]	0.78	-	1.93
W-Net [XK17]	0.81	0.57	1.76
ICM [SWW17]	0.82	0.58	1.75
SCMM	0.80	0.51	2.22
SCMM + PEM	0.83	0.54	1.83
SCMM + AEM	**0.85**	0.59	1.67
SCMM (upper bound)	0.91	0.79	1.03

Table 5.5: Comparison of the proposed segmentation models to state-of-the-art approaches on the BSDS500. The first two groups are based on contour detection. While the first group is based on owt-ucm presented in [Arb+11], the second group is based on multiscale combinatorial grouping proposed in [Arb+14]. The latter is considered as the current state-of-the-art. The third group contains other approaches which are not based on contour detection, but on a region based analysis comparable to our approach. The last group contains the proposed models which are based on generatively modelling the observed image data and the last row depicts the upper bound of the SCMM. Best values are marked in bold font. Groups are separated by horizontal lines.

a comparison to the current state-of-the-art in image segmentation, only one segmentation is computed for every image. Afterwards, the metrics of all available ground truth annotations are averaged.

In summary, the proposed generative segmentation model fares very well against the other region-based methods. Spectral Cut [Tan+18b] estimates a segmentation in an unsupervised fashion. However, the number of segments is provided as a supervisory signal. In contrast to our approach, which estimates the number of segments with a DPMM, Spectral Cut performs worse. FBTS [YWC15] is an unsupervised method, but the performance is worse compared to the proposed SCMM with an edge model included. RFCL

and HFCL [Li+18] may be considered as unsupervised, but they contain several hyperparameters which have to be set accordingly. Again, the resulting segmentation metrics are worse compared to the SCMM in combination with an edge model. W-Net [XK17] is trained in a supervised fashion on the segmentation task of the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC) and then applied to the BSDS500. Therefore, W-Net makes heavily use of annotations and has several orders of magnitudes more parameters. However, the generative SCMM with the edge models included performs slightly better. Another disadvantage of all preceding approaches is their lack of quantifying the uncertainty with which a prediction is made, because these approaches are not generative like the proposed approach is.

Further, the presented method is able to outperform a great number of different approaches which are discriminative and trained in a supervised fashion. These approaches are either based on OWT-UCM [Arb+11] or on MCG [Arb+14] and tackle image segmentation from a fundamentally different perspective. They treat image segmentation as a boundary detection task. From the detected boundaries a hierarchy of segments is build by thresholding the boundary strengths at different levels. Therefore, these methods cannot quantify region membership probabilities for every pixel, like the proposed approach does. All these methods have learned in a supervised way how to detect contours, how to build a hierarchy of segmentations from contours, and how to threshold the hierarchy to achieve the best scores.

In comparison to the contour based approaches, the proposed generative method is comparable in terms of the performance, but especially regarding the VoI criterion, not on the same level. However, none of the competing methods is able to provide sensible uncertainty estimates of the segmentation and feature a probability that an observed pixel belongs to a specific region. The proposed framework is built on this reasoning, and we consider it as a major advantage in comparison to the competing methods.

5.6 SUMMARY

In this chapter, a novel probabilistic segmentation model has been presented. In a first step, a way of reducing the computational burden of the demanding parameter estimation has been presented. Superpixels allow reducing the amount of data to 1 % of the original data size and vastly reduce the amount of data needed to compute a segmentation while limiting the degradation of the segmentation accuracies through the subsampling. Additionally, other ways of reducing the computational burden have been explored. Afterwards, different ways to estimate the number of mixture components have been evaluated. The Dirichlet Process Mixture Model is among the best-performing methods in the given setting. It estimates the number of mixture components in an unsupervised way with the help of a probabilistic model and fits therefore nicely in the proposed framework. In the following experiments, the Dirichlet Process Mixture Model is chosen for the estimation of the number of mixture components. This analysis is followed by a complete probabilistic description of the used mixture model—the Sinh-asinh Copula Mixture Model. In the previous chapter, it has been shown that increasing the flexibility of the distribution significantly helps to improve the achievable segmentation accuracies. In this chapter, the results have been transferred to the unsupervised

setting and it has been shown again that the proposed Sinh-asinh Copula Mixture Model performs best—even in the unsupervised setting.

Building on these results, the integration of a probabilistic edge model is presented to further improve the quality of the segmentations. The key idea is to aid the mixture model at placing regions of high uncertainty at areas where an edge is likely to appear. To achieve this goal, a Passive Edge Model which rates the current segmentation according to an edge map and an Active Edge Movement scheme to speed up the convergence has been proposed. A simple post-processing can additionally be applied to further improve the segmentation metrics. The proposed method reaches state-of-the-art performance in generative image segmentation, surpasses several discriminative approaches from the literature, and reaches nearly the performance of the current state-of-the-art in image segmentation. The competing methods are, however, not generative and often trained in a supervised fashion. In fact, the upper bound of the proposed Sinh-asinh Copula Mixture Model presented in the previous chapter outperforms all discriminative approaches in image segmentation.

Since the upper bound has been computed by knowing the ground truth regions, the comparison is unfair. However, the upper bound verifies that the limiting factor is not the model itself, but the way how the parameters are estimated. Including the probable locations of edges is a first important step at improving the estimation and a great step towards reaching the upper bound. But, the upper bound is not reached yet, leaving open a broad range of possible improvements in the future.

5.7 FUTURE RESEARCH DIRECTIONS

The primary goal of future research should aim at reaching the upper bound of the proposed SCMM. Using the proposed edge models has already been a significant contribution, but the current model leaves room for further improvements. This might be achieved through different approaches.

Currently, texture is implemented in a very rudimentary way. It is not directly included as a probabilistic model of different patterns occurring in the image, but passively through the projection of histogram representations to a one-dimensional subspace. However, texture has been shown to be an important cue in contour detection (see e. g., [MFM04]). Therefore, it should help in generative image segmentation as well. Further, the inclusion of positional features has been proven as an important feature in image segmentation. However, modelling the linearly increasing pixel indices by a probability distribution is rather limiting. Other approaches might be more sophisticated and integrate better into the generative segmentation model.

Another limiting factor is the superpixel representation itself. While it allows a huge reduction of the computational demand, the structure of the superpixels itself is visible in the computed segmentations (cf. Figure 5.9). To tackle these issues, different superpixel algorithms, an increased number of superpixels, or even using the full resolution can be tested. However, the computationally demanding evaluation of the model's likelihood needs to be taken into account.

6

WEAKLY SUPERVISED OBJECT LOCALIZATION AND SEMANTIC SEGMENTATION

The content of this chapter has been adapted and/or adopted from [Wil+17]. This publication is based on the author's idea to combine class activation maps with a generative segmentation model in order to improve the weakly supervised detection. The co-author Rene Grzeszick provided the trained CNN and the computation of the class activation maps.

In this chapter, a novel approach to the challenging task of weakly supervised segmentation and object localization is presented. The problem is tackled from a mixed perspective utilizing a discriminative and a generative approach. In the previous chapters, generative models for image segmentation have been presented and analysed. When transferring from image segmentation to the special case of semantic segmentation, Deep Neural Network (DNN) produce state-of-the-art segmentation accuracies. In this scenario, the parameter-rich DNNs can utilize their full discriminative power by exploiting a possibly large set of pixel-wise annotated ground truth segmentations. However, creating these annotations is cumbersome and time-consuming. According to [Bea+16] annotating every pixel of an image takes roughly ten times longer than simply annotating the presence or absence of a set of object classes. Additionally, by using discriminative methods the advantageous probabilistic interpretation of the generative approaches is lost. In this chapter, we propose a way to solve both issues and build a deep segmentation model by combining a DNN which is only supervised on the image-level and fuse it with a generative segmentation model. The proposed model thus exploits relations among different semantic objects and additionally yields a meaningful pixel-wise probabilistic classification. An overview of the proposed algorithm is given in Figure 6.1.

This chapter is structured as follows. First, related work regarding the proposed approach is presented in Section 6.1 and recent developments are sketched in Section 6.2. Second, the proposed fusion of density estimation and class activation maps is presented in Section 6.3. This is followed by an experimental evaluation and comparison to related approaches in Section 6.4. Lastly, the results are summarized in Section 6.5.

6.1 RELATED WORK

In recent years, rapid progress has been made in the field of deep learning. This has been a driving factor for the recognition and localization of objects, where they are very successful [LBH15]. Especially in prominent classification tasks like ImageNet [Rus+15] much progress has been made. In recent years, the error rates have been reduced by a large margin. Similarly, the detection of object instances has been improved. Accurate bounding box predictions were enabled by Regional Convolutional Neural Networks (RCNNs) and Fully Convolutional Networks (FCNs) for object detection [Gir+16; Liu+16]. However, most of these methods are learned in a supervised manner. While

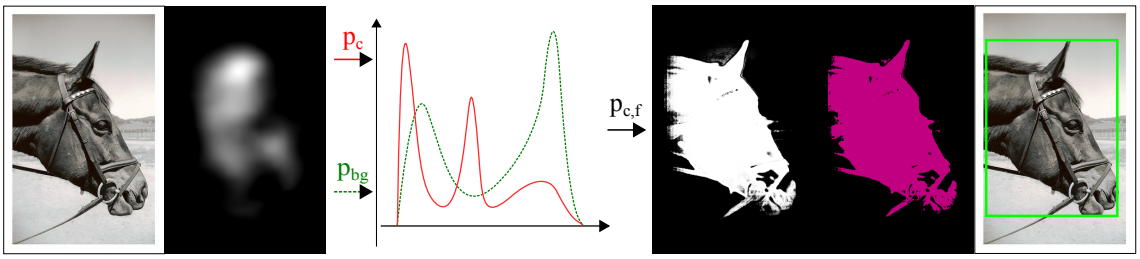


Figure 6.1: Overview of the proposed weakly supervised approach. From left to right: Raw Image, class activation map as produced by a CNN, sketch of kernel density estimate, probabilistic class membership, semantic segmentation, and bounding box.

this is easily possible for classification tasks where an image is assigned a single label, it becomes more complicated for detection and segmentation tasks. In the case of object detections, fine-grained annotations indicating the presence of an object instance have to be provided, usually in the form of bounding boxes.

Prominent examples of segmentation networks are based on FCN, like the residual networks (ResNets), as shown in [WSH16], or the SegNet [BHC15]. The SegNet has also been extended by a Bayesian approach in order to provide a certainty measure associated with the class labels [KBC15]. However, these uncertainty estimates are not a result of a generative model but approximated through a sampling scheme after the model has been trained. The sampling randomly cuts neurons from the full net and a statistic over the output is built. This technique is termed dropout at test time [KBC15].

In the case of semantic segmentation, each pixel of a training image has to be annotated such that deep architectures can enable a pixel-wise detection of objects in a given scene [Cor+16]. As previously mentioned, this takes roughly ten times longer than simply annotating the presence or absence of a set of object classes [Bea+16]. Therefore, the creation of the Cityscapes data set required a tremendous manual annotation effort [Cor+16].

As a result of the costly annotation acquisition, weakly supervised learning approaches became of broader interest as these methods require a lower level of supervision during training. However, semantic segmentation is a challenging task if the level of supervision is reduced to image-level annotations [Pap+15] because the supervisory signal does not include the positions and the extents of the objects any longer. Only the information contained in an object being present or absent is used to train the network. In [KL16] it is therefore proposed to use further micro-annotations in order to identify and prevent typical errors that are learned in a weakly supervised framework. In [KHH+17] a superpixel pooling network is used, which consists of two decoupled DNNs. Multiple passes through the unlabelled data are then used to iteratively refine the semantic segmentation based on images which probably belong to the same class. Another approach to reducing the annotation effort in segmentation is interactive multi-label segmentation [SPB10]. A human annotator draws scribbles, which are used for initializing the segmentation model. Additional information about different supervisory signals in image segmentation is, among others, presented in [HKH17].

In the context of this thesis, two principled approaches for segmentation are distinguished. The discriminative way, where regions are divided according to a trained discriminator or in a generative framework, where the appearances of the regions are

modelled by probability distributions. In the previous chapters, the appearance of image regions is modelled by mixture models. However, objects in an image often do not follow a parametric distribution, especially if an object is highly textured or subject to illumination changes. To some extent, this may be compensated by modifying the feature space or to include the probable locations of edges in an image as additional cues for the model. Another approach to circumvent the inability of parametric distributions to model complex scenes is the use of a non-parametric distribution. However, this only works if the extent of an object in an image is known, because otherwise, the latent representation used to derive the segmentation is meaningless (cf. Section 3.2.4). If, however, the size of objects is known or, as in this case, known to some extent, non-parametric methods like Kernel Density Estimation (KDE) (see Section 3.5.2) can be applied. While the appearance of objects is a-priori not known, it can be estimated after learning a weakly supervised object detector.

Such a weakly supervised object detector can be trained based on the image-level annotations which just indicate the presence of an object class within an image. The detector must then be able to recognize the regions that are discriminative for each of the object classes. Nowadays, these approaches are also based on Convolutional Neural Network (CNN). Recently, methods that incorporate information from region proposals showed state-of-the-art performance [BV16b; Kan+16]. While these methods require the computationally expensive pre-computation of region proposals, the approaches in [Oqu+15] and [Zho+16] tackle the task of weakly supervised detection in end-to-end systems. The work in [Oqu+15] applies a global max pooling to the feature map of the last filter in a fully convolutional VGG16 network architecture. The output of the max pooling is then used for predicting the location of objects in a weakly supervised manner. For each object class, a single point is predicted. A multi-scale training approach is proposed in order to learn the locations of objects of varying size more accurately. A similar approach is proposed in [Zho+16]. Instead of a global max pooling, a global average pooling is proposed as the authors argue that this captures the extent of the objects more accurately. A weighted combination of feature maps is then used in order to indicate the presence of an object in a scene. The resulting combination of feature maps is termed a class activation map (see also Section 2.8.3). The authors propose to use a simple heuristic in order to predict bounding box locations for objects. Each feature map is thresholded and binarized. Then, the largest connected component is chosen for predicting an object's location. While this forms a baseline to evaluate object detection, it partly neglects the pixel-level information provided by the class activation maps. A natural approach is to include pixel-level information and aid the weakly supervised object detector with a segmentation.

In this chapter, a combination of weakly supervised methods and a subsequent segmentation is proposed. It builds on the recent advances in weakly supervised detection using CNN. The segmentation is then initialized based on a class activation map which has been learned in a weakly supervised manner. The segmentation requires no further annotations and uses a generative approach. Partially, the proposed procedure is similar to the aforementioned interactive multi-label segmentation. However, the scribbles which are used to initialize the segmentation are not drawn by a human annotator, but rather by a CNN. Therefore, no further annotations are required for training the segmentation model and it can be nicely integrated into a weakly supervised detection

framework that is solely based on image-level annotations. Instead of scribbling the annotator is asked to tell what objects appear in an image and the proposed procedure estimates the probabilistic class membership of every pixel for each occurring object. In this chapter, it is shown how to initialize a probabilistic image segmentation using class activation maps generated by a weakly supervised object localization network. Further, a combination of this discriminative detection model and a generative image model is used to improve weakly supervised object detection. Lastly, the framework is expanded in order to derive semantic segmentations of natural images.

6.2 RECENT DEVELOPMENTS

Recently, weakly supervised methods gained further interest and an increased number of publications have been presented at major vision conferences, among others, [BMG18; AK18; Tan+18a; Fan+18]. The trends can be summarized as different ways to fuse information from different sources into the weakly supervised segmentation. In contrast to the proposed approach, they often rely on different data sets or need to be trained additionally. The proposed approach, however, directly operates on the class activation maps and the image data itself.

The first group of approaches fuses the image-level labels with saliency maps which are learned from a separate data set. However, the saliency maps are almost exclusively learned from pixel-level annotations and are therefore contradicting the weakly supervised paradigm to some extent. Typical representatives are, among others, [BMG18; Fan+18]. Other approaches, like [Qi+16; Wei+17; LAT18], generate surrogate pixel-level labels which are used to train a segmentation network or leverage pre-trained object detection networks [Sal+18] for semantic segmentation. More in line with the proposed approach are the methods which integrate pixel-level affinities [AK18] or modify the loss function of the DNN. For instance, in [Tan+18a] the normalized cut (see Section 2.5.2) is integrated into the loss function of a DNN. However, none of the above methods operates in a generative framework, as the proposed approach does.

6.3 FUSING CLASS ACTIVATION MAPS WITH DENSITY ESTIMATION

For localizing and segmenting objects in a weakly supervised fashion a combination of a discriminative and a subsequent generative approach is proposed. The key idea behind the combination is to maintain the benefits of a generative model even when using a discriminative classifier.

In a first step a discriminative classifier—a FCN—is solely trained on image-level annotations. The last convolutional layer of this network is designed such that it outputs a class activation map. Each map can then be leveraged in order to indicate the probable location of objects (see Section 2.8.3). Next, these class activation maps are used for initializing a generative segmentation model for every occurring class. From these models and an additional model for the background class of the image, a mixture model is built. A KDE is chosen as a probability distribution in order to be maximally flexible in the shapes the distribution can model. Since the class activation maps provide a rough localization and a coarse estimate of the extent of an object, the issue of an indirect interpretation of the KDE is circumvented. Recall, for image segmentation with mixture

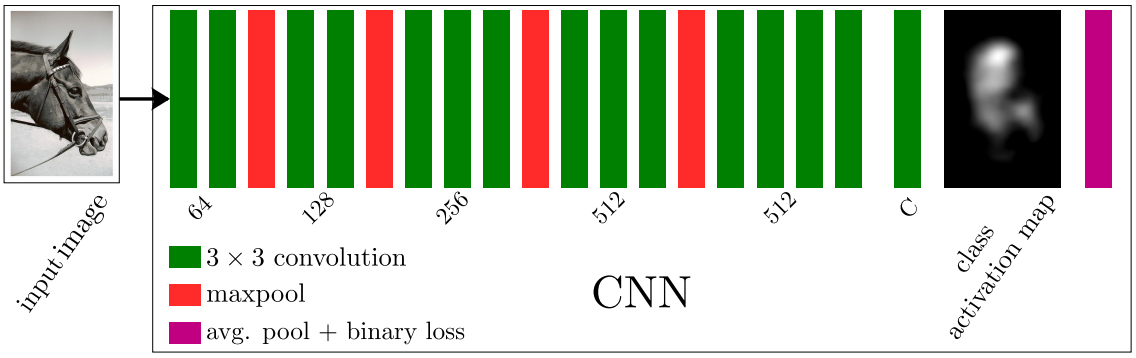


Figure 6.2: Overview of the proposed FCN architecture. Following the VGG architecture, the network is composed of 3×3 convolutions and subsequent max pooling operations. The last block of the network is designed such that a class activation map can be derived for each of the C classes from the last convolution layer. The network is trained using a binary logistic loss after a global average pooling.

models the latent interpretation of the mixture models is essential (cf. Section 3.5.2 and Section 3.2.4).

The derived segmentation is then evaluated based on global class similarities as well as local per instance knowledge about the object's appearance with respect to the background. An overview of the proposed algorithm is given in Figure 6.1. The approach can be applied for semantic segmentation as well as bounding box predictions.

In general, using class activation maps to initialize a segmentation model is similar to using scribble annotations. In this case, however, the scribbles are replaced by the output of the CNN. Therefore, one can think of the CNN as drawing the scribbles in the image instead of the human annotator. Technical details of the proposed approach are presented in the following sections.

6.3.1 *Class Activation Maps*

In this section, the concept of class activation maps in the context of the given setting is explained. General information about class activation maps are presented in Section 2.8.3. As already mentioned, a FCN has been trained for computing a class activation map. An overview of the architecture is given in Figure 6.2. It builds on the ideas proposed in [Oqu+15; Zho+16]. Following the VGG16 architecture, the network consists of blocks which contain 3×3 convolutions followed by subsequent pooling operations. The last of these blocks contains two additional convolution layers and the goal is to learn exactly one filter for each of the C object classes. The last layer is then followed by a global average pooling [Zho+16] so that a single prediction score is computed for each of the object classes. While the last step is essential and reflects the nature of weak supervision, the class activation maps cannot be directly used for computing the loss because the supervisory signal is only available on image-level and not on pixel-level.

Similar to the approach described in [Oqu+15], a binary loss function (cross-entropy) with

$$l_{ce} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{c=1}^{C} y_{j,c}\log\left(\hat{y}_{j,c}\right) \tag{6.1}$$

is applied for training, where $\mathbf{y}_j$ is the desired output indicating the presence of all object classes C and $\hat{\mathbf{y}}_j$ is the prediction of the network. N is the number of items in the training data. The network is thus trained based on a binary vector indicating the presence of multiple object classes simultaneously.

For localizing objects, the feature map of the last convolutional layer can easily be interpreted as a class activation map since the responses of each of the filters indicate the presence of exactly one class. Since the network contains four pooling layers, the final output of the class activation map is 1/16 of the original input image size.

6.3.2 *Segmentation*

As the class activation map is relatively coarse, recovering an object's boundaries is not directly possible. However, the network produces an educated guess about the position and extent of the objects appearing in the image. A natural approach for recovering the true object boundaries is segmenting the image into semantic regions.

The goal is to recover the joint probability distribution of the object extent and colour appearance from the raw data through a probabilistic model. This is achieved by combining the learned global view of the CNN with a local appearance model.

Modelling the local appearance of each region by KDE appears to be especially well suited because it does not make strong assumptions regarding the shape of the distribution. Alternatives, like the Gaussian distribution or any other parametric model, are in semantic segmentation often too limiting. In contrast to the previous chapters, where each region has been associated with a distribution that has one unique mode, this assumption needs to be relaxed in semantic segmentation. For certain classes, like the background class or the person class, assuming a unique mode for the colour distribution is too limiting. For instance, persons exhibit multiple colour modes through the differences in hair, skin, and the possibly large number of different colours of the clothing. Capturing this multimodality is easily achieved by KDE.

As pixel-level features, the coordinates of the Lab and RGB colour spaces are chosen. Additionally, the positions of the pixels within the image are used as positional features to enforce locality in the regions. This has been proven useful in the previous chapters (see Section 4.3). The joint probability $p(X, Y, L, a, b, R, G, B)$ is estimated by a KDE. In order to reduce the computational burden of the KDE, the joint distribution is rewritten as

$$p(X, Y, L, a, b, R, G, B) = p(X, Y)p(L)p(a)p(b)p(R)p(G)p(B)$$

by assuming independence between the probability distribution of the colour features. By only looking at the marginal distribution of the colour features the number of data points which need to be evaluated in the KDE is vastly reduced, since the number of data points grow exponentially with the number of features while keeping the resolution constant. However, possible correlations are an issue in the positional features and should

not be neglected, because the appearance of an object might be multi-modal. Assuming independence in the positional data leads to an overestimation of the object probability in regions where only one of the two marginals might have a high probability.

6.3.3 *Combining Class Activation Maps with KDE*

The class activation maps form the basis of the segmentation. In the first step, the class activations of an occurring class are rescaled to the size of the original input image using bicubic interpolation. Second, the activations a_j at pixel index j of the set of class activation maps $\mathcal{A} = \left\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(C)}\right\}$ are transformed to the interval $[0, 1]$ to measure the overall influence of different pixels with respect to all others. As the CNN learns to predict multiple classes at once, each class c is considered independently. The normalization is performed such that $\sum_{j=1}^{N} a_j = 1$ with the number of samples N being the pixels in the class activation map for a given class c. These activations are then used as weights ω_j for the KDE of each class occurring in the image separately, thus yielding an appearance model $p_c(X, Y, L, a, b, R, G, B)$ for each class c. For the sake of readability, the appearance model is simply denoted by p_c in the following.

The same procedure is repeated for a background model p_{bg} where the weights are computed by $\mathbf{Bg} = 1 - \mathbf{A}^{(c)}$. A separate background model is crucial to the success of the segmentation, because of the blurred predictions of the CNN. Since the class activation maps do not respect the object boundaries they almost always include a part of the background instead of just the foreground.

For deriving the final pixel-wise foreground probability map $p_{f|c}$ for a given class c, the probability model for class c and the background model, are both evaluated for each pixel. The resulting probabilities are compared with respect to how much of the sum of both models is explained by the model of class c:

$$p_{f|c} = \frac{p_c}{p_c + p_{bg}}. \tag{6.2}$$

The final segmentation is then achieved by thresholding $p_{f|c}$ at a given probability level θ_p for each class. Ambiguities in the case of equal probabilities for different classes are resolved by taking the class with the higher activation $\mathcal{A}_c$ by assuming that higher activations correspond to a higher certainty of the presence of an object at a given location. Therefore, the global view of the CNN is rated as more important than the local appearance.

6.3.4 *Localization*

For localizing an object, a bounding box can be extracted after computing the final segmentation. Traditionally, a bounding box can be extracted from the class activation maps by simply thresholding the class-specific scores at a given level and then finding the largest connected component. The extent of the resulting region is then computed and represents the bounding box. Following this reasoning, an improved bounding box, which takes the local instance knowledge into account, is extracted by thresholding the class probabilities $p_{f|c}$ and computing the extent of the largest connected component. The largest connected component is chosen in order to cope with multiple objects of the same class, which might be present in an image.

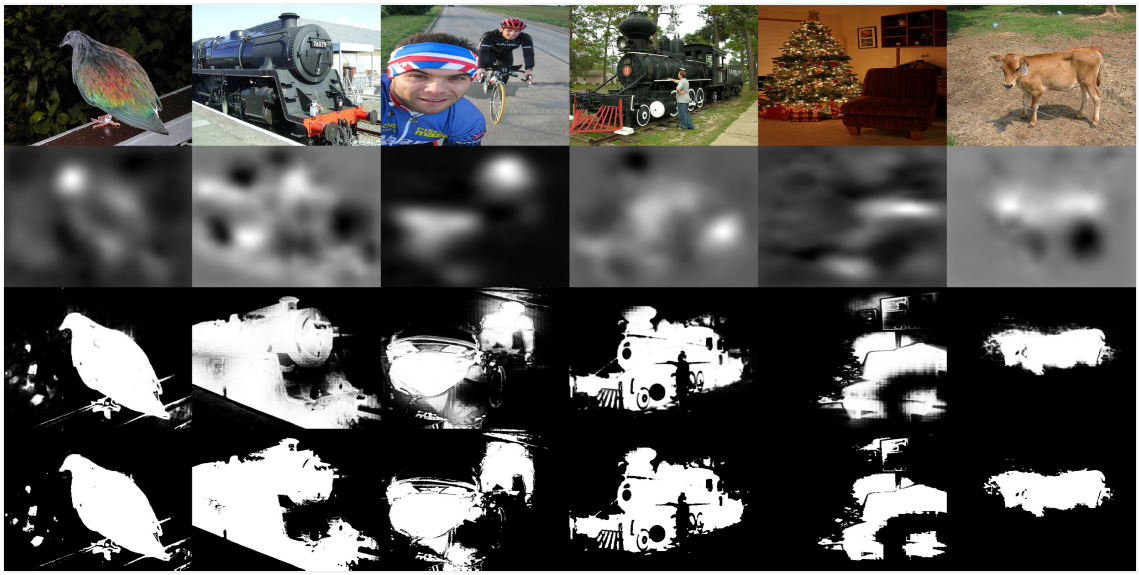


Figure 6.3: Segmentation results on the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC). The last two columns depict typical failure cases. From top to bottom: raw image, class activation map, probability map, and the resulting semantic segmentation.

6.4 EVALUATION

The proposed method is evaluated on two different tasks of VOC: the semantic segmentation task and the object localization task. Both approaches are tackled in a weakly supervised manner using the proposed approach. Additionally, the novel combination of density estimation and class activation maps is compared with state-of-the-art techniques in weakly-supervised learning.

The proposed FCN, which is used for computing the class activation maps, is trained in a weakly supervised fashion on the training set of VOC. Only image-level annotations are used which indicate the presence of an object class in a scene. For training and testing, the input images are rescaled so that the length of the shortest edge equals 512px. The network is trained with a batch size of 256 for $2,000$ iterations. This corresponds to $512,000$ images or 90 epochs on the training set. The weights are initialized from a VGG16 network trained on ImageNet. The first 600 iterations are trained with a learning rate of 10^{-4} which is then increased to 10^{-3}. Random data augmentations are applied to the original images. Namely, translations and rotations of up to 5% / 5 deg, Gaussian noise with $\sigma = 0.02$, and vertical mirroring.

The same network is used for both tasks, weakly supervised semantic segmentation and localization. First, the segmentation accuracy of the proposed approach is evaluated based on the probabilities which are computed by the KDE. Given a segmentation and a ground truth segmentation, the segmentation accuracy (see Section 2.11) is computed by treating each class as a binary classification problem and computing the confusion matrix. Recall, the segmentation accuracy is given by [Eve+15]:

$$\text{seg. acc.} = \frac{|\mathcal{TP}|}{|\mathcal{TP}| + |\mathcal{FP}| + |\mathcal{FN}|}. \tag{6.3}$$

		Mean Acc. [%]
	#labels	Validation
SPN (plain) [KHH+17]	10 582	40.0
EM-Adapt [Pap+15]	10 582	38.2
EM-Fixed [Pap+15]	10 582	20.8
GMM (this work)	5717 (0)	25.9
KDE (this work)	5717 (0)	**41.1**

Table 6.1: Segmentation accuracy on the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC) segmentation task. The column denoted by #labels indicates the number of annotations that are used for training the CNN in a weakly supervised manner. In brackets the number of annotations used for training the segmentation is depicted.

Note that, while the CNN is trained based on image-level annotations, the segmentation algorithm requires no further supervision.

Second, the Correct Localization (CorLoc) accuracy measure is evaluated [DAF12; Kan+16]. Given an image and a target class, the CorLoc describes the percentage of images where a bounding box has been correctly predicted for localizing an object of the said target class. The localizations are derived from the probabilistic image model as described in Section 6.3.4. In order to consider a prediction as correct, an Intersection over Union (IoU) of the two bounding boxes larger than 50% is required. The CorLoc metric is typically evaluated on the training set [Kan+16]. Here, the CorLoc is evaluated on the complete trainval set. Nevertheless, only the training split of the detection task has been used during the network's training. Note that the same network is used to evaluate both tasks—weakly supervised object detection and semantic segmentation.

6.4.1 *Qualitative Results*

Figure 6.3 and Figure 6.4 depict qualitative results for both tasks. It is evident that the computed probability maps capture the extent of the appearing objects precisely. However, the computed probability maps may fail to cover the complete extent of an object if the class activation maps are highly cluttered or the CNN is uncertain about the location of an object. Another frequently occurring drawback of the computed probability maps is that regions which are close to an object and are highly similar to parts of the annotated object are considered as a part of this object as well. This may be circumvented if texture features or shape priors are added to the appearance model in future work.

6.4.2 *Segmentation Accuracy*

In the first experiments, the segmentation accuracy of the proposed approach is evaluated. Therefore, the validation set of the segmentation task of VOC is used. In accordance with the CorLoc task, the visible classes are assumed to be known. The baseline

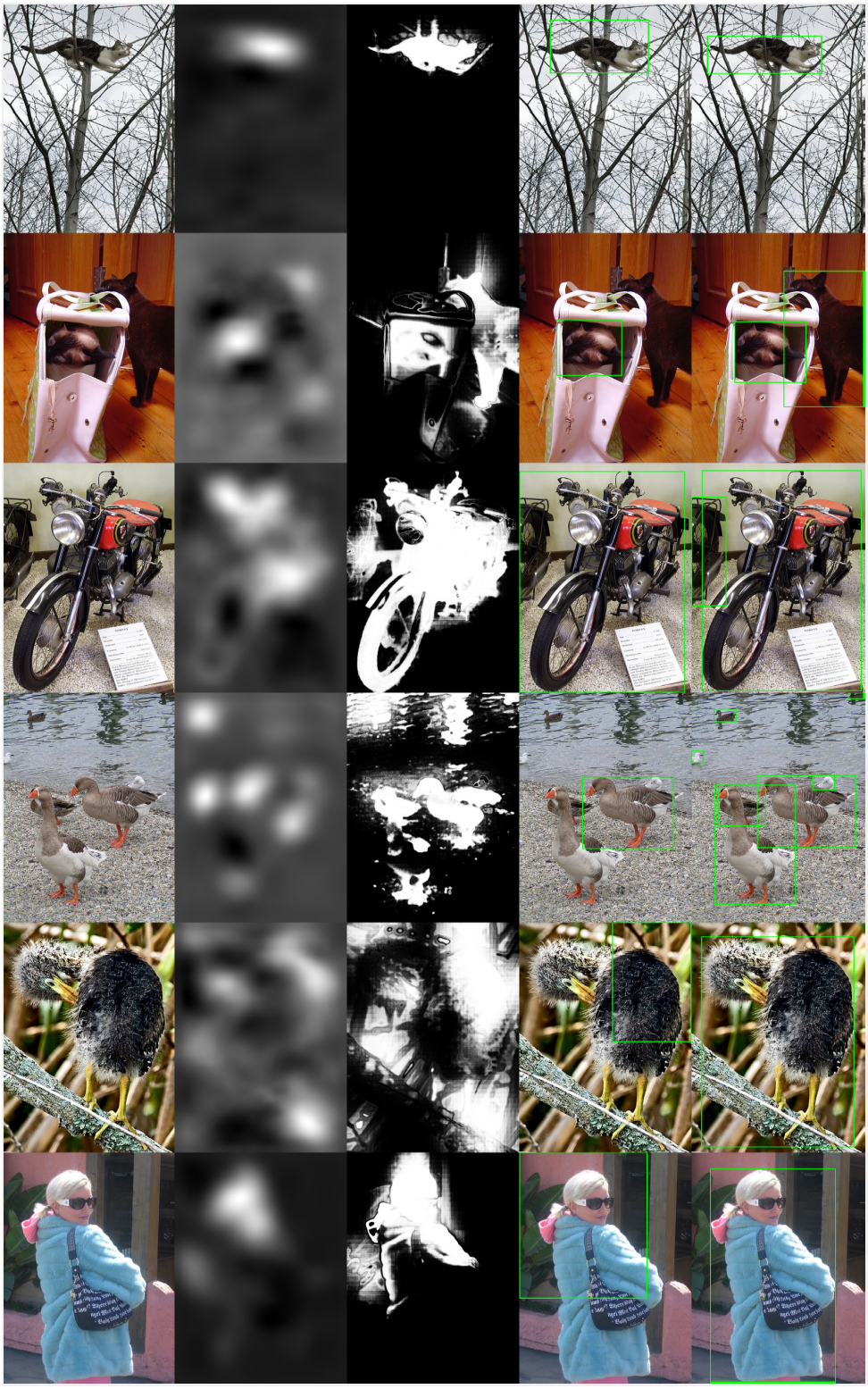


Figure 6.4: CorLoc results on the PASCAL Visual Object Classes Challenge 2012 [Eve+15] (VOC) trainval split. From left to right: raw image, class activation map, probability map, predicted bounding box, and ground truth bounding box. The first rows show good predictions and the last two rows depict typical failure cases.

	CorLoc [%]
Context LocNet [Kan+16]	54.8
Thresholding	37.9
KDE	41.9
KDE + oracle	53.3

Table 6.2: Correct Localization on VOC trainval split.

algorithm is a Gaussian Mixture Model (GMM) where each occurring object and the background are modelled by one component of a GMM, respectively. Furthermore, the presented KDE approach is evaluated in terms of segmentation accuracy. The results are summarized in Table 6.1. In addition to the accuracy, the number of training images is presented, as the presented approach uses only about half of the number of training images compared to methods from the literature.

The proposed approach is compared to a baseline using a GMM as a substitute for the generative image model. The application of a GMM greatly decreases performance. This confirms the assumption that regions of complex scenes do not follow Gaussian distributions and that a GMM is not a reasonable approximation for an appearance model in complex scenes. In terms of segmentation accuracy, the presented KDE approach is similar to state-of-the art approaches from the literature.

6.4.3 *CorLoc Metric*

The CorLoc metric is evaluated on the complete trainval split. Three different approaches are presented. Additionally, a competing method from the literature is evaluated. Note that the proposed methods are solely trained on the images from the training split of VOC, excluding the validation images. This is a deviation from the CorLoc protocol which is often evaluated in the literature where the complete training and validation data are used for training. All results are averaged over all 20 object classes and summarized in Table 6.2.

The baseline of our algorithm is thresholding the class activation maps, as discussed in Section 6.3.4. All values exceeding a normalized activation of 80% are considered as positive. This is followed by a connected component analysis (see e. g., [Sze10, pp. 131-132]) where only the largest connected component remains in order to reduce the influence of multiple objects, which may be present in an image. The proposed approach goes beyond the baseline method by using the thresholded class activation maps in order to initialize a kernel density estimator and by computing a probabilistic class model for each visible class in an image (see Section 6.3.3). In this approach, two thresholds need to be considered. The first is, as in the baseline approach, the level at which the class activation maps are thresholded and second the level at which the probability $\theta_{p_{f|c}}$ of a pixel is considered to belong to a specific class. The proposed approach improves the results by 4%. Note that recently the only other result on this task has been reported in [Kan+16]. While these show superior performance, the network is trained with fur-

ther samples and requires additional regional proposals as it is directly optimized for a bounding box evaluation, which is not the case here.

From the experiments, it was evident that the presented approach is sensitive with respect to the choice of the second threshold $\theta_{p_{f|c}}$, because the extent of an object influences the choice of an optimal threshold. Smaller regions require more confident predictions and vice versa. Therefore, a third approach is considered, where an appropriate threshold is determined by an oracle on a per image basis. This leads to a vastly increased performance compared to our previous approaches. It is, thus, reasonable to assume that the performance can be improved with a meta-recognition step which determines an appropriate threshold and allows for closing the gap between the proposed approach and [Kan+16].

6.5 SUMMARY

In this chapter, a novel approach to the task of weakly supervised segmentation and object localization has been presented. In contrast to the methods presented in the literature which are exclusively discriminative, the problem is tackled from a mixed perspective utilizing a discriminative and a generative approach.

The results show that the proposed weakly supervised semantic segmentation approach reaches state-of-the-art performance compared to other weakly supervised approaches from the literature. In contrast to other approaches, ours is generative and, therefore, offers uncertainty estimates. It is further shown that the application of a KDE is far superior to a GMM baseline which does not model the appearance in complex scenes accurately. In addition, the proposed segmentation approach is integrated into a weakly supervised object localization framework. Promising results are achieved by the proposed approach in terms of the CorLoc accuracy. The results may further be improved by tackling the problem from a Bayesian perspective and considering similarities across several test images, as in [KHH+17], where the posterior belief of the semantic labels is iteratively updated.

7 CONCLUSION

In contrast to the majority of approaches from the related work, image segmentation has been tackled from a generative perspective in this thesis. While the generative approach yields uncertainty estimates of the predictions as an implicit property, it comes at the cost of a higher modelling complexity. Therefore, these models have not been used if a high accuracy is required. However, this work has shown that generative approaches are a viable alternative. They additionally provide uncertainty estimates which are an important tool when assessing the quality of a segmentation, interpreting the prediction, and communicating the results to domain experts.

In order to model the observed data, mixture models are chosen. They have the benefit of yielding a direct interpretation (see Section 3.2.4) in which it is assumed that each region of an image is associated with exactly one probability distribution. Afterwards, the probability of a pixel belonging to a region is computed by evaluating the probability density function (pdf) of every mixture component. The result is then interpreted as a probabilistic segmentation. However, this approach poses several issues which are addressed in this thesis through rigorously testing. First, it is unclear how well image data can be described by probability distributions and which features shall be used. Second, it is unclear which pdfs should be used and which properties a suitable pdf should have. Lastly, the number of mixture components is a-priori unknown and has to be estimated or given, depending on the level of supervision which is desired.

In this thesis, it has been shown that the colour space used to describe the image data does not have a high impact on the achievable segmentation accuracy. However, including a model for the pixel positions is of central importance to achieve good results. Additionally, it is of utmost importance to have as flexible as possible distributions for modelling the observed image data. The proposed Sinh-asinh Copula Mixture Model (SCMM) has been proven as the best choice in terms of the considered segmentation metrics. These experiments have been conducted in a supervised setting in order to successfully establish an upper bound of the achievable segmentation accuracies. Afterwards, the experiments have been concerned with reaching the upper bound in an unsupervised setting.

In the unsupervised case, it is especially challenging to estimate a suitable number of mixture components. In this thesis, we have experimentally shown that a Dirichlet Process Mixture Model (DPMM) (see Section 3.5.1) performs best. Fortunately, it suits the Bayesian paradigm used throughout this thesis best. Afterwards, it has been shown that probabilistically including the probable location of edges into the mixture estimation is another crucial feature to leverage the proposed SCMM to surpass the state-of-the-art performance in generative image segmentation. Further, the proposed SCMM is on-par with many discriminative approaches which do not offer uncertainty estimates.

Lastly, an approach to weakly supervised semantic segmentation and object detection has been presented. In contrast to many other approaches to this challenging task, we compute a generative segmentation as an output although a discriminatively trained

Deep Neural Network (DNN) is used to detect the presence or absence of objects. This is achieved through combining the class activation maps of a weakly supervised DNN with a local appearance model. The local appearance is modelled generatively by non-parametric density estimation. The proposed approach outperforms a Gaussian Mixture Model (GMM) used as a generative baseline model significantly.

FUTURE WORK

While possible feature research directions have been briefly sketched at the end of the respective chapters, some of the most important aspects are summarized here. In the context of mixture models, the lack of a suitable texture model stands out. While this has been circumvented to some extent by using the proposed texture feature, it does not include texture as a thoroughly generative model. Future work should place more focus on this aspect because texture has often been proven to be an important cue in image segmentation. One particular interesting research direction is the integration of a generative model for repeatedly occurring texture patterns, like textons. By doing this, the resulting model would additionally consist of a mixture model of discrete probability distributions which generatively model the texture behaviour within every segment.

The presented Sinh-asinh Copula Mixture Model (SCMM) is modular. Therefore, parts of it can be replaced conveniently and new variants can be explored easily. While the use of the sinh-asinh distribution appears to be already well suited, because it is a very tractable and flexible distribution, the copula modelling the dependence structure deserves more attention. The rich copula literature offers alternatives to the used Gaussian copula which are worth exploring.

One of the most promising approaches to weakly supervised segmentation in a generative setting will be a better fusion of DNNs and a local appearance model based on probability distribution. While the results of the previous chapter have been an important first step, other ways can be explored. Probably most promising is the integration of another unsupervised approach which directly operates on the pixel-level. Therefore, a trifocal view on the data is achieved: the DNN's, the class activation map's, and the density's view on the image data. Another future challenge is to transfer the methods presented in this thesis to other domains like remotely sensed images or medical image data.

BIBLIOGRAPHY

[AS64] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied mathematics series. Dover Publications, 1964.

[Ach+12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282.

[AR13] C. Aggarwal and C. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis, 2013.

[AK18] J. Ahn and S. Kwak. "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4981–4990.

[Aka74] H. Akaike. "A new look at the statistical model identification." In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723.

[Alp+07] S. Alpert, M. Galun, R. Basri, and A. Brandt. "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2007.

[AD52] T. W. Anderson and D. A. Darling. "Asymptotic theory of certain goodness of fit criteria based on stochastic processes." In: *The annals of mathematical statistics* (1952), pp. 193–212.

[Arb06] P. Arbelaez. "Boundary extraction in natural images using ultrametric contour maps." In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE. 2006, pp. 182–182.

[Arb+11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. "Contour detection and hierarchical image segmentation." In: *IEEE transactions on pattern analysis and machine intelligence* 33.5 (2011), pp. 898–916.

[Arb+14] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. "Multiscale Combinatorial Grouping." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[Azz85] A. Azzalini. "A class of distributions which includes the normal ones." In: *Scandinavian journal of statistics* (1985), pp. 171–178.

[BHC15] V. Badrinarayanan, A. Handa, and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." In: *arXiv preprint arXiv:1505.07293* (2015).

[BLC18] Z. Ban, J. Liu, and L. Cao. "Superpixel Segmentation Using Gaussian Mixture Model." In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 4105–4117.

[Bar12] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[BDH14] R. Bardenet, A. Doucet, and C. Holmes. "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach." In: *International Conference on Machine Learning (ICML)*. 2014, pp. 405–413.

[BN77] O. Barndorff-Nielsen. "Exponentially decreasing distributions for the logarithm of particle size." In: *Proc. R. Soc. Lond. A* 353.1674 (1977), pp. 401–419.

[BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features." In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[Bay63] T. Bayes. "An essay towards solving a problem in the doctrine of chances." In: *Philosophical transactions of the Royal Society of London* 53 (1763), pp. 370–418.

[Bea+16] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. "What's the point: Semantic segmentation with point supervision." In: *European conference on computer vision*. Springer. 2016, pp. 549–565.

[BM98] S. Belongie and J. Malik. "Finding boundaries in natural images: A new method using point descriptors and area completion." In: *European conference on computer vision*. Springer. 1998, pp. 751–766.

[Bez81] J. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Advanced applications in pattern recognition. Plenum Press, 1981.

[BV16a] H. Bilen and A. Vedaldi. "Weakly supervised deep detection networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2846–2854.

[BV16b] H. Bilen and A. Vedaldi. "Weakly supervised deep detection networks." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2846–2854.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[BH14] J. K. Blitzstein and J. Hwang. *Introduction to probability*. Chapman and Hall/CRC, 2014.

[BG03] I. Borg and P. Groenen. "Modern multidimensional scaling: theory and applications." In: *Journal of Educational Measurement* 40.3 (2003), pp. 277–280.

[Bor+15] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. "Salient Object Detection: A Benchmark." In: *IEEE TIP* 24.12 (2015), pp. 5706–5722.

[BZM06] A. Bosch, A. Zisserman, and X. Munoz. "Scene classification via pLSA." In: *Proc. European Conference on Computer Vision (ECCV)* (2006), pp. 517–530.

[BHH+78] G. E. Box, W. G. Hunter, J. S. Hunter, et al. "Statistics for experimenters." In: (1978).

[BMG18] R. Briq, M. Moeller, and J. Gall. "Convolutional Simplex Projection Network (CSPN) for Weakly Supervised Semantic Segmentation." In: *Proc. British Machine Vision Conference (BMVC)* (2018).

[BM15] R. P. Browne and P. D. McNicholas. "A mixture of hyperbolic distributions." In: *Canadian Journal of Statistics* 43.2 (2015), pp. 176–198.

[CUF18] H. Caesar, J. Uijlings, and V. Ferrari. "COCO-Stuff: Thing and stuff classes in context." In: *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE. 2018.

[CH74] T. Caliński and J. Harabasz. "A dendrite method for cluster analysis." In: *Communications in Statistics-theory and Methods* 3.1 (1974), pp. 1–27.

[Can86] J. Canny. "A computational approach to edge detection." In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[Car+17] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. "Stan: A Probabilistic Programming Language." In: *Journal of Statistical Software, Articles* 76.1 (2017), pp. 1–32.

[Car+02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. "Blobworld: Image segmentation using expectation-maximization and its application to image querying." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.8 (2002), pp. 1026–1038.

[Che95] Y. Cheng. "Mean shift, mode seeking, and clustering." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), pp. 790–799.

[Cho+17] S. K. Choy, S. Y. Lam, K. W. Yu, W. Y. Lee, and K. T. Leung. "Fuzzy model-based clustering and its application in image segmentation." In: *Pattern Recognition* 68 (2017), pp. 141–157.

[CM02] D. Comaniciu and P. Meer. "Mean shift: a robust approach toward feature space analysis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619.

[Cor+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The cityscapes dataset for semantic urban scene understanding." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223.

[CV95] C. Cortes and V. Vapnik. "Support-vector networks." In: *Machine Learning* 20.3 (1995), pp. 273–297.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection." In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893.

[DB79] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[DAF12] T. Deselaers, B. Alexe, and V. Ferrari. "Weakly Supervised Localization and Learning with Generic Knowledge." In: *International Journal of Computer Vision (IJCV)* 100.3 (2012), pp. 275–293.

[Des14] C. Desrosiers. "A fast and adaptive random walks approach for the unsupervised segmentation of natural images." In: *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE. 2014, pp. 130–135.

[DZ15] P. Dollár and C. L. Zitnick. "Fast edge detection using structured forests." In: *IEEE transactions on pattern analysis and machine intelligence* 37.8 (2015), pp. 1558–1570.

[Dua+87] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. "Hybrid monte carlo." In: *Physics letters B* 195.2 (1987), pp. 216–222.

[DB94] K. L. Duffin and W. A. Barrett. "Spiders: a new user interface for rotation and visualization of n-dimensional point sets." In: *Proceedings Visualization '94*. 1994, pp. 205–211.

[Eve+15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes Challenge: A Retrospective." In: *International Journal of Computer Vision* 111.1 (2015), pp. 98–136.

[Fah+07] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz. *Statistik: Der Weg zur Datenanalyse*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007.

[Fan+18] R. Fan, Q. Hou, M.-M. Cheng, G. Yu, R. R. Martin, and S.-M. Hu. "Associating Inter-Image Salient Instances for Weakly Supervised Semantic Segmentation." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 367–383.

[FFP05] L. Fei-Fei and P. Perona. "A bayesian hierarchical model for learning natural scene categories." In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 524–531.

[Fer73] T. S. Ferguson. "A Bayesian analysis of some nonparametric problems." In: *The annals of statistics* (1973), pp. 209–230.

[FS89] I. Fogel and D. Sagi. "Gabor filters as texture discriminator." In: *Biological Cybernetics* 61.2 (1989), pp. 103–113.

[FH75] K. Fukunaga and L. Hostetler. "The estimation of the gradient of a density function, with applications in pattern recognition." In: *IEEE Transactions on Information Theory* 21.1 (1975), pp. 32–40.

[Fuk80] K. Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." In: *Biological Cybernetics* 36.4 (1980), pp. 193–202.

[Gei+13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets Robotics: The KITTI Dataset." In: *International Journal of Robotics Research (IJRR)* (2013).

[Gel+13] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.

[GH06] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[GG84] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[GRS95] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[Gir+16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Region-based convolutional networks for accurate object detection and segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (2016), pp. 142–158.

[GB10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 249–256.

[GW02] R. Gonzalez and R. Woods. *Digital Image Processing*. International Edition. Prentice Hall, 2002.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[GY19] S. N. Gowda and C. Yuan. "ColorNet: Investigating the importance of color spaces for image classification." In: *arXiv preprint arXiv:1902.00267* (2019).

[Gre17] P. J. Green. "Introduction to finite mixtures." In: *arXiv preprint arXiv:1705.01505* (2017).

[Haa+06] H. Haario, M. Laine, A. Mira, and E. Saksman. "DRAM: Efficient adaptive MCMC." In: *Statistics and Computing* 16.4 (2006), pp. 339–354.

[HST01] H. Haario, E. Saksman, and J. Tamminen. "An adaptive Metropolis algorithm." In: *Bernoulli* (2001), pp. 223–242.

[Has70] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications." In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109.

[HP17] R. Hettiarachchi and J. Peters. "Voronoi region-based adaptive unsupervised color image segmentation." In: *Pattern Recognition* 65 (2017), pp. 119 –135.

[HG14] M. D. Hoffman and A. Gelman. "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[HKH17] S. Hong, S. Kwak, and B. Han. "Weakly Supervised Learning with Deep Convolutional Neural Networks for Semantic Segmentation: Understanding Semantic Layout of Images with Minimum Human Supervision." In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 39–49.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." In: *Neural networks* 2.5 (1989), pp. 359–366.

[HW05] J. Hosking and J. Wallis. *Regional Frequency Analysis: An Approach Based on L-Moments*. Cambridge University Press, 2005.

[Hua+18] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. "The ApolloScape Dataset for Autonomous Driving." In: *arXiv: 1803.06184* (2018).

[HW62] D. H. Hubel and T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." In: *The Journal of physiology* 160.1 (1962), pp. 106–154.

[IKN98] L. Itti, C. Koch, and E. Niebur. "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (1998), pp. 1254–1259.

[Jai10] A. K. Jain. "Data clustering: 50 years beyond K-means." In: *Pattern recognition letters* 31.8 (2010), pp. 651–666.

[JF91] A. K. Jain and F. Farrokhnia. "Unsupervised texture segmentation using Gabor filters." In: *Pattern recognition* 24.12 (1991), pp. 1167–1186.

[Joe14] H. Joe. *Dependence modeling with copulas*. CRC Press, 2014.

[JP09] M. Jones and A. Pewsey. "Sinh-arcsinh distributions." In: *Biometrika* 96.4 (2009), pp. 761–780.

[Kan+16] V. Kantorov, M. Oquab, C. M., and I. Laptev. "ContextLocNet: Context-aware Deep Network Models for Weakly Supervised Localization." In: *Proc. European Conference on Computer Vision (ECCV)*. 2016.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models." In: *International journal of computer vision* 1.4 (1988), pp. 321–331.

[KBC15] A. Kendall, V. Badrinarayanan, and R. Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." In: *arXiv preprint arXiv:1511.02680* (2015).

[KM17] L. Khelifi and M. Mignotte. "A novel fusion approach based on the global consistency criterion to fusing multiple segmentations." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.9 (2017), pp. 2489–2502.

[KLL13] T. H. Kim, K. M. Lee, and S. U. Lee. "Learning full pairwise affinities for spectral segmentation." In: *IEEE transactions on pattern analysis and machine intelligence* 35.7 (2013), pp. 1690–1703.

[KWH14] J. Kivinen, C. Williams, and N. Heess. "Visual boundary prediction: A deep neural prediction network and quality dissection." In: *Artificial Intelligence and Statistics*. 2014, pp. 512–521.

[Kok15] I. Kokkinos. "Pushing the boundaries of boundary detection using deep learning." In: *arXiv preprint arXiv:1511.07386* (2015).

[KL16] A. Kolesnikov and C. H. Lampert. "Improving Weakly-Supervised Object Localization By Micro-Annotation." In: *Proc. British Machine Vision Conference (BMVC)* (2016).

[KCW14] A. Korattikara, Y. Chen, and M. Welling. "Austerity in MCMC land: Cutting the Metropolis-Hastings budget." In: *International Conference on Machine Learning*. 2014, pp. 181–189.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[Kuc+17] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. "Automatic differentiation variational inference." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 430–474.

[KL51] S. Kullback and R. A. Leibler. "On information and sufficiency." In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[Kum+12] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. "Leafsnap: A Computer Vision System for Automatic Plant Species Identification." In: *The 12th European Conference on Computer Vision (ECCV)*. 2012.

[KHH+17] S. Kwak, S. Hong, B. Han, et al. "Weakly supervised semantic segmentation using superpixel pooling network." In: *AAAI*. 2017, pp. 4111–4117.

[Lap+19] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. "Unmasking Clever Hans predictors and assessing what machines really learn." In: *Nature Communications* 10.1 (2019), p. 1096.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 2169–2178.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." In: *Nature* 521.7553 (2015), pp. 436–444.

[LeC+98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LKF+10] Y. LeCun, K. Kavukcuoglu, C. Farabet, et al. "Convolutional networks and applications in vision." In: *ISCAS*. Vol. 2010. 2010, pp. 253–256.

[LWW16] M. Lenoch, T. Wilhelm, and C. Wöhler. "Simultaneous Surface Segmentation and BRDF Estimation via Bayesian Methods." In: *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, (VISIGRAPP 2016)*. INSTICC. SciTePress, 2016, pp. 39–48.

[Les+93] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." In: *Neural networks* 6.6 (1993), pp. 861–867.

[LM01] T. Leung and J. Malik. "Representing and recognizing the visual appearance of materials using three-dimensional textons." In: *International journal of computer vision* 43.1 (2001), pp. 29–44.

[LCS11] S. Leutenegger, M. Chli, and R. Y. Siegwart. "BRISK: Binary robust invariant scalable keypoints." In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 2548–2555.

[LKJ09] D. Lewandowski, D. Kurowicka, and H. Joe. "Generating random correlation matrices based on vines and extended onion method." In: *Journal of Multivariate Analysis* 100.9 (2009), pp. 1989 –2001.

[Li00] D. Li. "On Default Correlation: A Copula Function Approach." In: *Journal of Fixed Income* 9.4 (2000), pp. 43–54.

[Li+18] K. Li, W. Tao, X. Liu, and L. Liu. "Iterative image segmentation with feature driven heuristic four-color labeling." In: *Pattern Recognition* 76 (2018), pp. 69–79.

[LAT18] Q. Li, A. Arnab, and P. H. Torr. "Weakly-and semi-supervised panoptic segmentation." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 102–118.

[LWC12] Z. Li, X.-M. Wu, and S.-F. Chang. "Segmentation using superpixels: A bipartite graph partitioning approach." In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 789–796.

[Lin+16] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3159–3167.

[LCY13] M. Lin, Q. Chen, and S. Yan. *Network In Network*. 2013. arXiv: 1312.4400 [cs.NE].

[Lit+17] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. "A survey on deep learning in medical image analysis." In: *Medical image analysis* 42 (2017), pp. 60–88.

[LYT09] C. Liu, J. Yuen, and A. Torralba. "Nonparametric scene parsing: Label transfer via dense scene alignment." In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 1972–1979.

[Liu+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "SSD: Single shot multibox detector." In: *Proc. European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 21–37.

[LDY18] X. Liu, Z. Deng, and Y. Yang. "Recent progress in semantic image segmentation." In: *Artificial Intelligence Review* (2018).

[Liu+14] Z. Liu, W. Zou, L. Li, L. Shen, and O. Le Meur. "Co-saliency detection based on hierarchical segmentation." In: *IEEE Signal Process. Lett* 21.1 (2014), pp. 88–92.

[Llo82] S. Lloyd. "Least squares quantization in PCM." In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[Low99] D. G. Lowe. "Object recognition from local scale-invariant features." In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[Lun+12] D. Lunn, C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter. *The BUGS Book: A Practical Introduction to Bayesian Analysis*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2012.

[Ly+17] A. Ly, M. Marsman, J. Verhagen, R. P. Grasman, and E.-J. Wagenmakers. "A tutorial on Fisher information." In: *Journal of Mathematical Psychology* 80 (2017), pp. 40–55.

[MA15] D. Maclaurin and R. P. Adams. "Firefly Monte Carlo: Exact MCMC with subsets of data." In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[Mal+99] J. Malik, S. Belongie, J. Shi, and T. Leung. "Textons, contours and regions: Cue integration in image segmentation." In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. IEEE. 1999, pp. 918–925.

[MMR05] J.-M. Marin, K. Mengersen, and C. P. Robert. "Bayesian Modelling and Inference on Mixtures of Distributions." In: *Bayesian Thinking*. Ed. by D. Dey and C. Rao. Vol. 25. Handbook of Statistics. Elsevier, 2005, pp. 459 –507.

[MFM04] D. R. Martin, C. C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using local brightness, color, and texture cues." In: *IEEE transactions on pattern analysis and machine intelligence* 26.5 (2004), pp. 530–549.

[Mar+01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. IEEE. 2001, pp. 416–423.

[Maš+14] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, et al. "A benchmark for comparison of cell tracking algorithms." In: *Bioinformatics* 30.11 (2014), pp. 1609–1617.

[Mat+04] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions." In: *Image and vision computing* 22.10 (2004), pp. 761–767.

[MP00] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2000.

[MFE15] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management: Concepts, techniques and tools*. Princeton university press, 2015.

[Mei05] M. Meilă. "Comparing clusterings: an axiomatic view." In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 577–584.

[Mig08] M. Mignotte. "Segmentation by fusion of histogram-based k-means clusters in different color spaces." In: *IEEE Transactions on image processing* 17.5 (2008), pp. 780–787.

[Min+16] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris. "Finely-grained annotated datasets for image-based plant phenotyping." In: *Pattern Recognition Letters* 81 (2016), pp. 80 –89.

[MP69] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

[MH10] V. Mnih and G. E. Hinton. "Learning to detect roads in high-resolution aerial images." In: *European Conference on Computer Vision*. Springer. 2010, pp. 210–223.

[Mur12] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[Nad05] S. Nadarajah. "A generalized normal distribution." In: *Journal of Applied Statistics* 32.7 (2005), pp. 685 –694.

[NS96] L. Najman and M. Schmitt. "Geodesic saliency of watershed contours and hierarchical segmentation." In: *IEEE Transactions on pattern analysis and machine intelligence* 18.12 (1996), pp. 1163–1173.

[Nea03] R. M. Neal. "Slice sampling." In: *Annals of statistics* (2003), pp. 705–741.

[NM65] J. A. Nelder and R. Mead. "A Simplex Method for Function Minimization." In: *The Computer Journal* 7.4 (1965), pp. 308–313.

[Nel06] R. B. Nelsen. *An Introduction to Copulas*. Springer Science & Business Media, 2006.

[NW12] T. M. Nguyen and Q. J. Wu. "Robust student's-t mixture model with spatial constraints and its application in medical image segmentation." In: *IEEE Transactions on Medical Imaging* 31.1 (2012), pp. 103–116.

[NW13] T. M. Nguyen and Q. J. Wu. "Fast and robust spatially constrained Gaussian mixture model for image segmentation." In: *IEEE transactions on circuits and systems for video technology* 23.4 (2013), pp. 621–635.

[Nie15] M. A. Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press USA, 2015.

[Nol03] J. Nolan. *Stable distributions: models for heavy-tailed data*. Birkhauser New York, 2003.

[Ntz11] I. Ntzoufras. *Bayesian modeling using WinBUGS*. Vol. 698. John Wiley & Sons, 2011.

[OD11] S. O'Hara and B. A. Draper. "Introduction to the bag of features paradigm for image classification and retrieval." In: *arXiv preprint arXiv:1101.3354* (2011).

[OG10] M. Ojala and G. C. Garriga. "Permutation tests for studying classifier performance." In: *Journal of Machine Learning Research* 11.Jun (2010), pp. 1833–1863.

[Oqu+15] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. "Is object localization for free?-weakly-supervised learning with convolutional neural networks." In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 685–694.

[Ots79] N. Otsu. "A threshold selection method from gray-level histograms." In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[Ozd+18] O. Ozdemir, T. G. Allen, S. Choi, T. Wimalajeewa, and P. K. Varshney. "Copula Based Classifier Fusion Under Statistical Dependence." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.11 (2018), pp. 2740–2748.

[Pap+15] G. Papandreou, L. Chen, K. Murphy, and A. L. Yuille. "Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation." In: *CoRR* abs/1502.02734 (2015).

[Pas01] G. Paschos. "Perceptually uniform color spaces for color texture analysis: an empirical evaluation." In: *IEEE transactions on Image Processing* 10.6 (2001), pp. 932–937.

[PM00] D. Peel and G. J. McLachlan. "Robust mixture modelling using the t distribution." In: *Statistics and computing* 10.4 (2000), pp. 339–348.

[PTM16] J. Pont-Tuset and F. Marques. "Supervised Evaluation of Image Segmentation and Object Proposal Techniques." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.7 (2016), pp. 1465–1478.

[Pri12] S. J. Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[PHB97] J. Puzicha, T. Hofmann, and J. M. Buhmann. "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval." In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE. 1997, pp. 267–272.

[Qi+16] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. "Augmented feedback in semantic segmentation under image level supervision." In: *European Conference on Computer Vision*. Springer. 2016, pp. 90–105.

[RR04] A. Rahimi and B. Recht. "Clustering with normalized cuts is clustering with a hyperplane." In: *Statistical Learning in Computer Vision* 56 (2004).

[RS61] H. Raïffa and R. Schlaifer. *Applied statistical decision theory*. Studies in managerial economics. Division of Research, Graduate School of Business Adminitration, Harvard University, 1961.

[RTV86] R. Rammal, G. Toulouse, and M. A. Virasoro. "Ultrametricity for physicists." In: *Reviews of Modern Physics* 58.3 (1986), p. 765.

[Ran71] W. M. Rand. "Objective criteria for the evaluation of clustering methods." In: *Journal of the American Statistical association* 66.336 (1971), pp. 846–850.

[RW06] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited, 2006.

[RM03] X. Ren and J. Malik. "Learning a classification model for segmentation." In: *Proceedings Ninth IEEE International Conference on Computer Vision*. Vol. 1. 2003, pp. 10–17.

[Rob77] A. R. Robertson. "The CIE 1976 color-difference formulae." In: *Color Research & Application* 2.1 (1977), pp. 7–11.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 234–241.

[Ros58] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), pp. 386–408.

[RD05] E. Rosten and T. Drummond. "Fusing points and lines for high performance tracking." In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. 2005, 1508–1515 Vol. 2.

[Rou87] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53 –65.

[Rus+15] O. Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge." In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[Sal+18] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, J. M. Alvarez, and S. Gould. "Incorporating Network Built-in Priors in Weakly-Supervised Semantic Segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1382–1396.

[Sal09] F. Salmon. *Recipe for disaster: the formula that killed wall street, Wired Magazine 17.03*. 2009.

[Sal01] D. Salsburg. *The lady tasting tea: How statistics revolutionized science in the twentieth century*. Macmillan, 2001.

[SGH98] S. Sanjay-Gopal and T. J. Hebert. "Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm." In: *IEEE Transactions on Image Processing* 7.7 (1998), pp. 1014–1028.

[SPB10] J. Santner, T. Pock, and H. Bischof. "Interactive Multi-Label Segmentation." In: *Proceedings 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand*. 2010.

[Sch01] C. Schmid. "Constructing models for content-based image retrieval." In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. 2001, II–39–II–45 vol.2.

[Sch15] J. Schmidhuber. "Deep learning in neural networks: An overview." In: *Neural Networks* 61 (2015), pp. 85 –117.

[Sch78] G. Schwarz. "Estimating the Dimension of a Model." In: *Ann. Statist.* 6.2 (Mar. 1978), pp. 461–464.

[Sco79] D. W. Scott. "On optimal and data-based histograms." In: *Biometrika* 66.3 (1979), p. 605.

[Sco15] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[Set+17] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, et al. "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge." In: *Medical image analysis* 42 (2017), pp. 1–13.

[SNG07] G. Sfikas, C. Nikou, and N. Galatsanos. "Robust image segmentation with mixtures of Student's t-distributions." In: *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. Vol. 1. IEEE. 2007, pp. I–273.

[SNG08] G. Sfikas, C. Nikou, and N. Galatsanos. "Edge preserving spatially varying mixtures for image segmentation." In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–7.

[Sfi+10] G. Sfikas, C. Nikou, N. Galatsanos, and C. Heinrich. "Spatially varying mixtures incorporating line processes for image segmentation." In: *Journal of Mathematical Imaging and Vision* 36.2 (2010), pp. 91–110.

[Sfi+11] G. Sfikas, C. Nikou, N. Galatsanos, and C. Heinrich. "Majorization-minimization mixture model determination in image segmentation." In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE conference on*. IEEE. 2011, pp. 2169–2176.

[She+15] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3982–3991.

[SM00] J. Shi and J. Malik. "Normalized cuts and image segmentation." In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[SJC08] J. Shotton, M. Johnson, and R. Cipolla. "Semantic texton forests for image categorization and segmentation." In: *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[SL15] P. Sibbertsen and H. Lehne. *Statistik: Einführung für Wirtschafts- und Sozialwissenschaftler*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[Sil+12] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. "Indoor segmentation and support inference from rgbd images." In: *European Conference on Computer Vision*. Springer. 2012, pp. 746–760.

[SZ14] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556* (2014).

[Skl59] M. Sklar. "Fonctions de repartition an dimensions et leurs marges." In: *Publ. Inst. Statist. Univ. Paris* 8 (1959), pp. 229–231.

[Smi39] N. V. Smirnov. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples." In: *Bull. Math. Univ. Moscou* 2.2 (1939), pp. 3–14.

[SAC07] M. D. Smucker, J. Allan, and B. Carterette. "A comparison of statistical significance tests for information retrieval evaluation." In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM. 2007, pp. 623–632.

[Stu08] Student. "The Probable error of a mean." In: *Biometrika* 6.1 (1908), pp. 1–25.

[SHL18] D. Stutz, A. Hermans, and B. Leibe. "Superpixels: An evaluation of the state-of-the-art." In: *Computer Vision and Image Understanding* 166 (2018), pp. 1 – 27.

[Swa+15] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna." In: *Scientific data* 2 (2015), p. 150026.

[SWW17] J.-H. Syu, S.-J. Wang, and L.-C. Wang. "Hierarchical Image Segmentation Based on Iterative Contraction and Merging." In: *IEEE Trans. Image Processing* 26.5 (2017), pp. 2246–2260.

[Sze10] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[Tan+18a] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers. "Normalized cut loss for weakly-supervised CNN segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1818–1827.

[Tan+18b] M. Tang, D. Marin, I. Ben Ayed, and Y. Boykov. "Kernel Cuts: Kernel and Spectral Clustering Meet Regularization." In: *International Journal of Computer Vision* (2018).

[TK09] S. Theodoridis and K. Koutroumbas. *Pattern Recognition (Fourth Edition)*. Fourth Edition. Boston: Academic Press, 2009, pp. 765 –862.

[The98] P. Theodossiou. "Financial Data and the Skewed Generalized T Distribution." In: *Management Science* 44.12-part-1 (1998), pp. 1650–1661.

[TM99] L. Tierney and A. Mira. "Some adaptive Monte Carlo methods for Bayesian inference." In: *Statistics in medicine* 18.1718 (1999), pp. 2507–2515.

[Tit+85] D. Titterington, P. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Applied section. Wiley, 1985.

[Tor+14] C. Tortora, B. C. Franczak, R. P. Browne, and P. D. McNicholas. "A mixture of coalesced generalized hyperbolic distributions." In: *arXiv preprint arXiv:1403.2332* (2014).

[TKW16] J. Townsend, N. Koep, and S. Weichwald. "Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation." In: *Journal of Machine Learning Research* 17.137 (2016), pp. 1–5.

[UH05] R. Unnikrishnan and M. Hebert. "Measures of similarity." In: *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*. Vol. 1. IEEE. 2005, pp. 394–394.

[UPH07] R. Unnikrishnan, C. Pantofaru, and M. Hebert. "Toward objective evaluation of image segmentation algorithms." In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007), pp. 929–944.

[Was13] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[Wei+17] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1568–1576.

[Wil45] F. Wilcoxon. "Individual Comparisons by Ranking Methods." In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83.

[Wil+17] T. Wilhelm, R. Grzeszick, G. A. Fink, and C. Wöhler. "From Weakly Supervised Object Localization to Semantic Segmentation by Probabilistic Image Modeling." In: *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2017, pp. 1–7.

[WW17a] T. Wilhelm and C. Wöhler. "Boundary aware image segmentation with unsupervised mixture models." In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3325–3329.

[WW17b] T. Wilhelm and C. Wöhler. "On the suitability of different probability distributions for the task of image segmentation." In: *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. 2017, pp. 1–6.

[Wil+19] T. Wilhelm, R. Grzeszick, G. Fink, and C. Wöhler. "Unsupervised Learning of Scene Categories on the Lunar Surface." In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,* INSTICC. SciTePress, 2019, pp. 614–621.

[WW16] T. Wilhelm and C. Wöhler. "Flexible Mixture Models for Colour Image Segmentation of Natural Images." In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2016, pp. 1–7.

[WW17c] T. Wilhelm and C. Wöhler. "Improving Bayesian Mixture Models for Colour Image Segmentation with Superpixels." In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, (VISIGRAPP 2017)*. INSTICC. SciTePress, 2017, pp. 443–450.

[Wis28] J. Wishart. "The generalised product moment distribution in samples from a normal multivariate population." In: *Biometrika* 20.1/2 (1928), pp. 32–52.

[Woh+18] K. Wohlfarth, C. Schröer, M. Klaß, S. Hakenes, M. Venhaus, S. Kauffmann, T. Wilhelm, and C. Wöhler. "Dense Cloud Classification on Multispectral Satellite Imagery." In: *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*. 2018, pp. 1–6.

[WSH16] Z. Wu, C. Shen, and A. v. d. Hengel. "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition." In: *arXiv preprint arXiv:1611.10080* (2016).

[XK17] X. Xia and B. Kulis. "W-Net: A Deep Model for Fully Unsupervised Image Segmentation." In: *arXiv preprint arXiv:1711.08506* (2017).

[Yan+08] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. "Unsupervised segmentation of natural images via lossy data compression." In: *Computer Vision and Image Understanding* 110.2 (2008), pp. 212–225.

[Yan+13] Y. Yang, S. Han, T. Wang, W. Tao, and X.-C. Tai. "Multilayer graph cuts based unsupervised color–texture image segmentation using multivariate mixed student's t-distribution and regional credibility merging." In: *pattern recognition* 46.4 (2013), pp. 1101–1124.

[YQG17] S. Yin, Y. Qian, and M. Gong. "Unsupervised hierarchical image segmentation through fuzzy entropy maximization." In: *Pattern Recognition* 68 (2017), pp. 245 –259.

[Yu+18] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. *BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling*. 2018. arXiv: `1805.04687 [cs.CV]`.

[YFL15] Y. Yu, C. Fang, and Z. Liao. "Piecewise Flat Embedding for Image Segmentation." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1368–1376.

[YWL12] J. Yuan, D. Wang, and R. Li. "Image segmentation using local spectral histograms and linear regression." In: *Pattern Recognition Letters* 33.5 (2012), pp. 615–622.

[YWC15] J. Yuan, D. Wang, and A. M. Cheriyadat. "Factorization-based texture segmentation." In: *IEEE Transactions on Image Processing* 24.11 (2015), pp. 3488–3497.

[ZFG08] H. Zhang, J. E. Fritts, and S. A. Goldman. "Image segmentation evaluation: A survey of unsupervised methods." In: *Computer Vision and Image Understanding* 110.2 (2008), pp. 260 –280.

[Zha+18] Z. Zhang, F. Xing, H. Wang, Y. Yan, Y. Huang, X. Shi, and L. Yang. "Revisiting graph construction for fast image segmentation." In: *Pattern Recognition* 78 (2018), pp. 344 –357.

[Zha+15] R. Zhao, W. Ouyang, H. Li, and X. Wang. "Saliency Detection by Multi-Context Deep Learning." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[Zho+16] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. "Learning Deep Features for Discriminative Localization." In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Zho+17] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. "Scene Parsing through ADE20K Dataset." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[ZZW13] H. Zhou, J. Zheng, and L. Wei. "Texture aware image segmentation using graph cuts and active contours." In: *Pattern Recognition* 46.6 (2013), pp. 1719–1733.

[Zhu+17] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. "Deep learning in remote sensing: A comprehensive review and list of resources." In: *IEEE Geoscience and Remote Sensing Magazine* 5.4 (2017), pp. 8–36.

[Zhu+09] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.