The Discriminative Generalized Hough Transform for Localization of Highly Variable Objects and its Application for Surveillance Recordings

by

Georg Ferdinand Hahmann

Thesis submitted in fulfillment of the requirements for the degree of

Doctor of Engineering (Dr.-Ing.)

at the

Technical Faculty of the
Christian-Albrechts-University Kiel

Supervisors:
Prof. Dr. Hauke Schramm
Prof. Dr. Reinhard Koch

2020

1. Gutachter: Prof. Dr. rer. nat. Hauke Schramm
Institut für Informatik
Christian-Albrechts-Universität zu Kiel
2. Gutachter: Prof. Dr.-Ing. Reinhard Koch
Institut für Informatik
Christian-Albrechts-Universität zu Kiel

Datum der mündlichen Prüfung: 10. Dezember 2020

Abstract

This work is about the localization of arbitrary objects in 2D images in general and the localization of persons in video surveillance recordings in particular. More precisely, it is about localizing specific landmarks. Thereby the possibilities and limitations of localization approaches based on the Generalized Hough Transform (GHT), especially of the Discriminative Generalized Hough Transform (DGHT) will be evaluated. GHT-based approaches determine the number of matching model and feature points and the most likely target point position is given by the highest number of matching model and feature points. Additionally, the DGHT comprises a statistical learning approach to generate optimal DGHT-models achieving good results on medical images. This work will show that the DGHT is not restricted to medical tasks but has issues with large target object variabilities, which are frequent in video surveillance tasks.

As all GHT-based approaches also the DGHT only considers the number of matching model-feature-point-combinations, which means that all model points are treated independently. This work will show that model points are not independent of each other and considering them independently will result in high error rates. This drawback is analyzed and a universal solution, which is not only applicable for the DGHT but all GHT-based approaches, is presented. This solution is based on an additional classifier that takes the whole set of matching model-feature-point-combinations into account to estimate a confidence score. On all tested databases, this approach could reduce the error rates drastically by up to 94.9%.

Furthermore, this work presents a general approach for combining multiple GHT-models into a deeper model. This can be used to combine the localization results of different object landmarks such as mouth, nose, and eyes. Similar to Convolutional Neural Networks (CNNs) this will split the target object variability into multiple and smaller variabilities.

A comparison of GHT-based approaches with CNNs and a description of the advantages, disadvantages, and potential application of both approaches will conclude this work.

Zusammenfassung

Diese Arbeit beschäftigt sich im Allgemeinen mit der Lokalisierung von Objekten in 2D Bilddaten und im Speziellen mit der Lokalisierung von Personen in Videoüberwachungsaufnahmen. Genauer gesagt handelt es sich hierbei um die Lokalisierung spezieller Landmarken. Dabei werden die Möglichkeiten und Limiterungen von Lokalisierungsverfahren basierend auf der Generalisierten Hough Transformation (GHT) untersucht, insbesondere die der Diskriminativen Generalisierten Hough Transformation (DGHT). Bei GHT-basierten Ansätze wird die Anzahl an übereinstimmenden Modelpunkten und Merkmalspunkten ermittelt und die wahrscheinlicheste Objekt-Position ergibt sich aus der höchsten Anzahl an übereinstimmenden Model- und Merkmalspunkte. Die DGHT umfasst darüber hinaus noch ein statistisches Lernverfahren, um optimale DGHT-Modele zu erzeugen und erzielte damit auf medizinischen Bilder und Anwendungen sehr gute Erfolge. Wie sich in dieser Arbeit zeigen wird, ist die DGHT nicht auf medizinische Anwendungen beschränkt, hat allerdings Schwierigkeiten große Variabilität der Ziel-Objekte abzudecken, wie sie in Überwachungsszenarien zu erwarten sind.

Genau wie alle GHT-basierten Ansätze leidet auch die DGHT unter dem Problem, dass lediglich die Anzahl an übereinstimmenden Model- und Merkmalspunkten ermittelt wird, was bedeutet, dass alle Modelpunkte unabhängig voneinander betrachtet werden. Dass Modelpunkte nicht unabhängig voneinander sind, wird im Laufe dieser Arbeit gezeigt werden, und die unabhängige Betrachtung führt gerade bei sehr variablen Zielobjekten zu einer hohen Fehlerrate. Dieses Problem wird in dieser Arbeit grundlegend untersucht und ein allgemeiner Lösungsansatz vorgestellt, welcher nicht nur für die DGHT sondern grundsätzlich für alle GHT-basierten Verfahren Anwendung finden kann. Die Lösung basiert auf der Integration eines zusätzlichen Klassifikators, welcher die gesamte Menge an übereinstimmenden Model- und Merkmalspunkten betrachtet und anhand dessen ein zusätzliches Konfidenzmaß vergibt. Dadurch konnte auf allen getesteten Datenbanken eine deutliche Reduktion der Fehlerrate erzielt werden von bis zu 94.9%.

Darüber hinaus umfasst die Arbeit einen generellen Ansatz zur Kombination mehrere GHT-Model in einem tieferen Model. Dies kann dazu verwendet werden, um die Lokalisierungsergebnisse verschiedener Objekt-Landmarken zu kombinieren, z. B. die von Mund, Nase und Augen. Ähnlich wie auch bei Convolutional Neural Networks (CNNs) ist es damit möglich über mehrere Ebenen unterschiedliche Bereiche zu lokalisieren und somit die Variabilität des Zielobjektes in mehrere, leichter zu handhabenden Variabilitäten aufzuspalten.

Abgeschlossen wird die Arbeit durch einen Vergleich von GHT-basierten Ansätzen mit CNNs und einer Beschreibung der Vor- und Nachteile und mögliche Einsatzfelder beider Verfahren.

Acknowledgements

During this work, I received the support of many people and I would like to express my gratitude to them. Especially,

Hauke Schramm

Reinhard Koch

Carsten Meyer

Carsten Rosemann

Nina Hafer-Hahmann

Cris Lovell-Smith

Heike Ruppertshofen

Gordon Böer

Ralf Stannarius

Eric Gabriel

Inga Berger

I would also like to thank my family and my friends who supported me throughout this time.

LaTeX code inspired by the LaTeX Thesis Template by Manuel Kuehner
www.bedienhaptik.de/latex-template/

Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

This work was funded by the Innovation Foundation Schleswig-Holstein under the grant 2010-90H and the Central Innovation Program SME from the Federal Ministry for Economic Affairs and Energy.

Contents

Contents

List of Symbols

$\mathscr{X}$	Feature image
κ	Image height
$\mathbf{q}$	Position in image
$\hat{\mathbf{q}}_n$	Annotated ground truth landmark position of image n
$\tilde{\mathbf{q}}_n$	Predicted target landmark position of image n
Υ	Landmark
$\varnothing$	Mean value
sd	Standard deviation
d	Distance function
$\mathbf{x}$	Feature point
$\phi_{\mathbf{x}_l}$	Value of feature point
$\mathscr{H}$	Hough-space
$\mathbf{c}$	Hough-cell
$\hat{\mathbf{c}}_n$	Annotated ground truth target Hough-cell of image n
$\tilde{\mathbf{c}}_n$	Predicted target Hough-cell of image n
ϱ	Hough-space quantization
$\mathscr{M}$	(D)GHT model
$\mathbf{m}$	Model point
$\phi_{\mathbf{m}_j}$	Value of model point
λ_j	Weight of model point
Λ	Weights of all model points
g	Classification function
$\mathbf{r}$	Feature vector
$\mathbf{f}$	Feature vector
f_j	Feature function
r_j	Feature function
ε	Error calculation function
ϵ_n	Error of image n
$\ddot{\epsilon}_n$	Normalized error of image n
Γ	Allowed error tolerance
Ξ	Localizatin accuracy
ϑ	SCM parameter for neighboorhood size
Ω_r	Regular shape
Ω_i	Irregular shape
ξ_1	Allowed error distance for regular shape
ξ_2	Minimal error distance for irregular shape
N	Number of images
L	Number of landmarks
Ψ	Domain definition
ψ	Domain definition

List of Figures

List of Tables

Chapter 1

Introduction

Computer Vision

The visual sense is the most important sense for humans. Therefore, Computer Vision, the machine understanding of images, has a long tradition going back to 1966. In this year, the MIT planned image understanding as a summer project [150] with mostly a foreground/background segmentation with the final task of object identification, naming "objects by matching them with a vocabulary of known objects". Today, this summer project is often used as an example of how easy and natural visual (object) recognition and identification comes to humans resulting in underestimating that the underlying processes are very complex and insufficiently understood [195, 200]. Computer Vision is a more complex task than it seems at first glance.

Object recognition

One subfield of Computer Vision is object recognition. The recognition of objects is mostly important because the understanding of images is always based on an understanding of the visible objects and their interactions. However, the definition of an object is very unclear since everything can be considered as an object. Therefore, a system returning all objects in an image is impossible, while returning foreground or salient objects is possible, but defining them is highly subjective or task dependent [48].

Object localization

This work deals with the task of object localization, more precisely, the localization of specific key points, or landmarks, on a specific object. Which landmarks and objects should be located is task specific circumventing the problem of a potential definition of an object in general. The challenge here is the precision of the localization. Landmark localization is frequently a first step to perform other tasks, e.g. the detection of facial landmarks such as eyes, mouth, nose etc. can be used for face recognition [98], gender classification [217, 130, 176] or driver drowsiness detection [88, 58].

History of Computer Vision

In the early years, Computer Vision consisted of the implementation of heuristic rules trying to describe what a certain object in a scene would have to look like. In 2001, Viola&Jones [206] used a machine learning technique, Adaboost, for image processing, which can be seen as the beginning of a new epoch of Computer Vision. With the usage of machine learning techniques, it is not necessary to develop heuristic rules. In this case, objects are described automatically but based on manually selected feature extraction methods. Therefore, the image preprocessing, especially feature extractions, remains an important aspect. For example Viola&Jones used Haar-like features. Following their invention, there is a long history of developments trying to find and analyze good feature extraction methods (see Section 2.1.1). The DGHT (see Section 5.1.2), which is mainly used in this work, falls within this type of object detection approaches. In 2012 the success of AlexNet [109], a Convolution Neural Network (CNN), revolutionised the field of Computer Vision yet again after the introduction of machine learning techniques. CNNs have the advantage that feature extraction is also part of the learning

process, allowing joint optimization of the whole process pipeline. To sum up, the development effort has been increasingly reduced by new improvements. Initially, the definition of heuristic rules for how an object looks like (e.g. eyes have a dark, circular pupil) required high development effort and a lot of expertise for each task. With the usage of machine learning approaches, the development effort was reduced to feature extraction and combination with an appropriate machine learning approach. Therefore, more general systems could be developed, which could easily be trained for a larger range of tasks. Nowadays, with the success of CNNs, most of the time only the input and the expected output has to be defined. Whereas the required development effort has been reduced, the drawback is that the number of training images needed has significantly increased. Heuristic rules need no or only a small number of images for development. Machine learning techniques need more training images and the more general a system is, the more training images are required so that the learning algorithm can select the correct parameters.

Task definition

The main task in this work will be the localization of people and faces in video surveillance recordings. There are many tasks for which the localization of people is required, such as analyzing customer behavior [216], predicting the required number of checkouts in stores, counting people [182], or improving alarm systems. Although the main task is related to human and face localization, a major goal of this work is the development of object localization approaches which can easily be adapted to new tasks. Therefore, a medical task, the precise localization of epiphyses in left hand radiographs, will also be addressed here to ensure the generality and transferability of the outcomes. Nevertheless, the main constraints and requirements are derived from the task of person detection in video surveillance recordings.

Project requirements

This work focuses on developing a detection algorithm for surveillance. In line with this task, there were two project specific requirements, which had to be considered:

Number of training images

1. The number of training images is limited. In this project, obtaining real data was difficult due to privacy laws and the costs of annotating these data. Therefore, at most only a few thousand training images were available. Though big companies like Google or Facebook may have the resources to label hundreds of thousands of images, for small or medium size companies this may be a large investment. Furthermore, publicly available databases are mostly restricted to research work only and do not allow for commercial products.

Feature extraction

2. Since people are identifiable in the recordings, this kind of data falls under privacy protection laws. Hence, only edge images, generated by applying the Canny Edge Detector [17], were available and could be used in this project. This also allowed the use of a cloud concept, whereby the data is transferred to a server for faster processing, without privacy concerns.

Requirements in scientific context

Both requirements are unusual for current scientific developments, since due to the Internet publicly available databases are increasing in size. Furthermore, during the last one or two decades, the opinion has taken hold that improving feature extraction is more constructive than improving algorithms on bad features.

Method selection

As this work aims to use object localization approaches easily adaptable to new tasks, heuristic approaches are not suitable due to their high demand on development effort for each task and task specific knowledge. At the same time, the limited number of available training images

and the restriction to Canny Edge features in our settings, precludes the usage of CNNs, whose breakthrough occurred after the beginning of this work [109]. The strong power of CNNs results from learning convolution filters which require the original image as input.

Discriminative Generalized Hough Transform

On these grounds, the Discriminative Generalized Hough Transform (DGHT) [178, 133, 172, 169] will be used in this work. The DGHT has been developed for object localization in medical images such as knee localization in long-leg radiographs, mammilla localization in mammographies or femur localization in MRIs and has achieved good results with a small number of training images in this context [169]. Furthermore, the DGHT has been developed using Canny Edge features albeit other features can also be used. However, the use of the DGHT has been mostly restricted to medical image processing. In this context, the variability of the target object is comparatively small and the background is in general very uniform. In contrast, the variability of people in surveillance recordings is very large and comprises for example different poses and sizes (infants, children, adults, different distances to the camera). Furthermore, the background is very challenging with many static and non-static objects, varying lightning conditions, etc. This results in many edge features visible in the background which could confuse the algorithm, resulting in a high number of false positives.

Voting-based approaches

The DGHT is a voting-based method. This means that a model represents the shape of the target object by a set of model points in relation to the target landmark. For each feature (here edge) point all matching model points vote for potential target point locations and the greater the number of votes the more likely the object is at the corresponding location. Obviously, an important questions is how to generate such a model. In the DGHT, the model generation is based on overlaying labeled training images on the target point location. As long as the variability of the target object is small and/or the background is at least static and therefore learnable, this approach works very well [169]. However, a large target object variability leads to a large model with many model points required to cover all the different variations. This increases the likelihood of an accidental match between the model and the background. Similarly, a variable background results in many edges and therefore, again, the risk of an accidental match with the model increases.

Main tasks of this work

This work addresses this drawback of the DGHT, mainly by investigating two different, but complementary approaches. The first one is to analyze the structure of model points that voted for a certain hypothesis. Because model points vote independently from each other, it is possible that model points from mutually exclusive variations votes for the same hypothesis. By analyzing the voting structure, votes from mutually exclusive variations are weighted down. The second approach is a stacked combination of multiple (D)GHT models, where each level extracts different detailed features, i.e. the lower-level models localize specific landmarks whereas the higher-level models combine the localization results. This concept, which is also applied in CNNs, splits the complete variation, present in the training corpus, into multiple but smaller variations.

1.1 Project field

Rosemann Software GmbH

The outcome of this work should support the Rosemann Software GmbH in extending the software CamIQ. CamIQ is an intelligent recording and analyzing system allowing the handling

of large numbers of cameras simultaneously and informing the operator only about relevant events.

Privacy law

Due to strong privacy laws, video surveillance is strongly regulated. For example, the period for storing such videos is very short and does not allow long term customer analyzes. However, most of the times, it is not relevant to identify people, i.e. video streams in which customer's faces are blurred out work perfectly fine as long as the customer's actions are recognizable. Therefore, this work intends to localize faces in such surveillance videos. These faces can be encrypted or blurred out so that an identification of the person is not possible anymore. However, in suspected cases of criminal acts it is possible to decrypt the face region to identify the person. The ability to decrypt the face is required for security reasons, but after the time allowed to save the data, this possibility to decrypt the data can be destroyed so that privacy laws no longer apply for the remaining part of the video.

CamIQ Cloud Services

The current version of CamIQ allows the transfer of data to a Cloud service for further analysis. However, transferring video recordings is critical in terms of data and privacy protection and requires a certain amount of network bandwidth. The usage of only Canny Edge features could again be a potential solution. Transferring only edge images increases data protection and decreases the required amount of bandwidth. Since certain applications, requiring person or face detection, should also be applicable on the Cloud, this underlines the restriction to Canny Edge features in the present work.

Chapter 2
State of the Art

2.1 Review of generic machine learning object detection approaches

General object detection procedures

In the last two decades, there has been a lot of scientific development in the field of generic machine learning object detection. Still, in general, each detection and localization task can be considered as a binary classification task in which each potential hypothesis, which can be either a bounding box around the object or specific landmark coordinates, will be classified into one of the two classes "Target-Object" or "Non-Target-Object". Therefore, in some way, each object detection approach works as follows:

1. Potential hypotheses are selected, which can be done either with or without taking the image into account. Around each potential hypothesis a Region of Interest (ROI) is extracted. For detection of object bounding boxes, the bounding box is usually identical to the ROI, but in theory it could also be different.

   Proposal generation

2. Either the whole input image or at least the potential ROIs are transformed into one or multiple features. These features are described by a value vector and potentially a coordinate vector.

   Feature extraction

3. Each potential ROI is classified into "Target-Object" or "Non-Target-Object", whereby the input of the classifier are the extracted features from the previous step.

   Proposal classification

Independent developments

Until the success of CNN based object detection, proposal generation was usually an independent step. The feature extraction and the object classification were mostly developed and optimized together, even if sometimes the order of proposal generation and feature extraction could be interchanged. With CNN based object detection approaches, a joint optimization of all three steps has now become possible. Therefore, here, different object detection approaches will be reviewed in general by considering all steps together.

Proposal generation

Apart from CNN based approaches, the potential hypotheses are selected mostly using the sliding window approach [74, 83], which slides the ROI over the image. Since the sliding window approach is little more than a brute force approach, there are also more sophisticated methods, like measuring the objectness, the probability that a given area contains a salient or foreground object, of image parts [131, 3, 3, 24], Selective Search [203], Active Search Strategy [75], or task specific heuristical approaches [23]. These approaches, however, only reduce the number of proposals and therefore the computational power required and usually

do not improve the quality of the object detection tasks. This may be the reason why the sliding window approach is still the most used proposal generator.

2.1.1 Rigid Object Detection

Viola & Jones

In 2001 Viola & Jones (VJ) [206] proposed a face detector by using Haar-Features [149] with a boosted cascade of classifiers. In contrast to previous approaches, VJ uses a machine learning technique, Adaboost [57], in a cascade architecture, for ROI classification. Therefore, it is the first well-known generic object detection approach based on machine learning techniques.

Haar-like features

VJ considered as features adjacent rectangular regions in which the pixel intensities are summed up and the difference between the sum of intensities of both regions is the feature value, mostly referred as Haar-like features. Using an integral image, these features can be calculated very fast. VJ use only four different rectangular combinations, but theses combinations are used in different scales and positions inside the ROI so that in a 24×24 pixel ROI over 180,000 features can be generated.

Histogram of oriented Gradients

In 2005, Dalal and Triggs published a new feature calculating method, called Histogram of oriented Gradients (HOG) [35]. As the name suggests, it generates a histogram of gradient orientation on a dense grid of uniformly spaced cells. The HOG features are similar to edge orientation histograms [55, 56], scale-invariant feature transform (SIFT) [124], and shape contest [9]. The basic concept even dates back to 1986 [135]. Dalal and Triggs, however, were the first to demonstrate the usefulness for object detection in combination with a Support Vector Machine (SVM) [26] on the task of human detection [35].

Integral Channel Features

At first glance, VJ and HOG+SVM seem very different. However, in 2009, Dollár *et al.* presented Integral Channel Features (ICF) [42] which generalized the Haar and HOG features to a common base. Initially, Dollár *et al.* calculated multiple image channels with linear and non-linear transformations of the input image, which are for example the gray values required for VJ or gradient histograms for HOG features. Using integral images from these channels, it is possible to efficiently calculate different features like local sums, histograms or Haar-Features. With a single rectangle on the gradient histogram channel, it is possible to also approximate HOG features. Consequently, Dollár *et al.* made a deep evaluation of different features. A conclusion of this paper, which was also confirmed in subsequent publications [219, 11], was that gradient histograms and color channels in combination achieve the best results. This procedure was subsequently mostly refered to as HOG+LUV channels. Up until now, these channels are considered the best feature channels for generic object detection. Nevertheless, it is still unclear, what good features are, which is also a problem for deep learning [11].

First- vs. higher-order features

Dollár *et al.* also evaluated different types of features. First-order features are single rectangles randomly placed inside the ROI, which is, e.g. , used for the calculation of HOG features. Higher-order features are weighted combinations of multiple first-order features such as Haar-Features from VJ which consist of two or more rectangles weighted with 1 and -1 from the grey channel. Another important finding in [42] was that using higher-order features leads to only a marginal improvement in detection accuracy, albeit [219, 220] presented subsequent results in which higher-order features still improved detection quality.

The Fastest Pedestrian Detector in the West

With "the Fastest Pedestrian Detector in the West" [41] Dollár *et al.* extended the ICF. Until then, size variability had been addressed by scaling the input image multiple times and calculating the features for each scale. Therefore, calculating the features was the computational bottleneck. In [41, 40] it was shown that the ICF is partially size invariant, more precisely that from the features calculated for a single scale the features for nearby scales can be approximated in a faster way without significantly losing detection accuracy.

Feature cell pattern

In the original implementation of HoG Features according to [35], the cells are arranged in a regular pattern whereas in ICF the cell pattern can be randomly selected. [10] evaluated the influence of the cell patterns in depth and came to the conclusion that it is better to discriminatively learn an irregular pattern instead of using a regular cell pattern. The best results can be achieved by using all possible rectangles and allowing the classifier to decide which ones to use, as long as training time and computational resources allow. This analysis was done on the task of pedestrian detection, but the outcome was also confirmed in [134] on the task of face detection.

Higher-order features

The idea of using higher-ordered features, as described in ICF, was addressed again in [219] by using Haar-like features on gradient histograms and color channels. The Haar-Features used were called "informed Haar-like features" and were heuristically designed only for the task of pedestrian detection. However, this idea evolved in [220] by designing filtered channel features. Whereas in ICF and subsequently, the pixel intensities of rectangles are summed up on an integral image, [220] generalized this framework by integration of a convolution filter bank. This filter bank was not restricted to only the sum of rectangles and thus this framework integrates the approaches presented in [40, 10, 11, 219, 140]. The best results where achieved with the so called "Checkerboards" filters, which are similar to Haar-like features on the HOG+LUV channels.

2.1.2 Part-based Object Detection

Part-based object detection

For rigid object detection, the target object is considered as a whole. In contrast, part-based object detection subdivides the target object into multiple parts. Each part is detected independently and thereafter the different detection results are combined to detect the whole target object. The advantage is that the detection process is divided into multiple steps which also distributes the object variability. Human detection presents a good example. The human body consists of hands, feet, a head and so on. Normally, all of these parts exist, but how they are arranged in relation to each other is very flexible. Hence, if the variability within the different parts and the variability in the arrangement of these parts are modeled independently, the resulting task becomes less complex.

Deformable Part Models

The best known approach is "Deformable Part Models" (DPM) [49] which is a direct extension of the work of [35] by combining HoG-Features with a Support Vector Machine. To train different model parts, [49] use an extended SVM version, called "Latent-SVM", which learns latent variables. These latent variables are the model parts and therefore it is possible to train model parts without manually specifying these parts. In [151] a faster DPM was introduced for the task of pedestrian detection.

Rigid vs. part based object detection

At least until 2014, DPM was the defacto standard for generic object detection and a properly trained vanilla DPM reached top performances for face detection, albeit detectors based

on rigid templates also reached similarly good performances[134]. Overall, Mathias *et al.* [134] concludes that DPM performs better at generalizing unseen views but with the enlarged databases common at that time, modeling of object parts is no longer critical [134].

2.1.3 Generalized Hough Transform

Generalized Hough Transform

The Generalized Hough Transform (GHT) [6], introduced by Ballard in 1981, is a general and well known voting-based object localization approach. As image features, Ballard used the Canny Edge Detector to first transform the original image into a feature image. This means that the target object is described by a model, consisting of model points, whereby each model point represents a small object part in relation to a reference point, the target landmark. Furthermore, each edge point represents a separate feature location which could, in combination with a model point, represent a part of the target object. Therefore, each model-feature point combination votes for a possible target point location. More precisely, a model-feature point combination votes in a parameter space, called Hough-space, for possible model transformation parameters like translation, scaling or rotation. Since each additional model transformation requires an additional dimension in the Hough-space, the computational complexity increases exponentially with the number of transformation parameters used and therefore most of the current GHT-based approaches restrict themselves to a translation only. Note, GHT-based approaches detect specific landmark points instead of the bounding box around the whole object. However, since landmark position and bounding box are correlated to some extend, it is generally possible to determine one from the other.

Proposal generation and classification

The concept of the GHT is different from the approaches discussed previously. In the GHT no direct proposal generator is required, though it is possible to see the GHT as a sliding window approach in which the model is translated over the potentially transformed (feature) image and at the position of the reference point, the number of matching model and feature points is counted. Furthermore, the discrimination between "Target object" and "Non-Target object" is not performed by a standard classification method like Adaboost or SVM, but by a very simple classifier: A threshold that is applied to votes for whether the localization in question is a target point or not.

GHT for object detection

The GHT has been used for many different approaches like [1, 129]. [102, 169] provide a good overview of the GHT and its applications. However, two approaches are particularly worth mentioning: Hough-Forests and the Discriminative Generalized Hough Transform.

Hough-Forests

Gall *et al.* developed the Hough-Forests [68, 69], as an enhancement of the Implicit Shape Model (ISM) from Leible *et al.* [114, 116, 115]. The main difference between Hough-Forests and the original GHT is the feature extraction. Whereas in [6], edge features were used, Hough-Forests use a sophisticated feature extraction. A Random-Forest classifier [14] learns a direct mapping between the appearance of image patches and votes in the Hough-space. The idea of modeling the appearance of image patches as features for a GHT similar voting procedure can also be found in other, previous publications [38, 129, 144]. Hough-Forests have been used for various applications such as mouth localization, classification of facial expressions [47] or object detection in medical images [28, 30, 29, 173]. Lindner *et al.* [121] and Donner *et al.* [43] slightly adapted the basic algorithm for medical image analysis.

Feature extraction using Hough-Forests

The main success of the Hough-Forests is based on the feature extraction. Whereas 2009, to my best knowledge, any other object detection approach tried to optimize the feature extraction by manually defined features, the Hough-Forests already used a machine learning approach for feature extraction. Therefore, in [148] a method for fast keypoint recognition is presented based on the idea of the Hough-Forests. Schulter *et al.* [180] use features calculated similarly as in Hough-Forests for a foreground vs. background classification in combination with a bounding box regressor for accurate object detection. Furthermore, the basic idea of the Hough-Forests can be found in some face alignment applications [166, 103], whereby these applications do not detect facial landmark points directly but only refine the positions of points estimated from a face bounding box.

Discriminative Generalized Hough Transform

Whereas Hough-Forests utilize a sophisticated feature extractor, the Discriminative Generalized Hough Transform (DGHT) [178, 133, 172, 169] can also be used with simple edge features. Instead of a sophisticated feature extractor, the DGHT uses a steepest descent approach to weight each model point individually according to its relevance for correct localization and to reduce false positives. Furthermore, the DGHT includes an iterative training approach, in which an a-priori model is generated by overlaying features from a few training images, weighting the model points and determining images with a high localization error. Since the model does not fit well on these images, they might contain unseen target object variations and therefore the model will be extended by features from these images in the next iteration. The DGHT was developed for the detection of anatomical structures in medical images [169, 173], but can also be used for facial landmark detection [81] or pedestrian detection [61, 64].

Independent model point voting

The GHT as well as its extensions handle each model point independently. During testing, this reduces the complexity of the task and allows for parallel voting. Furthermore, it also allows parts from different training samples to support the same localization hypotheses which is useful for independent object variations, e.g. different shapes of nose, eyes and mouth, and has made the model point weighting in the DGHT possible. At the same time, this advantage is also the DGHT's biggest disadvantage since it also allows parts from mutually exclusive variations, like different head poses, to support the same localization hypothesis. In [161] it was assumed that this was the main reason for the poor performance of GHT-based approaches. Therefore, this problem has been addressed by using latent variables to enforce consistency among votes [161], further assessments of the voting pattern [160, 13], or by clustering the training images [36] to reduce the variability in each cluster. Clustering of training images has the disadvantage that for each cluster a specific model is required, which to some extend results in the loss of the advantages of the individual voting procedure.

2.1.4 CNN

Deep learning and CNNs

The idea behind Convolutional Neural Networks (CNNs) is to follow vision and data processing in living organisms. Therefore, the idea sounds promising, though we are far away from understanding all details of the human brain and from artificially providing the same computational power the human brain provides. The first idea of artificial Convolutional Neural Networks dates back to 1998 [112], but due to a lack of computational power and training images, these early neural networks did not reach the quality and performance of other machine learning algorithms. The eventual breakthrough was achieved by Alex-Net [109] in 2012 by training a CNN on two GPUs and thereby wining the difficult ImageNet competition by a large margin.

R-CNN

The ImageNet challenge is on object recognition in images, i.e. the identification of one or multiple objects in the image but not on object detection, i.e. finding the position of a specific object. In 2014, Girshick *et al.* published a CNN based object detection approach called R-CNN [72]. This approach used the Selective Search algorithm [203] to identify potential ROIs containing any kind of object of any size. This corresponds to the first step in the general object detection process described above. The potential ROIs are scaled to a fixed size. For feature extraction, a CNN is used. However, the use of this CNN is restricted to feature extraction, i.e. the transformation of the input image patch (scaled ROI) to an output feature vector of 4096 neurons. SVMs use this output feature vector to classify it into an object type such as face, car etc. For each object a specifically trained SVM is required, but since the main computational part is the feature extraction, which is not object dependent, the overhead for a large number of SVMs to detect a large number of different object types is marginal. However, to detect one specific object, like people or faces as in this work, the overhead for feature extraction is still high.

R-CNN Improvoements

Since inaccurate localization was a major source of detection error, Zhang *et al.* [221] suggest a better proposal generator based on a Bayesian optimization framework [139, 188] and trained the CNN with a loss function considering also the localization error. Ouyang *et al.* [146, 147] extend the R-CNN with an additional CNN modeling object part, which contains geometric constraints for improving the detection accuracy.

SPPnet and Fast R-CNN

R-CNN has one big bottleneck regarding processing time, the feature generation. For each ROI, the whole CNN is applied, which means that the CNN is applied approximately a few thousand times per image. This drawback was solved by SPPnet [85] as well as Fast R-CNN [71] which applies the CNN on the whole image and generates a whole image feature map. The ROI proposals are still estimated on the original image with the Selective Search algorithm [203], but so called ROI pooling layers extract feature vectors for each proposed ROI from the feature map. Whereas SPPnet uses a spatial pyramid pooling for handling different ROI sizes, Fast R-CNN scales all ROI proposal sizes to a fixed size output feature vector, which is a special case for pyramid pooling by using only one pyramid layer. Furthermore, whereas R-CNN uses a SVM for object proposal classification, in Fast R-CNN the classification step is also performed by a CNN, more precisely inside the CNN, which estimates the object probability as well as its bounding box.

Proposal Generation

R-CNN, SPPNet and Fast R-CNN use an external algorithm for proposal generation and therefore work without a sliding window approach. Since analyzing a sliding window is computationally intensive the usage of an external algorithm keeps the number of object proposals small compared to a sliding window approach. However, in convolution layers filters slide over the image and therefore a convolution layer is some kind of sliding window approach. OverFeat [181, 46] uses advantages of this and can therefore apply object detection in a sliding window way with a CNN. The competition between these two proposal generation approaches was ended with Multibox [194] by introducing the idea of a Region Proposal Network (RPN), which is defacto a CNN for proposal generation.

Faster R-CNN

In Faster R-CNN [167] the idea of an RPN is used. After the convolution layers, the generated feature map is used in a sliding window way. For each window, anchor points with different scales and aspect ratios are used, on which a classifier and a regressor are trained. The classifier returns the probability that the anchor point with the comprised scale and aspect ratio contains any object and the regressor optimizes the bounding box, i.e. the scale and aspect ratio to the

containing object. However, this only applies to proposal generation and does not contain any information about the object type. For object detection, including a bounding box refinement, a CNN is used in a similar way as in Fast R-CNN. The main difference is that the proposal comes from a CNN. Therefore, every step from proposal generation through feature extraction to object (type) classification is combined in a complete CNN pipeline, which allows an End-to-End optimization.

YOLO

Faster R-CNN has an (internal) proposal generator, whose proposals are used for object detection. Object detection and object classification use only a certain image region which is the outcome from the proposal generator. By contrast, YOLO [162, 163, 164] directly uses the whole image for the detection task. In other words, YOLO extends the proposal generator so that the object class is an additional output. More precisely, this is done by splitting the image into a $S \times S$ grid where each grid cell predicts N object bounding boxes with their confidence scores. Since YOLO does not additionally process the output of the proposal generator, it is faster than Faster R-CNN, but sometimes at the cost of precision (see Figure 4 in [163]).

SSD

Single Shot Multibox Detector (SSD) [123] works in a similar fashion to YOLO and uses the outcome of the proposal network directly as detection outcome. However, in SSD the region proposal network is applied on multiple scaled convolution layers to improve the object detection for objects with different sizes. In YOLOv3 [164] the multiple scaled convolution layers are applied, too.

RetinaNet

CNN-based detection approaches can be divided into two different categories: one-stage and two-stage detectors. Two-stage detectors, such as Faster R-CNN and its predecessors, apply one method for generating object proposals i.e. regions which could contain an object. In a second stage, these proposals are classified into the different object classes or background. One-stage detectors, such as YOLO or SSD combine both steps into one network, which leads to a faster processing time albeit less accurate performance [120]. In [120] it was assumed that the main reason is that two-stage detectors could handle the class imbalance problem better than one-stage detectors. Therefore, they proposed a new loss function, which could handle class imbalance very well and therefore the proposed network, called RetinaNet, achieved high performance results with less processing time. The focal loss is also integrated into YOLOv3 [164] achieving good results for an IoU of 0.3. The authors propose that an IoU of 0.5 might outperform human labeling performance and therefore proposed a less restrictive IoU than 0.3.

Segmentation

CNNs can also be used for object segmentation as in [50, 226] and in a postprocessing step an object bounding box can be estimated from this segmentation. In [226], the bounding box prediction is obtained iteratively so that the bounding box prediction also improves the segmentation and so on. In Mask R-CNN [84], Faster R-CNN was extended by a branch for predicting the pixelwise mask of the detected object. Similar to per-pixel prediction in semantic segmentation, FCOS [198] is an object detection approach without the need for anchor boxes or proposal generator reducing the complexity and number of hyperparameters for object detectors. Pixel Consensus Voting [207] achieved an instance segmentation based on the GHT, whose voting is based on CNN feature patches inspired by [119]. Similar to how segmentation works, object detection is also possible by keypoint detection. In this case, the feature heatmap has peaks at the keypoint position. CornerNet [111], ExtremeNet [224], and CenterNet [223] detect specific keypoints of the target object. CornerNet and ExtremeNet detect keypoints on

the bounding box, whereas CenterNet detects the center point of the bounding box and the bounding box is regressed from the features at the center location.

Scaling

One challenge of object detection is the detection of objects with different scales. Some solutions work by constructing pyramides either on the image level [34, 91, 122] or on feature levels [7, 108, 185, 186, 20]. The trident network [118] consists of three branches with different dilated convolution layers increasing the receptive field to different degrees. Hence, each branch covers different object sizes.

Machine learning approaches and CNNs

Due to the success of CNNs, there is some research trying to convert other machine learning or object detection approaches into CNNs. Girshick *et al.* [73] shows that a DPM can be formulated as a CNN. Therefore, it also became possible to replace the HoG-Features used in DPM with features computed by a deep convolution network. The well-known Random-Forest classifier can also be converted into a CNN for better optimization [211, 92, 168].

2.1.5 Task specific adaptations

Task specific adaptations

Many of the presented methods have also been customized for face detection [225, 222], facial feature detection [76, 189, 117, 93, 228, 192], or face alignment [18, 103, 202, 5, 215]. Other have been specifically developed for specific tasks such as eye detection [208, 110, 153, 189, 199, 204, 31, 32, 201, 96, 106]. Similarly, some of the approaches have been customized or specifically developed for medical tasks like epiphyses localization [52, 53, 113, 196, 190, 54, 19].

2.2 Edge based object detection

Rigid Object Detection

Almost all publications mentioned above focus mainly on a better feature extraction, especially the approaches mentioned in Section 2.1.1. As these frameworks first extract the features, the original images may not be required. However, a lot of features are extracted and therefore a reconstruction of the original image may be possible. Besides that, in many approaches, grey values, which are the original image, are a very useful feature channel.

DPM

The development of the DPM [49] focused more strongly on splitting the target object into parts by using HOG features. The requirement to use features that protect privacy, would also permit the usage of HoG features. However, in comparison to edge features, HoG features have the main drawback that the HoG features are scarcely human readable. Whereas a visual error analysis (and also detection of incorrect annotations) is possible on Canny Edge Images, it is almost impossible on HoG features.

CNN

Even if the CNN based approaches do not explicitly focus on feature extraction, it is an integral component of CNNs to learn the feature extraction. Therefore, also for CNN based approaches, the original image is required.

Hough-Forests

The GHT in it original version only uses edge features as does the DGHT extension. However, the main component of ISM or the Hough-Forests is also based on feature learning and therefore the original image is required.

DGHT

Except for the DGHT, none of the methods mentioned above uses only edge features. Hence, when restricted to the use of edge features, only the DGHT as a general and efficient object localization approach with promising results is available. It should be noted, however, that in general, the DGHT is not restricted to Canny Edge features. In the past also other edges features [61] or statistical features [169] have been used. Furthermore it is also possible to use more sophisticated features, like HoG or LBP and also using multiple feature extractors together.

Edge image as normal input

Technically, it is always possible to apply any of the aforementioned approaches on an image containing only the edges extracted e.g. by the Canny Edge Detector. However, except for the DGHT, one of the key components in each mentioned approach was feature extraction. Transforming an image into an edge image results in the severe loss of information which cannot be recovered. For approaches based on this lost information, this will lead to worse results.

Chapter 3
Scientific Goals

DGHT for video surveillance

The main goals of this thesis are to improve the DGHT, to transfer it from medical image domain to natural images, and to evaluate its usability for surveillance applications. One strong advantage of the DGHT is its easy transferability to new tasks. It is also a major objective to ensure easy transferability also for the new improvements, developed in this work. An additional aim is to achieve these goals while keeping the effort for ground truth generation at a minimum, i.e. in the ideal case only the target landmark needs to be annotated.

Requirements

Two additional requirements emerge from the application to surveillance recordings:

Limited number of training images

1. Due to privacy issues, obtaining data from surveillance applications is difficult, so the algorithm has to be able to work with a small amount of training images. It may be possible to obtain a few hundred or even a few thousand images, but a much higher number of images, such as the hundreds of thousands or even millions of images, as required by current CNN approaches, is not available.

Canny Edge Features

2. Since privacy laws in Germany are very strict, storage of personal data is difficult. The data needs to be well protected against unauthorized access and the data generally has to be deleted after a few days. Therefore, we decided to use Canny Edge Images only since they have previously been shown to be useful by various authors like [169]. Furthermore, edge images are anonymized, but for a human observer it is still possible to recognize structures and objects contained in the image. Considering the lack of access to the original images for the present work, features that still allow for the recognition of some details in the images by a human are required for evaluation and debugging of the developed methods. The Canny Edge Images offer a suitable trade-off between the demands for privacy and suitability for development.

High target object variability

In the novel field of application for the DGHT, video surveillance, it can be expected that the variability of the target object, namely humans, is much larger than for the tasks the DGHT has previously been evaluated on. Additionally, prior to this work, the main focus of the DGHT was medical image processing. Medical images mostly have no, little, or at least static backgrounds. Therefore, in the past, the shape of the background could be learned for better discrimination between the background and the target object. Such a learnable background cannot be assumed for the tasks addressed in this work. Therefore, this work will mainly focus on how large target object variability with a dynamic background can be handled by the DGHT. Large target object variability presents one of the main challenges for GHT-based approaches. While the main focus of this work is to improve the DGHT, the potential for any improvements to be transferred to other GHT-based approaches will also be considered.

Chapter 4
Data

In this work, three different databases will be used for training and validation. Two of them are publicly available and aimed at eye localization, whereas the third task is based on inhouse hand radiographs and focuses on the localization of 12 epiphyses for supporting a subsequent automatic bone age assessment.

Data

4.1 Color FERET Face Database

The Color FERET Face Database [154] is managed and distributed by the National Institute of Standards and Technology (NIST). It contains a total of 14126 images of 1199 subjects that differ in ethnicity, age, and gender (see Figure 4.1). The images additionally vary in lighting conditions, face size, and head pose and have a resolution of 512 × 768 pixels. In this work, the 2409 frontal images from series fa and fb with annotated eye positions were used.

FERET Face Database

The data has a large variation in ethnicity, but all images are front looking without any noticeable background. The average eye distance is 135.3 pixel with a minimum of 86.1 and a maximum of 240.1 pixels. This means that the eye distances varies between 0.636 and 1.774 around the average eye distance with a normalized standard deviation of 0.135 (see Table 4.1).

Data description

For each image used, the annotation for both eyes, the nose, and the mouth were provided by NIST. Due to slight incorrectnesses in seven images, the provided annotations have been manually adjusted.

Annotations

Figure 4.1: Example images of the FERET Face Database

Table 4.1: Eye distance on the FERET Face Database

	Minimum	Mean	Maximum	Standard deviation
Absolute values	86.1	135.3	240.1	18.3
Normalized by average eye distance	0.636	1.000	1.774	0.135

Corpora

A subset of 594 images from 220 randomly selected subjects define the training corpus. Validation was performed on 1815 images from 647 subjects, which were not included in the training corpus.

4.2 RWTH Hand Database

RWTH Hand Database

For the task of epiphysis localization, an inhouse corpus from the University Hospital RWTH Aachen, consisting of 812 unnormalized hand radiographs was used. The average size of the images is 1185 × 2066 pixels and the image height ranges from 1347 to 2964 pixels.

Data description

The main variation in this data is the age of the subjects resulting in different sizes of the epiphyses, the distances between the epiphyses and the finger bones, and the relative size of

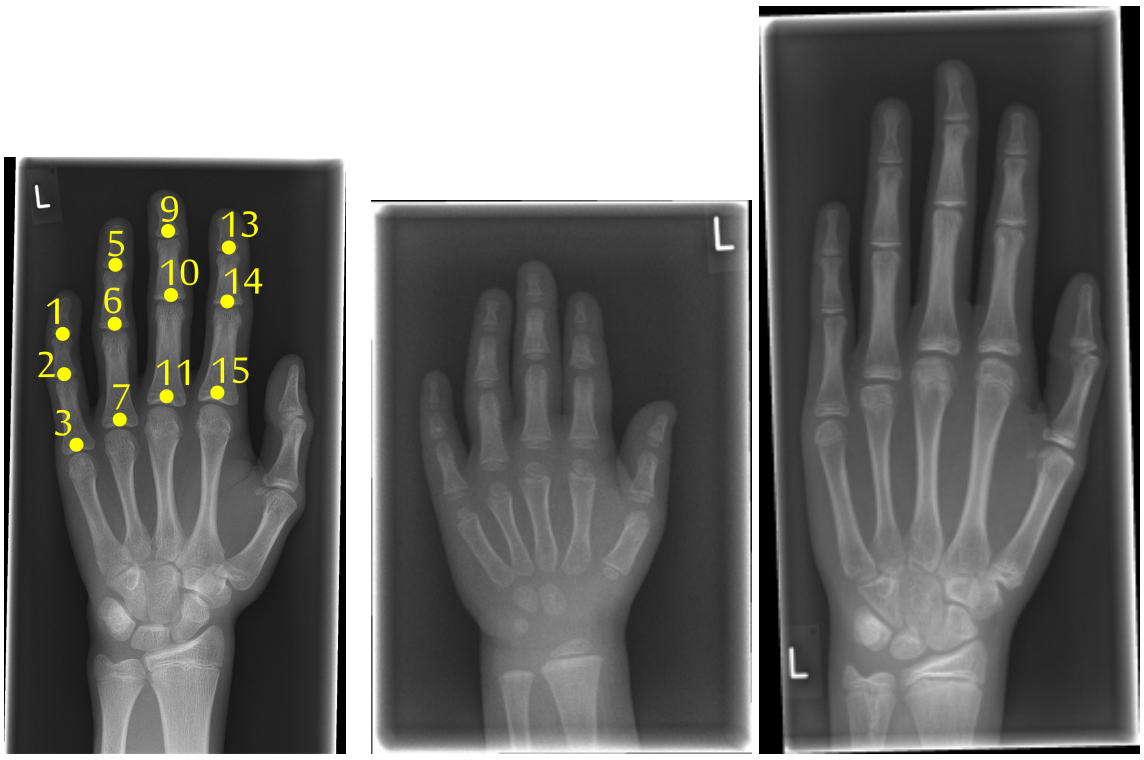


Figure 4.2: Identification number of epiphyses used in this work (a) and examples of the RWTH Hand Database (b-c). The size of the circels is $\frac{6}{256}$ of the image height. According to [52] this is the allowed error tolerance for a human observer. Note that only the 12 epiphyses which are used in this paper have been labeled.

the bones (see Figure 4.2). Furthermore, also the hand pose varies sometimes but there is no noticeable background in the images.

Annotations

The 12 finger epiphyses, as displayed in Figure 4.2a, were manually annotated. Furthermore, the bone age was manually estimated by an expert and ranges from 3 to 19 years.

Corpora

400 images were randomly selected for training, whereas the remaining 412 images define the validation corpus.

4.3 Chokepoint Dataset

Chokepoint Dataset

The Chokepoint Dataset [214] is a publicly available database, mainly designed for person identification and verification under real-world surveillance conditions and it is therefore also applicable for person detection. Cameras were placed above 4 portals capturing subjects walking through the portal in a natural way. For each portal between 3 and 4 sequences were recorded on which the same persons walks through the portal but in slightly different ways.

Data description

The faces, captured while the person walked through the portal, have variations in terms of illumination, pose, sharpness, and eye distance, mainly resulting from the distance to the

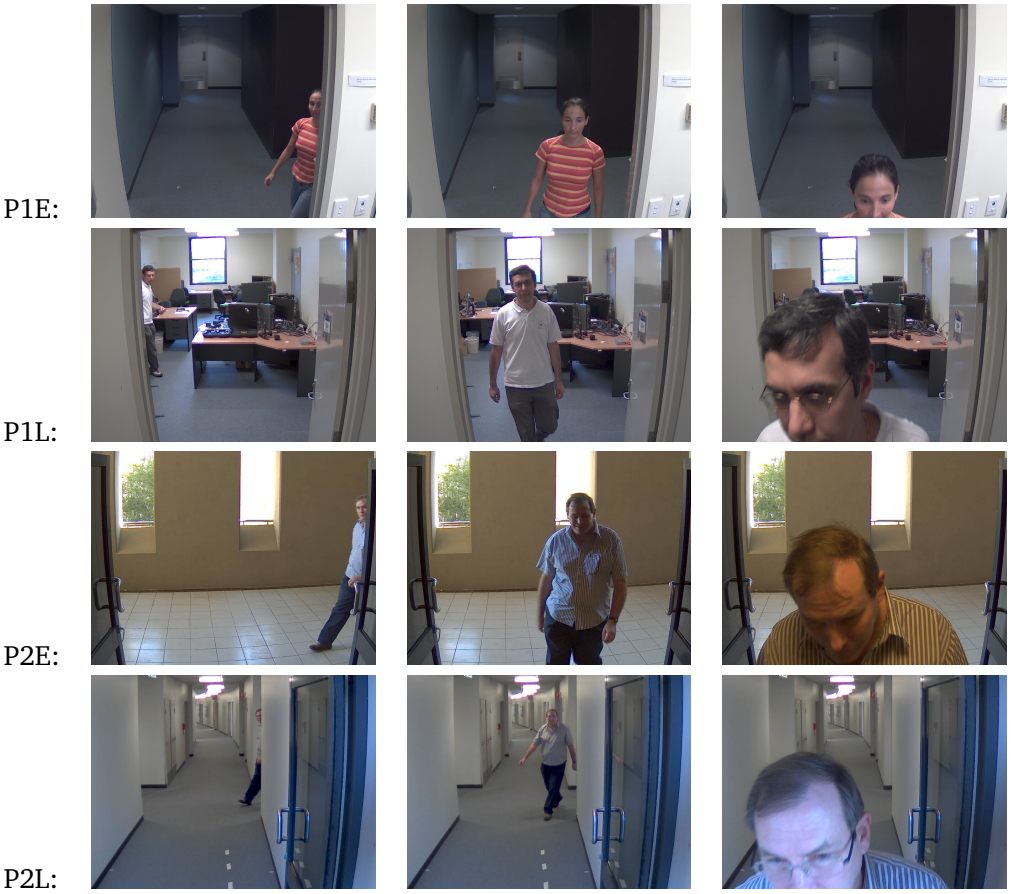


Figure 4.3: Example images from the 4 sequences in the Chokepoint Dataset

Table 4.2: Eye distance on the Chokepoint Dataset

	Portal	Minimum	Mean	Maximum	Standard deviation
Absolute values	P1E	15.3	39.1	116.0	15.1
Normalized by average eye distance	P1E	0.391	1.000	2.967	0.386
Absolute values	P1L	13.2	31.9	105.4	15.1
Normalized by average eye distance	P1L	0.412	1.000	3.300	0.473
Absolute values	P2E	11.0	32.6	119.9	16.2
Normalized by average eye distance	P2E	0.338	1.000	3.682	0.496
Absolute values	P2L	13.0	30.6	109.0	16.0
Normalized by average eye distance	P2L	0.425	1.000	3.568	0.523

Table 4.3: Number of training and validation images per portal in the Chokepoint Dataset

Portal name	#Training images	#Validation images
P1E	3940	1037
P1L	5290	1444
P2E	5636	2092
P2L	6058	2112

camera, and therefore face size (see Figure 4.3). The four different portals allows testing of the system under different illumination and background conditions. The eye distances have stronger variations than in the FERET Face Database as shown in Table 4.2. Therefore, it is expected that this database is more challenging than the FERET Face Database or RWTH Hand Database.

Annotations

Only the eye positions were manually annotated in addition to a unique identifier for each person allowing a separation into training and validation subjects.

Corpora

The database contains 29 different persons, from which 7 are used exclusively for validation. The number of images per sequence is shown in Table 4.3. During training as well as testing each portal is handled separately.

Object background discrimination

On the RWTH Hand Database and the FERET Face Database, the image contains almost only the target object, the left hand or the face respectively, and the challenge is a precise detection of the epiphyses or facial landmarks inside the target object. In other words, there is no relevant background, which could confuse the localization algorithm. This means that all visible shapes in the image theoretically contain information about the position of the target landmark. By contrast, the Chokepoint Dataset contains images, which not only contain the target object, in this case the person, but also a highly structured and non-static background. Additionally, the target object is located at different positions in relation to the background. This means that these images contain visible shapes, from the background, which do not even theoretically contain any information about the position of the target landmark. In this case, a good localization approach needs to internally separate the target object, whose shapes support the localization of the target landmark, from the background, which is meaningless.

Portal description

As can be seen in Figure 4.3 and in Table 4.2, the four portals differ in terms of background and target object variation, as measured by eye distance. Therefore, it can be expected that the results for the four portals will differ.

4.4 Validation

Validation error

On all databases the results are usually achieved on the validation corpus. Results on the training corpus are explicitly noted. $\tilde{\mathbf{q}}_n^{\Upsilon}$ gives the prediction localization from the system for landmark Υ in image n. $\hat{\mathbf{q}}_n^{\Upsilon}$ gives the correct (i.e. annotated) localization for Υ. The error

Error calculation per image

$$\epsilon_n^{\Upsilon} = \left\| \tilde{\mathbf{q}}_n^{\Upsilon} - \hat{\mathbf{q}}_n^{\Upsilon} \right\|_2 \tag{4.1}$$

is measured as the Euclidean distance between the predicted and the annotated localization in pixels.

Eye distance normalization

On the FERET Face Database and the Chokepoint Dataset the error is normalized to

$$\ddot{\epsilon}_n^{\Upsilon} = \frac{\epsilon_n^{\Upsilon}}{\left\| \hat{\mathbf{q}}_n^{\text{left eye}} - \hat{\mathbf{q}}_n^{\text{right eye}} \right\|_2} \tag{4.2}$$

by the eye distance. A normalized error of less than or equal to 0.05 corresponds approximatly to the size of the pupil, 0.1 to the size of the iris, and 0.25 to the size of the eye. Furthermore, for eye localization, the maximum error of the left and the right eye

Eye distance error

$$\ddot{\epsilon}_n^{\text{eye}} = \max(\ddot{\epsilon}_n^{\text{left eye}}, \ddot{\epsilon}_n^{\text{right eye}}) \tag{4.3}$$

is used so that e.g. an eye localization error of 0.1 means that the irides of both eyes were correctly localized.

RWTH Hand Database normalization

On the RWTH Hand Database the error is normalized by the image height

$$\ddot{\epsilon}_n^{\Upsilon} = \frac{\epsilon_n^{\Upsilon}}{\kappa} \tag{4.4}$$

with the image height κ. According to [52], a human observer perceives an epiphyseal localization as correct if the Euclidean distance to the center is less than 6 pixel for hand radiographs normalized to an image height of 256 pixels, which is why we consider $\ddot{\epsilon}_n^{\Upsilon} \leq \frac{6}{256}$ as correct from a human observer's perspective.

Accuracy calculation

On each validation corpus the accuracy is estimated by

$$\Xi^{\Upsilon}(\Gamma) = \frac{1}{N} \sum_{n=1}^{N} \begin{cases} 1 & \text{if } \ddot{\epsilon}_n^{\Upsilon} < \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

and gives the percentage of correctly localized images within an error tolerance of Γ and over all validation images n, whereas N giving the total number of validation images.

Chapter 5

Baseline method

5.1 Theory

5.1.1 Generalized Hough Transform

History

The Generalized Hough Transform (GHT) was introduced by Ballard in 1981 [6] as a general version of the Hough Transform [90]. The Hough Transform is a voting method for detection of analytical objects, such as lines, circles, or ellipses. Feature points vote for possible object parameters, e.g. slope and y-intercept for lines or center and radius for circles. The more feature points vote for the same parameters the more likely the object in question exists with these parameters. Ballard extended the voting idea for detection of arbitrary objects. An object shape is described by a model which is transformed by translation, scaling, and rotation. All feature points in an image vote for possible model transformation parameters and the higher the number of votes the more likely the object can be found at the position given by the translation parameters with the corresponding size and rotation.

Landmark detection approach

More precisely, the GHT is a landmark detection approach, i.e. it detects a specific landmark within the target object instead of the target object itself. However, knowing the specific position of a landmark usually also reveals the position of the corresponding object.

Model

In contrast to the Hough Transform, the GHT requires a model which describes the target object in relation to a so-called reference point, representing the target landmark. Hence, the model

$$\mathcal{M} := \{(\mathbf{m}_1, \phi_{\mathbf{m}_1}), (\mathbf{m}_2, \phi_{\mathbf{m}_2}), ..., (\mathbf{m}_j, \phi_{\mathbf{m}_j}), ..., (\mathbf{m}_J, \phi_{\mathbf{m}_J})\} \subset \Psi \times \psi \tag{5.1}$$

consists of J model points $\mathbf{m}_j$, with

$$\Psi \subset \mathbb{R}^2 \tag{5.2}$$

for 2D images, which represent specific parts of the target object. Each model point has a value $\phi_{\mathbf{m}_j}$ which describes to some extend the appearance of the target object at the position of the model point $\mathbf{m}_j$. With edge features only, as in this work, $\phi_{\mathbf{m}_j}$ is the gradient direction of the edge feature point $\mathbf{m}_j$, defining

$$\psi \subset [0, 2\pi] \tag{5.3}$$

in this work. However, by utilizing more sophisticated features, as in Hough-Forests [68], the model point value $\phi_{\mathbf{m}_j}$ can be a more detailed representation, e.g. the color or gray-value appearance in a surrounding area which potentially results in a higher dimensional domain.

Hough-space

By comparing the feature image

$$\mathscr{X}_n := \{(\mathbf{x}_1, \phi_{\mathbf{x}_1}), (\mathbf{x}_2, \phi_{\mathbf{x}_2}), ..., (\mathbf{x}_l, \phi_{\mathbf{x}_l}), ...(\mathbf{x}_L, \phi_{\mathbf{x}_L})\} \subset \Psi \times \psi, \tag{5.4}$$

given as a set of feature points $\mathbf{x}_l$ with corresponding feature values $\phi_{\mathbf{x}_l}$, with the model $\mathscr{M}$, the Hough-space

$$\mathscr{H} : \Psi \to \mathbb{R} \tag{5.5}$$

is generated. In this work, the Hough-space

$$\mathscr{H} := \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_i, ...\mathbf{c}_I\} \subset \Psi \tag{5.6}$$

consists of I Hough-cells $\mathbf{c}_i$, representing possible target point locations. Each cell $\mathbf{c}_i$ reflects the degree of matching between the GHT model and the feature image at the coordinates represented by the Hough-cell in question. Therefore, the higher the value, the more likely the target object can be found at this position and with this transformation. Note that in the original form, the GHT can cope with variations in size and rotation of the target object. This is achieved by a higher dimensional Hough-space, where the additional dimensions provide the scaling and rotation of the model. However, in this work the used GHT is restricted to a model translation since moderate object variability with respect to shape, rotation, and size is learned into the model.

Hough-space calculation

More mathematically, the Hough-space $\mathscr{H}$ is calculated by

$$\mathscr{H}(\mathbf{c}_i, \mathscr{M}, \mathscr{X}_n) = \sum_{j=1}^{|\mathscr{M}|} f_j(\mathbf{c}_i, \mathscr{X}_n) \tag{5.7}$$

with[1]

$$f_j(\mathbf{c}_i, \mathscr{X}_n) = \sum_{\forall \mathbf{x}_l \in \mathscr{X}_n} \begin{cases} 1, & \text{if } \mathbf{c}_i = \lfloor (\mathbf{x}_l - \mathbf{m}_j)/\varrho \rfloor \text{ and } d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) < \Delta\varphi. \\ 0, & \text{otherwise.} \end{cases} \tag{5.8}$$

Quantization

In Equation (5.8), the Hough-space is quantized by the quantization parameter ϱ. This serves to cover slight variations. Therefore, the coordinates of a Hough-cell $\mathbf{c}_i$ in the image space are given by

$$\mathbf{q}_i = \lfloor (\mathbf{c}_i + 0.5) \cdot \varrho \rfloor. \tag{5.9}$$

Model point and feature point value

$d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j})$ determines the difference between the model point value $\phi_{\mathbf{m}_j}$ and the feature point value $\phi_{\mathbf{x}_l}$. Since in this work only edge features in two dimensional space are used, $d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) = |\phi_{\mathbf{x}_l} - \phi_{\mathbf{m}_j}|$ represents the absolute difference between the gradient directions. $\Delta\varphi$ defines the allowed difference between the feature and model point value.

Voting procedure

Equation (5.7) is a simple template matching by using a sliding window approach. However, due to performance reasons, a voting procedure is implemented which is a loop over all feature points $\mathbf{x}_l$. Based on the value of the feature point $\phi_{\mathbf{x}_l}$, all potential model points $\mathbf{m}_j$ with $d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) < \Delta\varphi$ are selected from a lookup table. These model points vote for the Hough-cell $\mathbf{c}_i = \lfloor (\mathbf{x}_l - \mathbf{m}_j)/\varrho \rfloor$, i.e. increase the number of matching model-feature-point combinations for the Hough-cell $\mathbf{c}_i$ by one.

[1] Note that $\lfloor \mathbf{a} \rfloor$ denotes the floor of each component of $\mathbf{a}$.

Standard R-Table as introduced in [6]

$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [0, \Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [\Delta\varphi, 2*\Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [2*\Delta\varphi, 3*\Delta\varphi)\}$
...

Redundant R-Table as introduced in [169]

$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [-\frac{1}{2}*\Delta\varphi, \frac{1}{2}*\Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [0, \Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [\frac{1}{2}*\Delta\varphi, \frac{3}{2}*\Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [\Delta\varphi, 2*\Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [\frac{3}{2}*\Delta\varphi, \frac{5}{2}*\Delta\varphi)\}$
$\{\mathbf{m}_j \in \mathcal{M} \mid \phi_{\mathbf{m}_j} \in [2*\Delta\varphi, 3*\Delta\varphi)\}$
...

Figure 5.1: Illustration of the R-Table. Modified after Table 5.1 in [169].

Lookup table

To reduce computational complexity, Ballard introduced a lookup table, called R-Table [6]. One row of the R-Table contains all model points $\mathbf{m}_j$ with a $\phi_{\mathbf{m}_j}$ in a specific range. The idea is that given a certain feature point $\mathbf{x}_l$, only one row needs to be selected representing $\phi_{\mathbf{x}_l}$ and for all model points from the selected row $d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) < \Delta\varphi$ applies. Depending on $\phi_{\mathbf{m}_j}$ the R-Table row

$$k = \left\lfloor \frac{\phi_{\mathbf{x}_l}}{\Delta\varphi} \right\rfloor \tag{5.10}$$

is selected. However, the R-Table concept cannot guarantee that each model point $\mathbf{m}_j$ with $d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) < \Delta\varphi$ is selected, especially if $\phi_{\mathbf{x}_l}$ is near the boarder of one R-Table row as can be seen in Figure 5.1. For example if $\phi_{\mathbf{m}_j} = 1.1$ and $\Delta\varphi = 1.0$ all model points with $\{\mathbf{m}_j \mid \phi_{\mathbf{m}_j} \in [0.1, 2.1)\}$ should be selected, but the corresponding R-Table row contains only model points with $\{\mathbf{m}_j \mid \phi_{\mathbf{m}_j} \in [1, 2)\}$. To achieve more robust results, in [169] a concept called redundant R-Table is introduced. Whereas in the standard R-Table the first row has a range from $[0, \Delta\varphi)$, the redundant R-Table contains twice as many rows as the standard R-Table and the first two rows have a range from $[-\frac{1}{2}\Delta\varphi, \frac{1}{2}\Delta\varphi)$ and $[0, \Delta\varphi)$. Hereby the R-Table row is selected by

$$k = \left\lfloor \frac{2 * \phi_{\mathbf{x}_l}}{\Delta\varphi} + 0.5 \right\rfloor. \tag{5.11}$$

R-Table

This ensures a more robust selection of the model points allowed to vote (see Figure 5.1) as in the aforementioned example the model points with $\{\mathbf{m}_j \mid \phi_{\mathbf{m}_j} \in [0.5, 1.5)\}$ will be chosen from the R-Table, which is more robust albeit not completely accurate. Due to performance reasons, this inaccuracy was accepted in this work.

Probability space

After the voting procedure is applied and under the assumption that exactly one target object is visible in the image, the Hough-space can be considered as a probability space by

$$p(\mathbf{c}_i = \hat{\mathbf{c}}_n \mid \mathcal{X}_n, \mathcal{M}) = \frac{\mathcal{H}(\mathbf{c}_i, \mathcal{M}, \mathcal{X}_n)}{\sum_{\mathbf{c}_k} \mathcal{H}(\mathbf{c}_k, \mathcal{M}, \mathcal{X}_n)}. \tag{5.12}$$

given the probability that $\mathbf{c}_i$ is the target cell $\hat{\mathbf{c}}_n$.

Bayes classifier

Using the Bayes classifier, the most likely target point location results from the Hough-cell $\tilde{\mathbf{c}}_n$ with the highest number of votes, corresponding to the best match between the model $\mathcal{M}$ and the feature image $\mathcal{X}_n$:

$$\tilde{\mathbf{c}}_n(\mathcal{X}_n, \mathcal{M}) = \arg\max_{\mathbf{c}_i} p(\mathbf{c}_i = \hat{\mathbf{c}}_n \mid \mathcal{X}_n, \mathcal{M}) = \arg\max_{\mathbf{c}_i} \frac{\mathcal{H}(\mathbf{c}_i, \mathcal{M}, \mathcal{X}_n)}{\sum_{\mathbf{c}_k} \mathcal{H}(\mathbf{c}_k, \mathcal{M}, \mathcal{X}_n)} \tag{5.13}$$

Simplification of Bayes classifier

Since the normalization term in (5.13) has no influence on the arg max function, it can be simplified to

$$\tilde{\mathbf{c}}_n(\mathscr{X}_n, \mathscr{M}) = \arg\max_{\mathbf{c}_i} \mathscr{H}(\mathbf{c}_i, \mathscr{M}, \mathscr{X}_n) \tag{5.14}$$

Model generation

In Equation (5.14) it is obvious that the result of $\tilde{\mathbf{c}}_n$ depends on the image and on the model $\mathscr{M}$. Therefore, the quality of the localization highly depends on the quality of the model. Hence, the DGHT comprises a statistical learning approach for generating optimal models as described below.

5.1.2 Discriminative Generalized Hough Transform

Model point weighting

A naive way of generating a model is to overlay feature points of training images on the target points. Given enough training images, covering all object variations, such a naive model should achieve a good overlap on the target point location. However, such a model might also fit similar or confusing objects leading to wrong localizations. Therefore to generate good GHT models it is necessary to analyze the importance of each model point. It is quite evident that the model points of a given shape model are of different importance for the localization problem. While points that allow for good discrimination between the target object and other confusable structures are particularly useful, others may even mislead the detection procedure by fitting to wrong image parts. Therefore, the application of a model point specific discriminative weighting scheme appears to be a reasonable measure when using the GHT.

Hough-space as probability distribution

The theory is based on describing the GHT as a probabilistic framework, in which the Hough-space is interpreted as a posterior probability distribution $p(\mathbf{c}_i|\mathscr{X}_n)$ as mentioned in Equation (5.12). This distribution can be estimated, for example, from the relative frequencies of votes in each Hough-cell $\mathbf{c}_i$. The GHT-based localization task, which searches for the cell with the highest number of votes, can be formulated as the Bayes classifier

$$\tilde{\mathbf{c}}_n(\mathscr{X}_n, \mathscr{M}) = \arg\max_{\mathbf{c}_i} p(\mathbf{c}_i|\mathscr{M}, \mathscr{X}_n). \tag{5.15}$$

Separation of individual model point importance

In order to identify the individual importance of each single model point, it is necessary to split the Hough-cell votes into model point specific parts, which is done by the characteristic function[2] $f_j(\mathbf{c}_i, \mathscr{X}_n)$ (see Equation (5.8)), which denotes the number of votes from model point $\mathbf{m}_j$ in Hough-cell $\mathbf{c}_i$ for a given feature image $\mathscr{X}_n$.

Recombination into a maximum entropy distribution

Since the feature functions only consider the contributions of single model points, they must be recombined in order to preserve the constraints from the GHT voting procedure for the entire model. In the DGHT framework this is achieved by using the Maximum-Entropy distribution [95], which assures maximum objectivity and introduces model point specific weights λ_j.

$$p_\Lambda(\mathbf{c}_i|\mathscr{X}_n) = \frac{\exp\left(\sum_j \lambda_j \cdot f_j(\mathbf{c}_i, \mathscr{X}_n)\right)}{\sum_k \exp\left(\sum_j \lambda_j \cdot f_j(\mathbf{c}_k, \mathscr{X}_n)\right)} \tag{5.16}$$

[2] Here I follow the same definition as in [174]. It is also possible to refer to the "characteristic function" as feature, but to avoid confusion with features in the images processing, I chose the term "characteristic function".

Optimization via minimal classification error

The estimation of the free parameters $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_J\}$ with the side conditions (5.8) can be done by the method of Lagrange multipliers. Since this leads to an optimal approximation of the training data distribution but not necessarily to a minimal error rate, the parameter optimization in the DGHT follows a Minimal Classification Error (MCE) training approach [97], as first applied in the field of automatic speech recognition [12]. This technique, which was for object detection first introduced by [133], minimizes a smoothed error measure over a set of N training images and I Hough-cells.

$$E(\Lambda) = \sum_{n=1}^{N} \sum_{i=1}^{I} \varepsilon(\mathbf{c}_i, \tilde{\mathbf{c}}_n) \cdot \frac{p_\Lambda(\mathbf{c}_i|\mathscr{X}_n)^\eta}{\sum_k p_\Lambda(\mathbf{c}_k|\mathscr{X}_n)^\eta} \tag{5.17}$$

Loss function

Here, η controls the influence of alternative localization hypotheses on the error measure and $\varepsilon(\mathbf{c}_i, \tilde{\mathbf{c}}_n)$ denotes the error between the Hough-cell $\mathbf{c}_i$ and the target cell $\tilde{\mathbf{c}}_n$, which may be determined e.g. by the Euclidean distance $\|\mathbf{c}_i, \tilde{\mathbf{c}}_n\|_2$.

Steepest descent optimization

The optimization of the model point weights Λ over the error measure $E(\Lambda)$ is achieved in this work by applying the method of steepest descent. Although this technique does not guarantee that a global minimum will be reached, it is a frequently used method in machine learning and especially the success of Convolution Neural Networks demonstrates that steepest descent can be very powerful.

Incorporation of weights in GHT voting procedure

The estimated model point weights are directly incorporated into a standard GHT voting procedure by incrementing the value of a Hough-cell $\mathbf{c}_i$ by $\lambda_j \cdot f_j(\mathbf{c}_i, \mathscr{X}_n)$ for each model point $\mathbf{m}_j$.

$$\mathscr{H}(\mathbf{c}_i, \mathscr{M}, \mathscr{X}_n) = \sum_{j=1}^{|\mathscr{M}|} \lambda_j * f_j(\mathbf{c}_i, \mathscr{X}_n) \tag{5.18}$$

No influence on arg max function

The localization result $\tilde{\mathbf{c}}_n(\mathscr{X}_n, \mathscr{M})$ is then given by

$$\tilde{\mathbf{c}}_n(\mathscr{X}_n, \mathscr{M}) = \arg\max_{\mathbf{c}_i} \mathscr{H}(\mathbf{c}_i, \mathscr{M}, \mathscr{X}_n). \tag{5.19}$$

This leads to the same results as applying the log-linear feature combination (5.16), used for training, since neither the normalization term in the denominator nor the exponential function has an influence on the result of the arg max function.

5.1.3 Iterative training

Iterative training procedure

Since the assigned weights are directly incorporated into the GHT weighting scheme, it is obvious that the elimination of model points with a small absolute weight does not have a significant influence on the final localization result. Consequently, an initial shape model can be substantially reduced, keeping only a small amount of the most relevant positively and negatively weighted structures. In an iterative procedure [175], illustrated in Figure 5.2, this technique can be applied in order to repeatedly expand the model with the shapes of unrecognized target objects and, for negative weighting, the most important confusing structures contained in the training corpus.

Detailed process

In more detail, the first step of the iterative training is to select a small subset of training images to generate an initial shape model by overlaying feature points from a predefined region

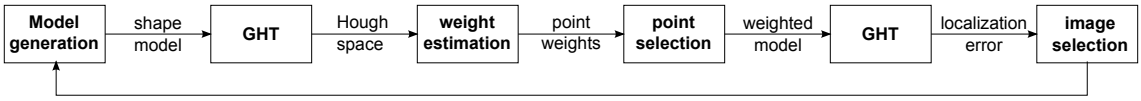


Figure 5.2: Illustration of the iterative training procedure.

(ROI) around the annotated target point. This model is subsequently used with equal point weights in a standard GHT procedure to localize the target points in all training images. The features $f_j(\mathbf{c}_i, \mathscr{X}_n)$ and error measures $\varepsilon(\mathbf{c}_i, \hat{\mathbf{c}}_n)$ are extracted from the resulting Hough-spaces and utilized to compute the updated weights. In the next step, model points with a low absolute weight are removed from the model, which is afterwards tested on the whole training dataset. Since the estimation of this first shape model is based on very few images, it can most likely not cover the whole variability contained in the training data. Therefore, in our framework, the model is expanded by additional structures taken from images with high localization error. To this end, feature points from a region around the target object and the most confusable objects are added to the model for the next iteration. The integration of structures from confusable objects into the shape model allows for the identification of *anti-shapes*, i.e. confusable structures, since the weighting scheme is capable of assigning negative weights to these model parts, thus increasing the discrimination capabilities. In the next iteration the expanded model is again applied for target point localization on the training corpus, and new weights are estimated using the described method. The iterative training procedure stops when the localization error on all training images is below a given threshold or if all training images have been used for model generation.

Idea behind iterative training

The basic idea behind the iterative training process is that assigning model point weights can indirectly remove model points, but it cannot create them. Therefore, a precondition of the weighting scheme is to have a model already covering all target object variations. The easiest way to achieve this is to use all feature points from all training images as model points. However, this would create a huge number of model points. Since the weighting scheme jointly optimizes all model points this results in a high-dimensional optimization problem, which is difficult to solve. Therefore, the iterative training procedure starts with only a few training images from which the initial model is generated. The weighting scheme ensures that model points with a negligible influence are recognized and removed from the model to keep the number of model points small.

Implementation details

The iterative training was implemented to stop at the latest after 99 iterations even if the aforementioned stop criterion was not reached. From a theoretically point of view more iterations are possible and such a fixed number of maximum iterations is not required. In most cases, hoever, much less iterations are needed as can be seen in [174].

5.1.4 Multi-Level-Approach (MLA)

High vs. low resolution

A high performance detection of very small structures, like the pupil, can only be achieved in high resolution images. However, a clear drawback of using the highest available resolution level lies in a large processing time and memory demand, which hampers the utilization of the method in practical applications. Additionally, feature extraction in high resolution images will produce many noisy details which may mislead the localization procedure. Therefore, a reasonable trade-off between the level of detail, required for reliable localization, and the

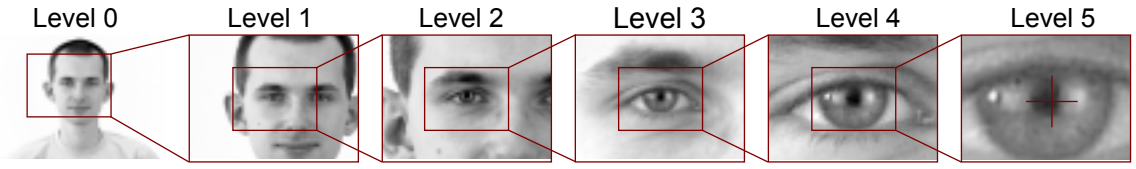


Figure 5.3: Image extracts with different resolutions in the multi-level approach

necessary suppression of irrelevant structures is needed. To this end, the DGHT framework uses a coarse-to-fine strategy based on a Gaussian image pyramid as described in [171].

Validation

The procedure begins with a low resolution image, called zoom level 1, which ensures fast processing time and provides the most relevant structures for coarse orientation. Since reliable localization with high accuracy is not possible on this zoom level, an image extract is selected around the detected point (hereafter called anchor point) and is further processed on an increased resolution (see Figure 5.3). The refined search region has half the size of the original image and twice the resolution such that the number of pixels stays approximately the same. The procedure can be repeated several times, each time using a level-specific model for object localization, selecting an image extract half the previous size and doubling the resolution. Due to the gradual increase of the resolution, this method can be viewed as a zooming procedure. Therefore, each level is called a zoom level and assigned an index. If a zoom level is named x/y then it is the xth from in total y zoom levels and the resolution is x/y.

Image extract generation during training

Generally, the MLA is independent of the localization approach. The training process is the same as without the MLA. The only difference is that for each zoom level a specific model needs to be trained to the given resolution and image extract, which depends on the anchor point. To get realistic image extracts during training, different corpora are required for each zoom level. This allows the training of a model for zoom level x and the application of the model to a new training corpus to obtain realistic anchor points for zoom level $x+1$. Then the model for zoom level $x+1$ can be trained on image extracts around these anchor points, which theoretically have the same distribution as during inference. However, this would require extra training corpora for each zoom level increasing the necessary number of training images drastically. Using the same training corpus for each zoom level (i.e. same images but resolution and extraction depends on the zoom level) would be more efficient. In this case, there are three options to obtain anchor points for generating the image extracts:

1. Applying the model for zoom level x on the same images on which it was trained to obtain anchor points for zoom level $x+1$. The resulting anchor points will usually be closer to the ground truth than realistically obtained points and hence their distribution will be unrealistic.

   Anchor points from training corpus

2. Adding a random shift to the ground truth points to obtain anchor points. This requires knowledge of the approximate shift during validation to simulate the same image extracts.

   Random shift of ground truth

3. Using the ground truth points as anchor points. In this case the image extracts fit perfectly. It is also the simplest approach and it ensures that the target landmark is always inside the image extract. Therefore, this approach was used in the original implementation from [171]. However, it results in a mismatch between training and inference.

   Using ground truth points

Method in this work

In this work, I use approach 3 as described in [171]. Although, the image extracts fit perfectly and therefore there is a mismatch between training and validation, the influence of this was considered to be small. For generation of the DGHT models, a region of interest (ROI) around the ground truth point is used from which the initial GHT model is generated. In the first zoom level, the ROI is manually defined but in the following zoom levels the ROI is a fraction of the image extract. In this work, I used a fraction of 75% of the image extract to avoid that the DGHT model contains information from the border of the image extract giving some tolerances for imprecise localization during validation. Another reason for using the ground truth as anchor points in combination with a ROI was that it is the simplest approach and has been well-proven in previous publications [171, 173, 81]. Furthermore, during validation there was no reason to assume that using different image extracts would strongly influence the localization performance.

5.2 Experiments

Experiments

To test the DGHT, experiments on all three database were conducted. Since most of the parameters are evaluated in [169], the main goal of the experiments was to explore the quality and limitations of the baseline system.

5.3 Results and Discussion

RWTH Hand Database

Table 5.1 shows that the best mean accuracy over all landmarks is 97.1% on the RWTH Hand Database achieved with five zoom levels. Therefore, we consider 5 zoom levels as the optimal setup for RWTH Hand Database. The result in terms of the mean accuracy is comparable with the results in [169, 173]. In these papers, an error tolerance of one centimeter was used. Due to a lack of spacing information in the RWTH Hand Database, it is not possible to directly calculate the errors in centimeter. The error tolerances plotted in Figure 4.2a suggest that they are usually smaller than one centimeter. Therefore, this result shows that the initial DGHT framework, applied here, was working correctly and in agreement with previous works such as [173] it confirms again that the DGHT can be used for medical landmark localization.

FERET Face Database

On the FERET Face Database, based on the accuracy rate for iris localization, the best result was achieved with three zoom levels. With this setup, both irides could be localized with a success rate of 96.4% and both pupils with 74.7% (see Table 5.2). The drastic reduction of the success rate for the pupil localization can partly be explained by slightly inaccurate ground truth annotations, as described in [81], which makes model training as well as model evaluation difficult and error prone. For mouth and nose localization the results were comparably good (see Table 5.2). They demonstrate that the DGHT is useful for facial landmark localization in general and not restricted to eye localization only. However, since the eye center is clearly defined and therefore easier to annotate, it is frequently localized with higher accuracies for small error tolerances (Table 5.2) than mouth and nose.

Chokepoint Dataset

As expected, localization results on the Chokepoint Dataset are less accurate than on the FERET Face Database (see Table 5.3). Depending on the portal, both eyes could be correctly localized

Table 5.1: The localization accuracy ($\Xi(\frac{6}{256})$, see Equation 4.5) for the different landmarks (1D,2D, ... 15D) on the RWTH Hand Database, mean accuracy over all landmarks ($\varnothing(\Xi(\frac{6}{256}))$), and mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels

	$\Xi(\frac{6}{256})$													
	1D	2D	3D	5D	6D	7D	9D	10D	11D	13D	14D	15D	$\varnothing(\Xi(\frac{6}{256}))$	$\varnothing(\epsilon_n)$
1/4	88.1%	88.3%	92.0%	93.4%	96.4%	93.7%	93.7%	96.8%	96.1%	90.8%	93.9%	94.2%	93.1	23.7
2/4	93.0%	95.4%	97.1%	97.8%	96.1%	95.6%	95.4%	99.0%	98.3%	95.6%	96.6%	97.6%	96.5	17.4
3/4	92.7%	95.4%	96.4%	97.8%	96.1%	95.1%	95.9%	99.3%	98.3%	95.9%	96.8%	97.6%	96.4	15.2
4/4	92.7%	95.4%	96.6%	97.8%	96.4%	95.4%	95.9%	99.3%	98.3%	95.9%	96.8%	97.6%	96.5	14.7
1/5	64.1%	70.9%	81.6%	68.4%	83.3%	85.9%	82.0%	81.8%	91.0%	71.6%	82.5%	83.3%	78.9	37.6
2/5	88.1%	92.2%	95.1%	94.9%	96.1%	96.4%	92.2%	98.1%	98.5%	91.5%	95.1%	94.7%	94.4	20.9
3/5	95.9%	94.7%	97.1%	96.8%	97.8%	97.6%	96.1%	98.5%	99.8%	93.2%	96.6%	97.8%	96.8	15.1
4/5	96.4%	95.6%	97.3%	97.3%	98.1%	97.6%	96.1%	98.5%	99.8%	93.7%	96.6%	97.8%	97.1	12.8
5/5	96.4%	95.9%	97.3%	97.3%	98.1%	97.6%	96.1%	98.5%	99.8%	93.4%	96.6%	97.8%	97.1	12.4

Table 5.2: The localization accuracy (Ξ, see Equation 4.5) on the FERET Face Database for different error tolerances (0.05, 0.1, 0.25, and 0.5) and the mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels

	eye					nose					mouth				
	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$
1/3	23.7%	74.4%	98.6%	99.6%	9.1	44.5%	82.7%	98.8%	99.7%	9.6	45.5%	83.0%	97.5%	99.1%	10.2
2/3	59.4%	94.9%	98.1%	98.1%	6.7	56.5%	91.4%	98.9%	99.5%	8.1	64.4%	92.5%	98.2%	98.6%	8.2
3/3	74.7%	96.4%	97.9%	98.1%	5.9	70.1%	95.8%	98.7%	99.5%	6.9	76.1%	94.3%	98.5%	98.7%	7.2
1/4	4.0%	35.5%	95.3%	99.2%	14.3	23.0%	56.9%	95.3%	99.4%	14.9	24.8%	61.4%	93.8%	99.1%	14.5
2/4	11.7%	61.0%	97.4%	98.8%	11.1	32.8%	78.0%	98.0%	99.6%	10.9	38.1%	82.9%	97.5%	99.1%	10.6
3/4	55.2%	94.5%	98.5%	98.7%	6.5	57.1%	92.6%	98.7%	99.6%	8.1	66.4%	92.3%	98.5%	99.0%	7.7
4/4	73.9%	96.1%	98.5%	98.7%	5.5	73.4%	96.4%	98.6%	99.6%	6.6	70.1%	92.8%	98.4%	99.0%	7.3
1/5	0.8%	7.3%	59.2%	91.2%	29.3	5.9%	22.1%	73.9%	95.9%	28.3	4.1%	16.1%	62.1%	94.9%	32.7
2/5	2.0%	21.8%	87.2%	96.6%	18.9	13.4%	42.3%	94.0%	98.5%	18.5	19.3%	55.2%	92.5%	97.1%	18.3
3/5	10.2%	60.1%	95.5%	97.1%	12.9	33.3%	76.8%	96.6%	98.5%	12.8	33.5%	77.0%	95.6%	97.3%	14.2
4/5	49.8%	89.3%	96.1%	97.1%	8.8	52.7%	88.0%	96.0%	98.4%	10.6	52.0%	84.5%	95.9%	97.2%	12.3
5/5	61.3%	89.8%	96.0%	97.1%	8.2	56.7%	87.5%	95.7%	98.4%	10.4	55.2%	82.0%	95.5%	97.2%	12.4

Table 5.3: Localization accuracy (Ξ, see Equation 4.5) for different error tolerances and mean localization error in pixels ($\varnothing(\epsilon_n)$) for the four portals in the Chokepoint Dataset and each zoom level

Portal	Zoom level	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\Xi(1)$	$\varnothing(\epsilon_n)$
P1E	1/4	0.2%	4.9%	28.7%	65.6%	27.4
	2/4	9.4%	56.9%	81.5%	90.1%	12.6
	3/4	31.4%	78.5%	85.9%	91.1%	9.6
	4/4	53.5%	84.8%	86.9%	89.2%	8.7
P1L	1/4	0.4%	6.5%	26.2%	61.6%	32.5
	2/4	4.3%	32.9%	61.6%	81.9%	21.8
	3/4	16.3%	67.3%	86.1%	88.7%	16.9
	4/4	33.9%	82.3%	87.5%	88.4%	15.9
P2E	1/4	0.3%	7.1%	33.6%	71.2%	24.5
	2/4	3.1%	29.4%	67.4%	89.3%	16.6
	3/4	11.2%	61.5%	85.2%	90.5%	13.5
	4/4	23.5%	76.6%	88.0%	89.8%	12.5
P2L	1/4	0.4%	2.8%	13.9%	55.0%	29.6
	2/4	2.8%	26.5%	65.7%	82.2%	17.7
	3/4	12.2%	61.0%	83.8%	85.5%	14.1
	4/4	27.9%	79.5%	84.5%	85.4%	12.9

in 76.6% to 84.8% of the images. A correct face localization can be assumed if both eyes were localized with an error smaller than the eye distance, i.e. the error tolerance is one eye distance. This was achieved for 85.4% to 89.8% of the images.

Chokepoint Dataset challenges

As described in Chapter 4, the Chokepoint Dataset is more challenging due to larger target object variability and non-static backgrounds. Therefore, we assume that the unsatisfying performance of the DGHT on the Chokepoint Dataset is related to these challenges. In the following paragraphs, I will analyze this assumption.

Analyzing large target object variability

I will start by analyzing the large target object variation, exemplary on the P2E portal. As we can see in Chapter 4, the eye distance variation is much larger on the Chokepoint Dataset than on the FERET Face Database (compare Table 4.1 and 4.2). Though the target object varies not only in eye distance, it is a good simplification to measure the extent of target object variability. Thus, I am capable of restricting the training and validation corpus to a comparable target object variation as on the FERET Face Database by using the minimum and the maximum eye distance from the FERET Face Database normalized by the mean eye distance. The restricted Chokepoint corpus contains only images for which the eye distance is in the same normalized range, i.e. the eye distance ranges from 21.0 to 57.4 with a standard deviation of 8.9. The normalization by the mean eye distance is necessary since it clearly differs from 135.3 on FERET Face Database to 32.6 on the Chokepoint Dataset for the P2E portal.

Results on Chokepoint Dataset for medium target object variability

The DGHT model generated on this subset achieves an accuracy of 90.3 for correct eye localization and 96.9 for a less restrictive error tolerance of 0.5 of the eye distance (see Table 5.4). Firstly, we can see that by reducing the target object variability, the localization performance improves. Secondly, considering the different average eye distance for the FERET Face Database and the Chokepoint Dataset, an error tolerance of 0.1 for the FERET Face Database is comparable to an error tolerance of 0.5 on the Chokepoint Dataset in terms of pixel, re-

Table 5.4: Localization accuracy (Ξ, see Equation 4.5) for different error tolerances and mean localization error in pixels ($\varnothing(\epsilon_n)$) for portal P2E. "No restrictions" are the standard experiments as shown in Table 5.3. "Medium head size" are the results for the corpus where the eye distance range is comparable to the FERET Face Database. "No background" are the results when a correct localization at zoom level 1 was assumed and therefore the background has only a negligible influence.

		$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\Xi(1)$	$\varnothing(\epsilon_n)$
No restrictions	1/4	0.3%	7.1%	33.6%	71.2%	24.5
	2/4	3.1%	29.4%	67.4%	89.3%	16.6
	3/4	11.2%	61.5%	85.2%	90.5%	13.5
	4/4	23.5%	76.6%	88.0%	89.8%	12.5
Medium head size	1/4	0.5%	6.9%	40.7%	87.4%	16.0
	2/4	3.5%	36.6%	81.0%	96.8%	9.0
	3/4	14.0%	80.1%	95.8%	98.5%	5.7
	4/4	34.4%	90.3%	96.9%	98.1%	4.4
No background	1/4	0.5%	10.0%	47.2%	100.0%	13.6
	2/4	4.0%	35.9%	78.8%	97.4%	8.6
	3/4	13.6%	68.4%	93.1%	97.7%	5.8
	4/4	29.6%	85.4%	95.9%	97.4%	4.5
Medium head size and no background	1/4	0.5%	7.9%	46.6%	100.0%	13.5
	2/4	3.2%	38.1%	82.8%	98.4%	8.3
	3/4	15.2%	81.3%	97.2%	99.7%	4.8
	4/4	36.5%	91.8%	98.4%	99.4%	3.5

spectively Hough-cell errors. Considering the different error tolerances and restricting the Chokepoint Dataset to the same normalized eye distance range, the accuracies for FERET Face Database and the exemplary tested P2E portal on Chokepoint Dataset are similar.

Analyzing Non-static background

Besides the large target object variation, also the non-static background might be challenging for the DGHT on the Chokepoint Dataset. Due to the MLA approach, the background mainly influences zoom level 1, where the complete image needs to be analyzed. Assuming a correct localization on zoom level 1, we can assume that the following zoom levels mainly contain the target object and the influence of the background is minor. Since the training process was implemented to use perfect image extracts, the background should not have a strong influence on training process, either. Therefore, I analyzed the system performance under the assumption that the localization at zoom level 1 was correct, i.e. the error was smaller than one eye distance. For the exemplary P2E portal, the accuracy increases to 85.4 for correct eye localization and to 95.9 for an error tolerance of 0.5 of the eye distance. This shows, that the DGHT can also handle large target object variations, if there is no noteworthy background.

Large target object variability and non-static background

Last but not least, restricting target object variability and removing the background, i.e. combining both aforementioned analyses, leads, for the exemplary P2E portal, to an accuracy of 91.8 for correct eye localization and 98.4 for an error tolerance of 0.5 eye distance. Again, this is an improvement, albeit not such a strong one as seen before. Altogether this leads to the conclusion that the DGHT has issues with large target object variability especially in combination with non-static backgrounds. This is perfectly explainable since a larger target object variation requires a higher relative number of model points. The absolute number of model points depends on many other factors, e.g. in the training process the maximum number of

Table 5.5: Relative sum of model points voting for the correct hypotheses on different databases at zoom level 1.

	Average relative sum of voting model points	Standard deviation of relative sum of voting model points
Chokepoint Dataset P2E portal	0.079	0.028
Chokepoint Dataset with restricted eye distance	0.102	0.035
FERET Face Database	0.140	0.059

Table 5.6: The table shows the average number of iterations per database and zoom level

Database	Zoom level	∅ iterations
FERET Face Database	1/3	2.75
	2/3	3.5
	3/3	7.0
RWTH Hand Database	1/5	12.33
	2/5	9.17
	3/5	6.75
	4/5	11.00
	5/5	35.67
Chokepoint Dataset	1/4	99.0
	2/4	74.0
	3/4	61.0
	4/4	80.0

model points can be restricted, but the important value is how many model points vote for correct localization hypotheses in relation to the complete number of model points. In the DGHT not only the number of model points is relevant, but their weights also play an important role. Table 5.5 shows that, as expected, with larger target object variability, the relative sum of model points voting for correct hypotheses is reduced. In combination with non-static and non-learable backgrounds, this increases the risk that the model will accidentally fit somewhere in the background. However, the results also show that the DGHT can handle large target object variability or non-static background to some extend, but not both together.

Meaning of large number of iterations during training

The DGHT uses an iterative training procedure in which training images with a high localization error are integrated into the model for the next iteration (see Section 5.1.3). Therefore, the number of iterations varies between the databases and zoom levels. Less iterations means that less training images are sufficient to cover the target object variations in a way that ensures an optimal discrimination between the target object and similar and confusable objects. After the training process, at least the majority of training samples should have a low localization error. However, it is still possible that a few training images continue to have a high localization error. This happens, e.g. , for so called outlier images. The best example is a wrong annotation. A good DGHT model will always have a high localization error on a single image with a wrong annotation. During the training procedure, this image will be detected very early and therefore integrated into the model. Since the influence of this image is very small compared to all other training images, the weighting procedure will assign a low weight or even a negative weight to the model points generated by the image with the faulty annotation.

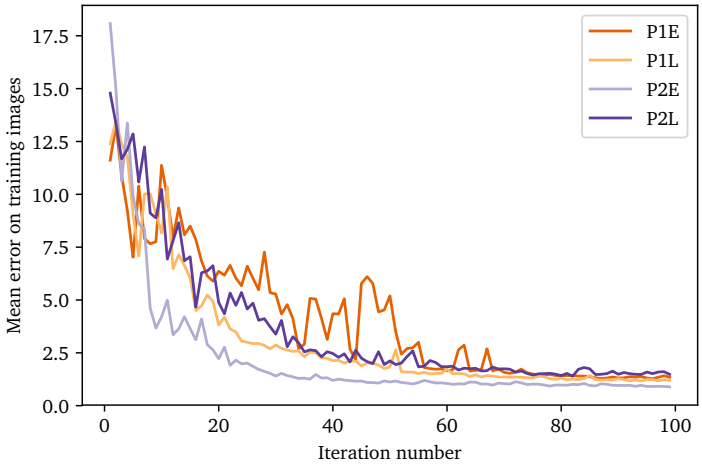


Figure 5.4: Mean error per iteration on the Chokepoint Dataset

In this case, the image with the wrong annotation still has a high localization error, but since it has already been integrated into the model, it will not be integrated again. Of course, this could also happen with images with correct annotations, but with a very different appearance of the target object, e.g. shape, in comparison to the large majority of the training samples. By contrast, a larger number of iterations is required if a larger number of training images needs to be integrated into the model so that the model can cover each specific variation.

Analysing number of iterations

I analyzed the number of iterations for the training procedure, as can be seen in Table 5.6. On the FERET Face Database and on the RWTH Hand Database the number of iterations is higher for the last zoom level than for a previous one. On the FERET Face Database it is clearly visible how the number of iterations increases with the number of zoom levels. This means that the variability of the target object increases with each zoom level. Since in a later zoom level more details are visible, due to the higher resolution, it is explainable that also the target object variation increases. On the Chokepoint Dataset we see a different pattern. Here, the first zoom level requires the most iterations, always 99, after which it stops automatically. This is, again, an indication of the special challenge of the Chokepoint Dataset. As already mentioned, the challenge of discrimination between target object and background occurs mainly at the first zoom level and the large number of iterations there shows that it is difficult to generate a model which ensures a good discrimination.

Analysing iterative training stop criterion

Furthermore, at the first zoom level of the Chokepoint Dataset until 40 to 60 iterations, depending on the portal, the mean error on the training data decreases before it saturates (see Figure 5.4). Due to this saturation, we can expect that a larger number of iterations would not increase the localization performance significantly and therefore, stopping the training after 99 iterations seems sufficient.

Chokepoint Dataset Training

The fact that the Chokepoint Dataset is more challenging is further illustrated by a longer training time. Whereas on the RWTH Hand Database and the FERET Face Database, the model training for one landmark was approximately one day, on the Chokepoint Dataset the overall training time for the 8 models (4 portals with 2 landmarks) was approximately one month. Since training time was not a requirement in this work, it was not measured in a comparable

Table 5.7: Comparison of the accuracy (Equation 4.5) for different maximum number of zoom levels for the Chokepoint Dataset portal P2E on the left eye at zoom level 1

Zoom Levels	$\Xi^{lefteye}(0.5)$	$\Xi^{lefteye}(1)$
1/3	24.5	41.4
1/4	54.1	83.4
1/5	29.4	59.0

Table 5.8: Mean image size after downscaling and/or image patch extraction using the optimal number of zoom levels

Database	#Zoom Levels	Mean image size
RWTH	5	74×129
FERET	3	128×192
Chokepoint	4	100×75

way and therefore only very rough estimates are available, but they still clearly illustrate the difference in training times.

Zoom Levels

Comparing different zoom levels per task revealed that the last zoom level with the original resolution achieved almost always the best results. Sometimes, also early zoom levels achieved similar or slightly better results, but such better results provide little improvement over what could be explained by random effects. On the Chokepoint Dataset, the results at zoom level 0 with a large error tolerance of a factor of 1.5 of the eye distance shows similar results as in the last zoom level when using a factor of 0.25 of the eye distance (see Table 5.3). This indicates again that the main issue at the first zoom level is background discrimination rather than precise landmark localization.

Number of Zoom Levels

Comparing the results of experiments with different number of zoom levels (see Table 5.1, 5.2, and 5.7) revealed that the total number of zoom levels should be chosen so that the input image at zoom level 0 has, after downsampling, an approximate size of around 100 pixel (see Table 5.8). On the FERET Face Database, the input image at zoom level 1 is slightly larger, but the decision between three and four zoom levels in total is to some extend inconclusive, depending on landmark identity and error tolerance (see Table 5.2). Note however, due to long training times on the Chokepoint Dataset, the number of zoom levels was only compared for the P2E portal for detecting the left eye and only within zoom level 0. Since the results (see Table 5.7) show very clearly that four zoom levels is the best setup, no further experiments were performed.

5.4 Conclusion

Conclusion

As show in previous publications, the DGHT is a good and general object localization method for medical images. Furthermore, this chapter shows that it can also be easily transfered to natural image processing, in particular facial landmark localization. However, this chapter reveals also the main issue of the DGHT: Large target object variability in combination with difficult backgrounds. Theses difficulties are expected in surveillance situations and cannot be handled by the DGHT with satisfying accuracy without further modifications.

Chapter 6

Shape Consistency Measure

6.1 Introduction

DGHT weighting procedure useful for medium object variability

The underling idea of the DGHT is to concatenate all object variations, seen in training, into a single model and to individually weight the model points based on their contributions to the correct localization. Let us consider the example of eye localization with two different poses: in one the subject is looking to the right, in the other to the left. Both feature images will be overlaid with respect to the eye, e.g. left eye, to generate a model which covers both variations (see Figure 6.1 a-c). The weighting procedure of the DGHT will estimate a higher weight for the model points around the eyes, the mouth and the nose since these are in the middle of the face and therefore at similar positions for both variations, i.e. to some degree they are pose-invariant. By contrast, the back of the head is different in both variations and therefore the DGHT will give the corresponding model points a lower weight. If we extend the example by adding pose variations of looking up and down, then also the influence of the model points around the nose and the mouth will be reduced because now these poses are not invariant anymore. When considering additional face sizes, the model points around the eyes will also not fit very well. This means that, theoretically, by increasing the amount of variability, the importance of each model point for correct localization will be reduced and therefore model point weights will become more similar. To summarize, the DGHT weighting procedure is good as long as the variation is small enough so that a crucial part of the model is invariant to most of the variations.

Mutually exclusive variations might support the same localization hypothesis

For simplification, we will only consider the two variations of looking to the left and to the right. If the variation is very large, the individual contribution of each model point and therefore also its weight will become similar. Therefore, we will ignore the model point weights for now and consider each model point as having the same influence. In the aforementioned example, the model points from the back of the head only fit for one pose. However, the voting procedure, defined in Equation (5.7), allows all model points to vote individually and independently from all other points. Yet, in the example, we can clearly see that if the model points from the back of the head looking to the left vote for a specific Hough-cell $\mathbf{c}_i$ then the model points from the back of the head looking to the right will not vote for $\mathbf{c}_i$, if $\mathbf{c}_i$ is the correct position of the left eye. In regions with many feature points it is possible that model points from both, mutually exclusive, variations will vote for the same $\mathbf{c}_i$. In GHT-based approaches, like the DGHT, this would result in a false localization (see Figure 6.1 d-f).

General issue

The independent voting is a general problem of GHT-based localization approaches and, according to [161], the main reason for their poor performance, leading to a high false positive

rate. Accordingly, in Chapter 5.3, we saw poor performance on the Chokepoint Dataset. The large target object variability in combination with the non-static background leads to models with a large number of model points from mutually exclusive variations. Hence, on the target object only a small part of the model fits and in combination with the non-static background there is a high risk that the model will accidentally fit better at a wrong localization.

Potential solutions

Since the independent voting is a very well-known problem, many potential solutions exist. Most of them reduce the number of model points allowed to vote. In the Hough-Forests this is, for example, done by feature extraction which allows a more detailed description of target object parts represented by a model point. With a more detailed description, it is less likely that this may fit to another part of the target object or to the background. By contrast, the gradient direction of edge features is a rather poor description of an object part and may therefore easily vote with feature points not belonging to the target object. A different, frequently applied solution is to reduce the target object variability by separating the images into different classes as in [170, 161] and training a specific model for each class.

Shape Consistency Measure

The basic issue of GHT-based approaches is that dependencies between model points are not taken into account and none of the mentioned approaches fixes that issue itself. Therefore, in this chapter, I will present a novel solution called Shape Consistency Measure (SCM), which directly analyses the set of model points voting for a specific hypothesis (hereafter called voting pattern) to rate their quality, i.e. the likelihood that the shape of the voting pattern represent the target object.

6.2 Theory

6.2.1 Overview

Components of the SCM

The principle of the SCM is to model constraints between model points, test if these constraints are violated and use this information to improve the localization. Therefore, the development of the SCM can be structured into three steps:

Voting pattern

1. Firstly, it is important to know, which model points have voted for the Hough-cell in question. This is hereafter called the voting pattern. Therefore, we define a feature vector $\mathbf{r}$, which contains for each model point $\mathbf{m}_j$ the information whether $\mathbf{m}_j$ has voted for the Hough-cell $\mathbf{c}_i$ in question or not.

Correctness of voting pattern

2. Then, the voting pattern needs to be compared to a set of rules or constraints to judge if the voting pattern is correct, i.e. if these model points are allowed to vote for the same localization hypothesis. Here, we introduced a function g, which returns the probability that the given feature vector $\mathbf{r}$ represents the target object.

Integration into localization procedure

3. As a last step the returned probability from function g needs to be incorporated into our localization function given in Equation (5.13).

In the following, I will investigate these three steps in more detail.

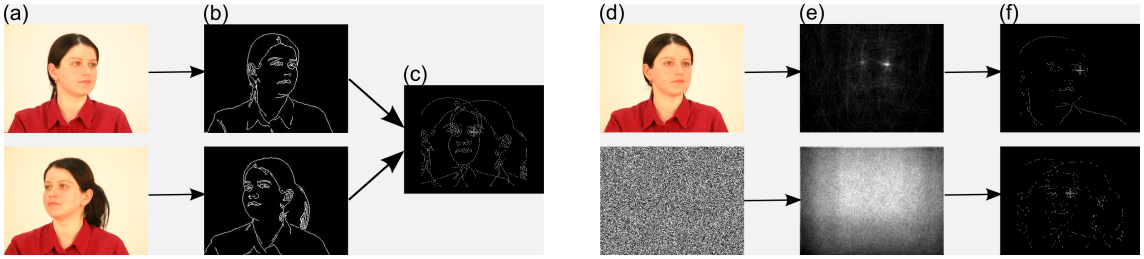


Figure 6.1: (a) shows two images for different head poses and the corresponding feature images (b). Both variations are integrated into one model (c) by fusing the corresponding edge points. (d) shows a head image and an artificially generated noise image. (e) is the corresponding Hough-space, generated by applying the model (c) for detection of the left eye in the corresponding image and (f) shows those model points which have contributed to the votes of the Hough-cell corresponding to the maximum in the Hough-space. For the head image in (d) a clear peak in (e) is visible at the correct position of the left eye and (f) shows that mostly model points from one variation only voted for this cell. This is in contrast to the noise image, which leads to higher votes in the Hough-space. However, (f) shows that model points of both, mutually exclusive, variations have contributed to the maximum.

6.2.2 Feature vector

SCM feature vector

The model points which have voted for a specific localization hypothesis are directly given by $\mathbf{f}(\mathbf{c}_i, \mathscr{X}_n) = [f_1(\mathbf{c}_i, \mathscr{X}_n), f_2(\mathbf{c}_i, \mathscr{X}_n), ..., f_J(\mathbf{c}_i, \mathscr{X}_n)]^t$, which can be used as a feature vector for analyzing the voting pattern. Another issue of GHT-based approaches is that a model-feature-point-combination only votes for a specific Hough-cell and therefore $\mathbf{f}(\mathbf{c}_i, \mathscr{X}_n)$ is sensitive to small target object variations. To overcome this issue, it is reasonable to consider the voting pattern for neighboring Hough-cells as well within a certain distance. If this neighborhood distance is too large, however, the set of model points, voting for this area, loses its explanatory power (see Figure 6.2). Therefore, making the voting pattern more generic, a feature function r_j is used which captures the closest distance of the vote of model point $\mathbf{m}_j$ in a given neighborhood area as

$$r_j(\mathbf{c}_i, \mathscr{X}_n) = \min_{\mathbf{c}_k} \begin{cases} d(\mathbf{c}_i, \mathbf{c}_k), & \text{if } f_j(\mathbf{c}_k, \mathscr{X}_n) \geq 1 \\ & \text{and } d(\mathbf{c}_i, \mathbf{c}_k) \leq \vartheta \\ \vartheta + 1, & \text{otherwise.} \end{cases} \tag{6.1}$$

with

$$d(\mathbf{a}, \mathbf{b}) = \max_t |a_t - b_t|. \tag{6.2}$$

Minimum neighborhood distance measurement

Thus, a value $\alpha = r_j(\mathbf{c}_i, \mathscr{X}_n) < \vartheta$ specifies the minimum neighborhood of $(2\alpha + 1) \times (2\alpha + 1)$ around $\mathbf{c}_i$ in which the model point $\mathbf{m}_j$ has voted. Also other distance metrics for $d(\mathbf{a}, \mathbf{b})$, such as the Euclidean distances, are possible but were not considered in this work. In case a distance $d(\mathbf{c}_i, \mathbf{c}_k)$ exceeding ϑ, a link between $\mathbf{m}_j$ and $\mathbf{c}_i$ cannot be assumed and so the exact distance is purely coincidental. Experiments show that the choice of parameter ϑ has only a small effect on the overall localization rate and is more relevant for runtime performance (see Section 6.6.2).

Feature vector for the SCM

With the feature function $r_j(\mathbf{c}_i, \mathscr{X}_n)$ the GHT voting pattern is extended as

$$\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n) = [r_1(\mathbf{c}_i, \mathscr{X}_n), r_2(\mathbf{c}_i, \mathscr{X}_n), ..., r_J(\mathbf{c}_i, \mathscr{X}_n)]^t, \tag{6.3}$$

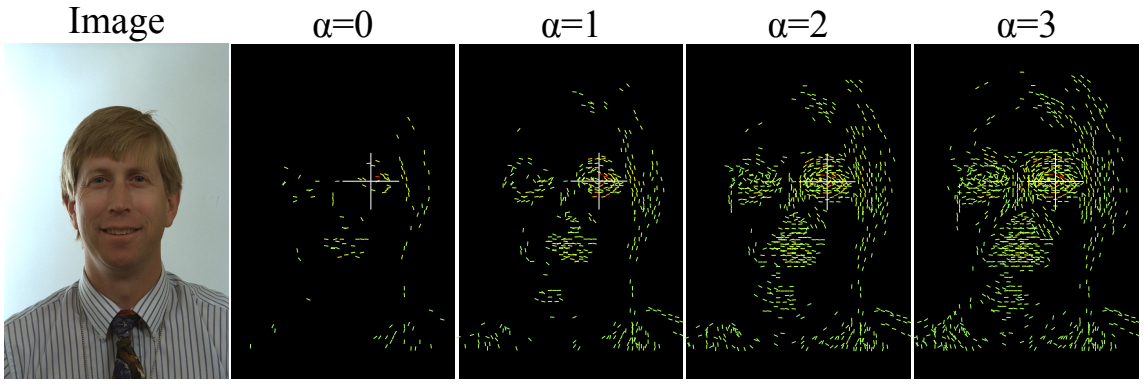


Figure 6.2: Illustration of model points voting for the target localization hypothesis $\tilde{\mathbf{c}}$ and its $(2\alpha+1)\times(2\alpha+1)$ neighborhood. When considering the target cell only (i.e. $\alpha=0$), a very sparse structure with a strongly random character is visible. However, incrementing the size of the neighborhood ($\alpha>0$) gradually increases the density and robustness of the facial structures.

containing information about the voting behavior in cell $\mathbf{c}_i$ and its neighborhood. By setting $\vartheta=0$, $\mathbf{r}$ is identical to not considering any neighborhood cells.

6.2.3 Classification function

Requirements for the detection function

The detection function is a crucial part of the SCM, but at the same time it is difficult to manually set the rules for whether a voting pattern $\mathbf{r}$ is correct or not. Therefore, a supervised machine learning approach can be used, which needs to fulfill two requirements:

Probability output

- The output shall be a confidence value, e.g. probability, of how likely $\mathbf{r}$ is correct. Also other output types, such as a binary output, might be usable as long as it can be converted into some kind of confidence value.

Modeling dependencies between attributes

- The machine learning approach needs to be capable of modeling dependencies between attributes r_j. This is crucial since otherwise no new information would be added to the GHT-based localization function.

Detection function

Therefore, we define a detection function

$$g(\mathbf{r}(\mathbf{c}_i,\mathscr{X}_n)) \tag{6.4}$$

which returns a confidence value that the voting pattern $\mathbf{r}(\mathbf{c}_i,\mathscr{X}_n)$ corresponds to the target object. In the light of this work a more intuitive and simplified description would be that the confidence value represents how likely a given $\mathbf{c}_i$ is the annotated target point localization.

Class definition

Since a supervised machine learning approach is used, a training procedure is required for which a ground truth label Ω needs to be defined. We define two classes: Ω_r as a regular voting pattern and Ω_i as an irregular pattern. To define what a regular or an irregular voting pattern is, we assume that voting patterns for Hough-cells at or near the target Hough-cell are regular shapes whereas voting patterns from Hough-cells with large errors are irregular.

Therefore, the decision whether a GHT voting pattern belongs to class Ω_r or Ω_i is based on the Euclidean distance between the Hough-cell in question and the target Hough-cell. If the distance is smaller than or equal to a parameter ξ_1, the corresponding voting pattern belongs to class Ω_r. If the distance is larger than a parameter ξ_2, it belongs to class Ω_i. Hough-cells with a distance between ξ_1 and ξ_2 are undefined and will not be considered during training. Since during evaluation and real application the target Hough-cell is unknown, the class definition is only relevant during the SCM model training.

6.2.4 Integration into localization procedure

Integration

After generating the feature vector and obtaining a confidence value that this feature vector corresponds to the target object, we need to integrate this confidence value into the DGHT framework.

Combining DGHT and SCM

$\mathcal{H}(\mathbf{c}_i, \mathcal{M}, \mathcal{X}_n)$ can be interpreted as the probability (see Equation (5.12)) that $\mathbf{c}_i$ is the target point location using the DGHT. With $g(\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n))$ we have, in addition, a confidence value that $\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n)$ is the shape of the target object and therefore that $\mathbf{c}_i$ is the target point location. To achieve the combination of the two knowledge sources, the Hough-cells $\mathbf{c}_i$ are weighted by the confidence values $g(\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n))$. Hence, the final localization is given by

$$\tilde{\mathbf{c}}_n(\mathcal{X}_n, \mathcal{M}) = \arg\max_{\mathbf{c}_i}(g(\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n)) \cdot \mathcal{H}(\mathbf{c}_i, \mathcal{M}, \mathcal{X}_n)). \tag{6.5}$$

6.3 Implementation details

Used Classifier

For the detection function g in this work a Random-Forest classifer is used since this method is very efficient [28] and well suited to model dependencies between attributes. This is required in order to capture the voting characteristics of groups of model points at correct and incorrect positions. Additionally, the Random-Forest classifier can determine the probability $p(\Omega_r|\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n))$ for a regular pattern, i.e. assumed to be a correct localization, given the attribute vector $\mathbf{r}(\mathbf{c}_i, \mathcal{X}_n)$. Therefore, the Random-Forest fulfills the given requirements.

Random-Forest classifier

The Random-Forest classifier is an ensemble method consisting of multiple decision trees. These decision tress are trained on randomly selected subsets of samples. Attribute selection is also based on random subsets. These random factors result in weaker trees but ensure that all trees are different and the combination of multiple trees leads to more robust results. In this work, the Random-Forest is trained by balanced subsampling, i.e. the subset of samples used to train a single tree is generated in a way that each class has the same number of samples.

Number of hypotheses

To train the classifier, a set of training samples from the training images is required. Theoretically, all hypotheses from a training image could be used. However, since the Hough-space votes are included into Equation (6.5), localization hypotheses with very few votes are unlikely to be selected as localization results. To avoid the classifier training being distracted by hypotheses which have, due to the small number of votes, practically no influence in Equation (6.5), only the N hypotheses with the highest number of votes are used to train the classifier.

SCM training procedure

In summary, during the training of the SCM, a previously generated DGHT model is applied to all training images. For each image, the N hypotheses with the highest votes, are used as training samples for the Random-Forest classifier. For these hypotheses, the feature vector $\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n)$ is generated according to Equation (6.3). To determine the class label for a given hypothesis, the Euclidean distance $\varepsilon(\mathbf{c}_i, \hat{\mathbf{c}}_n) = \|\mathbf{c}_i, \hat{\mathbf{c}}_n\|_2$ between the hypothesis in question $\mathbf{c}_i$ and the ground truth localization $\hat{\mathbf{c}}_n$ is used. Hypotheses with an error smaller than or equal to ξ_1 Hough-cells considered as regular structures and determined as class Ω_r, whereas hypotheses with an error larger than ξ_2 Hough-cells belong to class Ω_i (irregular shape). Hypotheses with an error between ξ_1 and ξ_2 Hough-cells are not considered during training to ensure a better discrimination between both classes. Subsequently, the Random-Forest classifier is trained as described in [14], to separate the two classes Ω_r and Ω_i (Figure 6.3).

SCM localization procedure

To localize the target object in an unknown image, at first the DGHT model is applied. Then, for the N hypotheses with the highest DGHT-votes, the feature vector $\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n)$, generated according to Equation (6.3), is used as input for the previously trained Random-Forest classifier. This determines the posterior probability $p(\Omega_r|\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n))$ that the hypothesis in question belongs to class Ω_r, i.e. that it is the target object. Finally, according to Equation (6.5), the best localization hypothesis is selected as the estimated target landmark (Figure 6.4).

6.4 Experiments

What to evaluate

The performance of the SCM will be evaluated on the FERET Face Database, the RWTH Hand Database and the Chokepoint Dataset (see Section 4) on all zoom levels. On some of these zoom levels, the baseline results, using the DGHT only, are already very good while on other ones the results need to be improved. Therefore, the question is whether the SCM can improve the DGHT results if they are not satisfying and whether it can at least maintain results that are already very good.

Evaluation-Method

The evaluation of individual zoom levels was performed on image extracts around the ground truth coordinates. Except for the first zoom level, where the whole image is used, this is not a fair experiment since it assumes that the previous zoom level has returned perfect results. Nevertheless, it is useful to estimate the overall functionality of the SCM and helps to evaluate it with different parameter settings. Additionally, fair experiments on real image extracts were performed to evaluate the overall performance of the system.

6.5 Results and Discussion

FERET Face Database

On the FERET Face Database, the accuracy (see Equation (4.5)) clearly increases from 96.4% to 98.6% for the correct localization of both irides, when using additional SCM based weighting. This means that 61.1% of the errors that occurred when using only the DGHT could be corrected by the SCM (see Table 6.2). Figure 6.5 shows an example of how the SCM improves the Hough-space and the localization results.

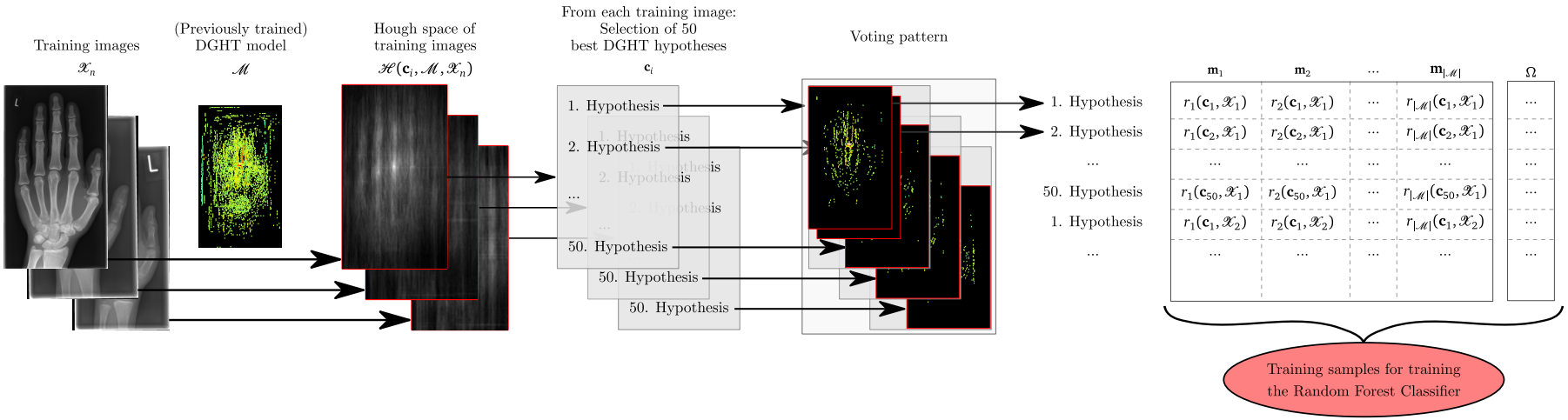


Figure 6.3: Scheme of the SCM training procedure: For each training image, a feature image $\mathscr{X}_n$is generated for which the feature vectors and corresponding class labels from the N best DGHT hypotheses, in this figure $N = 50$, constitute the training samples for training the Random-Forest classifier.

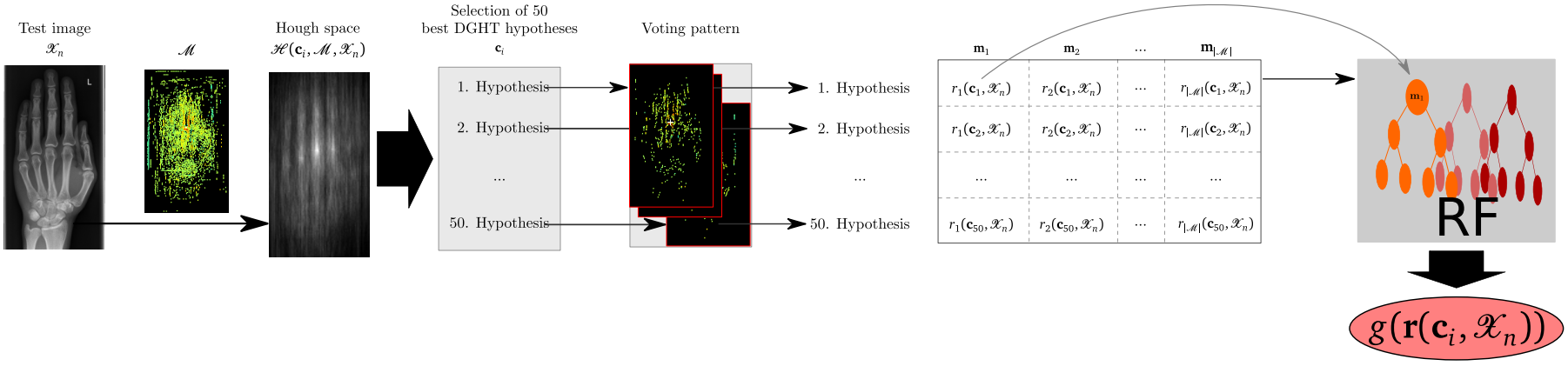


Figure 6.4: Scheme of the SCM test procedure: For a feature image $\mathscr{X}_n$created from an unknown test image, the feature vectors $\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n)$ from the N best DGHT hypotheses, 50 in this figure, are fed into the Random-Forest classifier which determines $p(\Omega_r|\mathbf{r}(\mathbf{c}_i, \mathscr{X}_n))$. Note, the used DGHT model $\mathscr{M}$ is the same as during the training procedure (see Figure 6.3).

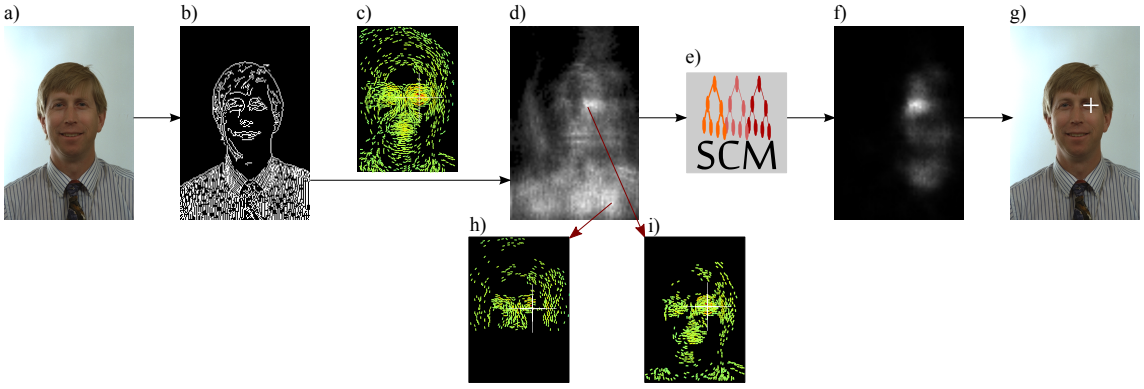


Figure 6.5: Illustration of the combined (D)GHT / SCM framework: An input image (a) is transformed into a feature, e.g. edge, image (b). The (D)GHT-Model (c) is applied onto the feature image and generates a Hough-space (d). The high number of feature points in the shirt of the subject leads to a mislocalization which is reflected in a random set of model points voting for this hypothesis (h). By contrast, the GHT voting pattern for the correct localization hypothesis (i) reveals a face-like structure. Rating the different sets of voting model points for all hypotheses with the SCM (e) and incorporating this information as weighting factor into the Hough-space leads to a much more focussed result (f) and produces a correct localization (g).

RWTH Hand Database

The improvements are similar on the RWTH Hand Database, on which the mean success rate for localizing the 12 epiphyses increases from 97.1% to 99.4% (see Table 6.3), which is an error reduction of 79.4%. Furthermore, on 95.8% of the validation images, all landmarks were correctly localized and on 98.0% no more than one landmark was incorrectly localized. For comparison, with the DGHT baseline system, on 83.3% of the images all landmarks were correct and on 92.2% no more than one landmark was incorrect.

Chokepoint Dataset

On the Chokepoint Dataset, the DGHT baseline system could only reach an average accuracy over the four portals of 80.8% for the detection of both eyes. With the SCM, the average accuracy improved to 97.2% now reaching a mature level also for datasets with strong target object variability (see Table 6.1). This means that 85.7% of the erroneous results from the DGHT system could be corrected by the SCM.

Individual zoom levels

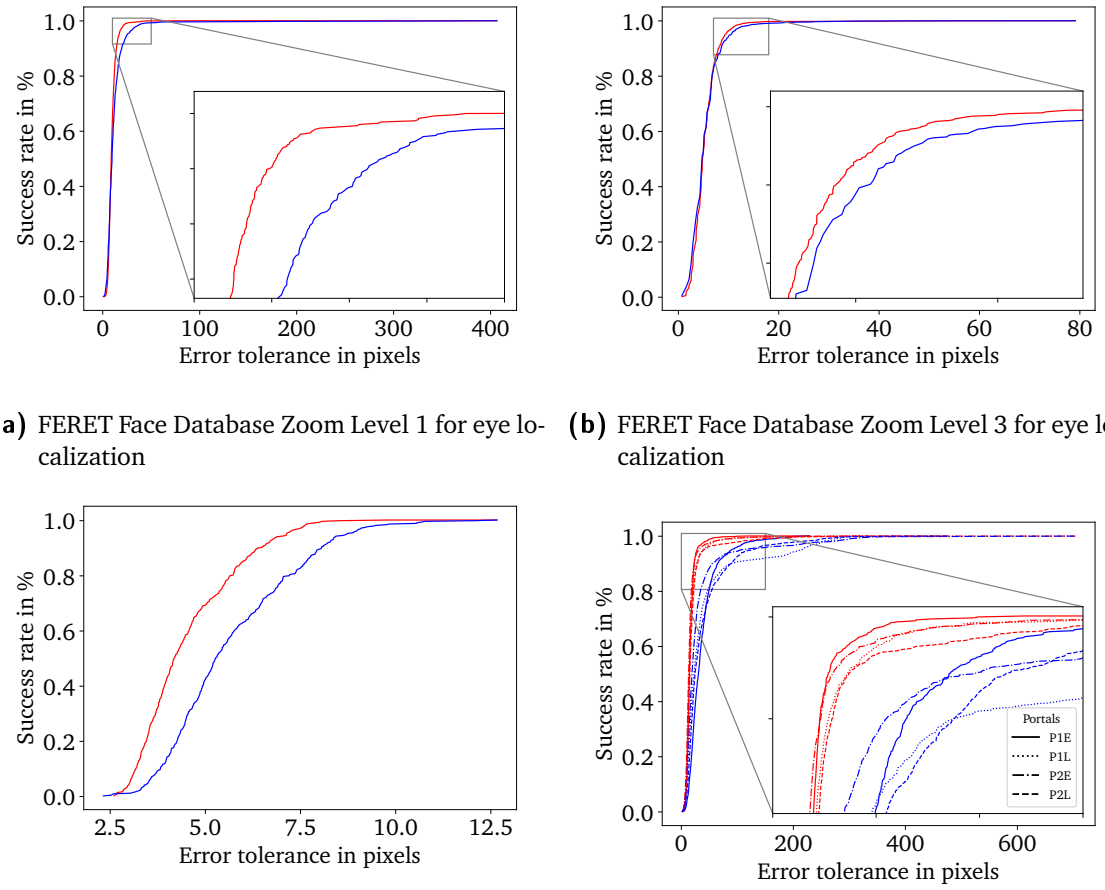
On all three databases, there is a clear improvement of the accuracy and the mean error in pixels. Analyzing the results of each zoom level individually reveals that there is an improvement on each zoom level. This improvement can be seen on image extracts based on the previous localization result as well as on image extracts generated around the annotated ground truth position.

Improvements for large target object variability

It was expected that the SCM is useful for a better discrimination between target object and background. We can see that this assumption is correct on the Chokepoint Dataset in the first zoom level. As described in Chapter 5.3, the main reason for the poor performance of the DGHT on the Chokepoint Dataset in the first zoom level is the combination of large target object variability with a non-static background. The large target object variability leads to a comparably larger number of model points from mutually exclusive variations, which might support the same localization hypothesis, e.g. in the non-static background. This issue is directly addressed by the SCM and solved as we can see on the Chokepoint Dataset in the first zoom level (see Figure 6.6d).

Table 6.1: Comparison of results for the Chokepoint Dataset between DGHT baseline results and DGHT+SCM for different portals, i.e. P1E, P1L, P2E, P2L. The mean error is given in pixels.

System	Portal	Zoom level	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\Xi(1)$	$\varnothing(\epsilon_n)$
DGHT	P1E	1/4	0.2%	4.9%	28.7%	65.6%	27.4
		2/4	9.4%	56.9%	81.5%	90.1%	12.6
		3/4	31.4%	78.5%	85.9%	91.1%	9.6
		4/4	53.5%	84.8%	86.9%	89.2%	8.7
DGHT	P1E	1/4	2.2%	29.2%	72.1%	96.0%	12.5
+ SCM		2/4	10.8%	69.5%	88.2%	98.3%	7.6
		3/4	43.4%	90.1%	95.9%	99.1%	4.2
		4/4	79.4%	97.6%	98.2%	98.7%	2.6
DGHT	P1L	1/4	0.4%	6.5%	26.2%	61.6%	32.5
		2/4	4.3%	32.9%	61.6%	81.9%	21.8
		3/4	16.3%	67.3%	86.1%	88.7%	16.9
		4/4	33.9%	82.3%	87.5%	88.4%	15.9
DGHT	P1L	1/4	1.4%	16.0%	51.4%	88.9%	13.6
+ SCM		2/4	6.7%	48.7%	87.1%	98.4%	7.2
		3/4	29.7%	85.0%	98.2%	99.2%	4.3
		4/4	67.2%	97.8%	99.2%	99.3%	2.8
DGHT	P2E	1/4	0.3%	7.1%	33.6%	71.2%	24.5
		2/4	3.1%	29.4%	67.4%	89.3%	16.6
		3/4	11.2%	61.5%	85.2%	90.5%	13.5
		4/4	23.5%	76.6%	88.0%	89.8%	12.5
DGHT	P2E	1/4	1.1%	17.5%	56.5%	91.5%	12.6
+ SCM		2/4	6.9%	49.0%	86.8%	98.0%	7.7
		3/4	21.3%	82.5%	97.5%	98.4%	5.3
		4/4	59.9%	96.5%	97.9%	98.2%	3.8
DGHT	P2L	1/4	0.4%	2.8%	13.9%	55.0%	29.6
		2/4	2.8%	26.5%	65.7%	82.2%	17.7
		3/4	12.2%	61.0%	83.8%	85.5%	14.1
		4/4	27.9%	79.5%	84.5%	85.4%	12.9
DGHT	P2L	1/4	1.1%	11.5%	43.9%	85.7%	15.5
+ SCM		2/4	6.9%	44.6%	88.1%	97.5%	8.3
		3/4	25.9%	83.2%	98.2%	98.4%	5.4
		4/4	63.5%	97.7%	98.3%	98.4%	4.0



(a) FERET Face Database Zoom Level 1 for eye localization

(b) FERET Face Database Zoom Level 3 for eye localization

(c) RWTH Hand Database Zoom Level 5 averaged over all landmarks

(d) Chokepoint Dataset Zoom Level 1

Figure 6.6: Comparison of the success rates for the DGHT (blue lines) with DGHT+SCM (red lines). For the Chokepoint Dataset the four portals are shown with different line styles. The red lines always show higher success rates demonstrating the strong improvement achieved by the SCM. Note, the image extracts used in panels (b) and (c) have been chosen to be located around the ground truth coordinates. These figures presents examples. See the appendix (Figure A.1) for all zoom levels on all databases.

Improvements for small target object variability

Furthermore, the SCM improves the previous localization result on all databases and at each zoom level using image extracts based on the previous localization result as well as image extracts generated around the annotated ground truth position (see Table 6.1, Table 6.2, Table 6.3 and Figure 6.6). At the last zoom level on the FERET Face Database or RWTH Hand Database, especially when using image extracts based on the annotated ground truth position, the DGHT results are already very good. This was to be expected since the object variability is limited and the image extract does not contain any background. Nevertheless, also in these scenarios, where the DGHT is already working very well, the SCM improves the performance of the DGHT.

Feature importance

Each node of the Random-Forest in the SCM was trained to decrease the impurity of the samples in the child nodes, i.e. to increase the homogeneity. This means that the decreasing impurity, weighted by the relative number of samples reaching that node, is an indicator of the importance of that node [15]. Since the test in this node is based on a feature, here

Table 6.2: The localization accuracy (Ξ, see Equation 4.5) on the FERET Face Database for different error tolerances (0.05, 0.1, 0.25, and 0.5) and the mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels

	Zoom	eye					nose					mouth				
System	level	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$
	1/3	23.7%	74.4%	98.6%	99.6%	9.1	44.5%	82.7%	98.8%	99.7%	9.6	45.5%	83.0%	97.5%	99.1%	10.2
DGHT	2/3	59.4%	94.9%	98.1%	98.1%	6.7	56.5%	91.4%	98.9%	99.5%	8.1	64.4%	92.5%	98.2%	98.6%	8.2
	3/3	74.7%	96.4%	97.9%	98.1%	5.9	70.1%	95.8%	98.7%	99.5%	6.9	76.1%	94.3%	98.5%	98.7%	7.2
DGHT	1/3	29.9%	83.7%	99.6%	99.9%	7.4	47.2%	87.2%	99.5%	99.5%	8.0	46.9%	86.0%	98.8%	99.7%	8.5
+	2/3	65.1%	98.1%	99.6%	99.8%	4.9	68.4%	97.3%	99.7%	99.7%	5.7	69.7%	95.7%	99.0%	99.4%	6.2
SCM	3/3	76.9%	98.6%	99.6%	99.8%	4.3	72.1%	97.6%	99.6%	99.6%	5.3	79.2%	96.6%	99.1%	99.3%	5.4

Table 6.3: The localization accuracy ($\Xi(\frac{6}{256})$, see Equation 4.5) for the different landmarks (1D,2D, ... 15D) on the RWTH Hand Database, mean accuracy over all landmarks ($\varnothing(\Xi(\frac{6}{256}))$), and mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels

	Zoom	$\Xi(\frac{6}{256})$													
System	level	1D	2D	3D	5D	6D	7D	9D	10D	11D	13D	14D	15D	$\varnothing(\Xi(\frac{6}{256}))$	$\varnothing(\epsilon_n)$
	1/5	64.1%	70.9%	81.6%	68.4%	83.3%	85.9%	82.0%	81.8%	91.0%	71.6%	82.5%	83.3%	78.9%	37.6
	2/5	88.1%	92.2%	95.1%	94.9%	96.1%	96.4%	92.2%	98.1%	98.5%	91.5%	95.1%	94.7%	94.4%	20.9
DGHT	3/5	95.9%	94.7%	97.1%	96.8%	97.8%	97.6%	96.1%	98.5%	99.8%	93.2%	96.6%	97.8%	96.8%	15.1
	4/5	96.4%	95.6%	97.3%	97.3%	98.1%	97.6%	96.1%	98.5%	99.8%	93.7%	96.6%	97.8%	97.1%	12.8
	5/5	96.4%	95.9%	97.3%	97.3%	98.1%	97.6%	96.1%	98.5%	99.8%	93.4%	96.6%	97.8%	97.1%	12.4
	1/5	86.0%	87.8%	92.2%	90.4%	92.2%	95.6%	93.4%	96.1%	97.7%	92.4%	93.6%	95.7%	92.8%	23.9
DGHT	2/5	98.9%	98.8%	99.3%	99.5%	99.6%	99.5%	99.0%	99.7%	100.0%	99.0%	99.0%	99.9%	99.3%	11.9
+	3/5	99.6%	99.0%	99.7%	99.2%	99.7%	99.7%	99.0%	99.7%	100.0%	98.5%	99.0%	100.0%	99.4%	7.8
SCM	4/5	99.6%	99.0%	99.6%	99.2%	99.8%	99.7%	99.0%	99.7%	100.0%	98.5%	98.9%	100.0%	99.4%	6.5
	5/5	99.6%	99.0%	99.6%	99.2%	99.8%	99.7%	99.0%	99.7%	100.0%	98.4%	98.9%	100.0%	99.4%	6.5

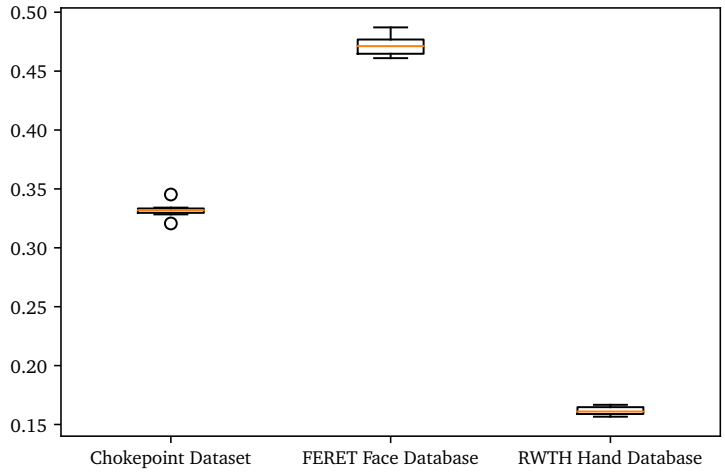


Figure 6.7: Correlation between DGHT weight and SCM feature importance for each dataset at zoom level 1. The correlation is calculated for each SCM model independently. The distributions of correlation factors results from different landmarks, portals and initialization for the random number generator used in the Random-Forest classifier (10 different initialization were used for each landmark).

the voting of a model point, it is possible to calculate the relative importance of each model point for the SCM. Also the DGHT estimates the importance of the model points given by their absolute weights. Therefore, it would be interesting to see, how the two importance values are correlated. On the first zoom level, the correlation is higher for the FERET Face Database than for the Chokepoint Dataset (see Figure 6.7). This might lead to the conclusion that the larger the error reduction by the SCM is, the more the SCM focuses on different model points compared with the DGHT. However, on the RWTH Hand Database the correlation is very low. To some extend, both can be explained. A strong correlation between absolute DGHT weights and the feature importance in the SCM could mean that the SCM could not improve the DGHT so much because both approaches focus on the same model points. However, it is also possible that the improvements of the SCM results from the combination of features, which cannot be modulated in the DGHT. By contrast, a low correlation means that the DGHT and the SCM focus on different features. This shows that there is no general rule defining how important a feature is. The importance of a feature, here model point, always depends on the algorithm used. Therefore, it is possible that two different approaches, focusing on different features, achieve almost the same results.

6.6 Parameter influence

6.6.1 Random Seeds

Setup

The Random-Forest classifier depends on a random number generator. Therefore, the results are also partly random. To analyze whether, against expectations, there is a significant influence from the initialization of the random number generator, each experiment was repeated

Table 6.4: Analysis of the influence of random seed initialization on the FERET Face Database for eye localization. The standard deviation over the mean pixel errors for all 10 experiments is very small, much smaller than the standard deviation within each experiment.

	$\varnothing(\varepsilon)$	$sd[\varepsilon]$	$sd[\varnothing(\varepsilon)]$
SCM Zoom Level 1	7.40497	6.64454	0.0436334
SCM Zoom Level 2	4.93438	6.21272	0.0351338
SCM Zoom Level 3	4.29833	6.38088	0.0296795

10 times with different initial values. For each initial value, the mean localization error $\varnothing(\varepsilon)$ and the standard deviation over the localization errors for each image $sd[\varepsilon]$ were calculated as well as the standard deviation over the mean localization errors $sd[\varnothing(\varepsilon)]$.

No significant influence

On average, $sd[\varepsilon]$ is higher by more than a factor of 100 compared to $sd[\varnothing(\varepsilon)]$ (see Table 6.4). This shows that the initialization of the random number generator has no significant effect on the localisation success. The image content has a much stronger influence on the localization error for the same initial value than the initialization of the random number generator.

Influence on final success rates

The final success rates for the localization of the iris on the FERET Face Database in these experiments varies between 98.4% and 98.8%. On the RWTH Hand Database, the mean success rate of all epipyhes varies between 99.3% and 99.5%. Lastly, on the Chokepoint Dataset, the eye localization rate varies between 97.3% and 97.5%. Hence, the influence of the random number generator on the final success rates is negligible.

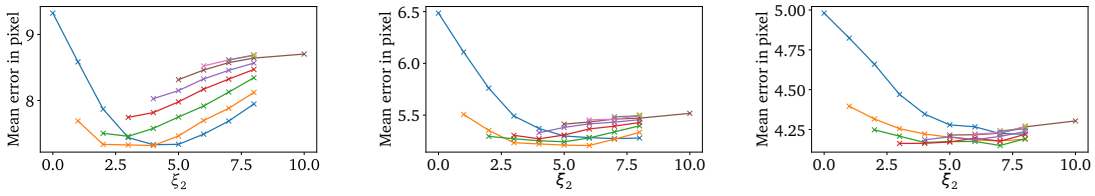
6.6.2 Class definition

Target object definition

The SCM utilizes a Random-Forest classifier which classifies hypotheses into one of the two classes "Target object" vs. "Non-Target object". As described above, this definition is simply based on the Euclidean distance between the hypothesis in question and the ground truth hypothesis. Strictly speaking, only the ground truth hypothesis can be the "Target object", i.e. there is only one point where the center of the left pupil is located. However, this assumes that the ground truth annotation is perfectly accurate. Most of the time this is not the case or it is impossible to define one specific point as the ground truth, e.g. which single point is the position of the nose? Therefore, two parameters were introduced into the SCM training framework, ξ_1 and ξ_2 (see Section 6.2.3) to compensate for small annotation errors, since they will not necessarily result in an assignment to the other class. For the same reason, we expect that this approach helps to establish a more robust classifier.

ξ_1 depends on the accuracy of the ground truth

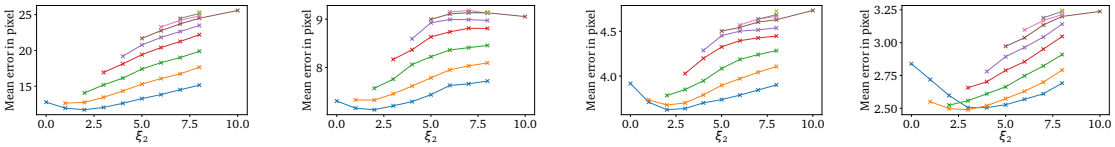
However, introducing these parameters requires that they are carefully set, mainly based on the allowed error tolerance and the accuracy of the ground truth. Since ξ_1 defines the region in which a Hough-cell is considered as the correct target object hypothesis, the optimal value for ξ_1 mainly depends on the precision of the ground truth. If the ground truth is precise enough to allow for a clear and correct definition of the target Hough-cell, ξ_1 can be set to 0. In this case only the correct target Hough-cell is considered as "Target object" class and the classifier can learn the shape of the target object without blurring by neighborhood cells. Setting ξ_1 to a higher value results in more Hough-cells per object being considered as correct. Since the classifier is trained to classify all of these cells as correct, the classifier has to find



(a) FERET Face Database, zoom level 1, eye localization

(b) FERET Face Database, zoom level 2, eye localization

(c) FERET Face Database, zoom level 3, eye localization



(d) Chokepoint Dataset portal P1E, zoom level 1

(e) Chokepoint Dataset portal P1E, zoom level 2

(f) Chokepoint Dataset portal P1E, zoom level 3

(g) Chokepoint Dataset portal P1E, zoom level 4

Figure 6.8: The mean error for different ξ_1 and ξ_2 for selected experiments. The x-axis shows the ξ_2 value, whereas each line represents one ξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, which is the value for ξ_1. See the appendix (Figure A.2, A.3, and A.4) for the results of all experiments.

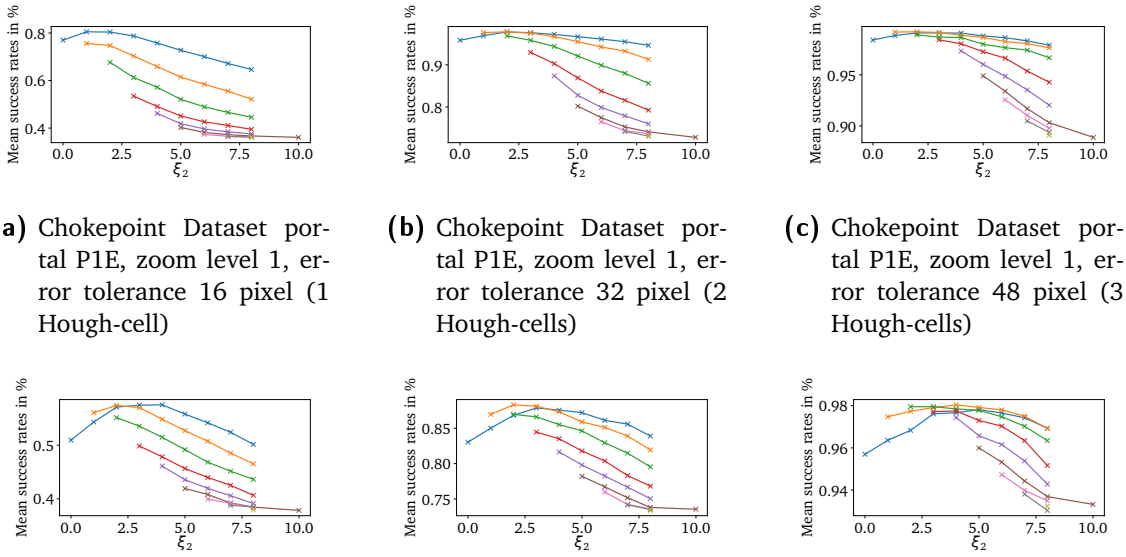
a more general model covering all of these samples. The larger ξ_1 is, the more fuzzy the resulting model will become, making a discrimination from the "Non-Target object" class more difficult. Therefore, if the ground truth is precise enough, it is better to set ξ_1 to 0 which will result in fewer samples for the "Target object" class. This is better than to spoil the model with almost correct samples. By contrast, if annotations are slightly incorrect so that the true target Hough-cell deviates slightly from the actual target Hough-cell, the model for the class "Target object" needs to be slightly fuzzy and therefore setting ξ_1 to a higher value ensures that the true target Hough-cell will always be included in the samples for training the "Target object" class. In conclusion, ξ_1 should be set to a value of approximately the variance of the accuracy of the ground truth annotations.

ξ_2 depends on the allowed error tolerance

ξ_2 defines the minimal distances at which a hypothesis belongs to the class "Non-Target object". This value therefore depends mainly on the acceptable error tolerance during validation. In theory, ξ_2 should be set to the minimal allowed error tolerance. In practice, however, sometimes no strict error tolerance is known or the aim is to minimize the mean error. Furthermore, a gap between ξ_1 and ξ_2 can be useful for better discrimination. This should be considered when choosing the value for ξ_2.

Number of samples

Note, each tree in the Random-Forest classifier was trained with an equal number of randomly selected samples of both classes. Changing ξ_1 or ξ_2 will result in a smaller or larger number of samples for one or both classes. This is compensated by using the remaining samples more or less often during training, ultimately still resulting in the same number of samples from both classes being used for training. Hence, the number of correct and incorrect hypotheses should have a negligible influence on training.



(a) Chokepoint Dataset portal P1E, zoom level 1, error tolerance 16 pixel (1 Hough-cell)

(b) Chokepoint Dataset portal P1E, zoom level 1, error tolerance 32 pixel (2 Hough-cells)

(c) Chokepoint Dataset portal P1E, zoom level 1, error tolerance 48 pixel (3 Hough-cells)

(d) Chokepoint Dataset portal P1E, zoom level 4, error tolerance 2 pixel (1 Hough-cell)

(e) Chokepoint Dataset portal P1E, zoom level 4, error tolerance 4 pixel (2 Hough-cells)

(f) Chokepoint Dataset portal P1E, zoom level 4, error tolerance 6 pixel (3 Hough-cells)

Figure 6.9: The success rate for different ξ_1 and ξ_2 for selected experiments and different error tolerances. The x-axis shows the ξ_2 value, whereas each line represents a different value for ξ_1. Since $\xi_2 \geq \xi_1$ the lines start at different ξ_2 values corresponding to ξ_1. It can be clearly seen that the higher the allowed error tolerance the higher the optimal ξ_2 becomes. See the appendix (Figure A.5, A.6, and A.7) for the results of all experiments.

Experiments

To confirm the aforementioned assumptions about setting ξ_1 and ξ_2, several experiments on the three Databases, FERET Face Database, RWTH Hand Database and Chokepoint Dataset on several zoom levels and with several error settings were conducted.

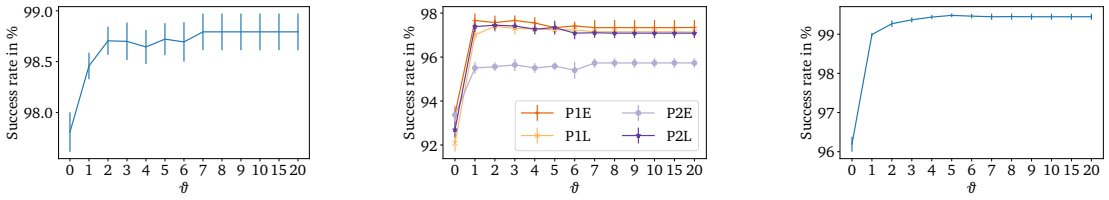
Evaluating ξ_1

Considering the mean pixel error per image shows that a higher ξ_1 produces better results for a higher zoom level (see Figure 6.8). For example on the FERET Face Database $\xi_1 = 3$ produces worse results at zoom level 1, whereas at zoom level 3 it produces almost the best results. Since in an early zoom level, the downsampling is larger and therefore small inaccuracies of the ground truth still lead to the same target Hough-cell, this is an indication that ξ_1 depends on the precision of the ground truth.

Chokpeoint vs FERET

Comparing the optimal settings for ξ_1 for eye detection on the FERET Face Database and Chokepoint Dataset reveals that ξ_1 should be smaller for the Chokepoint Dataset. Moreover, at both zoom levels 1 and 2, $\xi_1 = 0$ achieves much better results for the Chokepoint Dataset than for the FERET Face Database. Considering the downsampling factor and the eye distance on the FERET Face Database, the pupil has a size of 0.4 to 1.35 Hough-cells at zoom level 1. On the Chokepoint Dataset, the pupil has only a size of 0.05 to 0.3 Hough-cells.[1] Manually labeling the actual center point of the pupil is very difficult, but labeling some point inside the pupil is feasible. Therefore, it can be expected that labels of the eye centers have in general at least the precision of the pupil size. In this case, setting ξ_1 to 0 on the Chokepoint Dataset for the first two zoom levels is reasonable, since it can be assumed that the ground truth Hough-

[1] The size of the pupil is approximately 0.05 of the eye distance.



(a) Iris localization on the FERET Face Database ($\Gamma = 0.1$)

(b) Eye localization on the Chokepoint Dataset ($\Gamma = 0.25$)

(c) Mean success rate on the RWTH Hand Database ($\Gamma = \frac{6}{256}$)

Figure 6.10: Success rates for selected experiments and different neighborhood sizes (ϑ). See the appendix (Figure A.11) for the results of all experiments.

cell is independent from the slight variation of the ground truth annotations. On the FERET Face Database, however, the annotation is more imprecise due to the relatively higher pupil size and, accordingly, a higher ξ_1 leads to the best results.

Special case $\xi_1 = 0$

Furthermore, it also seems that $\xi_1 = 0$ is sometimes a special case. On the Chokepoint Dataset at early zoom levels it fits very well to the other settings, but for later zoom levels or on FERET Face Database the results for $\xi_1 > 0$ show a coherent picture. This can also be explained by the precision of the annotation since $\xi_1 = 0$ is more sensitive to imprecise annotations than for $\xi_1 > 0$. This does not necessarily mean that $\xi_1 = 0$ leads to worse results but it depends more on ξ_2. For example, on the FERET Face Database at zoom level 1, $\xi_1 = 0$ and $\xi_2 = 0$ is worse, but the combination of $\xi_1 = 0$ and $\xi_2 = 4$ produces almost the best results. Assuming that the annotations might be slightly incorrect, $\xi_1 = 0$ and $\xi_2 = 0$ result in a wrong class definition for some samples, whereas with $\xi_1 = 0$ and $\xi_2 = 4$ the risk of a wrong class definition is reduced.

Evaluating ξ_2

Since the optimal value for ξ_2 mainly depends on the allowed error tolerance, it is useful to compare the success rates for different error tolerances with different ξ_2. Especially at the first zoom level the influence of ξ_2 is clear. Mostly for all ξ_1, the optimal value for ξ_2 increases with increasing error tolerance (see Figure 6.9).

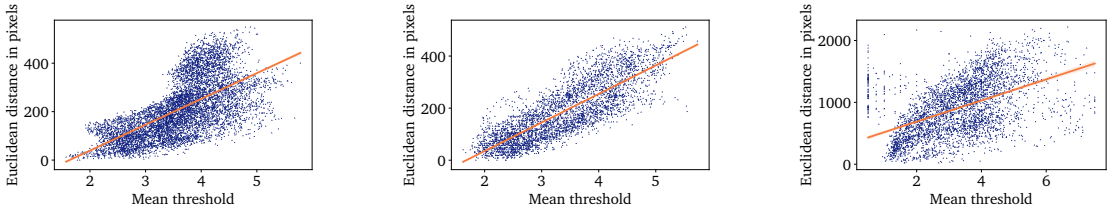
Conclusion

The experiments show that a good rule of thumb for setting ξ_1 is approximately the variation of the accuracy of the ground truth, translated into Hough-cell resolution. Additionally, it is a good advice to be careful when setting $\xi_1 = 0$. Setting ξ_2 to the value of the allowed error tolerance showed best results in the experiments. Furthermore, the experiments revealed that ξ_1 and ξ_2 can be optimized independently.

6.6.3 Neighboorhood

Parameter ϑ

In equation (6.1), a new parameter ϑ was introduced, defining the maximum neighborhood distance, in which GHT votes will be considered. The assumption is that a higher value is better, but at some point a saturation will be reached and therefore, the exact value of ϑ has, as long as it is large enough, a negligible influence on the localization accuracy. To evaluate this aspect, I conducted different experiments with different ϑ values from 0, i.e. no neighborhood, to 20.



(a) FERET Face Database, eye localization, zoom level 1

(b) Chokepoint Dataset, eye localization, zoom level 1

(c) RWTH Hand Database, 1D localization, zoom level 1

Figure 6.11: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for selected experiments. See appendix (Figure A.12, A.13, and A.14)for all experiments

Results

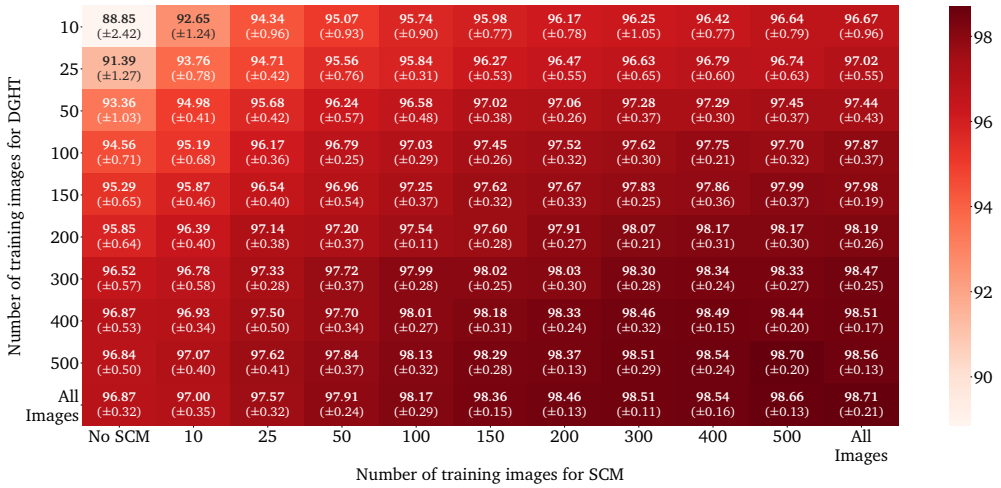
On the FERET Face Database, the iris localization rate ranges from 97.8%, with $\vartheta = 0$, to a maximum of 98.8% (see Figure 6.10). However, we can see that with $\vartheta \geq 2$ already a saturation is reached and the variance in iris localizing success rates is related to random effects rather than to ϑ. On the RWTH Hand Database the outcome is similar. On the Chokepoint Dataset for all portals a saturation is already reached with $\vartheta = 1$. Although it seems that on some portals the success rates decease slightly, this is not a significant decrease and could be the results of random effects. Still, also on the Chokepoint Dataset, it is clearly visible that the results with $\vartheta = 0$ are worse than with $\vartheta > 0$. Furthermore, on all databases, the results do not change for $\vartheta \geq 7$ and in fact the SCM models are the same. This shows that, in the conducted experiments, $\vartheta > 7$ does not add any additional information to $\mathbf{r}$ making $\vartheta = 7$ a good default value.

Importance of Hough-cell neighborhood

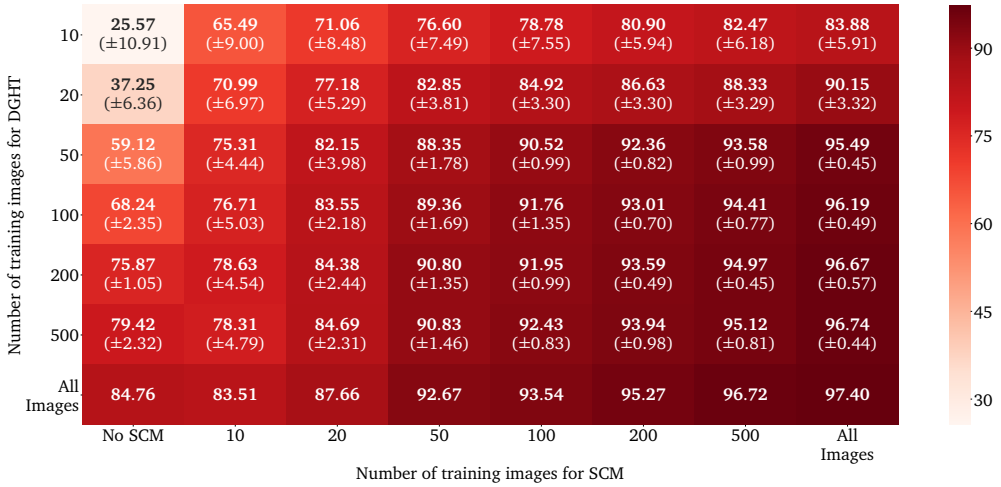
The results confirm the initial assumption that a larger ϑ is usually better, but a value of $\vartheta \in [2, 4]$ is normally a good choice. More importantly, the results show the importance of the neighborhood. Without considering a neighborhood ($\vartheta = 0$), the results clearly become worse. This is related to a general issue of GHT-based approaches. A model-feature-point-combination is only allowed to vote for one specific Hough-cell and small variations might result in voting for neighborhood cells. To some extend, this can be compensated, e.g. by binning the cells or using a Gaussian filter. However, the concept of considering the votes for neighborhood cells as in the SCM presents a more universal solution.

Optimal neighborhood size per model point

Binning of Hough-cells or using Gaussian filters will treat the votes of all model points equal. For example a model point with a distance to the reference point of e.g. 4 pixels with a binning factor of $\varrho = 2$ votes for hypotheses that are between 4 and 5 pixels away, which is a tolerance of 25%. If the distance between model point and reference point is 40 pixels, the model point votes for hypotheses that are between 40 and 41 pixels away, which is a tolerance of 2.5%. Therefore, model points with a larger distance to the reference points require a larger binning factor or a larger Gaussian filter. In [36], this is done heuristically by introducing a confidence weight depending on the distance between model and reference point. The Random-Forest classifier in the SCM can learn the optimal distance individually not only for each model point but also at each node of the decision trees using that optimal distance as a threshold in the test function. Although this is a more universal approach, we would still expect that with an increased distance between model and reference point also the average optimal neighborhood distance will increase. An analysis of the SCM models with $\vartheta = 7$ generally shows such a dependency. For some models, this dependency is quite strong and linear (see Figure 6.11), but there are always model points which do not fit that linear dependency. Furthermore,



(a) FERET Face Database eye localization with error tolerance of 0.1 eye distance

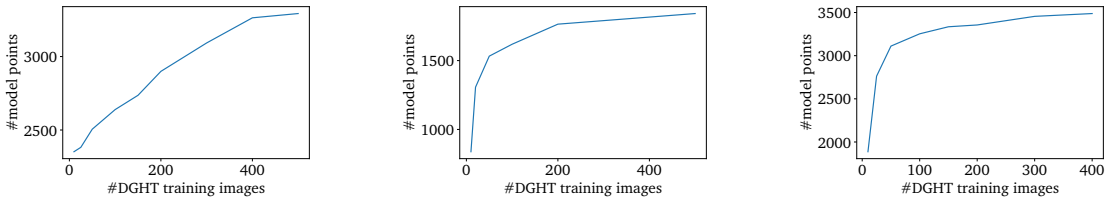


(b) Chokepoint Dataset P1E portal with error tolerance of 0.25 eye distance

Figure 6.12: Mean localization accuracy and standard deviation for selected experiments depending on the number of DGHT and SCM training images. The dark red signifies higher accuracy. This shows that keeping the number of SCM training images constant (columns), the accuracy varies less than keeping the number of training images for the DGHT constant (rows). See appendix (Figure A.8, A.9, and A.10) for more experiments.

I calculated the average threshold per model point over 10 SCM models over all trees and nodes using the model point in question. The standard deviation is on average 1.49, which is quite high considering that the neighborhood size ranges only from 0 to 7.[2] This shows that the assumption that a larger distance between model and reference point requires a larger threshold might be true in general, but for particular cases, the SCM benefits from selecting the threshold individually.

[2]The standard deviation normalized by the mean value is on average 0.51.



(a) FERET Face Database **(b)** Chokepoint Dataset **(c)** RWTH Hand Database

Figure 6.13: Average number of model points per database depending on the number of training images used for generating the DGHT models. The number of model points is averaged over all landmarks and zoom levels. The graphs show that with an increasing number of training images also the number of model points increases.

6.6.4 Number of training images

Setup

To evaluate the influence of the number of training images, multiple experiments were conducted in which the number of training images for both DGHT and SCM was varied in such a manner that each combination was evaluated (see Figure 6.12). For experiments with the same number of training images for the DGHT and the SCM, the same images were used for both methods. If the number of training images differed, images were selected to ensure maximum overlap between them for DGHT and SCM, i.e. all images used for the method which was trained on less images were part of the larger training corpus of the other method.

Results

On all three databases, when using the DGHT in combination with the SCM, increasing the number of training images for either the SCM or the DGHT does not improve overall localization performance equally. Increasing the number of training images for the SCM only increases performance more strongly than only improving the DGHT model through an increased number of training images (see Figure 6.12). This shows that to some extent the quality of the SCM is independent of the performance of the DGHT model and a well-trained SCM model (e.g. trained on a large number of training images) can compensate for a medium quality DGHT model.

Number of model points

Nevertheless, if too few training images are used to generate the DGHT model, the number of model points will decrease, which reduces the number of features for the SCM and in turn the potential discriminatory power of the SCM (see Figure 6.13).

Random influence

Note, the selection of the training images was done at random. Therefore, the experiments were conducted 10 times with different images and the average as well as the standard deviation of the localization accuracy was calculated. This allows to determine the influence of the random selection. The experiments using all training images for training the DGHT models on the Chokepoint Dataset were only performed once due to the required training time.

6.7 Conclusion

Clear improvement by the SCM

The SCM clearly improves the localization rates on all databases and at all zoom levels. Even the already very good results on the FERET Face Database and the RWTH Hand Database could

be further improved by the SCM. The improvement on the Chokepoint Dataset is much more pronounced; here the DGHT reached a mature level only due to the SCM.

Shape of voting model points

The results show very clearly the main drawback of GHT-based approaches: The independent voting of model points. Model points only based on edge features can be confused with too many model points so that additionally taking the shape of the voting model points into account is more informative than the total number of voting model points only. This chapter has shown that the set of model points voting for the same Hough-cell constitutes a learnable structure.

Advantages of the SCM

Since the independent voting of model points is a general issue of GHT-based approaches, there are some other techniques trying to solve this problem, mostly by grouping the data [170, 161]. An approach, developed at the same time as the SCM, and underlining the importance of this issue, is [13]. It also extracts more information from the set of model points that have voted than only the number of model points which have voted. However, this method is very basic and uses heuristic ideas such as the distance between these model points. By contrast, the SCM is a more sophisticated and general approach. Moreover, the SCM represents a framework transforming the model points which have voted into a feature space. Therefore, the SCM is neither restricted to the usage of the Random-Forest classifier nor is it restricted to a classification task discriminating only the classes "Target object" and "Non-Target-Object". It is possible to discriminate between different types of target objects or to apply additional classification tasks to the target object, such as gender or age classification.

Neighborhood

One important aspect of the SCM feature space is to consider not only the model points that voted for a specific Hough-cell, but also those that voted for its neighboring Hough-cells. This prevents another problem of GHT-based approaches, i.e. that each model-feature point combination only votes for a specific cell making the GHT very sensitive to small variations. To make GHT-based approaches more robust, some applications have used a Gaussian filter [70] so that votes for one cell also influence cells in the neighborhood. Binning the Hough-space [169] is another potential solution to combine the votes from neighboring Hough-cells. While such approaches may be partially successful, they decrease the precision of the localization. Furthermore, each vote is treated equally in these approaches. However, model points near the target landmark are less sensitive to small variations than model points far away, which has been addressed heuristically in [36]. The SCM, by contrast, directly addresses the issue of dealing with small variations by considering directly at which distance from the given Hough-cell a model point has voted. Thereby, it can preserve the precise information which model points have voted for this Hough-cell. The classifier within the SCM can also learn which model points have e.g. larger expected variation because they are further away from the target point.

Chapter 7

Stacked-GHT

7.1 Introduction

Target object description

An object is never atomic. Each object, which we want to detect, consists of multiple object parts in multiple hierarchical levels. For example, the human body consists of body parts such as head, arms, chest, etc. The head in turn consists of eyes, nose, mouth, etc. At a lower hierarchical level, each object consists of specific features, e.g. each edge point describes a specific part of the target object albeit very vaguely. In some way, each object detection algorithm can be considered as a combined detection of object parts. The success of CNNs is based on this idea: At first, pixel values are used and they are combined into very general and low-level features. These features are in turn spatially combined to form more complex features. This is repeated multiple times until a final description of the target object is achieved.

Example Hough-Forests

For better understanding, it is worth to look at Hough-Forest. An early version required the manual labeling of object parts, e.g. head, arms, etc. [68]. These parts were described and detected by a Random-Forest and each part voted for potential target point locations. In other words, the Random-Forest acts as a detection approach for object parts. The voting procedure, which is similar to the GHT, is a combination of object parts for detecting the target object. In its current version, the Hough-Forest does not need such manual labeling of object parts, but the object parts are automatically estimated due to their position relative to the target position. Whereas at the beginning the Hough-Forest resembled a method for object part combination, it later became a feature extractor albeit it also models the spatial distribution of, automatically determined, object parts.

Classical detection approaches

Before the success of CNNs, detection approaches usually contained two levels: One to create features from the raw pixel values and one for combining these features to describe the complete target object. For example in the DGHT, the raw image is used to generate edge features which are subsequently combined into the complete target object. However, the information about the spatial distribution of detection results of multiple object parts or landmarks was and is often used in various approaches. Sometimes heuristic knowledge about their distribution [31] is used. A very well-known detection approach is DPM whose name, "Deformable Part Models", already suggests that this approach contains not only one model, describing the whole target object at once, but has also partial models describing object parts independently. [134] concludes that a face detector that combines object part detection results is more useful if the number of training images is smaller. With a larger number of training images, object part combinations become less important.

Splitting target object variability

This is explainable by the fact that separating an object into different parts also splits the variability of the target object into object-part related variabilities. Using the human body as an example, this can be illustrated easily. All body parts such as head, chest, arms, and legs have different variations in appearance, e.g. the head looks different depending on whether one looks at it frontally or from the side, as does the chest, etc. However, it is possible that a person looks to the side resulting in a situation where the head is only visible from the side but the chest remains visible from the front. Jointly modeling the whole body requires training images for each possible variant combination. By contrast, when modeling each body part separately and combining these parts, it is sufficient to have training samples that contain each variation irrespectively of any other variation.

CNNs

If we follow this argument, CNNs should require very few training images since they have perfected these object part combinations. However, since CNNs do not use any preselected features but generate their features directly from the raw pixel values, the number of required training images increases strongly. Results show that, in general, deeper models with more layers perform better [86]. Therefore, we can assume that deeper models require less training images to achieve the same performance.

DGHT

As aforementioned, one precondition of this work is that the number of training images is limited. Hence, in this chapter I analyze how the information of the spatial distribution of landmarks could help to improve the localization results within the DGHT framework. The importance of this approach is shown in preliminary work [80, 79]. In this chapter, a general approach is presented, which uses the Hough-spaces of single landmark detections as features in a higher hierarchical level model. As we already saw for the example of the Hough-Forests, the difference between feature extraction and object part combination is indistinguishable. Although not analyzed here, it is also possible to use Hough-spaces from higher hierarchical level models and combine them in the same way building up a pipeline, hereafter called Stacked-GHT, which is similar to CNNs.

Modified Multi-Level Approach

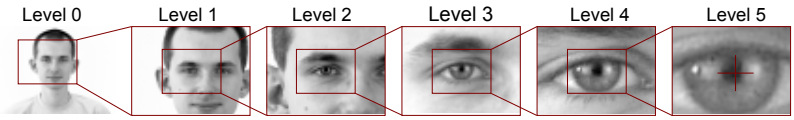
In addition to the Stacked-GHT, a modified multi-level approach is introduced here, which achieves an improved robustness by replacing the gradual reduction of the search space in [80] with a direct zooming into the eye region (Section 7.2.1). Both changes have been evaluated on the public PUT Face Database (Section 7.3) and led to a significant improvement over the standard method (Section 7.3.3).

7.2 Method

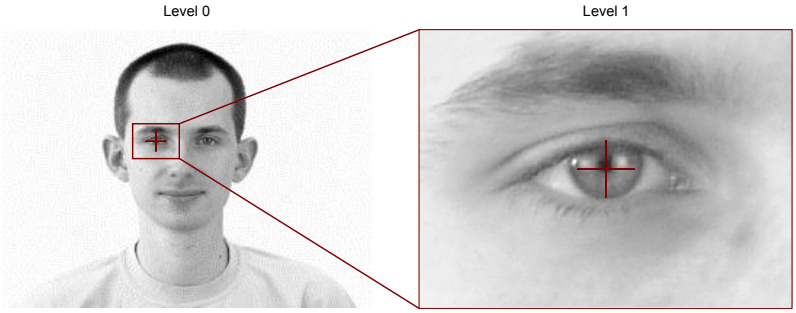
7.2.1 Modified Multi-Level-Approach

Multi-Level Approach

The Multi-Level-Approach (MLA) is a zoom-in strategy, in which the resolution is gradually increased around the suspected target point. By decreasing the image extract in question and increasing the resolution in each zoom level the visible structures range from global and coarse to local but fine structures. Since the different DGHT models, applied in the MLA, are specifically trained on the respective image extracts they learn relevant and discriminative structures in each zoom level. Therefore, the MLA is a good tradeoff between keeping sufficient target object details and suppressing noise and confusing objects.



(a) Standard Multi-Level-Approach with 6 zoom levels



(b) Modified Multi-Level-Approach with 2 zoom levels

Figure 7.1: Comparison of the standard Multi-Level-Approach with the modified Multi-Level-Approach

The MLA presented in prior publications [171] doubled the resolution and halved the size of the image extract in each zoom level, therefore keeping the number of pixels constant. For the task of eye localization on the public PUT Face Database this procedure was used with 6 zoom levels in [80] (Figure 7.1a).

MLA: double the resolution, half the size

It could be shown in [81], that the standard MLA procedure is prone to a confusion of the eyes at zoom levels, where both eyes might be visible while important discriminating structures are missing. Consequently, the modified MLA uses a higher resolution in the first zoom level in order to ensure a more accurate target localization than the standard approach. This especially aims at a reliable distinction between both eyes. In the second zoom level of the modified MLA the image extract is already restricted to a region containing only a single eye which excludes any confusion with the other eye. This image extract already has the full resolution and is used for the final localization (Figure 7.1b).

modified MLA

7.2.2 Stacked-GHT

Stacked-GHT

The Stacked-GHT occurs at two levels. In the first level, for each landmark Υ, special DGHT models $\mathscr{M}_{\Upsilon}^{1}$ are trained by using the standard DGHT procedure (section 5.1) and Canny Edge Images [17] as features. By applying these models to new images, individual Hough-spaces $\mathscr{H}^{1,\Upsilon}$ are generated, which are transformed into probability distributions by Equation (5.16) for target point localization of landmark Υ. Since (i) with the distribution of a landmark (e.g. left eye), the position of another landmark (e.g. right eye) can be estimated and (ii) the DGHT is neither restricted to edge images nor to 2D images, these landmark specific distributions are combined in a new 3D feature image

$$\mathscr{X}_n^2 = \{\mathscr{X}_n^{1,1}, ..., \mathscr{X}_n^{1,L}\}, \tag{7.1}$$

with L being the total number of landmarks, for the next localization level. For a given set of N training images, the corresponding 3D features $\mathscr{X}_1^2, ..., \mathscr{X}_N^2$ are used to train a higher-level 3D DGHT model $\mathscr{M}^2$ in the second level utilizing the standard DGHT training approach (section

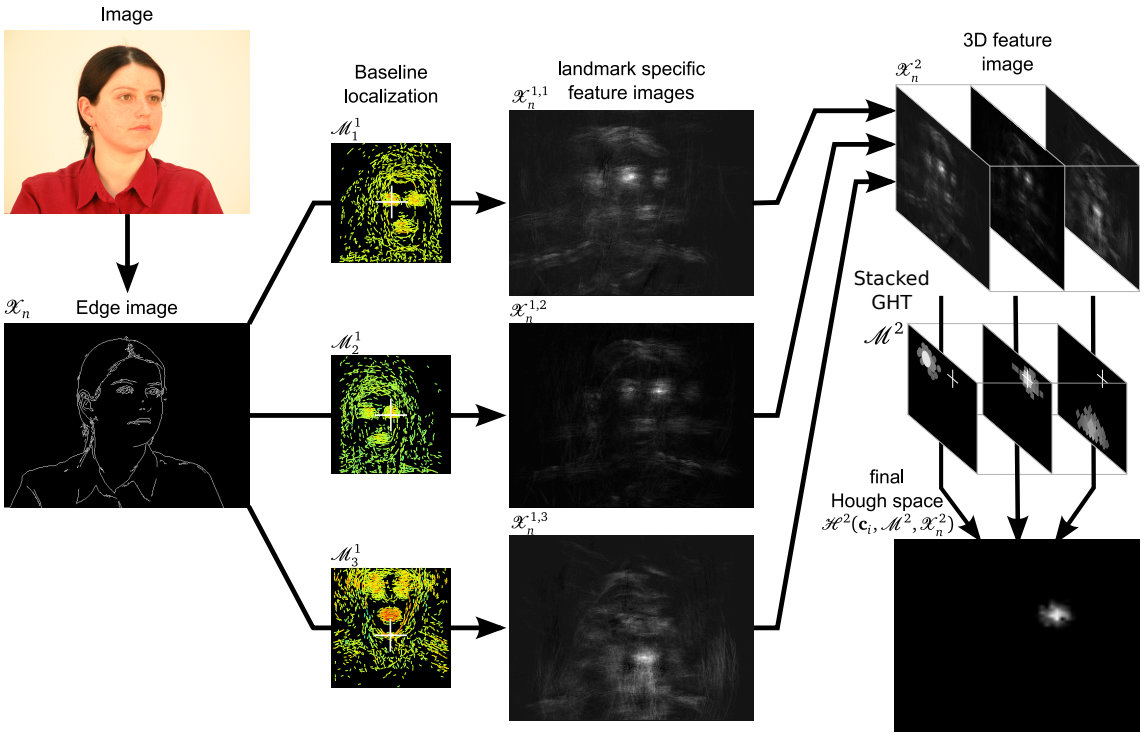


Figure 7.2: Illustration of the process of landmark combination: According to the standard procedure for landmark localization the image is transformed into a feature image (edge image). Subsequently, the edge-based DGHT models $\mathcal{M}_1^1$, $\mathcal{M}_2^1$, $\mathcal{M}_3^1$ are utilized for single localization of both eyes ($\mathcal{M}_1^1$ and $\mathcal{M}_2^1$) and the chin ($\mathcal{M}_3^1$). The thereby generated probability distributions $\mathcal{X}_n^{1,1}, \mathcal{X}_n^{1,2}, \mathcal{X}_n^{1,3}$ are combined into a 3D feature image $\mathcal{X}_n^2$. On this 3D feature image, a discriminatively trained 3D model $\mathcal{M}^2$ is applied for the final localization. Hence, $\mathcal{M}^2$ combines the information about the probable position of the individual facial landmarks related to the target landmark.

5.1). This model captures the relative position of the landmarks to each other and provides the final localization result.

Voting with feature values

The feature value $\phi_{\mathbf{x}_l^{1,\Upsilon}}$ specifies the probability of landmark Υ being localized at position $\mathbf{x}_l^{1,\Upsilon}$ for the given feature image $\mathcal{X}_n$and model $\mathcal{M}_\Upsilon^1$. Thus, it represents the certainty of the underlying localizer in GHT stack level one. This important source of information should be directly incorporated into the GHT voting procedure of level two in order to increase the influence of areas with high localization reliability. Therefore, the standard voting procedure (Equation (5.8)) is adapted to directly vote with the feature value $\phi_{\mathbf{x}_l^{1,\Upsilon}}$ instead of voting with the value 1. In addition to that, a summation over the L landmarks has to be done in order to combine the results from the different landmark localizations in level one. This leads to the following modified voting procedure for the GHT in level two:

$$\mathcal{H}^2(\mathbf{c}_i, \mathcal{M}, \mathcal{X}_n) = \sum_{j=1}^{|\mathcal{M}|} \lambda_j^2 f_j^2(\mathbf{c}_i, \mathcal{X}_n) \tag{7.2}$$

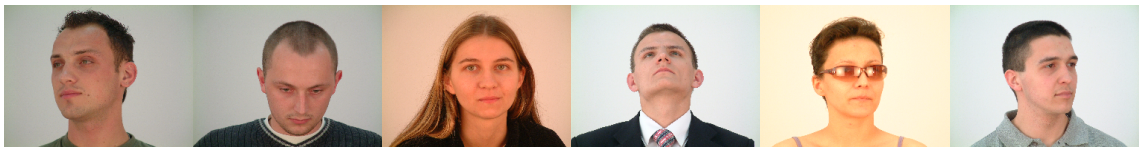


Figure 7.3: Illustration of the large head position variability contained in the PUT database.

with[1]

$$f_j^2(\mathbf{c}_i, \mathcal{X}_n) = \sum_{\forall \mathcal{X}_n^{1,\Upsilon} \in \mathcal{X}_n^2} \sum_{\forall \mathbf{x}_l^{1,\Upsilon} \in \mathcal{X}_n^{1,\Upsilon}} \begin{cases} \phi_{\mathbf{x}_l^{1,\Upsilon}}, & \text{if } \mathbf{c}_i = \lfloor (\mathbf{x}_l^{1,\Upsilon} - \mathbf{m}_j)/\varrho \rfloor \text{ and } \phi_{\mathbf{x}_l^{1,\Upsilon}} > \tau. \\ 0, & \text{otherwise.} \end{cases} \tag{7.3}$$

Note, a low $\phi_{\mathbf{x}_l^{1,\Upsilon}}$ has a negligible influence on the outcome of the voting processing. However, since each vote will be considered during model training, a large number of low votes will significantly increase the processing time. Therefore, only $\phi_{\mathbf{x}_l^{1,\Upsilon}}$ with a minimum probability τ will be allowed to vote for potential target point localizations. DGHT training

7.3 Experiments

7.3.1 Data

The experiments were conducted using the public PUT Face Database [100] in training and evaluation, which includes 9971 images from 100 subjects. The high resolution (2048×1536 pixels) color images were taken under controlled lighting conditions in front of a uniform background. Since 30 facial landmarks are provided for each image in this corpus it is very well suited for investigating the presented landmark combination technique. Despite of the neutral background, the corpus is challenging due to the strong variability of head positions (see Figure 7.3). PUT Face Database

As in [80, 81], the 100 different subjects in the corpus were divided into a training set, containing 60 subjects, and an evaluation set with the remaining 40 subjects. For better comparability the evaluation corpus is identical to [80, 81] and includes 3830 images. The training was performed on 600 images which have been randomly selected from the training set. Corpora

7.3.2 Setup

In the modified MLA (Section 7.2.1), the resolution is reduced by a factor of eight at zoom level 0 (see Figure 7.1b). Around the target point, localized in this level, an image extract with original resolution and the size of one-eight of the complete image is taken for the second and final localization step. The system works with Canny Edge features [17] and applies a standard DGHT training procedure for generating the specific GHT models for the two localization MLA

[1] Note that $\lfloor \mathbf{a} \rfloor$ denotes the floor of each component of $\mathbf{a}$.

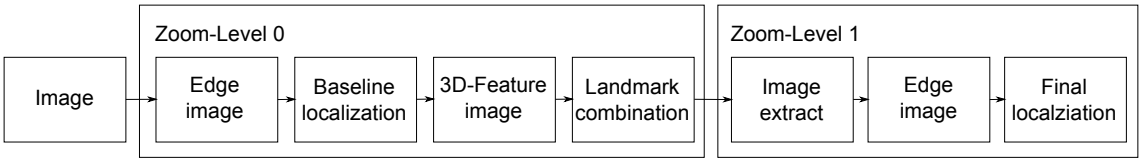


Figure 7.4: System overview of modified multi-level approach with landmark combination.

Table 7.1: Experimental results comparing different systems for different error tolerances.

	e<0.1	**e<0.15**	**e<0.2**	**e<0.25**
Kasinski et al. [101]	94.0%	-	-	-
Standard MLA with 6 zoom levels [81]	95.0%	95.4%	96.0%	96.5%
Standard MLA with model interpolation [80]	96.6%	97.1%	97.6%	98.1%
Modified MLA with 2 zoom levels	97.2%	97.6%	98.0%	98.2%
Modified MLA with landmark combination	97.9%	98.5%	98.9%	99.1%

levels. All described experiments have been performed using a 64 bit system with an Intel Xeon W3520 with 2.66 GHz and 24 GB RAM.

Stacked-GHT

To further enhance the robustness of the modified MLA at zoom level 0, a combination of three landmarks (both eyes and chin) is applied by the Stacked-GHT procedure described in Section 7.2.2: Using standard DGHT models, based on Canny Edge features, three probability distributions for the landmark locations are generated (see Section 7.2.2). These distributions are combined into a 3D feature image $\mathcal{X}_n^2$, ignoring values of less than 0.01 in order to decrease the processing time and to reduce noise. With a specifically trained 3D DGHT model a robust target localization at zoom level 0 is performed using the modified voting procedure (equation (7.2)) and the result is handed over to zoom level 1. Figure 7.4 gives an overview of the system with the modified MLA and landmark combination.

Validation

To determine the localization rate, the measurement explained in [96] is used, in which the larger localization error of both eyes is normalized with the eye distance. An error of less than 0.1 / 0.25 therefore corresponds to a localization result approximately located within the iris / eye. Due to slightly inaccurate annotations, provided by the PUT Face database, an error distance of less than 0.1 is not meaningful since the inaccuracy would be higher than the error distance.

7.3.3 Results

Modified MLA

By using the modified MLA a success rate of 97.2% for a localization within in the iris could be achieved on the evaluation corpus. This is an improvement of 0.6% compared to the previously best published result and a gain of 2.2% to the published result obtained with a standard method (Table 7.1). A good indicator for the localization robustness of zoom level 0 of the MLA is given by the number of detected anchor points lying outside the optimal image extract.

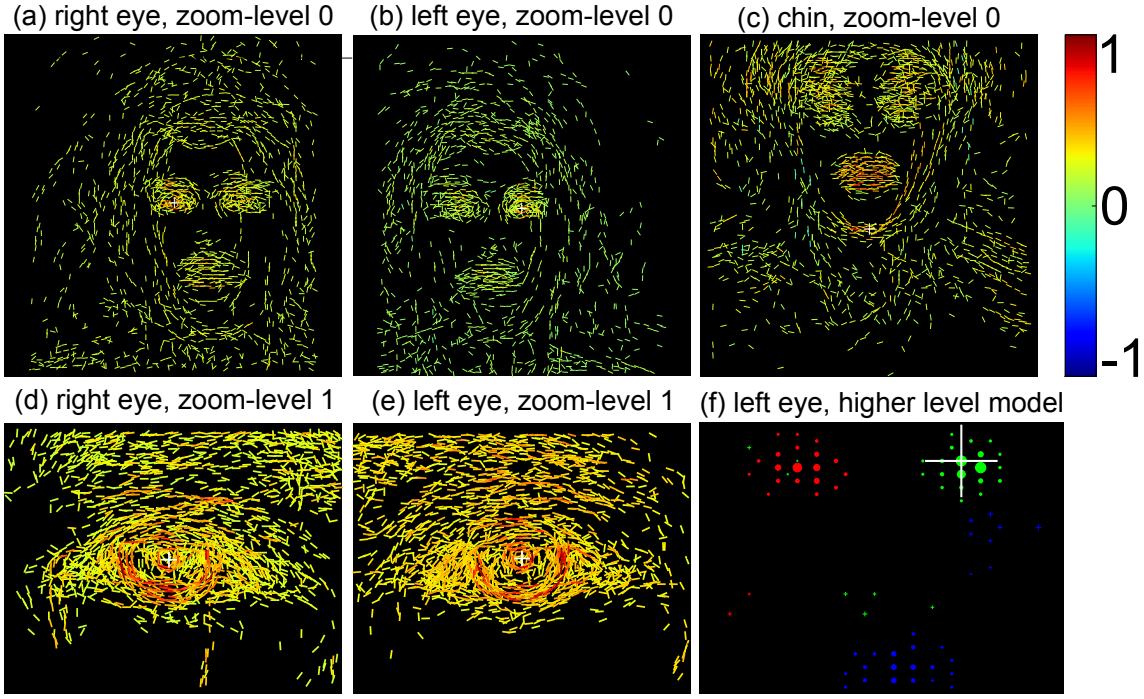


Figure 7.5: (a) to (e): DGHT models used for baseline landmark localization, where the color value denotes the individual model point weight. (f): 3D DGHT model used for landmark combination. The color illustrate the corresponding landmark layer (red: right eye layer, green: left eye layer, blue: chin layer) and the size represents the model point weight. Note, that model points with negative weights, which ensure a better discrimination of similar object, are shown as plus ('+').

In comparison to the standard MLA approach with a comparable image extract, this number could be reduced from 130 to 50 by applying the described modifications.

A further improvement of the localization robustness at zoom level 0 of the modified MLA could be achieved by using the described landmark combination technique for three facial landmarks. This measure reduced the number of detected anchor points lying outside the optimal image extract to 20 and therefore improved the error rate to 97.9% for iris localization. Considering a less restricted fault tolerance, a localization inside the eye was achieved in 99.1% (Table 7.1). The generated landmark localization models $\mathcal{M}_l$ are shown in Figure 7.5 (a) to (e). The model points are represented as lines to visualize their orientation while the color value illustrates their weight. Figure 7.5 (f) displays the 3D DGHT model of zoom level 0. Here, the symbol of a model point indicates the corresponding landmark and the gray value represents again the individual weight as obtained by the discriminative training process. Stacked-GHT

7.4 Discussion

The significant improvement of the modified MLA can be mostly explained by a better discrimination between both eyes. This is due to an improved localization robustness at zoom level 0 which may be assigned to a better and more detailed DGHT model with a strong focus on both eyes (e.g. see Figure 7.5(a)). Comparing the models of the standard and the modified Modified MLA

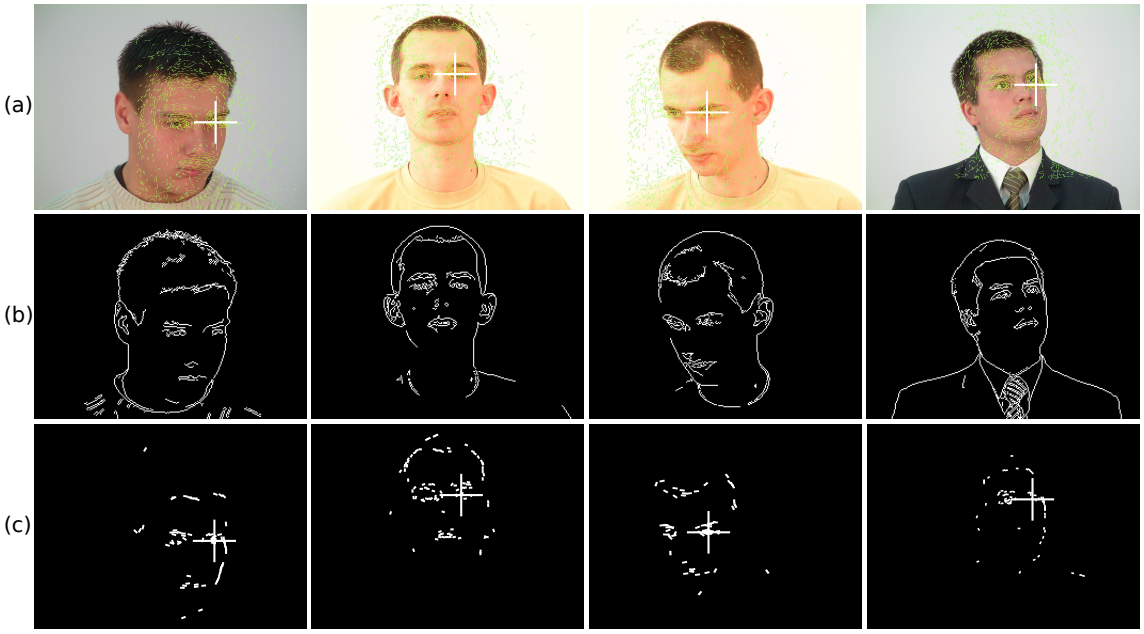


Figure 7.6: (a) Original images with overlayed model, (b) corresponding feature images, (c) model points which voted for the best localization hypotheses in the respective image. The used model is identical to Figure 7.5(b).

MLA, it is noticeable that the average number of model points has substantially increased from 357 to 1807. This rise results from the higher resolution in the modified MLA which leads to an increase of feature points and shape variation, compensated by a larger number of model points.

Number of voting model points

It is interesting to note that only a few model points of a given localization model are relevant for a single image. Therefore, the percentage of model points, voting for the best Hough-cell, is only 11% on average for the standard MLA. For the modified MLA, however, this number is even smaller and amounts to only 5% which underlines the fact that the overall size of the model results from the large amount of variation over all images.

Processing time

The higher number of feature and model points also explains an increase of the processing time from about 600 ms for the standard MLA to 970 ms for the modified approach. Note, the system has not been optimized for runtime performance, yet.

Visible analyze of DGHT models

A clear advantage of the DGHT approach in comparison to most other state-of-the-art localization techniques is the visual interpretability of the models, which reveals the shape of the most discriminative structures as well as the importance of each individual model point. In the localization models of zoom level 0 (Figure 7.5 (a) to (c)), for example, it can be seen that the localization heavily relies on both eyes and the mouth. The nose, is hardly represented by model points since it is a facial structure which is rarely visible in the feature images and, in addition to that, highly variable (Figure 7.6 (b)). Another interesting aspect, which can be seen in the model images, is that they represent different head positions at the same time to cope with the strong head pose variation contained in the PUT database. For demonstrating this aspect, Figure 7.6 shows (a) some original images with overlayed model, (b) the corresponding edge feature images, and (c) the model points which voted for the best localization hypotheses.

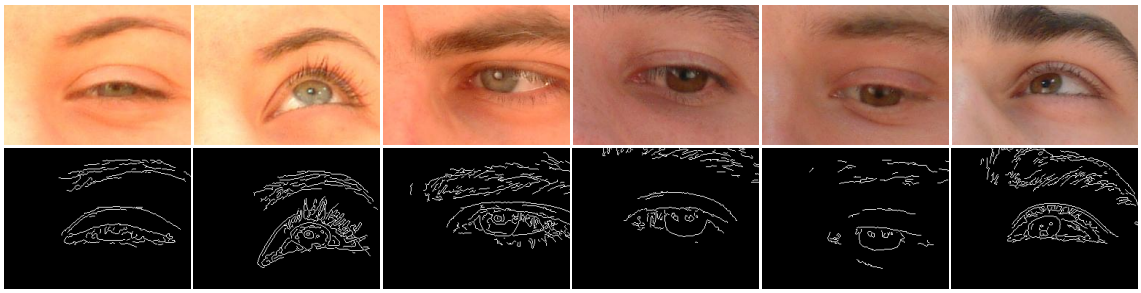


Figure 7.7: Examples of image extracts at zoom level 1 with corresponding feature images

Visual description of eyes

At zoom level 1 (Figure 7.5 (d) und (e)), the eye localization models clearly display two concentric circles, representing the iris and the pupil respectively. This search structure has also been integrated into many other systems by using expert knowledge [199, 44, 204, 143], which demonstrates that the DGHT may learn and incorporate this kind of knowledge fully automatically without the need for a detailed insight into the localization problem. Other model points, contained in the localization model, represent the eyebrows and eyelids, which have different positions depending on the viewing direction, and reflections of the flash on the eyeball (see Figure 7.7). This also contradicts the common modeling assumption that the sclera is always brighter than the iris, which in turn is brighter than the pupil.

Stacked-GHT model

When studying the model for the landmark combination (Figure 7.5 (f)), it is apparent that model points of the chin have a large scattering and very similar weights while the important points, representing the eye, are much more focussed. This is because of the lower reliability of the chin localizer, which has a mean error of 49 pixels in comparison to 21 and 23 pixels for left and right eye, respectively. It is also worth mentioning that the increased robustness of the landmark combination goes together with a loss in accuracy since the model is more blurred. The increase of the eye localization mean error to 29 and 31 pixels for the left and right eye at zoom level 0 after the landmark combination is compensated by the more precise edge based localization model applied at zoom level 1. At zoom level 1, the mean error was reduced to 12 and 10 pixel, respectively.

7.5 Stacked-GHT vs. CNNs

Stacked-GHT

In this chapter, I introduced a novel way, to use Hough-spaces as features for training and applying a DGHT model at a higher hierarchical level. This allows the stacking of multiple DGHT models and theoretically facilitates the use of different feature levels. For example, on the PUT Face database there are multiple landmarks annotated in the face, e.g. shape of the face, eyes, nose, mouth, eyebrows etc. (see Fig. 4 in [100]). Of course, many of these points are not directly facial landmarks but rather describe the contour of the face. It is not possible to precisely localize these points, but a DGHT model trained on these points will produce a Hough-space with high votes on the contour (see Figure 10.5 in [174]). Therefore, these high votes can be considered as feature points for a subsequent, higher level DGHT model. This can be repeated multiple times.

CNNs vs. Stacked-GHT

The basic idea behind CNNs is similar. Therefore, the question is, what is the difference between a stacked DGHT and CNNs? Generally, a DGHT, can be transformed into a convolution

layer. A convolution layer from a CNN takes all pixels in a moving window into account and multiplies them with the corresponding kernel value. The DGHT does the same with the difference that not all pixels are used but only pixels which seems to be more important. For better understanding lets have a look at the higher level model from this chapter, which uses a Hough-space as feature space. In this model there is no binning of the feature value as for edge features. Instead, the feature value is directly multiplied with the model point weight. Therefore, by contrast, a convolution layer in CNNs has a bias-value, but more importantly, the convolution layer multiples all pixel values with a trained weight value. In the DGHT this is theoretically also possible by setting the threshold value Γ for the feature space very low and ensuring that the model contains a model point at each position. However, this is not done in the DGHT for some good reasons. At first, there is a performance advantage of the DGHT since the number of multiplications is drastically lower than for CNNs. Furthermore, feature points with low values have also low information. This means that the lower level model does not fit well at this position, which means that the corresponding landmark is most probably not at this position. This lower level model hence provides very little information about this particular position. In this case, it is useful to not consider it as a feature point for performance reasons but also to avoid that the training process will focus on useless information. In this context, the DGHT takes advantage from developer knowledge that, without loss of generality, higher values are more important. By contrast, in CNNs each feature or pixel is treated as being equally important independent from its value. Considering the first convolution layer, which gets the raw image as input, this is correct. In this context, if a pixel value is low, medium, or high, the information entropy is usually the same.

Stacked-GHT vs. classical DGHT

In the classic DGHT, using equation (5.18) and (5.8), the feature value $\phi_{\mathbf{x}_l}$ is not taken into account. However, for different feature values, different model points might be selected. From a theoretical point of view, it is possible to have a specific model point $\mathbf{m}_j$ for each potential feature value and then λ_j can be set to a value correlating with $\phi_{\mathbf{x}_l}$. Therefore, equation (7.3) is a special case of (5.8). In equation (5.8), the model points are selected due the distance between the model point and the feature point value ($d(\phi_{\mathbf{x}_l}, \phi_{\mathbf{m}_j}) < \Delta\varphi$)). Compared to the convolution layer, this reduces the complexity since it shrinks the potentially relevant values without loosing much information. Albeit, it is possible that the DGHT has a model point for each possible feature value at each position, this is usually not the case, which gives the DGHT more options to reduce complexity. This has advantages in terms of computational costs, number of training images and might be more robust against adversarial attacks.

All DGHT-models are independent from each other

In the presented work, each DGHT model, at lower levels as well as at higher levels, was optimized independently. This means that also the lower-level model tries to achieve the best possible localization result. However, for the higher-level model, it is not so important that the lower-level models localize a point very precisely, it is more important that they consistently localize the same point or area. For example, a specific point on the face boundary is hard to localize but in this case, it is more helpful if the higher-level model can trust that the detected points will be on the face boundary. Of course, in the DGHT it is possible to change the loss function of the optimizer, as can be seen in [174], so that e.g. also lines or boundaries receive high votes in the Hough-space. The disadvantage is that this requires developer knowledge. At the same time, whereas on CNNs it is not clear what is really learned, a developer has more options to guide the Stacked-GHT or DGHT to learn as expected.

Manual labeling of landmarks required by Stacked-GHT

Last but not least, the Stacked-GHT framework requires the manual labeling of important landmarks and their manual arrangement. CNNs do this automatically. Although in the Stacked-

GHT framework it is possible to identify lowever level models which have less influence and to remove them, it does not have any possibility to find out that a new, not yet considered, landmark might help. At the same time, this reduces the risk that the training approach is misled by useless landmarks. In this case, again, the Stacked-GHT requires more developer knowledge, which means that CNNs are a more generic approach. Yet, if such developer knowledge exists, it might help to have the possibility to integrate it.

7.6 Conclusion

Improvements of results

This chapter presented two novel techniques for an improved eye localization in portrait images based on the Discriminative Generalized Hough Transform. By using a task-specific multi-level strategy and a novel facial landmark combination technique it was possible to increase the iris localization rate from 96.6 to 97.9%. This result is promising, since the variation of the head pose in the public PUT face database is quite large and the error measure considers the worst left and right eye localization attempt.

MLA

The general standard MLA, which gradually zooms into the target object by halving the search space in each level, turned out to be suboptimal. A more task-specific approach, adjusting the zooming strategy with respect to the relevant structures and confusable objects, may significantly improve the success rate. For the given task of eye localization, with two very confusable objects, a good strategy is an early limitation of the search space to a region, covering only a single eye.

Stacked-GHT

The novel approach for facial landmark detection, which has been introduced in this paper, could be combined with the modified MLA and further increases the robustness of the system in the first zoom level. With this framework, it could be shown for the first time that the DGHT is applicable for both, the individual localization of various landmarks and combined usage in a higher-level localization model. This comes with the possibility to visually interpret the DGHT models in the different stages unveiling discriminative structures and important model parts.

Large number of landmarks

Although in this contribution only three facial landmarks, both eyes and the chin, have been combined with the novel method, the approach may theoretically incorporate an unlimited number. Since the applied discriminative training procedure identifies and penalizes model points of weak landmarks, not supporting the localization, it is possible to select the most discriminative ones from a large set of candidates. A systematic evaluation of this idea, selecting optimal landmarks in an iterative training procedure might be a logical next step.

Stacked-GHT vs. CNNs

The Stacked-GHT is a generic approach for detecting low level landmarks and combining them to achieve more robust results. The importance of this can also be seen on subsequent work [127, 128]. The Stacked-GHT approach has also many things in common with CNNs and a comparison of both approaches reveals that CNNs still are more generic whereas the Stacked-GHT requires more developer knowledge. However, the more generic an approach is the more important the data is. The usage of developer knowledge can help to force the system to focus more on the important parts, which reduces the risk of overfitting or adversarial attacks. In the end, as always, which approach is better depends on the tasks at hand and on their requirements.

Chapter 8

Application on surveillance recordings

Introduction

In the previous chapters, the DGHT was analyzed on public databases to seek out constraints, weaknesses and to improve the DGHT. We saw that the DGHT has clear weaknesses when dealing with large target object variability in combination with non-static backgrounds - a situation which is the norm in surveillance videos. By introducing the SCM in Chapter 6, I have solved this issue resulting in clear improvements of the DGHT+SCM framework over the DGHT only system. In this chapter, I will analyze if the DGHT+SCM framework has the potential to be applied in real world surveillance situations.

8.1 Data

Data description

The data used in this section was recorded by Rosemann Software GmbH in a supermarket in the Netherlands. Similar to the Chokepoint Dataset the camera was mounted above the entrance so that people have to walk towards the camera and pass under it to enter the sales area. Therefore, there is a large variation in size between the subjects, resulting from approaching the camera, as visible in the Chokepoint Dataset. In addition, the camera is also recording a larger entrance area, where people could be standing around waiting for someone or moving sidewise in relation to the camera etc. The usage of shopping trollies or buggies is also possible, which in total results in a very high scene variability.

Featue image description

For evaluating the DGHT+SCM framework, I had access only to preprocessed images. The preprocessing consists of two parts. Initially, the images were downsampled up to 3 times reducing the resolution from 1280×800 to 160×100 pixels. On the original images as well as after each downsampling step, the Canny Edge Detector [17] was applied to generate the feature images. Hence for each image, I had access to four different feature images generated on different resolutions (see Table 8.1).

Annotation

The annotation of the images was done by Rosemann Software GmbH. Since they had access to the original, and not only preprocessed, images the annotation was done on the original images at the original resolution. The position of both eyes and a bounding box around the head and the body were manually labeled. In contrast to the other databases used, multiple target objects per image are possible.

Table 8.1: Overview of the Rosemann data

#Downsampling steps	Resolution	Average number of edge points	
		Rosemann Restricted	Rosemann Full
0	1280 × 800	-	-
1	640 × 400	17.58%	-
2	320 × 200	21.7%	21.68%
3	160 × 100	26.87%	26.50%

Rosemann Restricted Database

The data consist of two different datasets. The first one (the Rosemann Restricted Database) contains only images in which all persons are completely visible, meaning no body part is outside the image. This results in smaller variations in the size of people. This database contains 234 images with 49 persons as training corpus and 270 images with 59 different persons as validation corpus.

Rosemann Full Database

The other dataset (the Rosemann Full Database) does not have any restrictions and contains persons of all sizes and possible variations. In this set 724 images from 150 different persons are available for training and 735 images from 156 persons for validation.

Eye distance

In the Rosemann Restricted Database the eye distance varies between 14 and 32 pixels, with an average of 21.9 pixels. In the Rosemann Full Database the eye distance varies between 7 and 83.1 pixels, with an average of 28.5 pixels, which shows a much larger variation on the Rosemann Full Database than on the Rosemann Restricted Database.

Number of edge points

The number of edge points relative to the number of pixels increases with the number of down sampling steps from 17.5% to 26.9%. For comparison, on the FERET Face Database the relative amount of edge points was 13.1%, on the Chokepoint Dataset 14.1% of the image pixel. On the RWTH Hand Database, it was slightly higher with 22.7%. This shows that on the Rosemann data many structures are visible.

8.2 Setup

Validation criteria

Even if multiple subjects may have been visible in an image, I applied a localization procedure for one person only for better comparability with the approach applied throughout this work. In this case, the distance to the nearest annotated person was used as error distance. Of course, this depends to some degree on the number of persons per image. On 25 validation images from the Rosemann Restricted Database there are 2 persons and on the remaining 245 there is only one person visible per image. On the Rosemann Full Database it is more likely to find multiple persons per image, i.e. the maximum number of visible persons is six and on average there are 1.68 persons visible per image. Since it is possible that the right and the left eye from different persons are detected, a evaluation based on the worst results of both eyes as done for the FERET Face Database and the Chokepoint Dataset was not feasible and therefore only the left eye was localized. Furthermore, a localization was considered correct if it was located inside the head since this is sufficient for solving the anticipated task of blurring faces. On the Rosemann Restricted Database the average area covered by heads, which is the area in which a localization is considered as correct, is 8737.4 pixels, which is 0.85% of the complete image.

On the Rosemann Full Database, due to a larger number of multiple persons, the covered area increases to 23736 pixels, which is an area of 2.32% of the image size.

MLA

For this database, no full MLA procedure was applied. Instead, only the first zoom level was used since it is sufficiently accurate for the given face blurring task. Additionally, some of the trainings took very long and therefore, focusing on the first zoom level helped to reduce training time.

DGHT

The DGHT model was trained with the iterative DGHT training procedure (see Section 5.1.3). Due to training time and hardware restrictions, it was not possible to train a DGHT model without downsampling for both datasets. For the Rosemann Full Database a DGHT training was not possible even when downsampling once. For the latter, one iteration e.g. took more than one day and it was expected that up to 99 iterations would be required resulting in a training time of more than three months. Although the relative number of edge points per image increases per downsampling step (see Table 8.1), the absolute number decreases. Each edge point votes for multiple potential target point locations, and all of these votes need to be considered during training. Therefore the large number of edge points increases the training time. Thus training without downsampling cannot be completed in a feasible time.

SCM

The SCM model was trained as described in chapter 6. For the SCM two parameters need to be adjusted to the specific database: ξ_1 and ξ_2.

Setting ξ_1

As described in Section 6.6.2, ξ_1 mainly depends on the precision of the annotations. Given the eye distance, as aforementioned, the size of the pupil for downsampling one or two times is mostly less than one Hough-cell, which should allow for a precise enough annotation for $\xi_1 = 0$. However, Section 6.6.2 shows that $\xi_1 = 0$ is only useful for precise annotations. Pupil size varied between less than one and more than four pixels. Since I was unable to determine the precision of the annotation with certainty as the original images were unavailable, I set $\xi_1 = 1$. In general, $\xi_1 = 0$ may lead to worse results if the annotations are imprecise. However, $\xi_1 = 1$ leads only to slightly worse results for precise annotations compared to $\xi_1 = 0$ (see Section 6.6.2). Therefore, $\xi_1 = 1$ is a more general choice.

Setting ξ_2

The parameter ξ_2 mainly depends on the allowed error distance, which is calculated as the minimum distance between the position of the left eye and the bounding box of the head. For the Rosemann Restricted Database this was estimated at 23.2 pixels and for the Rosemann Full Database at 28.2 pixels. Therefore, for downsampling three times, ξ_2 was set to three Hough-cells. This means that a localization result is considered wrong for the SCM training if the error is more than 24 pixels. This also achieves a good discrimination between both classes. For downsampling two times, ξ_2 was set to 6 to keep the error tolerance of 24 pixels. For downsampling one time, ξ_2 was set to 8 in order to reduce the error tolerance slightly, but keeping a gap between classes big enough to prevent mixing up the classes.

8.3 Results

Results on the Rosemann Restricted Database

On the Rosemann Restricted Database the best localization rate was 99.2%, achieved by one time downsampling and using the SCM. In this setup, the DGHT baseline result was 84.4%, which means that the SCM could correct 94.9% of the errors committed by the DGHT. When

Table 8.2: Results on the Rosemann Restricted Database with different numbers of downsampling steps

#Downsampling steps	DGHT only	DGHT + SCM	improvement by the SCM
1	84.4%	99.2%	94.9%
2	90.0%	96.0%	60.0%
3	79.0%	86.5%	35.7%

Table 8.3: Results on the Rosemann Full Database with different numbers of downsampling steps

#Downsampling steps	DGHT only	DGHT + SCM	improvement by the SCM
2	70.0%	85.0%	50.0%
3	77.3%	82.0%	20.7%

downsampling twice, the DGHT baseline system obtained a better accuracy of 90.0%, whereas the combination of DGHT and SCM dropped down to 96.0% (see Table 8.2).

Results on the Rosemann Full Database

The best results on the Rosemann Full Database, obtained by downsampling twice, was 85.0% using the SCM and 70.0% using the DGHT only. Similar to the results on the Rosemann Restricted Database, the DGHT performed better for three times downsampling (77.3%) while the combination with the SCM led to a result of 82.0% (see Table 8.3).

8.4 Human Classifier

Error analyzing with respect to feature extraction

Using edge features only results in a loss of information, especially if the images are downsampled a few times. As we could previously see for the Rosemann Full Database with downsampling three times, with an error rate of 18.0% the results show room for improvement. It is unclear if the errors are (mostly) related to the poor feature extractor or to the localization algorithm. A typical way of analyzing it would be to manually look at the images with high error and decide if a human could detect the visible subjects. However, the "human" in question usually is the developer of the algorithm who might be biased by nature. Furthermore, only analyzing images with a high error does not take into account that the automatic classifier was able to correctly localize images on which a human might have failed.

Human performance study

To avoid these issues it is necessary to evaluate all images by a human, which should not be involved in the development of the algorithm. Therefore, I conducted a comparison study of human performance. A human labeler, who was not involved in any part of the development of the algorithm presented in this work, had the task to manually localize the eyes of each person in each image by only looking at the Canny Edge Image without having seen the original images before. To get a feeling for how the images and the persons would look like, the human labeler was presented the training images with annotated eyes during a training phase. It was followed by a test phase, during which the human labeler had to mark the position of the eyes on the test corpus.

Labeling procedure

The human labeler had to mark all images from the Rosemann Full Database with downsampling three times. Since a resolution of 160×100 is very small for the human perceptual system, the presented images were resized to 320×200. The resizing has only a visual effect

and does not have any influence on the information contained in the edge images. Furthermore, all images were presented in a random order to prevent the human labeler from deriving information from movements over time.

Results

The human labeler achieved an accuracy of 40%. Since the human labeler reported that the used annotation tool shifted the marked position by a few pixels, without exactly specifying how many pixels, I additionally analyzed the accuracy by shifting the x and y coordinates. Since, it was not possible to reproduce the reported issue and exactly calculate the extent of the shifting, all possible shifts between 10 pixels to the bottom right and 20 pixels to the top left side were tested and the best result was 80%. Using a machine learning algorithm this would be similar to optimizing parameters on validation data. Therefore, the only correct conclusion from these results is that a human labeler has a performance of at most 80%, which is not better than the accuracy of the DGHT in combination with the SCM.

8.5 Discussion

Tracking

On the Rosemann Restricted Database the DGHT in combination with the SCM achieves a success rate of 99.2% with one-time downsampling. One major issue with the Rosemann Full Database is the strong variation in the size of the subjects resulting from the fact that they are walking towards and underneath the camera. As we can also see on the Chokepoint Dataset there is almost always a situation where the whole person is visible in the image, which means these images would fulfill the requirements for the Rosemann Restricted Database. We expect a similar situation on the Rosemann data, where at some point almost each person will be completely visible, i.e. is part of the Rosemann Restricted Database. Due to the good localization results, it could be assumed that almost every person could be localized when entering the supermarket. This is at least a good starting point for further tracking helping to improve the localization on further frames, on which the person is only partly visible. In general, the integration of temporal information, such as tracking or optical flow, could be very helpful to increase the accuracy on the Rosemann Full Database.

SCM

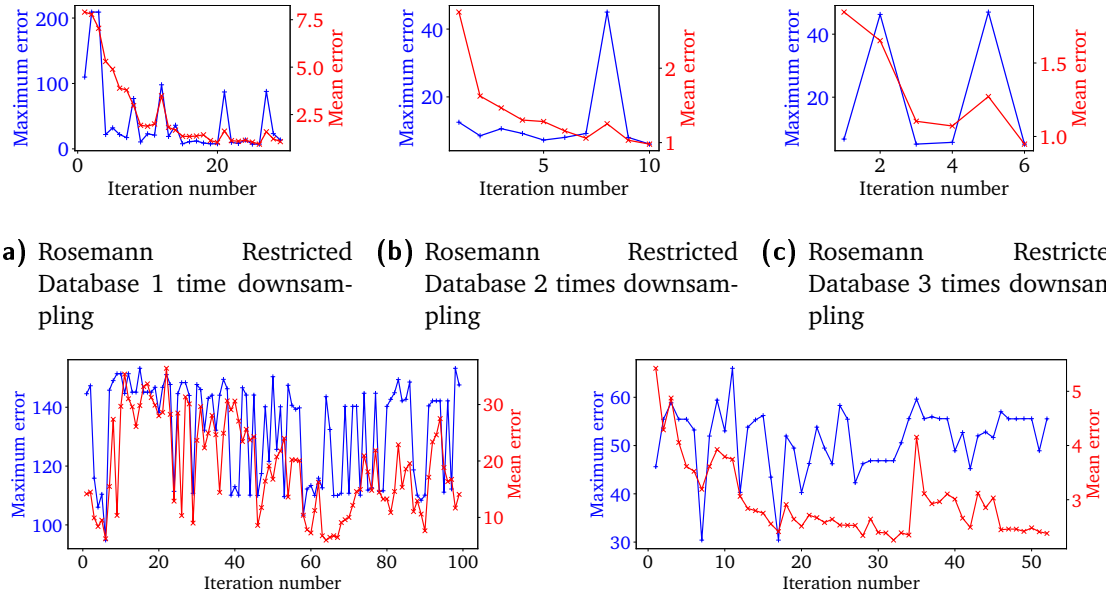
Depending on the dataset and the number of downsampling steps, the SCM corrected between 20.7% and 94.9% of the incorrect localizations performed by the DGHT. Since a smaller number of downsampling steps works better for the SCM, whereas for the DGHT it is better to use two or three downsampling steps, this shows that the quality of the SCM model is to some extent independent from the quality of the DGHT model. On the Rosemann Restricted Database, the results show that the SCM is capable of localizing almost every person.

Feature extraction

On the Rosemann Full Database, the improvements of the SCM are smaller than on all other datasets tested in this work. Taking into account that also the localization success from the human classifier (with an accuracy of at most 80%) is equally poor leads to the conclusion that the remaining errors are mostly due to insufficient feature extraction.

Number of iterations for the DGHT training procedure

In general, DGHT training is not well suited for such strong variations as in the datasets used here. The iterative training procedure (see Section 5.1.3) selects the images with the highest localization error from the previous iteration to extend the current DGHT model until the localization error for all training images is below a defined threshold or already integrated into the DGHT model. Technically, the training procedure is implemented to stop after 99 iterations



(a) Rosemann Restricted Database 1 time downsampling

(b) Rosemann Restricted Database 2 times downsampling

(c) Rosemann Restricted Database 3 times downsampling

(d) Rosemann Full Database 2 times downsampling

(e) Rosemann Full Database 3 times downsampling

Figure 8.1: These diagrams show the mean (red curve with red axis on the right side) and maximum (blue curve with blue axis on the left side) error on the training corpus per iteration during the training process.
For the Rosemann Restricted Database, the mean error tends to decrease per iteration. The maximum error only decreases with one-time downsampling demonstrating that for higher downsampling on the Rosemann Restricted Database there might be outliers which still could not be correctly localized after integration into the model.
For the Rosemann Full Database the maximum error does not decrease and the mean error only if the images were downsampled three times. For two times downsampling, the mean error as well as the maximum error varies between the iteration mostly in a random manner and shows that the iterative DGHT training procedure is not able to generalize a suitable DGHT model.

and on the Rosemann corpus sometimes all 99 iterations were required. From a theoretical point of view, there is no reason to stop after 99 iterations. However, normally fewer iterations are required (see Section 5.3 and Table 5.6). In some instances on the Chokepoint Dataset, 99 iterations are used. In this case, the mean error during training is reduced with an increasing number of iterations, albeit the improvements become increasingly marginal (see Section 5.3 and Figure 5.4). The idea behind the iterative training procedure is to include images with the highest localization error in the next iteration since these images have the greatest potential to improve the model. However, since on the Rosemann Full Database with downsampling twice neither the maximum error nor the mean error is reduced with a higher number of iterations (see Figure 8.1), the iterative training does not result in the expected improvements on this database and the DGHT training procedure was not able to abstract a general model. By contrast, on the Rosemann Restricted Database and the Rosemann Full Database with three times downsampling, the mean error is reduced over the iterations. The variability on the Rosemann Restricted Database is smaller than on the Rosemann Full Database and a higher downsampling factor further reduces variability. Since the iterative DGHT process is not capable of abstracting a sufficient model on the Rosemann Full Database with downsampling twice, it would not be able to do so with only downsampling once.

Training time

Additionally, the iterative DGHT training procedure may require a long processing time. On databases with small target object variability, such as the FERET Face Database or the RWTH Hand Database, the training time was at most a few days on a 64 bit system with Intel Xeon E5-1650 with 3.20 GHz and 32 GB Ram. On the Chokepoint Dataset the training time was a serious factor due to runtimes of multiple days or even weeks. On the Rosemann data the training with a high downsampling factor was feasible. However, with higher resolution images to avoid a large loss of information on the edge images, the training time was too long for practical applications (see above).

Number of edge points

Besides the strong target object variability, also the comparatively large number of edge points per image might be an issue for the DGHT. The high proportion of edge points on the Rosemann data is an indicator that many structures are visible. A high number of edge points alone does not make a task more difficult for the DGHT, as the RWTH Hand Database demonstrates. Nevertheless, to model the large target object variability, the DGHT model requires more model points to cover all target variations. In combination with a large number of edge points in the images, the probability that the model accidentally fits "better" on background structures than on a target object increases.

SCM and DGHT models

As aforementioned, we could see in the results that the quality of SCM is to some extent independent from the quality of the DGHT model. This is an outcome which is confirmed by the experiments performed with the SCM and a reduced number of training images (see Section 6.6.4). There we saw that using less training images to generate the DGHT model has less influence than reducing the number of training images for the SCM. To reduce the training time on the Rosemann Full Database a potential option would be to use less training images for the DGHT only since most of the training time is required for the DGHT. Another option is to generate a model only by overlaying training images without using the iterative training procedure, with or without weighting the model points by the DMC. The generated DGHT model also would have a poor performance, but the SCM could be capable to compensate for it.

Number of images

Nevertheless, another general issue on the Rosemann Full Database is the small number of available training images. DGHT and SCM both require comparably few training images (see Section 6.6.4). However, this is only true as long as all variations are covered by the available training images. Here only 724 images are available for training. Considering that these recordings are made in a supermarket without any constraints, it is very likely that not all variations, e.g. body shapes, children, babies, buggies etc., are included in the training corpus. The validation corpus may not even contain all of these situations. Therefore, in order to obtain a reliable system, a much larger training and validation corpus is required.

8.6 Conclusion

Error sources

On the Rosemann Restricted Database, where the whole person is visible in the image, the combined DGHT+SCM framework achieves very good results. However, without any constraints on person size or, in general, shape variation, the achieved results need to be improved for a practical application. The mediocre performance is most likely due to the insufficient features used. Even a human did not achieve better performance on localizing the persons.

Furthermore, also the number of training images is most likely too small to cover all possible variations, which is the second reasons for not achieving a better accuracy.

Improvements The most promising options to increase the performance would be better feature extraction and a larger number of training images. Additionally, the integration of temporal information, such as tracking the persons or using optical flow, could be very helpful and would to some extent also work on the edge images.

Detection It is also important to mention that the results, presented here, address the localization of one person per image only. This means that the Hough-cell with the highest value from Equation (6.5) is used and the error is calculated as being the distance to the nearest person. For a practical application, a detection approach is required for handling multiple persons as well as no person per image. This could be done e.g. with a threshold on the values from Equation (6.5). However, the aim of this work was to analyze the technical feasibility, which could also be done on the localization task. In the subsequent work [65] the DGHT+SCM framework is used for the detection of multiple pedestrians.

SCM Last but not least, this chapter demonstrates the importance of the SCM, which could, depending on the setup, correct up to 94.9% of the errors made by the DGHT. Additionally, we also saw that the quality of the SCM is independent of the quality of the DGHT model, which is an indication that the SCM will also work well in combination with other GHT-based detection or localization approaches.

Chapter 9

Hough-Forests

9.1 Introduction

Alternative approaches

There is a broad range of different detection and localization approaches. Especially, due to the success of CNNs, the possibilities for object detection have increased a lot. Testing each of these approaches is neither feasible nor necessary. It is clear that using current state-of-the-art technology will outperform the results of the DGHT+SCM. However, under specific circumstances DGHT+SCM might still have their applications. In this work, I will use Hough-Forests as an alternative approach to compare the results. I chose Hough-Forests since they are also a GHT-based approach and therefore show some similarity with the DGHT.

Hough-Forests

The Hough-Forests are a combination of a Random-Forest Classifier with the GHT-Voting-Procedure. Each image patch of a fixed size, i.e. 16×16 is mapped by a Random-Forest-Classifier to a prediction vector, i.e. the patch is mapped to the potential target landmark location. Subsequently, each image patch votes for the specific target point localization in a GHT-manner. See [68, 70] for a detailed description.

Hough-Forests vs. DGHT

Therefore, Hough-Forests are a GHT-based approach with a sophisticated feature extractor. Since the application envisaged in this work is restricted to the Canny Edge feature extraction, the Hough-Forests are not applicable for the purpose of this application. Nevertheless, we will use them on the public databases (see Chapter 4) since they show similarities with the DGHT and allow some inference on the influence of the feature extraction process. Additionally, the comparison also helps to choose an appropriate approach in cases where the available features are not restricted to Canny Edge features.

9.2 Setup

Setup

To allow a fair comparison, the setup of the Hough-Forests was chosen to be as similar as possible to the DGHT setup. Hence, the same training and validation corpora were used for the Hough-Forests and the DGHT. Besides images with the target object and a given target point localization, Hough-Forests can also use negative training samples without the target object allowing a better discrimination between the target object and the background. Since no negative samples are required for the DGHT or the SCM, no negative samples were used for the Hough-Forests, either. Using negative samples is quite useful for detection approaches,

where a discrimination between the target object and other objects or background is required. However, this work focuses on the localization of a specific target landmark within the target object. Technically, it is possible to select negative samples in this scenario, but their selection strongly influences the quality of the Hough-Forests. Therefore, I decided not to use any negative samples facilitating the analysis of the Hough-Forests under the same conditions as the DGHT+SCM.

Bounding-box around target landmark

Technically, Hough-Forests require a bounding-box around the target object for each image which may differ in size between images. In contrast, the DGHT only requires the target landmark and a fixed sized bounding-box is fitted around this target landmark. We use the same approach, i.e. a fixed sized bounding box around the landmark, for Hough-Forests to ensure a fair comparison. Therefore, annotation effort is the same for both approaches.

MLA

Last but not least, the Hough-Forests were used in the same MLA manner as the DGHT (see Section 5.1.4). In brief, at the beginning the image is down-sampled a few times, depending on the database in question. Subsequently, the resolution around the detected landmark is increased step-by-step until the original resolution is reached. The number of zoom levels depends on the database and is the same as for the DGHT experiments (see Section 5.3). To the best of my knowledge, this is the first time that Hough-Forests were used in combination with the MLA.

9.3 Results

Results

On the FERET Face Database the Hough-Forests achieve better results than DGHT+SCM (see Table 9.1). By contrast, on the RWTH Hand Database as well as on most portals of the Chokepoint Dataset the results from the DGHT+SCM outperform the results of the Hough-Forets (see Table 9.2 and Table 9.3).

9.4 Discussion

Results

In these experiments, sometimes the Hough-Forests outperform the DGHT in combination with the SCM, sometimes vice versa. [174] mentioned that Hough-Forests and DGHT achieve similar accuracies, but for smaller error tolerances the DGHT outperforms the Hough-Forests. One explanation was that due to the MLA the DGHT could achieve more precise localization results. However, the Hough-Forests in combination with the MLA were not tested in that publication. In this work, this was done, to the best of my knowledge, for the first time. Although the DGHT+SCM achieves much better results than the DGHT only, the Hough-Forests in combination with the MLA still outperforms the DGHT+SCM in some situations. Especially on the FERET Face Database, the Hough-Forests achieve good results for localizing the pupil and the mean pixel error is smaller for the Hough-Forests than for the DGHT+SCM, which highlights the effectiveness of the MLA.

Influence of MLA

To analyze which influence the MLA has, I also made experiments with the Hough-Forests without MLA. In order to avoid that conclusions depend on a wrongly chosen downsampling

Table 9.1: The localization accuracy (Ξ, see Equation 4.5) on the FERET Face Database for different error tolerances (0.05, 0.1, 0.25, and 0.5) and the mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels. If zoom level starts with "#DS" no MLA was used and the value gives the number of downsampling steps (DS).

	Zoom	eye					nose					mouth				
System	level	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$	$\Xi(0.05)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(0.5)$	$\varnothing(\epsilon_n)$
DGHT	1/3	29.9%	83.7%	99.6%	99.9%	7.4	47.2%	87.2%	99.5%	99.5%	8.0	46.9%	86.0%	98.8%	99.7%	8.5
+	2/3	65.1%	98.1%	99.6%	99.8%	4.9	68.4%	97.3%	99.7%	99.7%	5.7	69.7%	95.7%	99.0%	99.4%	6.2
SCM	3/3	76.9%	98.6%	99.6%	99.8%	4.3	72.1%	97.6%	99.6%	99.6%	5.3	79.2%	96.6%	99.1%	99.3%	5.4
Hough-	MLA: 1/3	10.7%	50.8%	95.6%	99.1%	12.7	19.8%	51.3%	93.8%	99.2%	15.5	21.6%	54.2%	92.2%	98.6%	15.8
Forests	MLA: 2/3	61.8%	95.6%	99.5%	100.0%	5.1	54.7%	89.8%	99.5%	100.0%	7.6	78.0%	96.3%	99.7%	100.0%	5.3
	MLA: 3/3	88.5%	99.3%	100.0%	100.0%	3.5	81.0%	98.5%	99.8%	100.0%	4.7	80.1%	96.9%	99.9%	100.0%	4.9
Hough-	#DS: 0	34.2%	71.3%	93.6%	98.9%	9.5	27.9%	60.9%	92.5%	99.1%	14.5	38.1%	72.2%	91.6%	97.6%	13.4
Forests	#DS: 1	35.4%	75.3%	95.0%	98.6%	9.2	33.1%	67.4%	94.9%	99.1%	13.0	41.8%	74.4%	93.7%	98.2%	12.2
	#DS: 2	10.1%	51.2%	95.9%	99.1%	12.7	20.0%	52.7%	94.0%	99.2%	15.3	21.1%	52.8%	93.2%	98.6%	15.9
	#DS: 3	4.7%	39.2%	95.6%	99.7%	13.9	14.8%	44.0%	92.9%	99.7%	16.7	15.1%	44.6%	90.8%	99.3%	17.1

Table 9.2: The localization accuracy ($\Xi(\frac{6}{256})$, see Equation 4.5) for the different landmarks (1D,2D, ... 15D) on the RWTH Hand Database, mean accuracy over all landmarks ($\varnothing(\Xi(\frac{6}{256}))$), and mean error in pixels ($\varnothing(\epsilon_n)$) for different zoom levels and different number of maximum zoom levels. If zoom level starts with "#DS" no MLA was used and the value gives the number of downsampling steps (DS).

	Zoom	$\Xi(\frac{6}{256})$													
System	level	1D	2D	3D	5D	6D	7D	9D	10D	11D	13D	14D	15D	$\varnothing(\Xi(\frac{6}{256}))$	$\varnothing(\epsilon_n)$
DGHT	1/5	86.0%	87.8%	92.2%	90.4%	92.2%	95.6%	93.4%	96.1%	97.7%	92.4%	93.6%	95.7%	92.8%	23.9
+	2/5	98.9%	98.8%	99.3%	99.5%	99.6%	99.5%	99.0%	99.7%	100.0%	99.0%	99.0%	99.9%	99.3%	11.9
SCM	3/5	99.6%	99.0%	99.7%	99.2%	99.7%	99.7%	99.0%	99.7%	100.0%	98.5%	99.0%	100.0%	99.4%	7.8
	4/5	99.6%	99.0%	99.6%	99.2%	99.8%	99.7%	99.0%	99.7%	100.0%	98.5%	98.9%	100.0%	99.4%	6.5
	5/5	99.6%	99.0%	99.6%	99.2%	99.8%	99.7%	99.0%	99.7%	100.0%	98.4%	98.9%	100.0%	99.4%	6.5
Hough-	1/5	37.6%	46.1%	70.4%	45.1%	54.1%	77.4%	44.9%	55.8%	78.4%	43.0%	53.4%	75.2%	56.8%	53.6
Forests	2/5	95.1%	95.6%	98.8%	92.2%	93.0%	89.8%	97.8%	96.4%	96.6%	88.3%	91.7%	98.8%	94.5%	20.9
	3/5	94.7%	90.8%	99.3%	97.1%	55.1%	89.8%	96.4%	61.9%	97.3%	89.8%	41.3%	99.8%	84.4%	23.9
	4/5	95.6%	92.5%	99.3%	97.6%	93.7%	96.6%	98.1%	98.3%	99.5%	94.2%	72.6%	99.8%	94.8%	13.5
	5/5	95.6%	93.7%	99.5%	98.1%	95.1%	96.8%	98.1%	98.3%	99.5%	95.4%	76.0%	99.8%	95.5%	13.2
Hough-	#DS: 2	51.7%	65.3%	80.8%	78.4%	89.6%	93.4%	83.7%	91.5%	96.1%	70.1%	83.0%	92.5%	81.4%	35.0
Forests	#DS: 3	62.4%	71.4%	76.7%	82.8%	84.5%	91.0%	75.7%	88.3%	92.5%	68.7%	77.7%	87.1%	79.9%	38.7

factor, I analyzed different downsampling steps. Table 9.1, Table 9.2 and Table 9.3 show that the performance of the Hough-Forests drastically reduces without the MLA. This demonstrates the high performance of the MLA independent of the localization approach. As [171, 174] shows the good performance of the MLA in combination with the DGHT, this work additional shows it for the MLA in combination with the Hough-Forests. This allows us to assume that the MLA is in general a useful approach for object localization. To the best of my knowledge, the MLA has previously only been tested in combination with the DGHT but not with other approaches. Sometimes similar approaches have been used, e.g. in a first step the face was detected and then inside the face the facial landmarks. However, the MLA is more generic and a main strength is the increase of the resolution per step allowing to go from global but coarse to local but fine features.

Feature learning

In some way, Hough-Forests are a very simple object localization approach. The only sophisticated part is the feature extraction. Yet, the performance of the Hough-Forests is sometimes better than the DGHT+SCM. Considering that the Hough-Forests suffer from the same issue of independent voting of model points, which was for the DGHT solved by the SCM, the results are very good. This shows the importance of a good feature extraction and that automatically selected features, as used in the Hough-Forests, are better than manually hand-crafted and fine-tuned features, such as Canny Edge features used in this work. In this context, it is worth mentioning that Hough-Forests have been developed in 2009 [68]. This was long before the success of the CNNs and, whereas other detection approaches like [42, 41, 10, 134, 219, 11, 40, 220], tried to manually find good features, the Hough-Forests already used a machine learning approach avoiding the bottleneck of hand-crafted feature extraction methods. For the application in this work, we do not have the original images but only preprocessed images produced by the Canny Edge Detector. Therefore, it is not possible to apply Hough-Forests on this task. Nevertheless, the experiments show that without this restriction much better results might be achievable.

Combination of Hough-Forests and DGHT

The feature extraction process of the DGHT is generally not restricted to the Canny Edge Detector. Therefore, it is theoretically possible to integrate the feature extraction process from the Hough-Forests into the DGHT+SCM framework. This would be a very interesting approach since it would combine the strengths of both approaches. In fact, Hough-Forests suffer from the same issue of independent voting of model points, which is solved by the SCM. Therefore, a combination would be the logical next step.

Setup optimization

To allow a fair comparison, the setup of the Hough-Forests is as similar as possible to the DGHT. Yet, to some extent, this remains an unfair comparison since the setup used was optimized for the DGHT. Either way, the DGHT and the SCM work very well with this setup, which does not necessarily mean that this setup is also optimal for the Hough-Forests. For example, it was not analyzed if a different number of zoom-levels works better for the Hough-Forests, but for the DGHT it was analyzed. Furthermore, we have greater expertise in the DGHT and the SCM than we have for the Hough-Forests. Therefore, with better optimization, the results might have been even better for the Hough-Forests.

Table 9.3: The localization accuracy (Ξ, see Equation 4.5) for different error tolerances and mean localization error in pixels ($\varnothing(\epsilon_n)$) for the four portals in the Chokepoint Dataset and each zoom level. If zoom level starts with "#DS" no MLA was used and the value gives the number of downsampling steps (DS).

System	Zoom	P1E				P1L				P2E				P2L			
	level	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(1)$	$\varnothing(\epsilon_n)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(1)$	$\varnothing(\epsilon_n)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(1)$	$\varnothing(\epsilon_n)$	$\Xi(0.1)$	$\Xi(0.25)$	$\Xi(1)$	$\varnothing(\epsilon_n)$
DGHT	1/4	2.2%	29.2%	96.0%	12.5	1.4%	16.0%	88.9%	13.6	1.1%	17.5%	91.5%	12.6	1.1%	11.5%	85.7%	15.5
+	2/4	10.8%	69.5%	98.3%	7.6	6.7%	48.7%	98.4%	7.2	6.9%	49.0%	98.0%	7.7	6.9%	44.6%	97.5%	8.3
SCM	3/4	43.4%	90.1%	99.1%	4.2	29.7%	85.0%	99.2%	4.3	21.3%	82.5%	98.4%	5.3	25.9%	83.2%	98.4%	5.4
	4/4	79.4%	97.6%	98.7%	2.6	67.2%	97.8%	99.3%	2.8	59.9%	96.5%	98.2%	3.8	63.5%	97.7%	98.4%	4.0
Hough-	1/4	7.9%	44.5%	98.6%	10.5	3.3%	25.3%	83.5%	19.4	4.2%	27.5%	83.7%	18.8	3.0%	27.2%	87.6%	14.1
Forests	2/4	9.4%	62.4%	99.8%	7.3	6.9%	41.5%	94.1%	14.1	8.1%	56.2%	97.2%	11.9	5.4%	51.7%	97.7%	9.7
	3/4	50.6%	91.5%	99.8%	4.0	33.4%	79.8%	94.6%	10.8	22.8%	79.1%	97.6%	9.9	25.2%	85.3%	97.9%	6.7
	4/4	83.1%	97.7%	99.7%	2.7	62.7%	93.0%	94.7%	9.4	57.9%	94.9%	97.8%	8.2	57.6%	95.3%	98.0 %	5.2
Hough-	1/4	7.9%	43.9%	97.9%	10.8	2.8%	22.9%	79.3%	22.3	4.4%	28.6%	83.7%	21.3	2.1%	22.7%	85.0%	17.9
Forests	2/4	10.0%	62.7%	99.6%	7.5	6.2%	42.3%	93.1%	15.7	7.7%	56.7%	96.3%	14.6	6.2%	52.5%	96.0%	12.9
1000	3/4	51.9%	90.5%	99.6%	4.1	33.8%	80.0%	93.8%	12.3	23.8%	81.5%	96.7%	12.7	23.9%	82.6%	96.1%	10.2
images	4/4	83.4%	97.5%	99.4%	2.7	62.6%	92.8%	93.8%	11.0	57.0%	93.8%	96.8%	11.3	55.5%	92.3%	95.9%	8.7
Hough-	#DS: 0	19.4%	48.8%	61.5%	54.4	0.7%	5.7%	23.7%	69.2	1.3%	13.4%	45.9%	76.0	1.9%	12.1%	38.5%	60.9
Forests	#DS: 1	30.7%	70.6%	91.8%	16.6	0.8%	5.7%	33.6%	64.0	5.7%	29.2%	67.0%	48.4	2.2%	18.7%	52.4%	55.9
1000	#DS: 2	13.0%	59.6%	95.5%	12.8	0.8%	17.2%	69.3%	41.0	6.8%	38.5%	85.8%	25.1	1.5%	14.3%	67.2%	43.9
images	#DS: 3	9.0%	43.4%	98.2%	10.6	3.1%	19.7%	78.8%	24.0	0.0%	0.4%	7.2%	79.2	1.6%	20.8%	84.0%	18.3

Chapter 10

Comparison with Convolutional Neural Networks

CNN introduction

The major breakthrough of CNNs occurred during the course of this work [109]. Today, CNNs are powerful computer vision tools achieving outstanding results for almost every task of object detection or localization, segmentation etc. One of the main reasons for the power of CNNs is their universality. Although the images are gradually processed from the pixel level to low-level features to higher level features, neither different layers nor image regions are considered independent from each other. Instead, all parameters are jointly optimized, which is the major difference to other computer vision approaches. Albeit the idea was not new, a lack of training images and computational resources prevented outstanding results, which might be the reasons that feature generation and the subsequent tasks such as detection or segmentation, were optimized independently.

Feature learning

This feature learning is an enormous strength of CNNs but at the same time also its biggest weakness. Considering a small image with a size of 200×100 pixels and only 8 bit gray values, i.e. 256 potential gray values, per pixel, there are still $256^{100*200}$ different input possibilities. For comparison, the ImageNet database contains 14 million images[37], which is less than 256^3. Through processing the images gradually, weight sharing in convolution layers, and other optimization, the complexity is reduced. However, depending on the architecture, the number of learnable parameters usually ranges between 5 and 155 million. To jointly optimize all these parameters, a large number of training images and computational power is required.

Number of training images

The high number of training images required is one of the most common issues with CNNs. Hence, there have been some attempts to reduce it, mainly using two different approaches: Learning from experience (pretrained tasks) and the use of artificial data. For the former, the most common approach is the pretraining and subsequent refinement of CNNs [105, 141, 210, 25]. More generally, a variety of approaches directly incorporates something that can be described as experience, i.e. a large number of different tasks or databases are used to train a general CNN, which can be adapted to a new task with less training images [157, 45, 179, 22, 193]. This can be achieved, e.g. by comparing the features of a test image with training images [107, 87, 205, 187, 165, 145], having optimal initialization [51, 142, 94], or optimization steps [159, 4, 8]. This is sometimes also referred to as Meta-Learning [177, 138] or distance learning [191, 136]. Although for the specific task only a few training images are required, a large database for pretraining is nevertheless necessary. To create artificial data, a commonly used

approach is data augmentation [82, 209, 33, 227]. In 2015, Goodfellow introduced Generative Adversarial Nets (GANs) [77] for generating artificial data by competition networks, which is receiving increased attention [99, 132, 2, 213, 125, 21, 152]. However, this does not reduce computational power and its ability to create unforeseen situations is limited as a matter of principle.

Adversarial attacks

Aside from the number of required training images and the computational complexity, a pronounced drawback of CNNs is adversarial attacks [158, 27]. In some way, adversarial attacks are optical illusions for the CNNs, e.g. adding specific noise to an image [78], which is not recognizable by humans, or placing a sticker with a specific pattern [16] misleads the CNN into producing wrong predictions. Researchers successfully placed a specific sticker on T-Shirts which prevented the owner from being detected by a CNN [197]. To make matters even worse, these attacks are largely independent of the specific CNN model. This means that the same noise, sticker, or T-Shirt might not only work for the specific network under attack but also for other CNNs models or architectures. For surveillance applications, this is a big issue. If a thief or housebreaker only needs to wear such a T-Shirt or sticker to escape detection, the application is not very reliable.

Adversarial attacks countermeasures

As of today, there is no good solution to prevent such attacks, albeit there is some work in this direction [218, 137]. One idea is to simulate attacks during training and therefore make the CNN robust against a specific attack; adding a specific noise pattern which will mislead the CNN during training will make the CNN robust against this pattern. However, the attack still works. It just requires a new pattern. As mentioned, the number of possible input values is very large, and therefore even with GANs and simulated data, only a small part of the complete input space can be covered by training images always leaving options for adversarial attacks. In this context, adversarial attacks are nothing else than unknown images which shows that there is a lack of understanding how CNNs work internally.

DGHT+SCM and adversarial attacks

With enough effort and malicious intent, any computer vision algorithm is in some way vulnerable and therefore also the DGHT+SCM framework most likely is. However, the design in general is more robust against attacks than CNNs. Especially when using edge features but also for some other features which can be used in combination with the DGHT+SCM, the noise needs to be very strong in order to have a clear influence on the system outcome. The second line of defence is the GHT model itself. The input space is much smaller than for CNNs and therefore, it is more difficult to construct images, not seen in the training. This becomes more pronounced with stacked GHT or the SCM. Last but not least, through the combinations of feature extraction, (multiple) DGHT and SCM, the whole framework is highly non-linear; an attack on the DGHT might not work on the SCM and vice versa.

Scientific focus

At the moment, the scientific community is focusing on CNNs and deep learning. However, good results with alternative approaches do not become worse over time. There are tasks, which are too complex to be solved without CNNs, GANs and simulated data. However, not every task is that complex. In this work, an alternative approach has been presented, which is not as generic as CNNs, but still achieves good results for the given tasks avoiding some of the drawbacks of CNNs.

CNN initialization using DGHT+SCM

Another major challenge of CNNs is their initialization. The most general approach is random initialization, but due to the steepest-descent approach, a bad initialization can result in a local minimum leading to bad results. With enough training data, even a local minimum may lead

to good results, but finding such a minimum requires a long training time. Therefore, transfer learning presents a good approach in which the model is initialized with a model already trained on a different database [212, 89]. However, such model initialization can also be done with a different approach. The outcome of any GHT-based approach is the same as if the GHT-model was used as a spatial filter, which is exactly the same as a convolution layer in a CNN. A Random-Forest classifier can also be transferred to a CNN-based approach [211, 92, 168]. Hence, it should be possible to transfer the complete system consisting of DGHT and SCM, as presented in this work, into a CNN-based approach. This has the advantage that with a very small number of training images, a DGHT+SCM system could be trained and transformed into a CNN. This CNN would have a good initialization and could then be further optimized. This is helpful since the disadvantage of the DGHT+SCM is that each step is optimized independently from the subsequent steps. This means that the DGHT model is optimized to achieve good performance without the SCM but as we could see in Section 8.5 sometimes a worse DGHT model achieves better results in combination with the SCM than a better DGHT model in combination with the SCM. With the transformation to a CNN, all steps could be optimized together and additionally a better feature extraction could be integrated. Also the combination of multiple DGHT approaches (see Chapter 7) and the Hough-Forests could be transformed into a CNN.

DGHT+SCM for CNN proposal generation

Furthermore, the DGHT+SCM framework can be used for hypotheses selection, which will be classified by a CNN in a latter step as already used in subsequent work [67, 63, 60].

Point cloud

Last but not least, from a theoretical point of view, CNNs and GHT-based approaches work on different kinds of data. CNNs are optimized for image processing whereas GHT-based approaches require point cloud data. Point cloud data are a collection of data points consisting of X, Y and potentially Z coordinates in space and an optional point value. In contrast to images, point cloud data are sparse and unordered. Convolutions, which are a main part of CNNs, are defined for dense data and therefore not directly applicable for sparse point cloud data. By contrast, the GHT is designed for such sparse data and therefore a preprocessing step, transforming an image into a point cloud, is required. For example, in [6], as well as in this work, this is done by an edge detector, but, of course, this is a crucial step having a large influence on the success of the system. However, there are also data, which are by nature point clouds, such as the outcome from Lidar scanner. In this case, the presented DGHT+SCM framework is very straightforward to apply. Albeit there are some solutions for handling point cloud data with Neural Networks [155, 156, 39, 184, 183], these approaches are more complex and the handling and preprocessing of the point clouds is crucial. Applying the KISS principle, "keep it simple, stupid", the DGHT+SCM framework is an alternative for point cloud data.

Conclusion

In summary, CNNs and Deep Learning in general has proven its success recently. Especially because of this success, there is the risk that research is focused only in this area. However, the restriction to a specific kind of solution might prevent the discovery of other, more straightforward solutions. Therefore, the development and evaluation of other approaches should remain a valid research.

Chapter 11 Scientific contribution

11.1 Goal specific contributions

Transfer of DGHT from medical images to photographic images

The first goal of this thesis was to transfer the DGHT from medical image processing to the processing of normal photographic images or video frames. This task was addressed in Chapter 5.1. The performance of the DGHT for such images, shown on the FERET Face Database, was comparable to its performance on medical images, shown on the RWTH Hand Database. Therefore, this work shows that the DGHT is not restricted to medical images but can also be applied to non-medical images. The performance of the DGHT depends more on the difficulty of the task, mainly described by the target object variability and the variability of the background, than on the image acquisition method.

Usability of DGHT for surveillance applications

The second goal was to evaluate the usability of the DGHT for surveillance applications. Chapter 5.1 shows that for a large target object variability and dynamic background, the performance of the DGHT is not good enough for practical surveillance applications. Therefore, two improvements were developed during this work: the SCM to handle the issue of independent voting of model points and the combination of landmarks to reduce the target object variability. This has provided the foundation for further research and developments [65, 126, 67, 63, 60].

DGHT+SCM for surveillance application

In Chapter 8 the final evaluation of the DGHT and the SCM for surveillance applications was conducted showing that the DGHT+SCM could be successfully applied to these applications. This chapter also showed the limitations of the usability of the DGHT. Restricting the target object variability to a certain limit, e.g. by using only completely visible persons, an acceptable performance was achieved. However, without this restriction, additional work is required for suitable results.

Canny Edge Feature restriction

Still, the restriction to Canny Edge features remains the main limitation, which was a given requirement in this work. Complying with this requirement, in all experiments with the DGHT, only edge features were used. Even a human could not reach a better performance than the DGHT+SCM when only the Canny Edge features were provided. Additionally, the good performance of the Hough-Forests shows that better performance may be expected when using a more sophisticated feature extraction technique.

General object localization approach

The DGHT is a general object localization approach, as demonstrated by its good performance on the variety of different tasks in [169] and in Chapter 5.1. An additional goal of this work was to keep the developed extensions general and thereby easily transferable to new tasks.

The SCM was applied to the different tasks of non-medical and medical image processing (see Chapter 6 and [66, 62]). Since the Stacked-GHT (Chapter 7) repeatedly uses standard DGHT technology at different hierarchical levels (e.g. edges, eyes/nose/mouth, face, person), it is not restricted to specific applications.

Number of training images

In addition to the goals, one requirement was that only a small number of training images would be needed and the annotation effort would be minimal. This requirement was fulfilled since the available number of training images was limited. Only on the Chokepoint Dataset more than 1000 images were used for training. Additionally, also experiments with very small numbers of training images were performed, beginning with as few as 10. Of course, accuracy increases with the number of training images, but, if some inaccuracy is acceptable, a trade-off against accuracy could reduce the number of necessary training image drastically. Furthermore, the DGHT only requires the annotation of the target landmark in the training corpus and the addition of the SCM does not increase the annotation effort.

11.2 General contributions

Besides the goal specific contributions, there are three general contributions, which will be explained here.

11.2.1 SCM

Independent voting of model points

One main drawback of voting-based methods is the independent voting of model points. This results in a high false positive rate, which may, according to Razavi *et al.* [161], be the main reason for the lack of attention for voting-based methods in the scientific community. During this work, I proposed a method, called Shape Consistency Measure (SCM). The SCM analyzes the whole voting pattern for a localization hypothesis in question with a Random-Forest classifier. Since the whole voting pattern of all model points is jointly considered, the individual votings are no longer independent from each other. Therefore, votes from mutually exclusive model points, e.g. different shape variants, are recognizable, which is a clear indicator for false positive localization hypotheses.

SCM and other solutions

The SCM strongly reduces the error rate by up to 94.9% over all tested databases demonstrating the potential of the information contained within the voting pattern. The idea of analyzing the voting pattern has been implemented by several groups [36, 161, 13] during the same time as this work, which demonstrates the importance of this approach. However, compared to these approaches, the SCM is very straightforward and directly addresses the source of the problem. Other approaches address the problem in an indirect manner, e.g. by comparing the voting pattern with training images [36], the usage of heuristic features [13], or by grouping the data [169, 161]. By contrast and as a result of addressing the underlying problem directly, the SCM is also very universal. Not only does it address the task of discriminating between correct and incorrect localization hypotheses but rather transforms the voting pattern into a feature space. This can be used for any classification or regression task, e.g. to estimate the ROI around an object as in [64] or a classification in more than two classes.

Influence of neighborhood cells

Another general drawback of GHT-based methods is that a model-feature-point combination only votes for a specific Hough-cell without any influence on and from neighborhood cells, which makes GHT-based methods vulnerable to small variations. Therefore, the DGHT uses a binning factor which slightly reduces this risk but also makes the localization less precise. Gall *et al.* [68] suggest using a Gaussian filter for smoothing the Hough-space so that indirectly the votes also count for neighborhood cells. The SCM solves this problem by not only analyzing the votes for the Hough-cell in question but also the votes for nearby Hough-cells. The experiments show that this is a key element of the system. In contrast to other workarounds, the SCM directly extends the feature space by the information which model points vote in which distance to the cell in question. Therefore, the classifier can learn the spatial relationships of the model points. Again, the SCM uses a more universal and direct approach than previous solutions.

SCM as GHT extension

Last but not least, the SCM is a general extension of GHT-based approaches and not restricted to the DGHT. As shown in Section 6.6.4 and 8.5, the performance of the SCM is to some extent independent from the performance of the DGHT model. Furthermore, the difference between DGHT and GHT models is the individual weighting of model points. Since these model point weights are not considered by the SCM, it can be used in combination with other GHT-based approaches including Hough-Forests.

11.2.2 Stacked-GHT

Stacked-GHT

The combination of landmarks to improve detection or localization results is well-known and the most known approaches using this idea are DPM [49]. For the first time, I introduced a landmark combination approach into the DGHT, in a way that the output of one or multiple DGHTs is used as an input for a new DGHT. The quality of landmark combination has been demonstrated for several other approaches [49, 68], but has not been used in combination with the DGHT before.

11.2.3 Hough-Forests

Hough-Forests

The Hough-Forests as a localization approach were tested in combination with the MLA approach for the first time. In this combination, the Hough-Forests achieve very good results, better than the DGHT+SCM. Since without the MLA [173], the Hough-Forest achieve similar results as the DGHT (without SCM), the MLA has a strong influence not only on the DGHT, but also on the Hough-Forests. The experiments with the Hough-Forests and with the DGHT+SCM show the potential of GHT-based approaches.

Chapter 12
Conclusion & Outlook

Main achievements of this work

In this work, the potential of GHT-based approaches, mainly based on the DGHT but also on Hough-Forests, was explored. A main drawback of GHT-based approaches, the independent voting of model points, could be solved with a universal approach. Furthermore, this work shows a possible method for stacking multiple GHT-based approaches into a deeper model. Analyzing the potential of the DGHT for video surveillance application reveals that the restriction to Canny Edge features prevents a good system performance.

Potentially applications

Although CNNs have demonstrated powerful results, they suffer from some disadvantages such as adversarial attacks making research for other methods still useful. For medium difficult tasks, the DGHT+SCM framework can achieve good results with a small number of training images. There are many tasks, which do not require a high accuracy. For example, in scientific research, it is often necessary to analyze large amounts of images, but it may be sufficient and even advantageous to obtain preliminary data quickly and without major investment into annotation. With the method presented here only a small number of images needs to be manually annotated and a large number of images can be analyzed to test whether a deeper and more precise analysis is necessary. Also for semi-automatic labeling, GHT-based approaches can generate a good first model as a starting point. Last but not least, GHT-based approaches are optimized for point cloud data by nature. Therefore, an interesting next step would the evaluation of the presented framework with pure point cloud data, such as Lidar data. This could be a very exciting task, which can be directly handled by the DGHT+SCM framework.

Combinations with CNNs

Furthermore, it is possible to combine the DGHT and the SCM with CNNs in different ways. The success of using the DGHT+SCM as a proposal generator for CNNs has already been shown [67], but there are further possible combinations such as using the DGHT+SCM model to initialise a CNN training. To analyse such potential combinations would be a logical next step.

SCM next steps

Although using a Random-Forest classifier in the SCM shows good results, the SCM is not restricted to it. Other classification approaches, such as Gradient Boosting Decision Tree [59, 104], would also be possible and to evaluate them in more detail could be worthwhile. Additionally, in this work, the SCM has only been evaluated for refining the localization results. However, the SCM can also be used for other classification tasks, such as gender classification or bone age estimation.

Bibliography

[1] Alberto S. Aguado, Eugenia Montiel, and Mark S. Nixon. Invariant Characterisation of the Hough Transform for Pose Estimation of Arbitrary Shapes. *Pattern Recognition*, 35(5):1083–1097, 2002.

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative Adversarial Networks for Extreme Learned Image Compression. In *International Conference on Computer Vision (ICCV)*, 2019.

[3] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[4] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to Learn by Gradient Descent by Gradient Descent. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[5] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental Face Alignment in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] Dana H Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[7] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-Outside Net: Detecting Objects in Context With Skip Pooling and Recurrent Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V. Le. Neural optimizer search with reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

[9] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching Shapes. In *International Conference on Computer Vision (ICCV)*, 2001.

[10] Rodrigo Benenson, Markus Mathias, Tinne Tuytelaars, and Luc Van Gool. Seeking the Strongest Rigid Detector. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[11] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten Years of Pedestrian Detection, What Have We Learned? *arXiv:1411.4304*, 2014.

[12] Peter Beyerlein. Discriminative Model Combination. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[13] Thomas Blaffert, Cristian Lorenz, Hannes Nickisch, Jochen Peters, and Jürgen Weese. SVM-based Failure Detection of GHT Localizations. In *SPIE Medical Imaging Conference*, 2016.

[14] Leo Breiman. Random Forests. *Machine learning*, 45(1):5–32, 2001.

[15] Leo Breiman, Jreome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Chapman, 1984.

[16] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial Patch. *arXiv:1712.09665*, 2017.

[17] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[18] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face Alignment by Explicit Shape Regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[19] Hum Yan Chai, Lai Khin Wee, Tan Tian Swee, Sh-Hussain Salleh, and Lim Yee Chea. An Artifacts Removal Post-processing for Epiphyseal Region-of-Interest (ERoI) Localization in Automated Bone Age Assessment (BAA). *BioMedical Engineering OnLine*, 10:87, 2011.

[20] Kai Chen, Yuhang Cao, Chen Change Loy, Dahua Lin, and Christoph Feichtenhofer. Feature Pyramid Grids. *arXiv:2004.03580*, 2020.

[21] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, Wei Fan, and Xiaohui Xie. DGGAN: Depth-image Guided Generative Adversarial Networks forDisentangling RGB and Depth Images in 3D Hand Pose Estimation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[22] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A Closer Look at Few-shot Classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[23] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d Object Detection for Autonomous Driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[25] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best Practices for Fine-Tuning Visual Classifiers to New Domains. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[27] Christian Cosgrove and Alan Yuille. Adversarial Examples for Edge Detection: They Exist, and they Transfer. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[28] Antonio Criminisi and Jamie Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Science & Business Media, 2013.

[29] Antonio Criminisi, Jamie Shotton, and Stefano Bucciarelli. Decision Forests with Long-range Spatial Context for Organ Localization in CT Volumes. In *MICCAI Workshop on Probabilistic Models for Medical Image Analysis*, 2009.

[30] Antonio Criminisi, Jamie Shotton, Duncan Robertson, and Ender Konukoglu. Regression Forests for Efficient Anatomy Detection and Localization in CT Studies. In *Menze B., Langs G., Tu Z., Criminisi A. (eds) Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging. (MCV)*, number 6533, pages 106–117.

[31] David Cristinacce and Timothy F Cootes. Facial Feature Detection using Adaboost with Shape Constraints. In *British Machine Vision Conference (BMVC)*, 2003.

[32] David Cristinacce, Timothy F Cootes, and Ian M Scott. A Multi-stage Approach to Facial Feature Detection. In *British Machine Vision Conference (BMVC)*, 2004.

[33] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data. *arXiv:1805.09501v3*, 2019.

[34] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[35] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[36] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time Facial Feature Detection using Conditional Regression Forests. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[38] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Discriminative Training for Object Recognition using Image Patches. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[39] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet: A deep learning label fusion method for multi-atlas segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019.

[40] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[41] Piotr Dollár, Serge J. Belongie, and Pietro Perona. The Fastest Pedestrian Detector in the West. In *British Machine Vision Conference (BMVC)*, 2010.

[42] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral Channel Features. In *British Machine Vision Conference (BMVC)*, 2009.

[43] René Donner, Bjoern H. Menze, Horst Bischof, and Georg Langs. Global Localization of 3D Anatomical Structures by Pre-filtered Hough Forests and Discrete Optimization. *Medical Image Analysis*, 17(8):1304–1314, 2013.

[44] Tiziana D'Orazio, Marco Leo, Grazia Cicirelli, and Arcangelo Distante. An Algorithm for Real Time Eye Detection in Face Images. In *International Conference on Pattern Recognition (ICPR)*, 2004.

[45] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-Shot Learning With Large-Scale Diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable Object Detection using Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[47] Gabriele Fanelli, Angela Yao, Pierre-Luc Noel, Juergen Gall, and Luc Van Gool. Hough Forest-based Facial Expression Recognition from Video Sequences. In *European Conference on Computer Vision (ECCV)*, 2010.

[48] Jacob Feldman. What is a Visual Object? *Trends in Cognitive Sciences*, 7(6):252–256, 2003.

[49] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[50] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-Up Segmentation for Top-Down Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[51] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning (ICML)*, 2017.

[52] Benedikt Fischer, André Brosig, Thomas M. Deserno, Bastian Ott, and Rolf W. Günther. Structural Scene Analysis and Content-based Image Retrieval applied to Bone Age Assessment. In *SPIE Medical Imaging*, 2009.

[53] Benedikt Fischer, Michael Sauren, Mark O. Güld, and Thomas M. Deserno. Scene Analysis with Structural Prototypes for Content-based Image Retrieval in Medicine. In *SPIE Medical Imaging*, 2008.

[54] Benedikt Fischer, Petra Welter, Christoph Grouls, Rolf W. Günther, and Thomas M. Deserno. Bone Age Assessment by Content-based Image Retrieval and Case-based Reasoning. In *SPIE Medical Imaging*, 2011.

[55] William T. Freeman and Michal Roth. Orientation Histograms for Hand Gesture Recognition. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 1995.

[56] William T Freeman, Ken-ichi Tanaka, Jun Ohta, and Kazuo Kyuma. Computer Vision for Computer Games. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 1996.

[57] Yoav Freund and Robert E. Schapire. A Desicion-theoretic Generalization of on-line Learning and an Application to Boosting. In *European Conference on Computational Learning Theory (EuroCOLT)*, 1995.

[58] Lex Fridman, Philipp Langhans, Joonbum Lee, and Bryan Reimer. Driver Gaze Region Estimation without Use of Eye Movement. *IEEE Intelligent Systems*, 31(3):49–56, 2016.

[59] Jerome H Friedman. Stochastic Gradient Boosting. *Computational statistics & data analysis*, 38(4), 2002.

[60] Eric Gabriel. *Automatic Multi-Scale and Multi-Object Pedestrian and Car Detection in Digital Images Based on the Discriminative Generalized Hough Transform and Deep Convolutional Neural Networks*. PhD thesis, Christian-Albrechts Universität Kiel, 2019.

[61] Eric Gabriel, Ferdinand Hahmann, Gordon Böer, Hauke Schramm, and Carsten Meyer. Structured Edge Detection for Improved Object Localization using the Discriminative Generalized Hough Transform. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2016.

[62] Eric Gabriel, Carsten Meyer, and Hauke Schramm. The Discriminative Generalized Hough Transform as a Proposal Generator for a Deep Network in Automatic Pedestrian Localization. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2018.

[63] Eric Gabriel, Michael Schleiss, Hauke Schramm, and Carsten Meyer. Analysis of the Discriminative Generalized Hough Transform as a Proposal Generator for a Deep Network in Automatic Pedestrian and Car Detection. *Journal of Electronic Imaging*, 27(5):051228, 2018.

[64] Eric Gabriel, Hauke Schramm, and Carsten Meyer. Experiments on Pedestrian Localization Using the Discriminative Generalized Hough Transform. In *International Symposium on Ambient Intelligence and Embedded Systems (AmiEs)*, 2016.

[65] Eric Gabriel, Hauke Schramm, and Carsten Meyer. Analysis of the Discriminative Generalized Hough Transform for Pedestrian Detection. In *International Conference on Image Analysis and Processing (ICIAP)*. Springer, 2017.

[66] Eric Gabriel, Hauke Schramm, and Carsten Meyer. Analysis of the Discriminative Generalized Hough Transform for Pedestrian Detection. In *International Conference on Image Analysis and Processing (ICIAP)*, 2017.

[67] Eric Gabriel, Hauke Schramm, and Carsten Meyer. The Discriminative Generalized Hough Transform as a Proposal Generator for a Deep Network in Automatic Pedestrian Localization. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2018.

[68] Juergen Gall and Victor Lempitsky. Class-specific Hough Forests for Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[69] Juergen Gall and Victor Lempitsky. Class-specific Hough Forests for Object Detection. In *Criminisi A., Shotton J. (eds) Decision Forests for Computer Vision and Medical Image Analysis. Advances in Computer Vision and Pattern Recognition.*, pages 143–157. Springer, 2013.

[70] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough Forests for Object Detection, Tracking, and Action Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011.

[71] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015.

[72] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[73] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable Part Models are Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[74] NI Glumov, EI Kolomiyetz, and VV Sergeyev. Detection of Objects on the Image using a Sliding Window Mode. *Optics & Laser Technology*, 27(4):241–249, 1995.

[75] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An Active Search Strategy for Efficient Object Class Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[76] David González-Ortega, Francisco Javier Díaz-Pernas, Mario Martínez-Zarzuela, Miriam Antón-Rodríguez, Jose Fernando Díez-Higuera, and Daniel Boto-Giralda. Real-time Hands, Face and Facial Features Detection and Tracking: Application to Cognitive Rehabilitation Tests Monitoring. *Journal of Network and Computer Applications*, 33(4):447–466, 2010.

[77] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[78] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572*, 2014.

[79] Ferdinand Hahmann, Gordon Böer, Thomas M. Deserno, and Hauke Schramm. Epiphyses Localization for Bone Age Assessment Using the Discriminative Generalized Hough Transform. In *Bildverarbeitung für die Medizin (BVM)*, 2014.

[80] Ferdinand Hahmann, Heike Ruppertshofen, Gordon Böer, and Hauke Schramm. Model Interpolation for Eye Localization using the Discriminative Generalized Hough Transform. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2012.

[81] Ferdinand Hahmann, Heike Ruppertshofen, Gordon Böer, Ralf Stannarius, and Hauke Schramm. Eye Localization Using the Discriminative Generalized Hough Transform. In *Joint DAGM (German Association for Pattern Recognition) and OAGM (Austrian Association for Pattern Recognition) Symposium*, 2012.

[82] Bharath Hariharan and Ross Girshick. Low-Shot Visual Recognition by Shrinking and Hallucinating Features. In *International Conference on Computer Vision (ICCV)*, 2017.

[83] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining Efficient Object Localization and Image Classification. In *International Conference on Computer Vision (ICCV)*, 2009.

[84] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017.

[85] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision (ECCV)*, 2014.

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[87] Elad Hoffer and Nir Ailon. Deep Metric Learning using Triplet Network. In *International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*, 2015.

[88] Tianyi Hong, Huabiao Qin, and Qianshu Sun. An Improved Real Time Eye State Identification System in Driver Drowsiness Detection. In *International Conference on Control and Automation (ICCA)*, 2007.

[89] Shin Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep Convolutional Neural Networks for Computer-aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE transactions on medical imaging*, 35(5):1285, 2016.

[90] Paul V.C. Hough. Method and Means for Recognizing Complex Patterns, 1962. U.S. Patent 3069654.

[91] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[92] Yani Ioannou, Duncan Robertson, Darko Zikic, Peter Kontschieder, Jamie Shotton, Matthew Brown, and Antonio Criminisi. Decision Forests, Convolutional Networks and the Models In-between. *arXiv:1603.01250*, 2016.

[93] Seyed Mehdi Iranmanesh, Ali Dabouei, Sobhan Soleymani, Hadi Kazemi, and Nasser Nasrabadi. Robust Facial Landmark Detection via Aggregation on Geometrically Manipulated Faces. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[94] Mirantha Jayathilaka. Enhancing Generalization of First-Order Meta-Learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[95] Edwin T Jaynes. Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630, 1957.

[96] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. Robust Face Detection using the Hausdorff Distance. In *International Conference on Audio-and Video-Based Biometric Person Authentication (AVBPA)*, 2001.

[97] B-H Juang and Shigeru Katagiri. Discriminative Learning for Minimum Error Classification (Pattern Recognition). *IEEE Transactions on signal processing*, 40(12):3043–3054, 1992.

[98] Aniwat Juhong and Chuchart Pintavirooj. Face Recognition Based on Facial Landmark Detection. In *Biomedical Engineering International Conference (BMEiCON)*, 2017.

[99] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[100] Andrzej Kasinski, Andrzej Florek, and Adam Schmidt. The PUT Face Database. *Image Processing and Communications*, 13(3-4):59–64, 2008.

[101] Andrzej Kasinski and Adam Schmidt. The Architecture and Performance of the Face and Eyes Detection System based on the Haar Cascade Classifiers. *Pattern Analysis & Applications*, 13(2):197–211, 2010.

[102] Ashraf A. Kassim, Tiowseng Tan, and K. H. Tan. A Comparative Study of Efficient Generalised Hough Transform Techniques. *Image and Vision Computing*, 17(10):737–748, 1999.

[103] Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[104] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[105] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882*, 2014.

[106] Klaus J. Kirchberg, Oliver Jesorsky, and Robert W. Frischholz. Genetic Model Optimization for Hausdorff Distance-Based Face Localization. In *Biometric Authentication*, 2002.

[107] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-Shot Image Recognition. In *Deep Learning Workshop at International Conference on Machine Learning (ICML)*, 2015.

[108] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[109] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[110] Bart Kroon, Sander Maas, Sabri Boughorbel, and Alan Hanjalic. Eye Localization in Low and Standard Definition Content with Application to Face Matching. *Computer Vision and Image Understanding*, 113(8):921–933, 2009.

[111] Hei Law and Jia Deng. CornerNet: Detecting Objects as Paired Keypoints. In *European Conference on Computer Vision (ECCV)*, 2018.

[112] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[113] Thomas M. Lehmann, Daniel Beier, Christian Thies, and Thomas Seidl. Segmentation of Medical Images Combining Local, Regional, Global, and Hierarchical Distances into a Bottom-up Region Merging Scheme. In *SPIE Medical Imaging*, 2005.

[114] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *Workshop on statistical learning in computer vision at European Conference on Computer Vision (ECCV)*, 2004.

[115] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008.

[116] Bastian Leibe and Bernt Schiele. Interleaving Object Categorization and Segmentation. In *Christensen H.I., Nagel HH. (eds) Cognitive Vision Systems*. Springer, 2006.

[117] Shaoxin Li, Junliang Xing, Zhiheng Niu, Shiguang Shan, and Shuicheng Yan. Shape Driven Kernel Adaptation in Convolutional Neural Network for Robust Facial Trait Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[118] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-Aware Trident Networks for Object Detection. In *International Conference on Computer Vision (ICCV)*, 2019.

[119] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human Pose Estimation Using Deep Consensus Voting. In *European Conference on Computer Vision (ECCV)*, 2016.

[120] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision (ICCV)*, 2017.

[121] Claudia Lindner, Shankhar Thiagarajah, J. Mark Wilkinson, Gillian A. Wallis, and Timothy F. Cootes. Fully Automatic Segmentation of the Proximal Femur using Random Forest Regression Voting. *IEEE transactions on medical imaging*, 32(8):1462–1472, 2013.

[122] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path Aggregation Network for Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[123] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[124] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[125] Qixiang Ma, Longyu Jiang, Wenxue Yu, Rui Jin, Zhixiang Wu, and Fangjin Xu. Training with Noise Adversarial Network: A Generalization Method for Object Detection on Sonar Image. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[126] Alexander O Mader, Hauke Schramm, and Carsten Meyer. Efficient Epiphyses Localization Using Regression Tree Ensembles and a Conditional Random Field. In *Bildverarbeitung für die Medizin (BVM)*, 2017.

[127] Alexander Oliver Mader, Cristian Lorenz, Martin Bergtholdt, Jens von Berg, Hauke Schramm, Jan Modersitzki, and Carsten Meyer. Detection and Localization of Landmarks in the Lower Extremities using an Automatically Learned Conditional Random Field. In *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, pages 64–75. Springer, 2017.

[128] Alexander Oliver Mader, Cristian Lorenz, Martin Bergtholdt, Jens von Berg, Hauke Schramm, Jan Modersitzki, and Carsten Meyer. Detection and Localization of Spatially Correlated Point Landmarks in Medical Images using an Automatically Learned Conditional Random Field. *Computer Vision and Image Understanding*, 176:45–53, 2018.

[129] Subhransu Maji and Jitendra Malik. Object Detection using a Max-margin Hough Transform. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[130] Erno Makinen and Roope Raisamo. Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):541–547, 2008.

[131] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime Object Proposals with Randomized Prim's Algorithm. In *International Conference on Computer Vision (ICCV)*, 2013.

[132] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[133] Ana Belén Martin-Recuero, Peter Beyerlein, and Hauke Schramm. Discriminative Optimization of 3-D Shape Models for the Generalized Hough Transform. In *International Symposium on Ambient Intelligence and Embedded Systems (AmiEs)*, 2008.

[134] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face Detection without Bells and Whistles. In *European Conference on Computer Vision (ECCV)*, 2014.

[135] Robert K. McConnell. Method of and Apparatus for Pattern Recognition, 1986. Patent: U.S. Classification 382/170, 382/209; International Classification G06K9/48; Cooperative Classification G06K9/48; European Classification G06K9/48.

[136] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, 2013.

[137] David J. Miller, Zhen Xiang, and George Kesidis. Adversarial Learning in Statistical Classification: A Comprehensive Review of Defenses Against Attacks. *arXiv:1904.06292*, 2019.

[138] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A Simple Neural Attentive Meta-Learner. In *International Conference on Learning Representations (ICLR)*, 2018.

[139] Jonas Močkus. On Bayesian Methods for Seeking the Extremum. In *Optimization Techniques (IFIP Technical Conference)*, 1975.

[140] Woonhyun Nam, Piotr Dollar, and Joon Hee Han. Local Decorrelation For Improved Pedestrian Detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[141] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning. In *ACM International Conference on Multimodal Interaction (ICMI)*. ACM, 2015.

[142] Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. *arXiv:1803.02999*, 2018.

[143] Olegs Nikisins and Modris Greitans. Local Binary Patterns and Neural Network based Technique for Robust Face Detection and Localization. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2012.

[144] Ryuzo Okada. Discriminative Generalized Hough Transform for Object Dectection. In *International Conference on Computer Vision (ICCV)*, 2009.

[145] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task Dependent Adaptive Metric for Improved Few-shot Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[146] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, and Xiaoou Tang. DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[147] Wanli Ouyang, Xingyu Zeng, Xiaogang Wang, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Hongyang Li, Kun Wang, Junjie Yan, Chen-Change Loy, and Xiaoou Tang. DeepID-Net: Object Detection with Deformable Part Based Convolutional Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1320–1334, 2017.

[148] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast Keypoint Recognition Using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.

[149] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A General Framework for Object Detection. In *International Conference on Computer Vision (ICCV)*, 1998.

[150] Seymour A. Papert. The Summer Vision Project. *Massachusetts Institute of Technology (MIT)*, 1966.

[151] Marco Pedersoli, Jordi Gonzalez, Xu Hu, and Xavier Roca. Toward Real-Time Pedestrian Detection Based on a Deformable Template Model. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):355–364, 2014.

[152] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: One-Class Novelty Detection Using GANs With Constrained Latent Representations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[153] Claudio A. Perez, Carlos M. Aravena, Juan I. Vallejos, Pablo A. Estevez, and Claudio M. Held. Face and Iris Localization using Templates Designed by Particle Swarm Optimization. *Pattern Recognition Letters*, 31(9):857–868, 2010.

[154] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET Evaluation Methodology for Face-recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[155] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[156] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[157] Hang Qi, Matthew Brown, and David G. Lowe. Low-Shot Learning With Imprinted Weights. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[158] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Applied Sciences*, 2019.

[159] Sachin Ravi and Hugo Larochelle. Optimization as a Model for Few-Shot Learning. In *International Conference on Learning Representations (ICLR)*, 2016.

[160] Nima Razavi, Juergen Gall, and Luc Van Gool. Backprojection Revisited: Scalable Multi-view Object Detection and Similarity Metrics for Detections. In *European Conference on Computer Vision (ECCV)*, 2010.

[161] Nima Razavi, Juergen Gall, Pushmeet Kohli, and Luc Van Gool. Latent Hough Transform for Object Detection. In *European Conference on Computer Vision (ECCV)*, 2012.

[162] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-time Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[163] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. *arXiv:1612.08242*, 2017.

[164] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv:1804.02767*, 2018.

[165] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-Learning for Semi-Supervised Few-Shot Classification. In *Deep Learning Workshop at International Conference on Machine Learning (ICML)*, 2018.

[166] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[167] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[168] David L. Richmond, Dagmar Kainmueller, Michael Yang, Eugene W. Myers, and Carsten Rother. Mapping Stacked Decision Forests to Deep and Sparse Convolutional Neural Networks for Semantic Segmentation. In *British Machine Vision Conference (BMVC)*, 2015.

[169] Heike Ruppertshofen. *Automatic Modeling of Anatomical Variability for Object Localization in Medical Images*. PhD thesis, Universität Magdeburg, 2012.

[170] Heike Ruppertshofen, Thomas Bülow, Jens von Berg, Sarah Schmidt, Peter Beyerlein, Zein Salah, Georg Rose, and Hauke Schramm. A Multidimensional Model for Localization of Highly Variable Objects. In *SPIE Medical Imaging Conference*, 2012.

[171] Heike Ruppertshofen, Daniel Künne, Cristian Lorenz, Sarah Schmidt, Peter Beyerlein, Zein Salah, Georg Rose, and Hauke Schramm. Multi-level Approach for the Discriminative Generalized Hough Transform. In *Jahrestagung der Deutschen Gesellschaft für Computer-und Roboterassistierte Chirurgie (CURAC)*, 2011.

[172] Heike Ruppertshofen, Cristian Lorenz, Peter Beyerlein, Zein Salah, Georg Rose, and Hauke Schramm. Fully Automatic Model Creation for Object Localization utilizing the Generalized Hough Transform. In *Bildverarbeitung für die Medizin (BVM)*, 2010.

[173] Heike Ruppertshofen, Cristian Lorenz, Georg Rose, and Hauke Schramm. Discriminative Generalized Hough Transform for Object Localization in Medical Images. *International journal of computer assisted radiology and surgery*, 8(4):593–606, 2013.

[174] Heike Ruppertshofen, Cristian Lorenz, Georg Rose, and Hauke Schramm. Discriminative Generalized Hough Transform for Object Localization in Medical Images. *International journal of computer assisted radiology and surgery*, 8(4):593–606, 2013.

[175] Heike Ruppertshofen, Cristian Lorenz, Sarah Schmidt, Peter Beyerlein, Zein Salah, Georg Rose, and Hauke Schramm. Iterative Training of Discriminative Models for the Generalized Hough Transform. In *Menze B., Langs G., Tu Z., Criminisi A. (eds) Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging. (MCV)*, number 6533, pages 21–30. 2011.

[176] Yunus Saatci and Christopher Town. Cascaded Classification of Gender and Facial Expression using Active Appearance Models. In *International Conference on Automatic Face and Gesture Recognition (FGR)*, 2006.

[177] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with Memory-Augmented Neural Networks. In *International Conference on Machine Learning (ICML)*, 2016.

[178] Hauke Schramm. Automated Generation of Shape-Variant Hough Models for the Generalized Hough Transform, 2007. Patent: PCT/IB2006/054912.

[179] Simon-Martin Schröder, Rainer Kiko, Jean-Olivier Irisson, and Reinhard Koch. Low-Shot Learning of Plankton Categories. In *German Conference on Pattern Recognition (GCPR)*, 2018.

[180] Samuel Schulter, Christian Leistner, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Accurate Object Detection with Joint Classification-Regression Random Forests. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[181] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, Localization and Detection using Convolutional Networks. *arXiv:1312.6229*, 2013.

[182] Juan Serrano-Cuerda, José Carlos Castillo, and Antonio Fernández-Caballero. Indoor Overhead Video Camera for Efficent People Counting. *Jurnal Teknologi*, 63(3):17–22, 2013.

[183] Xiaoke Shen and Ioannis Stamos. Frustum VoxNet for 3D Object Detection from RGB-D or Depth images. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[184] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[185] Bharat Singh and Larry S. Davis. An Analysis of Scale Invariance in Object Detection SNIP. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[186] Bharat Singh, Mahyar Najibi, and Larry S Davis. SNIPER: Efficient Multi-Scale Training. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[187] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-Shot Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[188] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[189] Fengyi Song, Xiaoyang Tan, Songcan Chen, and Zhi-Hua Zhou. A Literature Survey on Robust and Efficient Eye Localization in Real-life Scenarios. *Pattern Recognition*, 46(12):3157–3173, 2013.

[190] Darko Stern, Thomas Ebner, Horst Bischof, Sabine Grassegger, Thomas Ehammer, and Martin Urschler. Fully Automatic Bone Age Estimation from Left Hand MR Images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2014.

[191] Juan Luis Suárez, Salvador García, and Francisco Herrera. A Tutorial on Distance Metric Learning: Mathematical Foundations, Algorithms and Experiments. *arXiv:1812.05944*, 2018.

[192] Keqiang Sun, Wayne Wu, Tinghao Liu, Shuo Yang, Quan Wang, Qiang Zhou, Zuochang Ye, and Chen Qian. FAB: A Robust Facial Landmark Detection Framework for Motion-Blurred Videos. In *International Conference on Computer Vision (ICCV)*, 2019.

[193] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-Transfer Learning for Few-Shot Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[194] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe. Scalable, High-Quality Object Detection. *arXiv:1412.1441*, 2014.

[195] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[196] Hans Henrik Thodberg, Sven Kreiborg, Anders Juul, and Karen Damgaard Pedersen. The BoneXpert Method for Automated Determination of Skeletal Maturity. *IEEE Transactions on Medical Imaging*, 28(1):52–66, 2009.

[197] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[198] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *International Conference on Computer Vision (ICCV)*, 2019.

[199] Fabian Timm and Erhardt Barth. Accurate Eye Centre Localisation by Means of Gradients. In *Computer Vision Theory and Applications (VISAPP)*, 2011.

[200] Doris Tsao. Primate Visual System, Face Patches, Segmentation and Tracking, Inferotemporal Cortex, July 2017. International Computer Vision Summer School (ICVSS), Sicily.

[201] Mehmet Türkan, MM Pardàs, and Ahmet Enis Cetin. Human Eye Localization using Edge Projections. In *Computer Vision Theory and Applications (VISAPP)*, 2007.

[202] Georgios Tzimiropoulos. Project-out Cascaded Regression with an Application to Face Alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[203] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[204] Roberto Valenti and Theo Gevers. Accurate Eye Center Location and Tracking using Isophote Curvature. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[205] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[206] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[207] Haochen Wang, Ruotian Luo, Michael Maire, and Greg Shakhnarovich. Pixel Consensus Voting for Panoptic Segmentation. *arXiv:2004.01849*, 2020.

[208] Jinglei Wang, Fei Long, Jiping Chen, and Junfeng Yao. A Novel Eye Localization Method Based on Log-Gabor Transform and Integral Image. *Applied Mathematics*, 6(2S):323S–329S, 2012.

[209] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-Shot Learning From Imaginary Data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[210] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A Survey of Transfer Learning. *Journal of Big Data*, 3(1):9, 2016.

[211] Johannes Welbl. Casting Random Forests as Artificial Neural Networks (and Profiting from It). In *German Conference on Pattern Recognition (GCPR)*, 2014.

[212] Georg Wimmer, Andreas Vécsei, and Andreas Uhl. CNN Transfer Learning for the Automated Diagnosis of Celiac Disease. In *International Conference on Image Processing Theory Tools and Applications (IPTA)*, 2016.

[213] Jelmer M. Wolterink, Konstantinos Kamnitsas, Christian Ledig, and Ivana Išgum. Deep learning: Generative adversarial networks and adversarial methods. In *Handbook of Medical Image Computing and Computer Assisted Intervention*. 2020.

[214] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011.

[215] Xuehan Xiong and Fernando De la Torre. Supervised Descent Method and Its Applications to Face Alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[216] Jumpei Yamamoto, Katsufumi Inoue, and Michifumi Yoshioka. Investigation of Customer Behavior Analysis Based on Top-View Depth Camera. In *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2017.

[217] Zhiguang Yang, Ming Li, and Haizhou Ai. An Experimental Study on Automatic Face Gender Classification. In *International Conference on Pattern Recognition (ICPR)*, 2006.

[218] Haichao Zhang and Jianyu Wang. Towards Adversarially Robust Object Detection. In *International Conference on Computer Vision (ICCV)*, 2019.

[219] Shanshan Zhang, Christian Bauckhage, and Armin B Cremers. Informed Haar-Like Features Improve Pedestrian Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[220] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Filtered Channel Features for Pedestrian Detection. *arXiv:1501.05759*, 2015.

[221] Yuting Zhang, Kihyuk Sohn, Ruben Villegas, Gang Pan, and Honglak Lee. Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction. *arXiv:1504.03293*, 2015.

[222] Zhishuai Zhang, Wei Shen, Siyuan Qiao, Yan Wang, Bo Wang, and Alan Yuille. Robust Face Detection via Learning Small Faces on Hard Images. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[223] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. *arXiv:1904.07850*, 2019.

[224] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-Up Object Detection by Grouping Extreme and Center Points. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

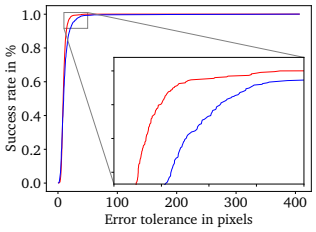
[225] Xiangxin Zhu and Deva Ramanan. Face Detection, Pose Estimation, and Landmark Localization in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[226] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

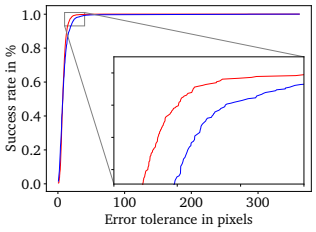
[227] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning Data Augmentation Strategies for Object Detection. *arXiv:1906.11172*, 2019.

[228] Xu Zou, Sheng Zhong, Luxin Yan, Xiangyun Zhao, Jiahuan Zhou, and Ying Wu. Learning Robust Facial Landmark Detection via Hierarchical Structured Ensemble. In *International Conference on Computer Vision (ICCV)*, 2019.

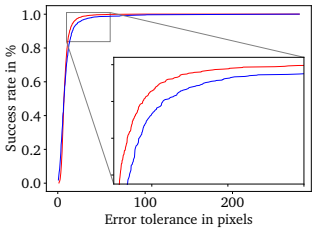
Appendix



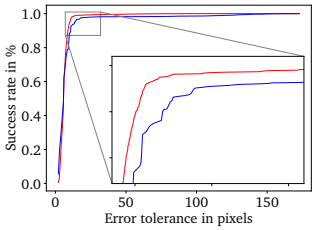
(a) FERET Face Database Zoom Level 1 for eye localization



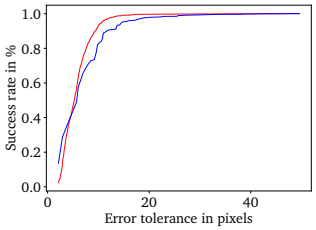
(b) FERET Face Database Zoom Level 1 for nose localization



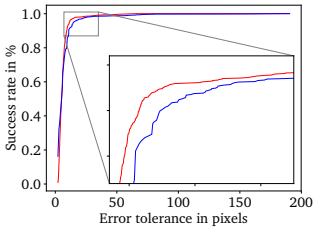
(c) FERET Face Database Zoom Level 1 for mouth localization



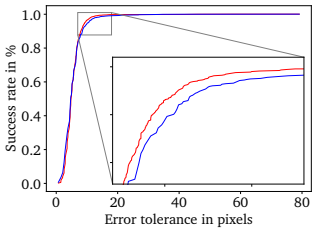
(d) FERET Face Database Zoom Level 2 for eye localization



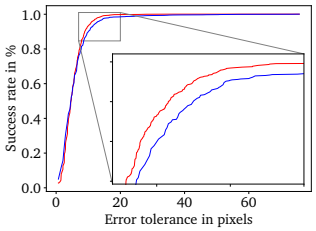
(e) FERET Face Database Zoom Level 2 for nose localization



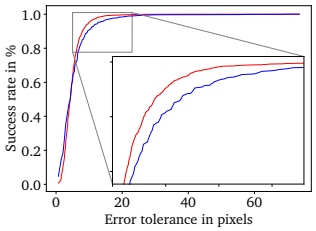
(f) FERET Face Database Zoom Level 2 for mouth localization



(g) FERET Face Database Zoom Level 3 for eye localization

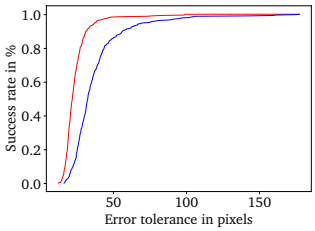


(h) FERET Face Database Zoom Level 3 for nose localization

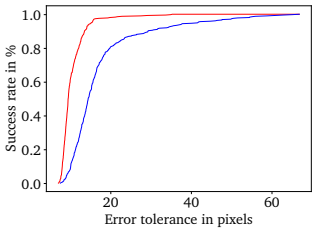


(i) FERET Face Database Zoom Level 3 for mouth localization

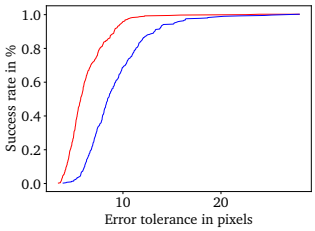
Figure A.1: Comparison of the success rates for the DGHT (blue lines) with DGHT+SCM (red lines). For the Chokepoint Dataset the four portals are shown with different line styles. The red lines always show higher success rates demonstrating the strong improvement achieved by the SCM. Note, the image extracts used in this figure are around the ground truth coordinates. Continued on next page.



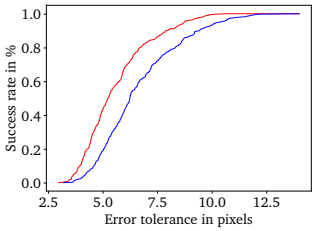
(j) RWTH Hand Database Zoom Level 1 averaged over all landmarks



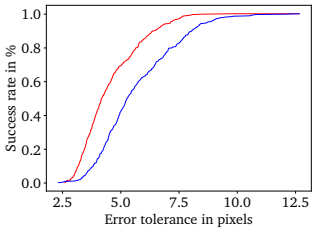
(k) RWTH Hand Database Zoom Level 2 averaged over all landmarks



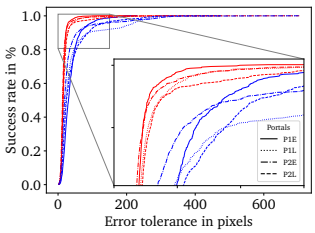
(l) RWTH Hand Database Zoom Level 3 averaged over all landmarks



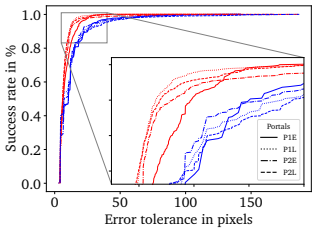
(m) RWTH Hand Database Zoom Level 4 averaged over all landmarks



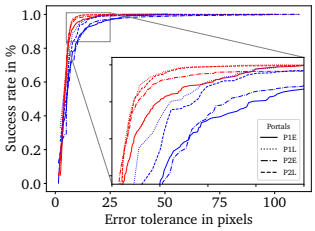
(n) RWTH Hand Database Zoom Level 5 averaged over all landmarks



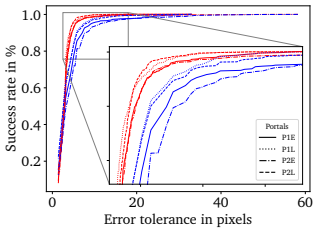
(o) Chokepoint Dataset Zoom Level 1



(p) Chokepoint Dataset Zoom Level 2



(q) Chokepoint Dataset Zoom Level 3



(r) Chokepoint Dataset Zoom Level 4

Figure A.1: Comparison of the success rates for the DGHT (blue lines) with DGHT+SCM (red lines). For the Chokepoint Dataset the four portals are shown with different line styles. The red lines always show higher success rates demonstrating the strong improvement achieved by the SCM. Note, the image extracts used in this figure are around the ground truth coordinates.

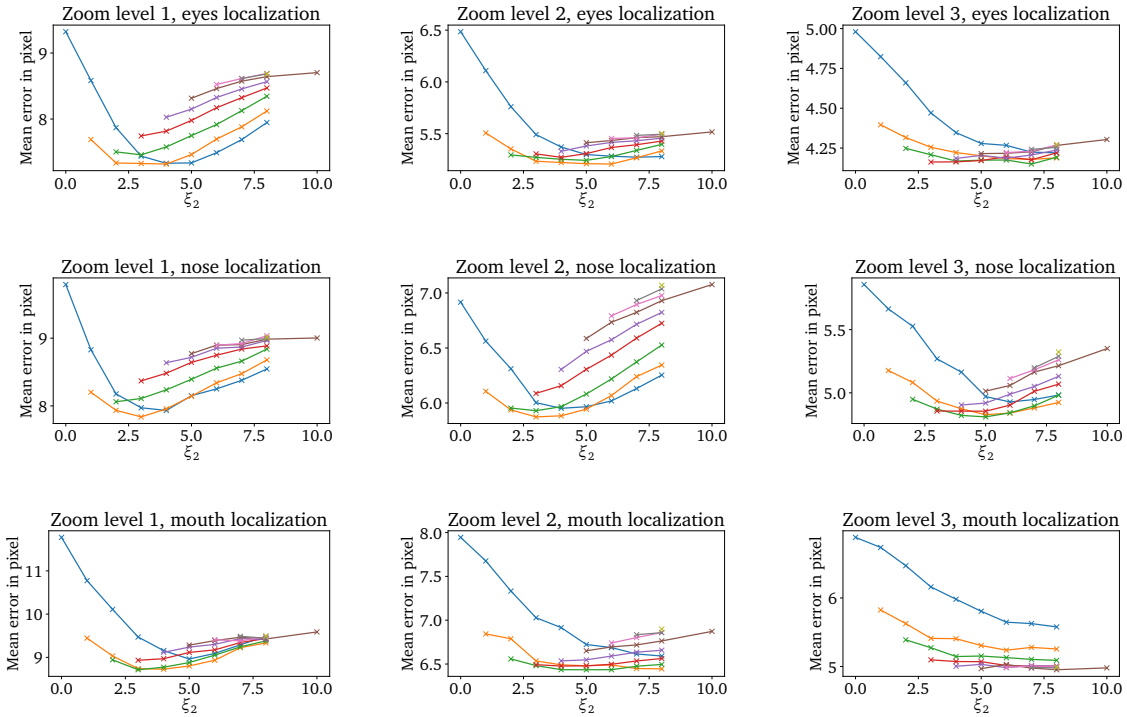


Figure A.2: The mean error for differentξ_1 and ξ_2 for the FERET Face Database. The x-axis shows the ξ_2 value, wheras each line represent one ξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, which is the value for ξ_1.

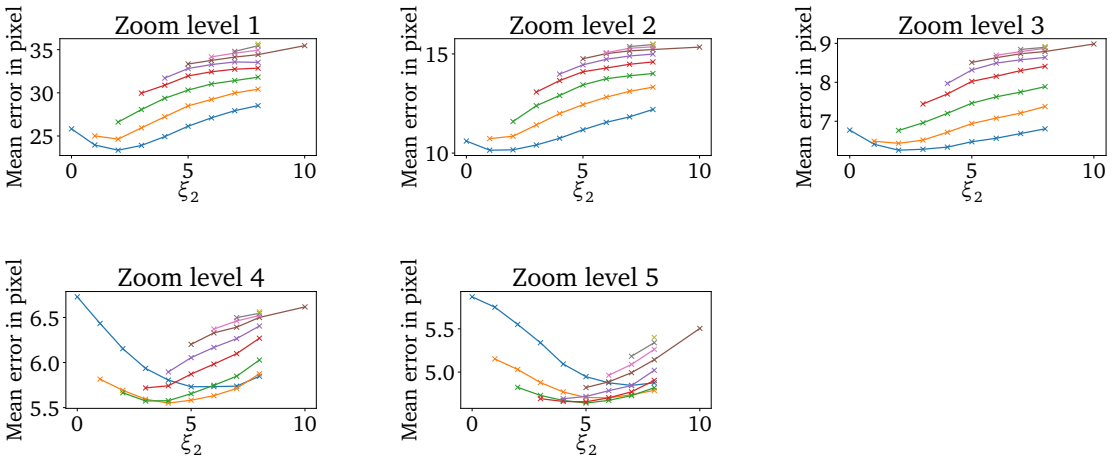


Figure A.3: The mean error for differentξ_1 and ξ_2 for the RWTH Hand Database. The x-axis shows the ξ_2 value, wheras each line represent one ξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, which is the value for ξ_1.

Appendix

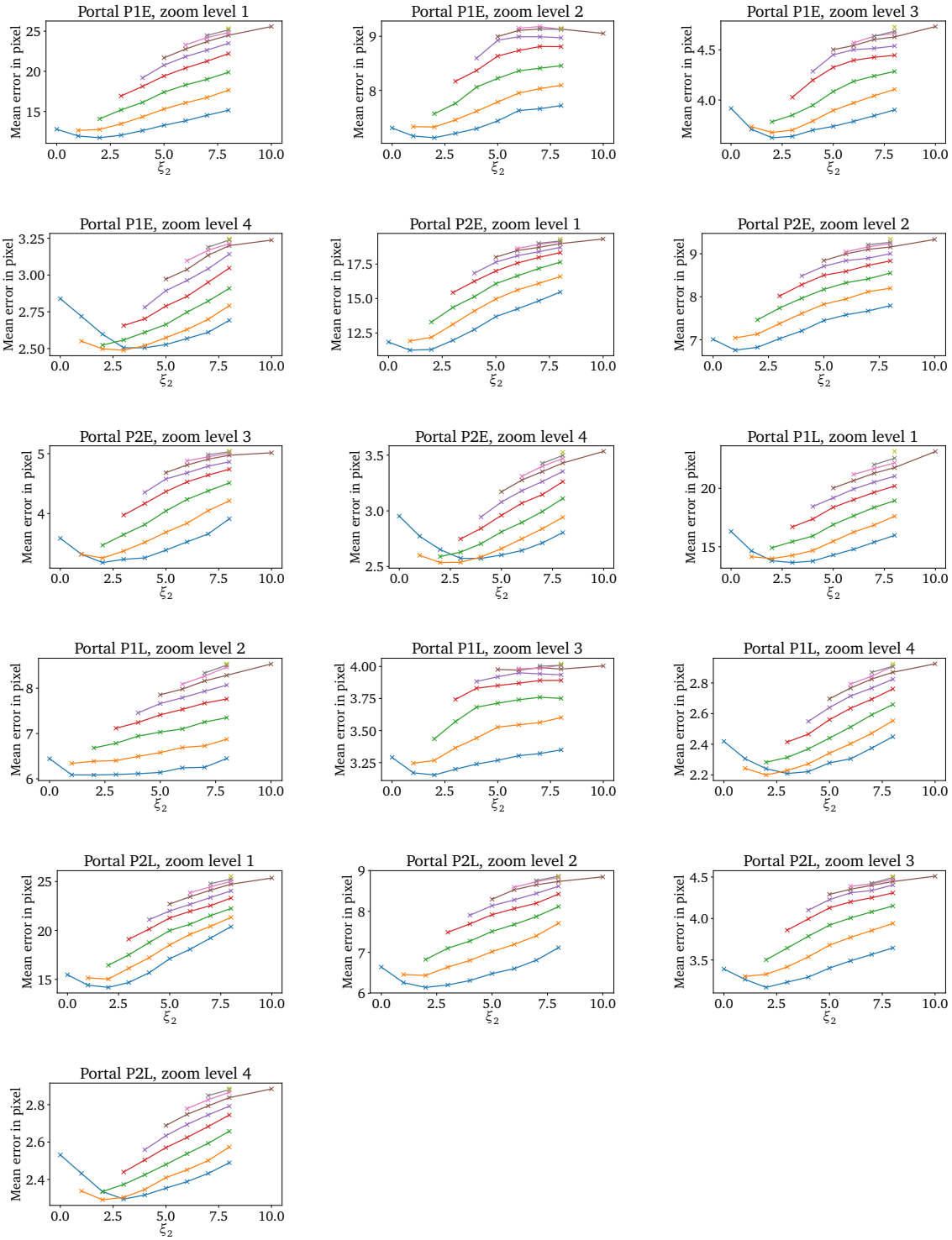


Figure A.4: The mean error for differentξ_1 and ξ_2 for the Chokepoint Dataset. The x-axis shows the ξ_2 value, wheras each line represent one ξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, which is the value for ξ_1.

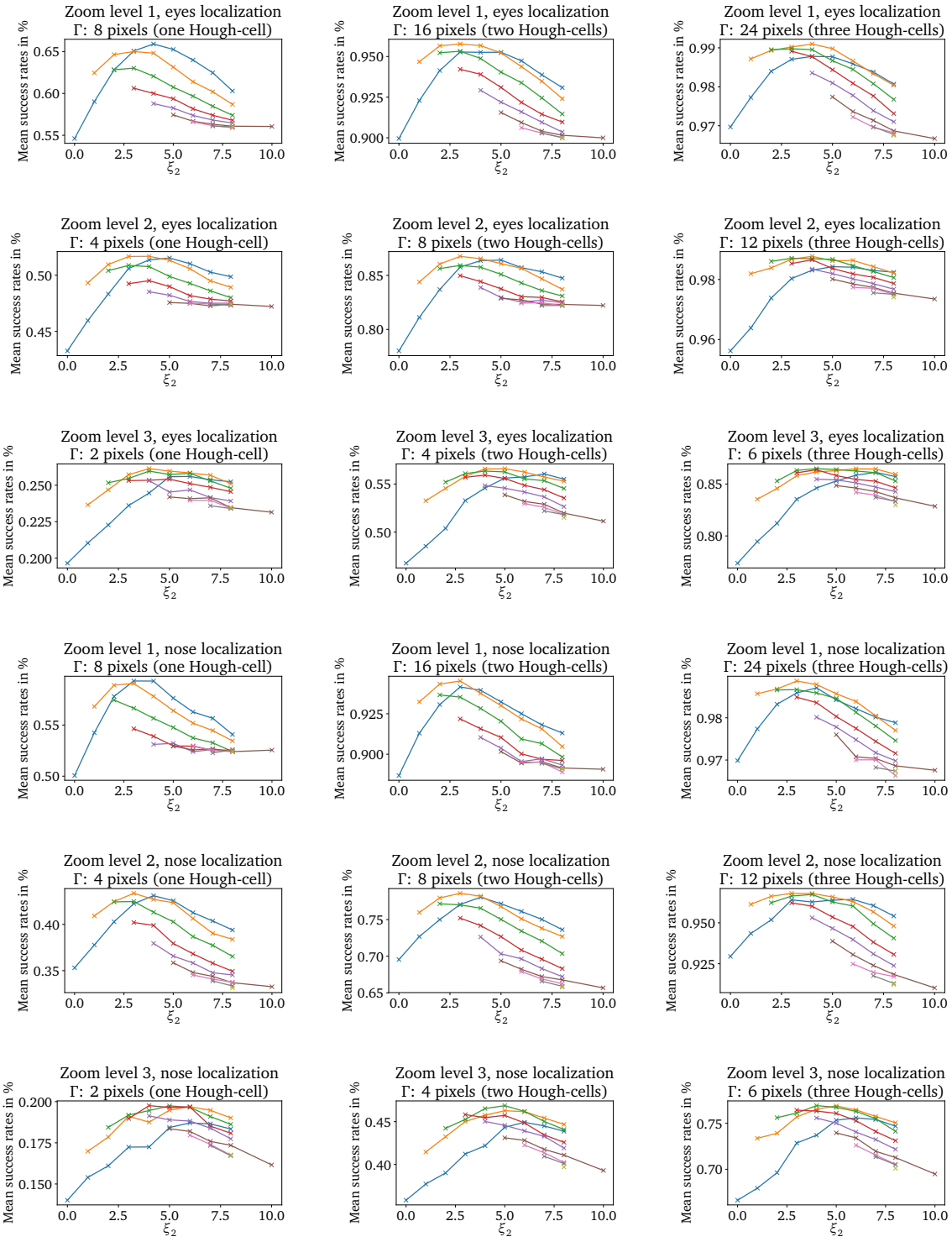


Figure A.5: The success rate for different ξ_1 and ξ_2 for the FERET Face Database and different error tolerances. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1. Continued on next page.

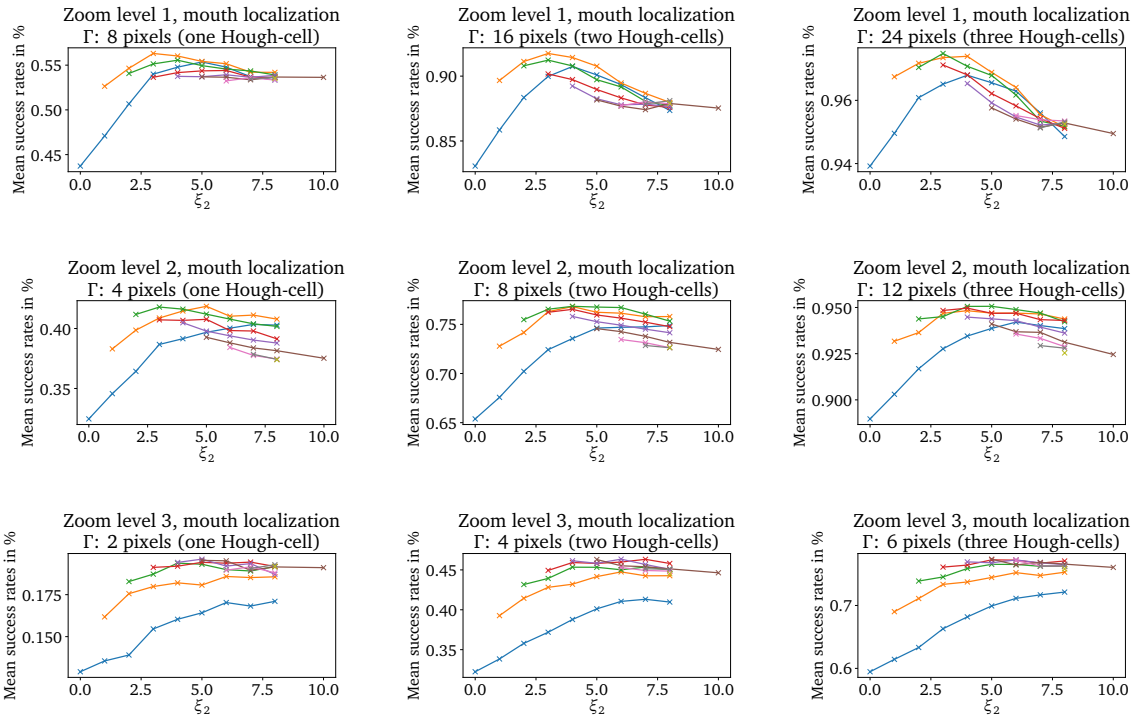


Figure A.5: The success rate for different ξ_1 and ξ_2 for the FERET Face Database and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1.

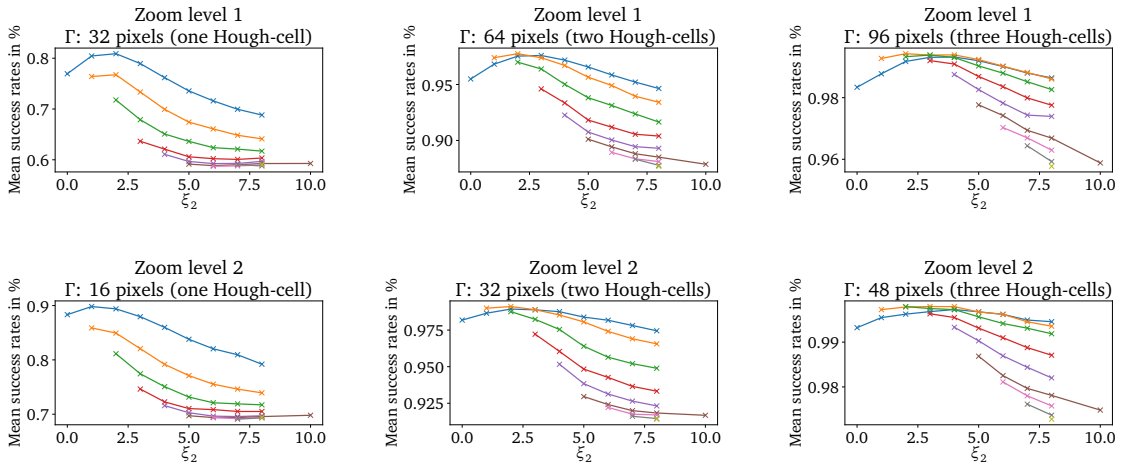


Figure A.6: The success rate for different ξ_1 and ξ_2 for the RWTH Hand Database and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1. Continued on next page.

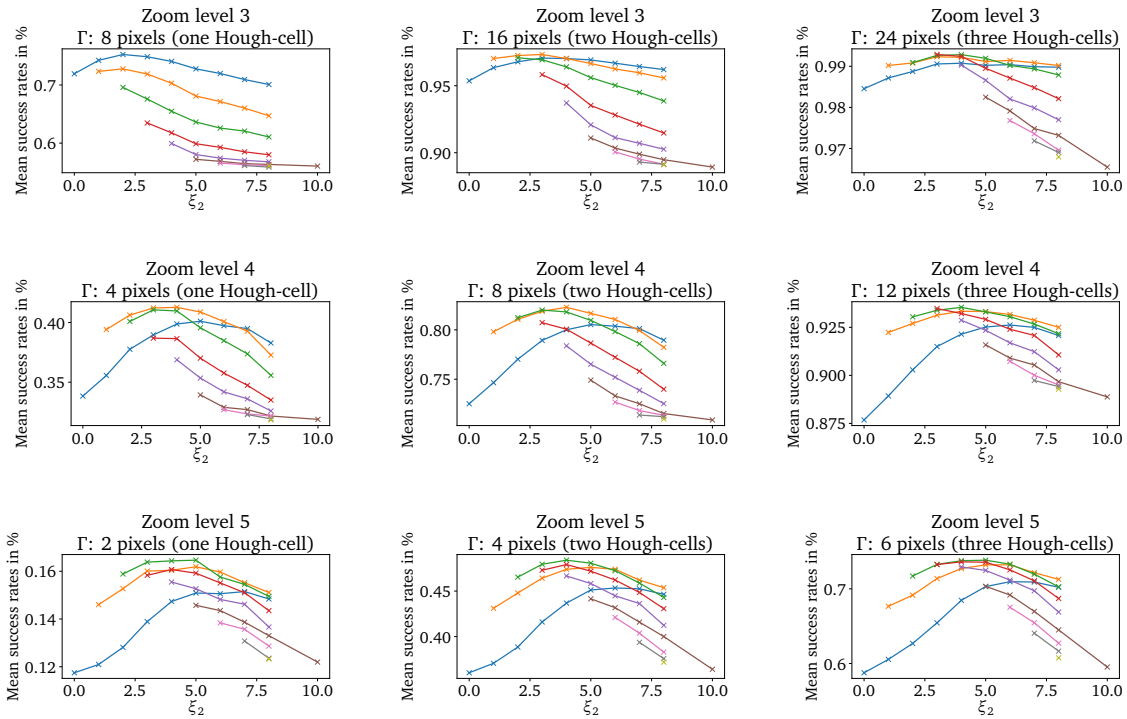


Figure A.6: The success rate for different ξ_1 and ξ_2 for the RWTH Hand Database and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1.

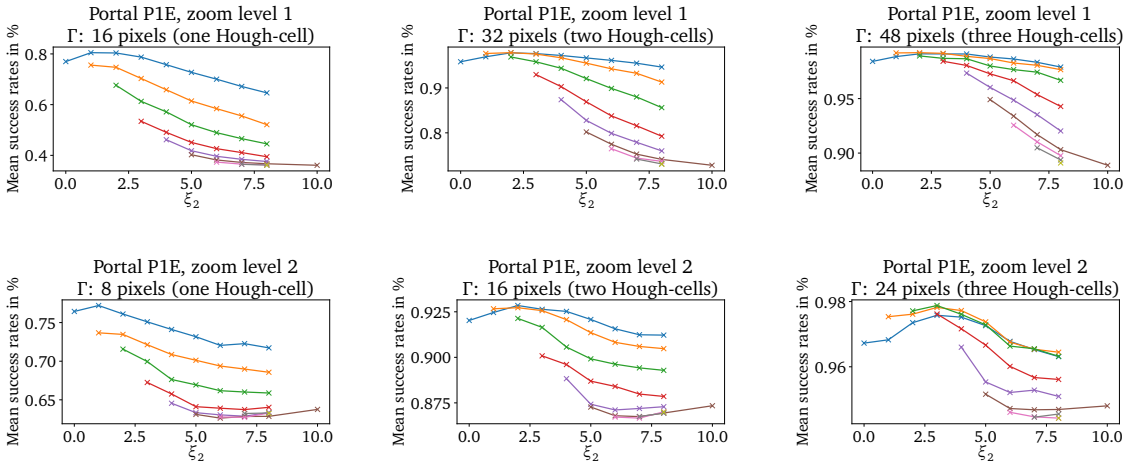


Figure A.7: The success rate for different ξ_1 and ξ_2 for the Chokepoint Dataset and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1. Continued on next page.

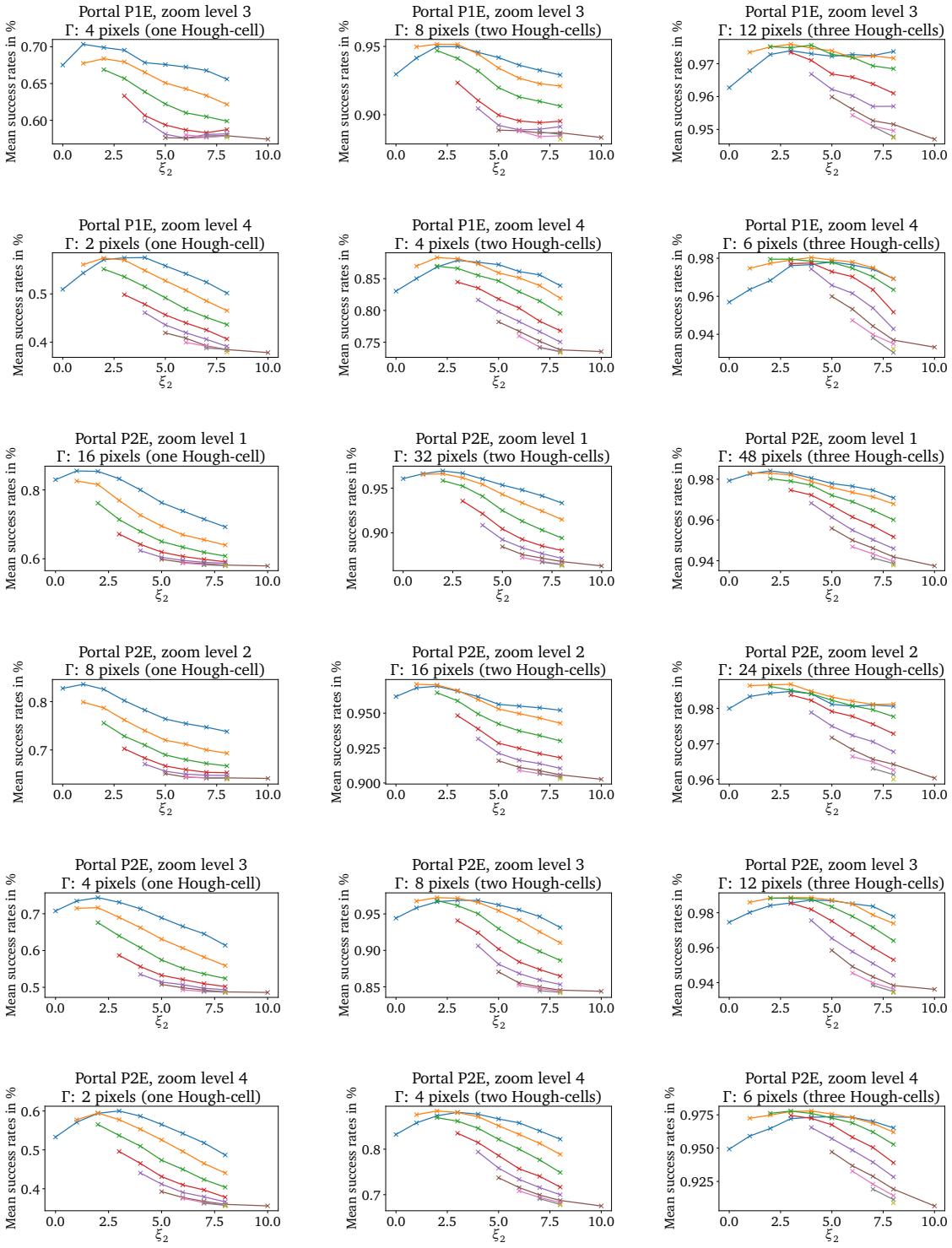


Figure A.7: The success rate for different ξ_1 and ξ_2 for the Chokepoint Dataset and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1. Continued on next page.

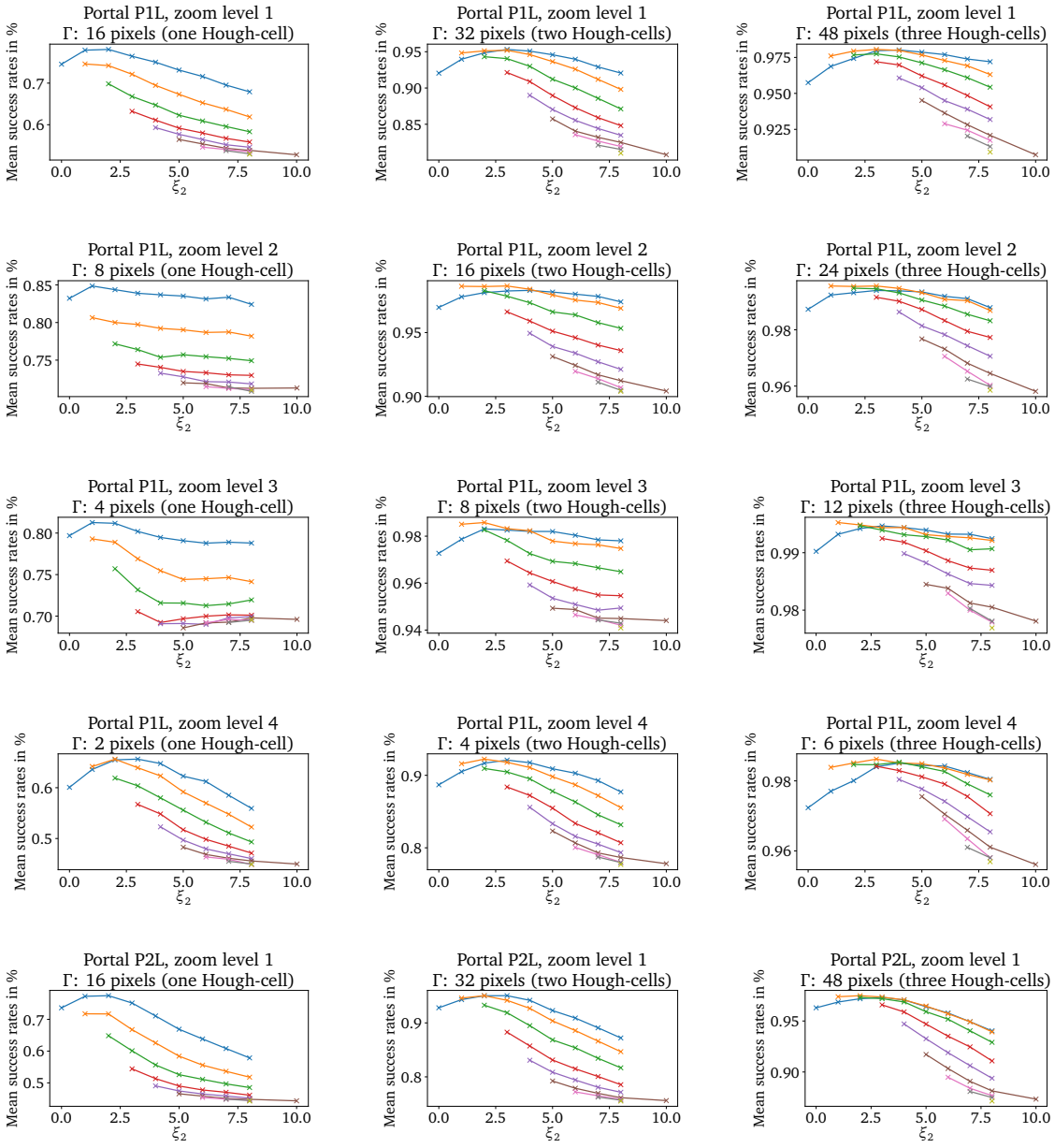


Figure A.7: The success rate for different ξ_1 and ξ_2 for the Chokepoint Dataset and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1. Continued on next page.

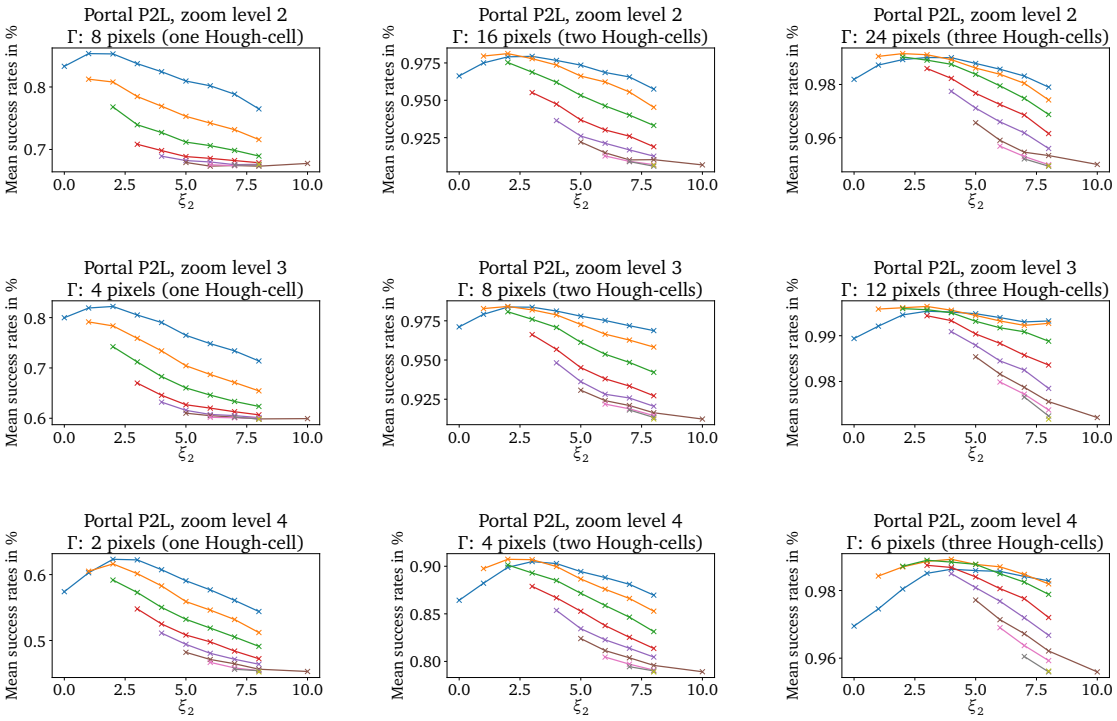
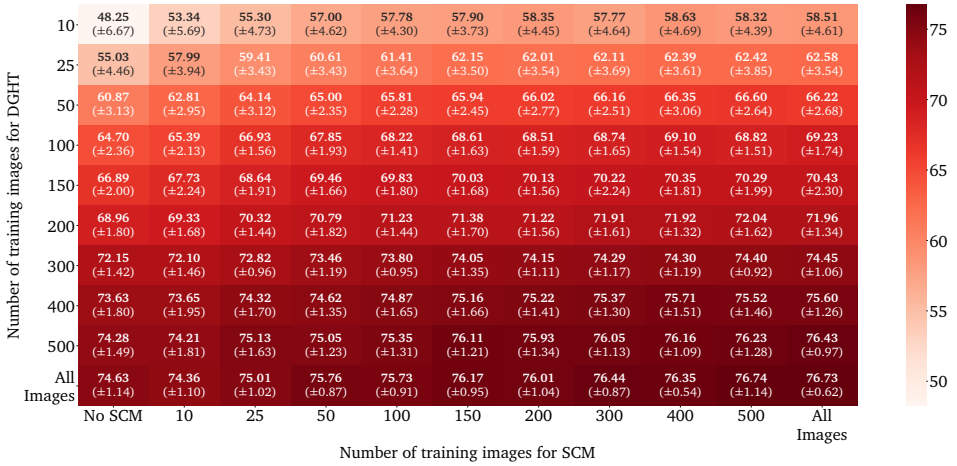
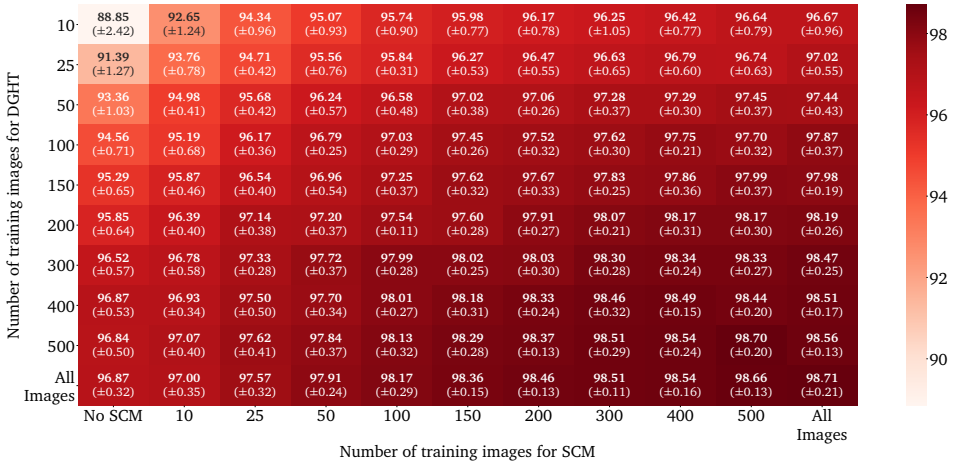


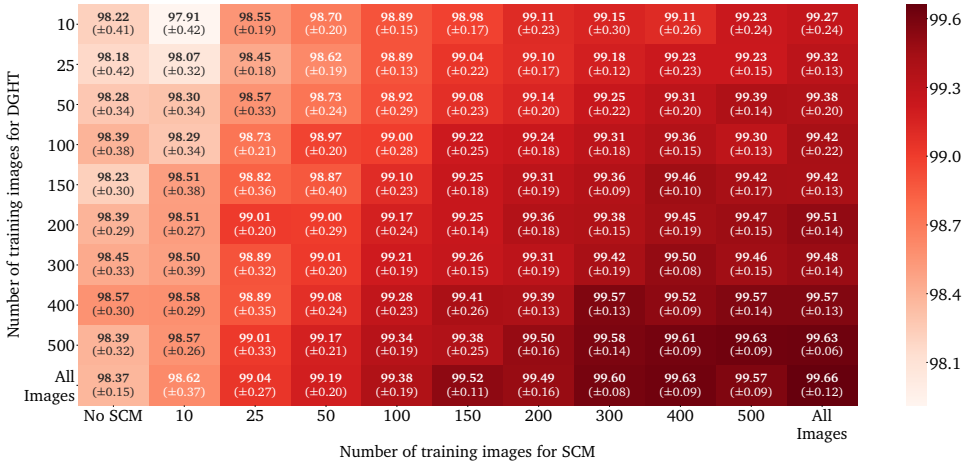
Figure A.7: The success rate for different ξ_1 and ξ_2 for the Chokepoint Dataset and different error tolerances Γ. The x-axis shows the ξ_2 value, whereas each line represents a different value forξ_1. Since $\xi_2 \geq \xi_1$, the lines start at different ξ_2 values, corresponds to ξ_1.



(a) Eye localization with error tolerance of 0.05 eye distance

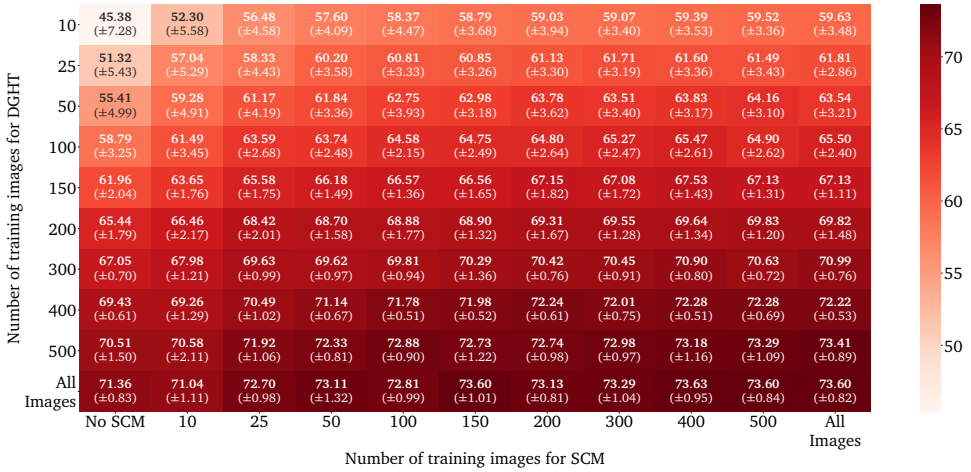


(b) Eye localization with error tolerance of 0.1 eye distance

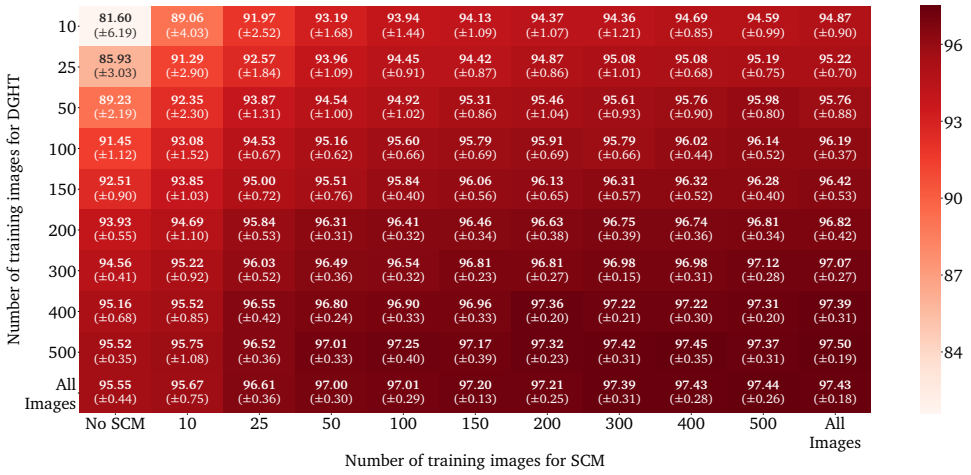


(c) Eye localization with error tolerance of 0.25 eye distance

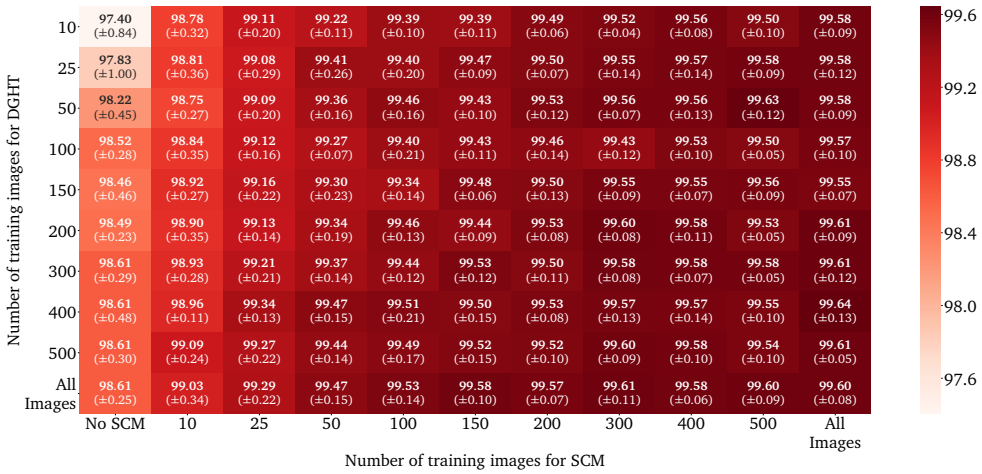
Figure A.8: Mean localization accuracy and standard deviation for FERET Face Database for different landmarks and error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(d) Nose localization with error tolerance of 0.05 eye distance

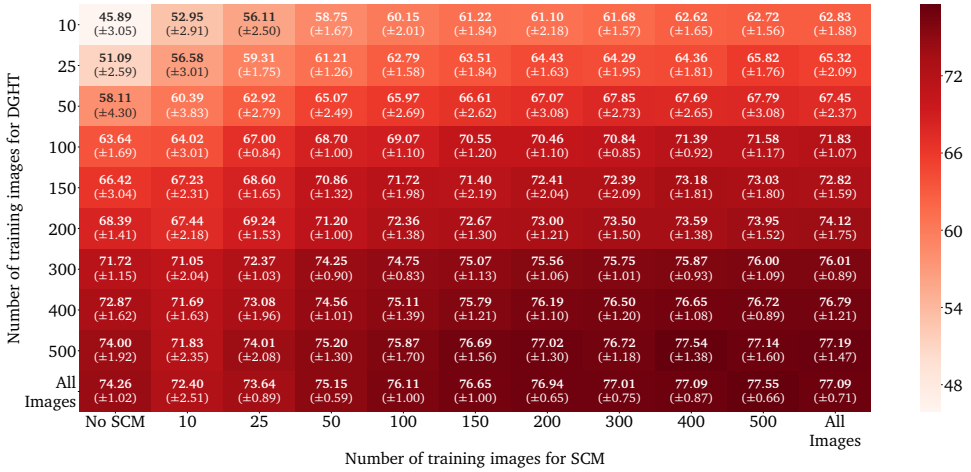


(e) Nose localization with error tolerance of 0.1 eye distance

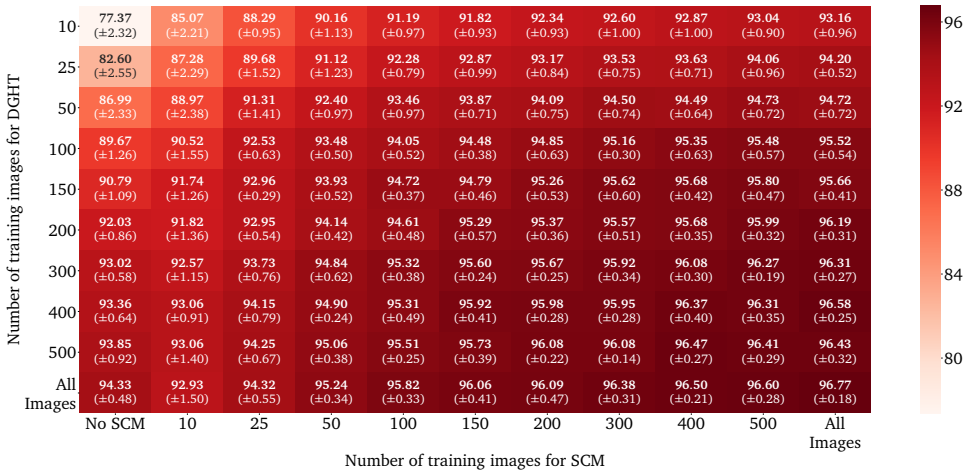


(f) Nose localization with error tolerance of 0.25 eye distance

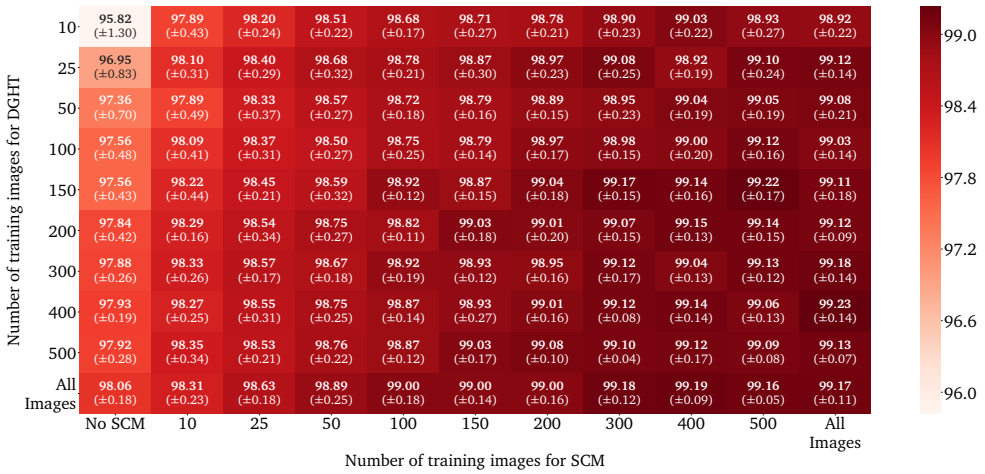
Figure A.8: Mean localization accuracy and standard deviation for FERET Face Database for different landmarks and error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(g) Mouth localization with error tolerance of 0.05 eye distance



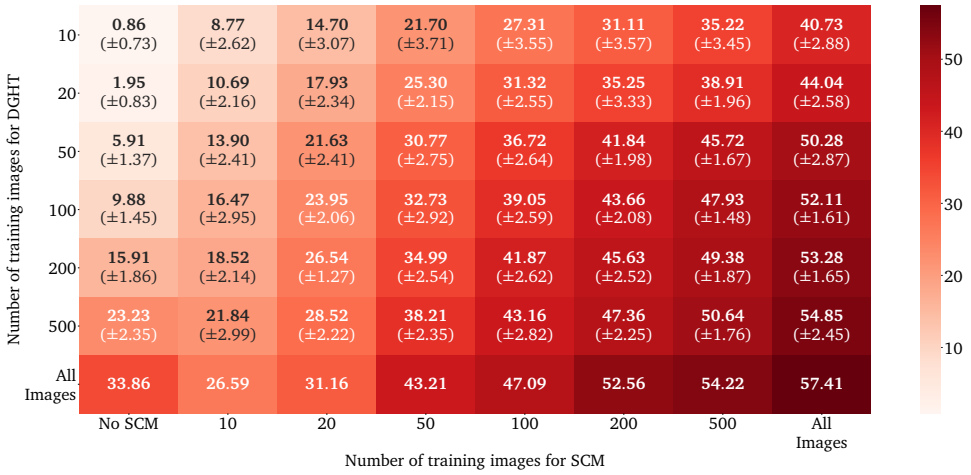
(h) Mouth localization with error tolerance of 0.1 eye distance



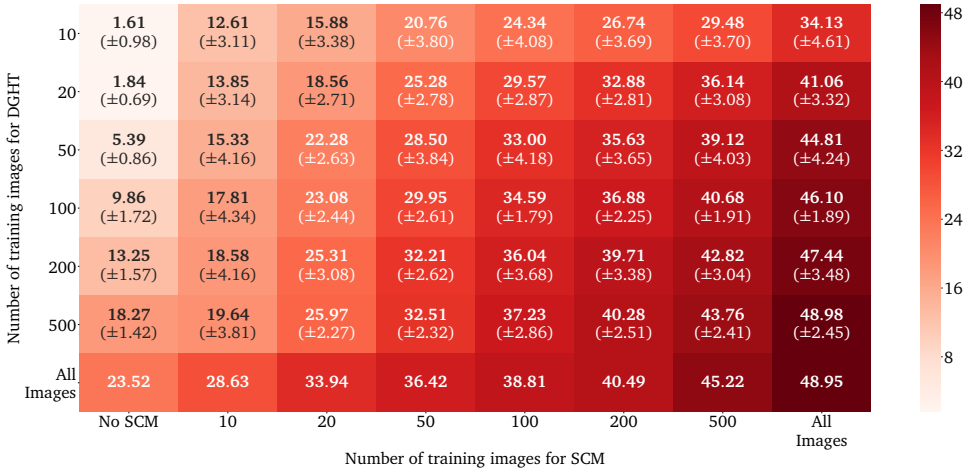
(i) Mouth localization with error tolerance of 0.25 eye distance

Figure A.8: Mean localization accuracy and standard deviation for FERET Face Database for different landmarks and error tolerances depending on the number of DGHT and SCM training images.

(a) P1E portal with error tolerance of 0.1 eye distance

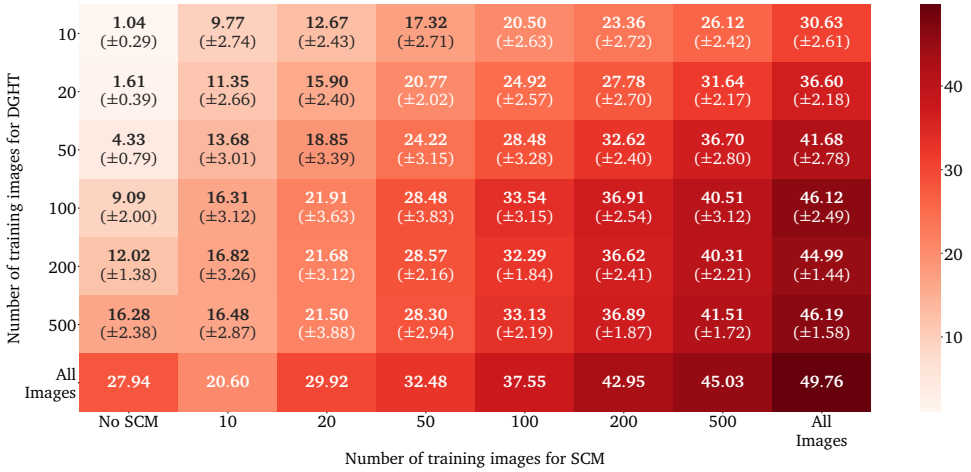


(b) P1L portal of 0.1 eye distance

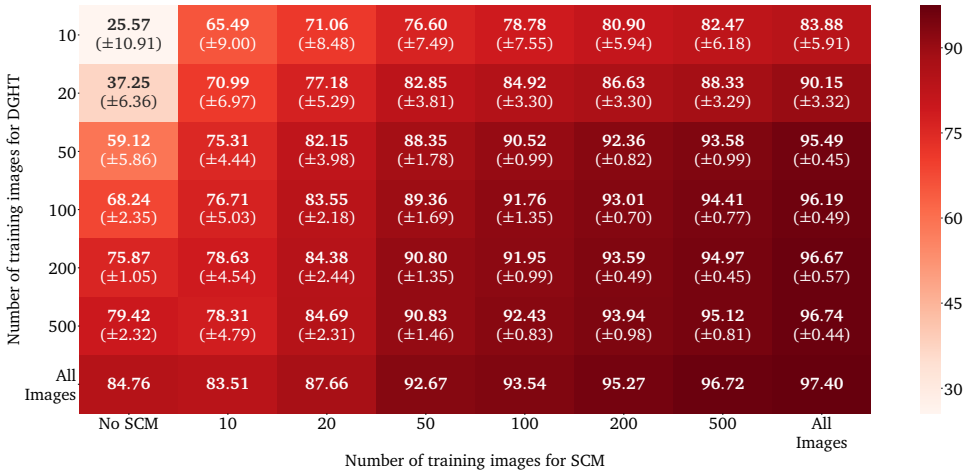


(c) P2E portal with error tolerance of 0.1 eye distance

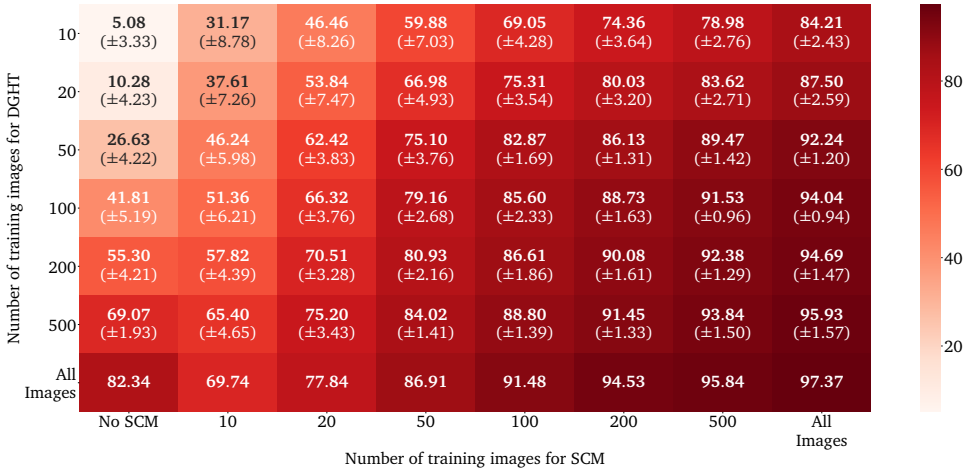
Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(d) P2L portal of 0.1 eye distance

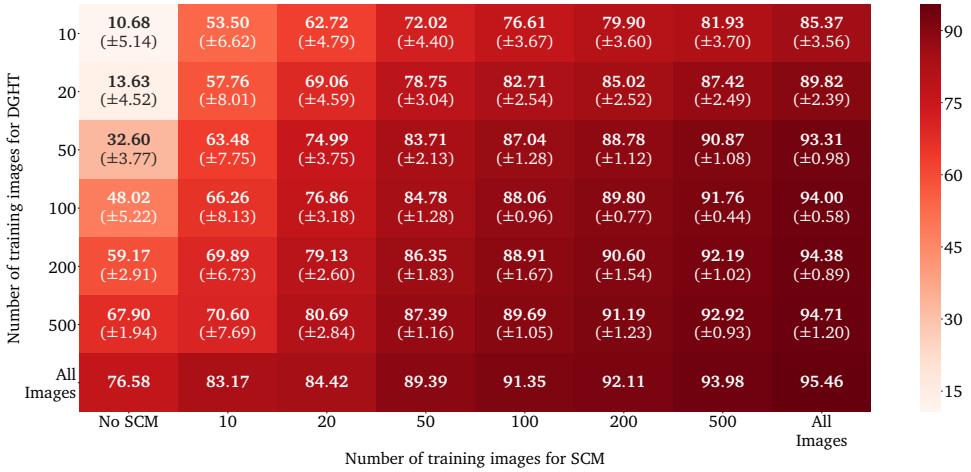


(e) P1E portal with error tolerance of 0.25 eye distance



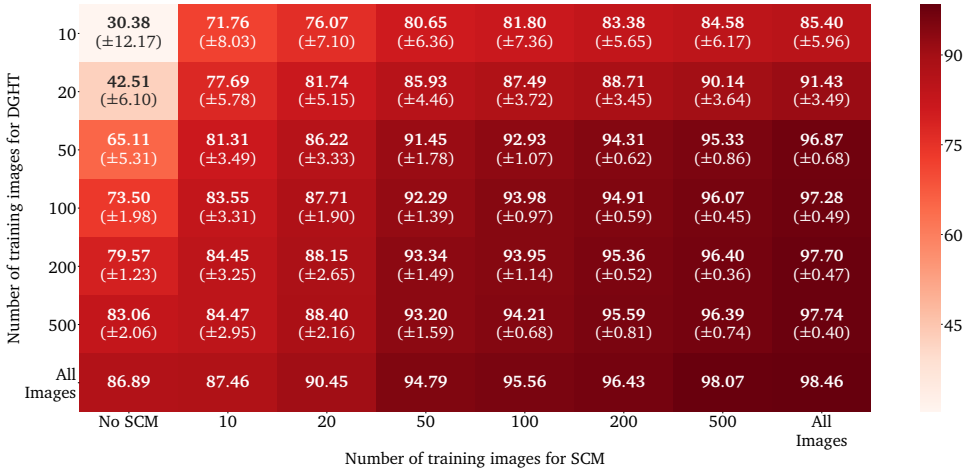
(f) P1L portal of 0.25 eye distance

Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



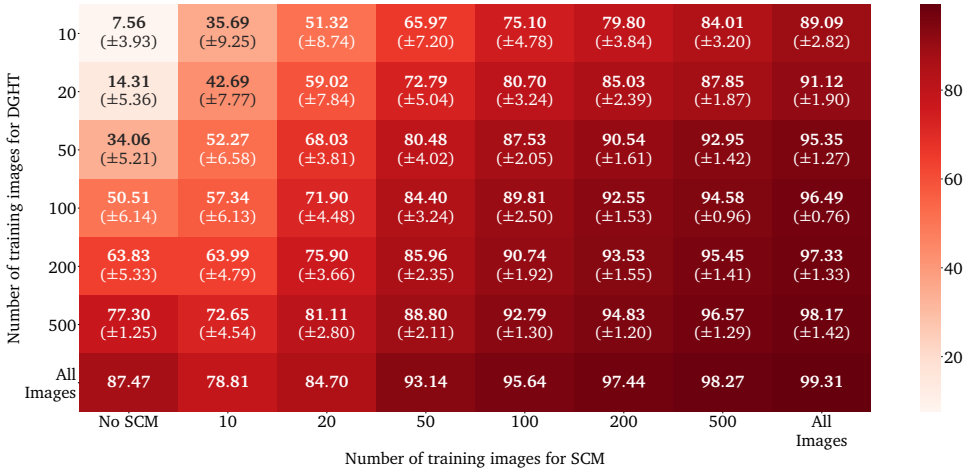
(g) P2E portal with error tolerance of 0.25 eye distance

(h) P2L portal of 0.25 eye distance

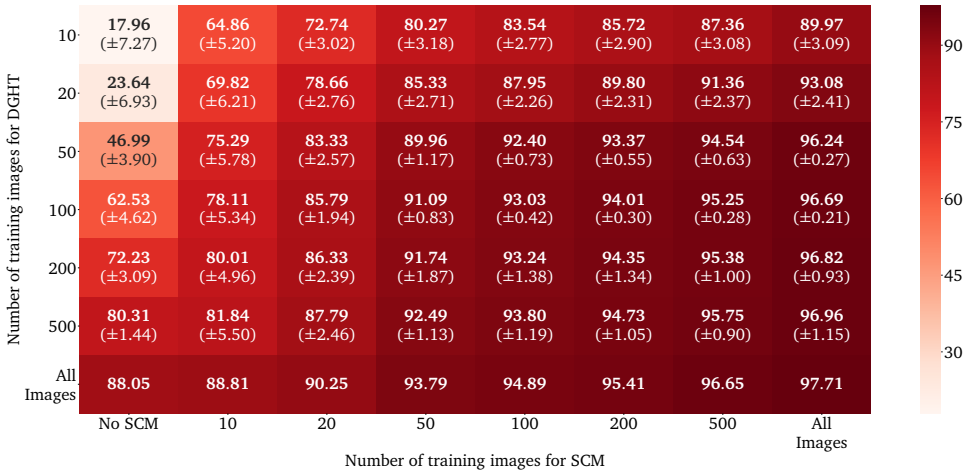


(i) P1E portal with error tolerance of 0.5 eye distance

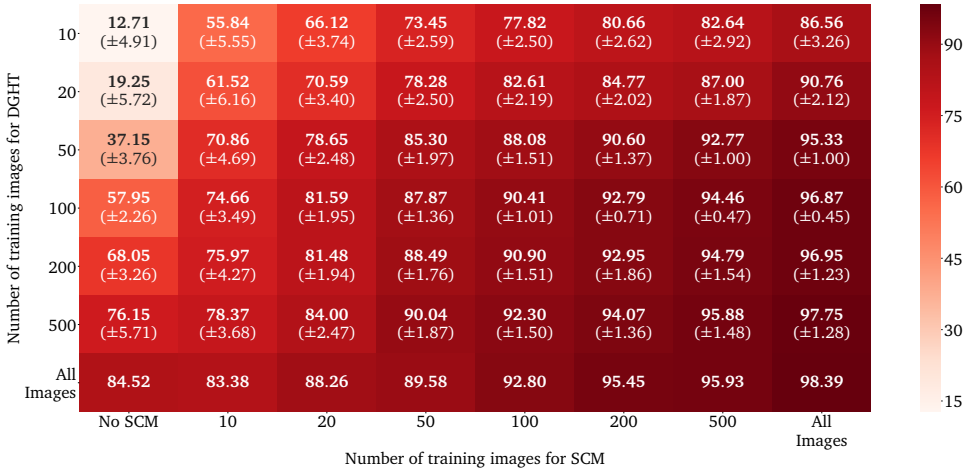
Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(j) P1L portal of 0.5 eye distance

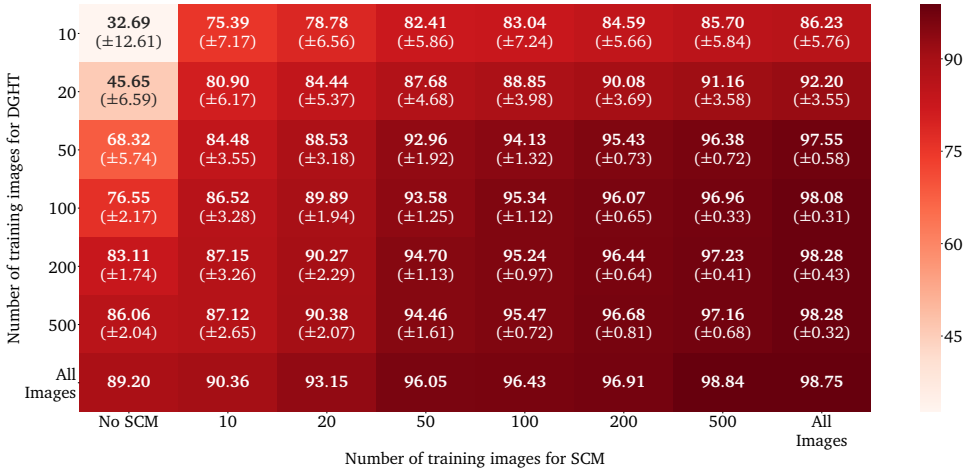


(k) P2E portal with error tolerance of 0.5 eye distance

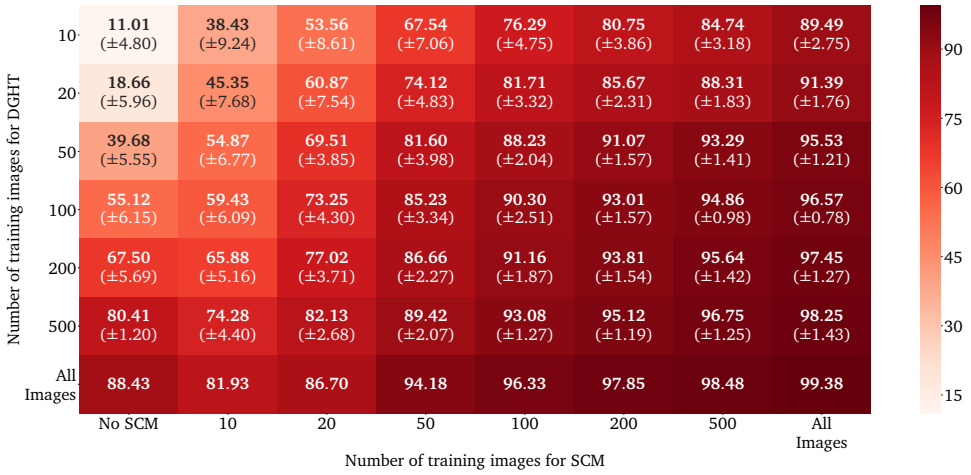


(l) P2L portal of 0.5 eye distance

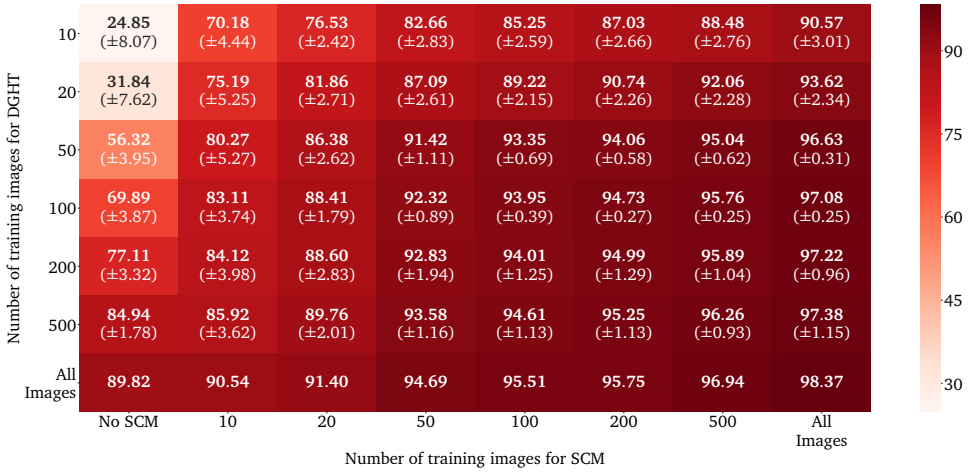
Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(m) P1E portal with error tolerance of 1.0 eye distance

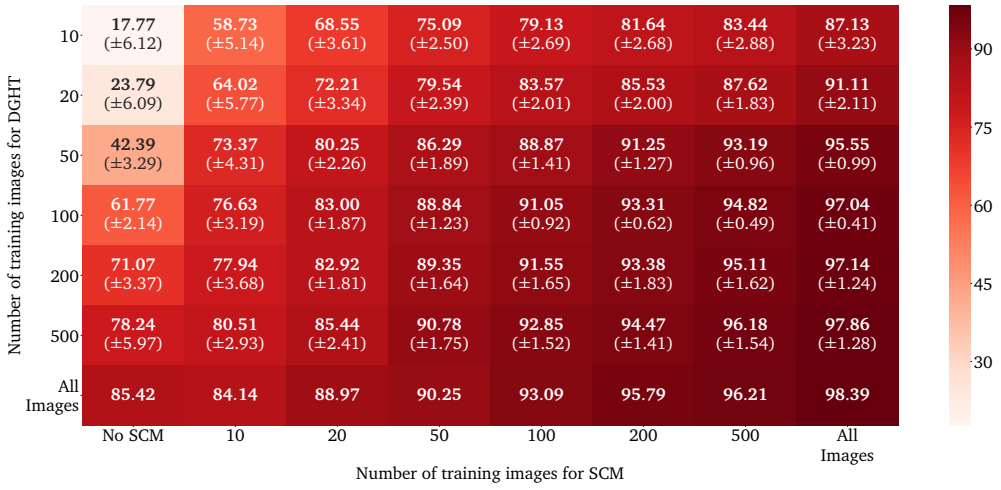


(n) P1L portal of 1.0 eye distance



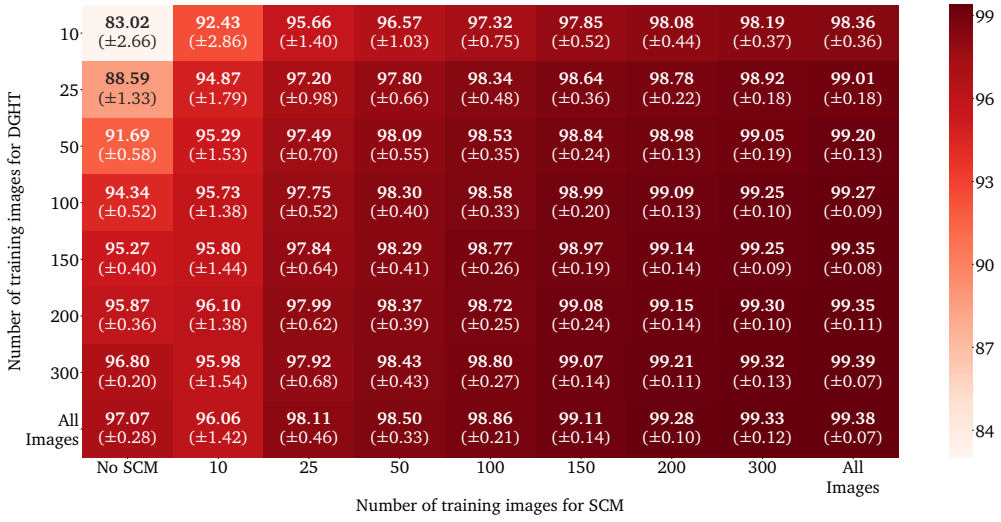
(o) P2E portal with error tolerance of 1.0 eye distance

Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images. Continued on next page.



(p) P2L portal of 1.0 eye distance

Figure A.9: Mean localization accuracy and standard deviation for Chokepoint Dataset for different error tolerances depending on the number of DGHT and SCM training images.

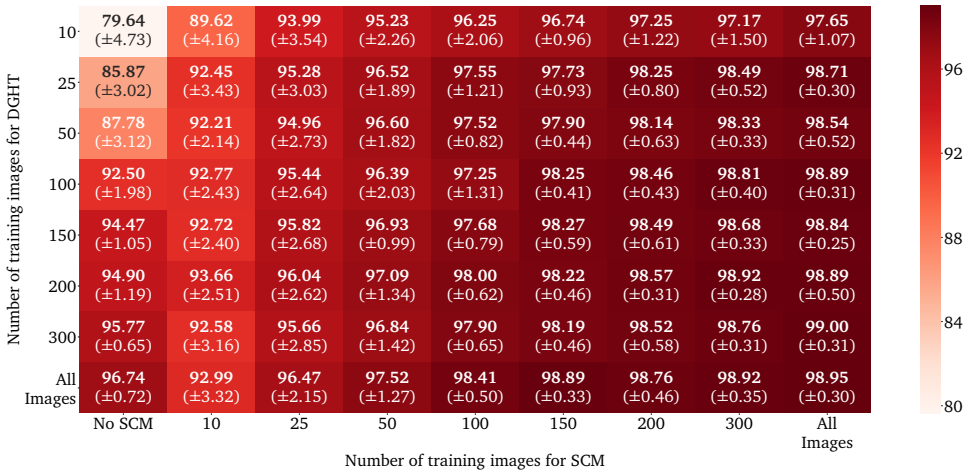


(a) Mean success rate

Figure A.10: Mean localization accuracy and standard deviation for RWTH Hand Database ($\Xi(\frac{6}{256})$) for different landmarks depending on the number of DGHT and SCM training images. Continued on next page.

Appendix

(b) Landmark 1D

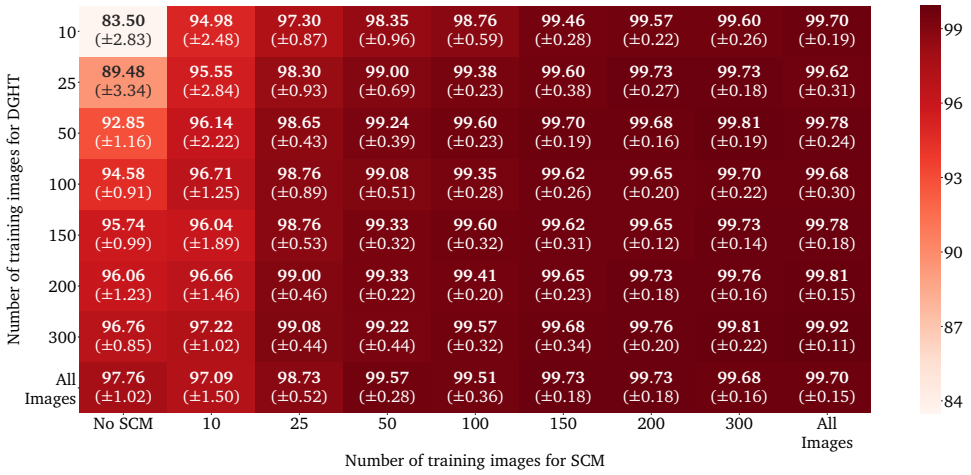


(c) Landmark 2D

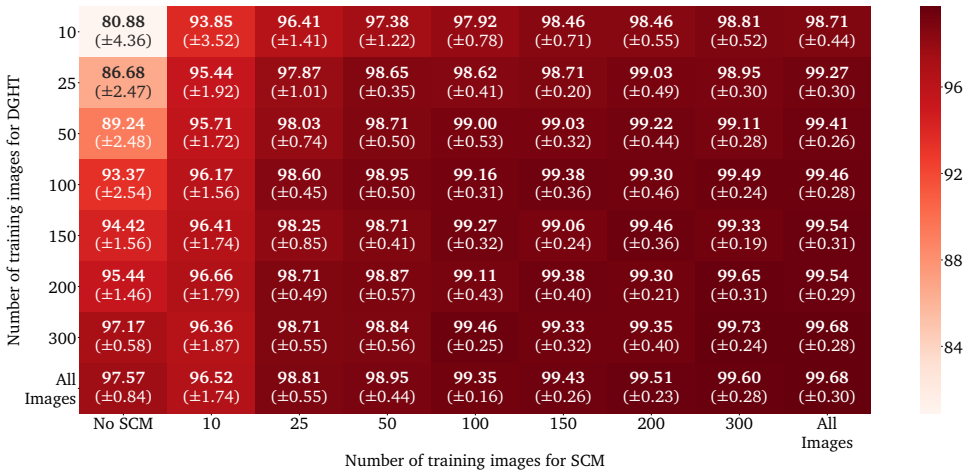
(d) Landmark 3D

Figure A.10: Mean localization accuracy and standard deviation for RWTH Hand Database ($\Xi(\frac{6}{256})$) for different landmarks depending on the number of DGHT and SCM training images. Continued on next page.

(e) Landmark 5D



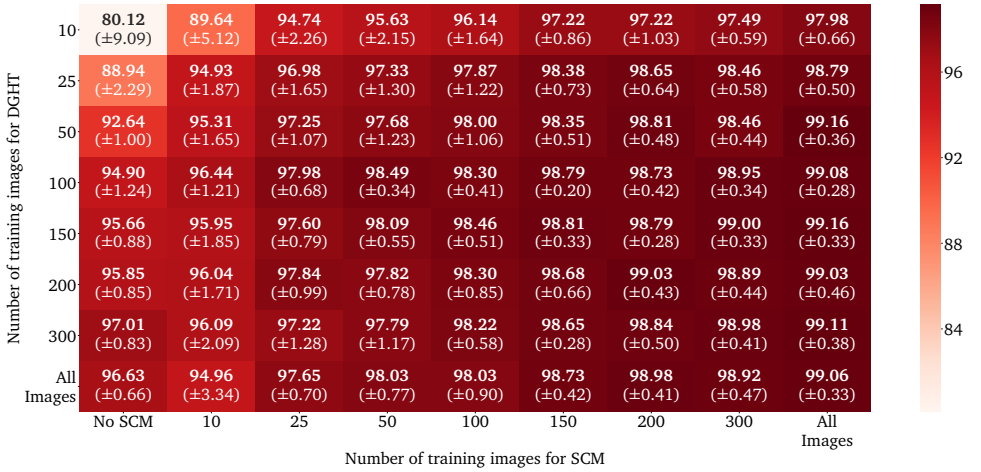
(f) Landmark 6D



(g) Landmark 7D

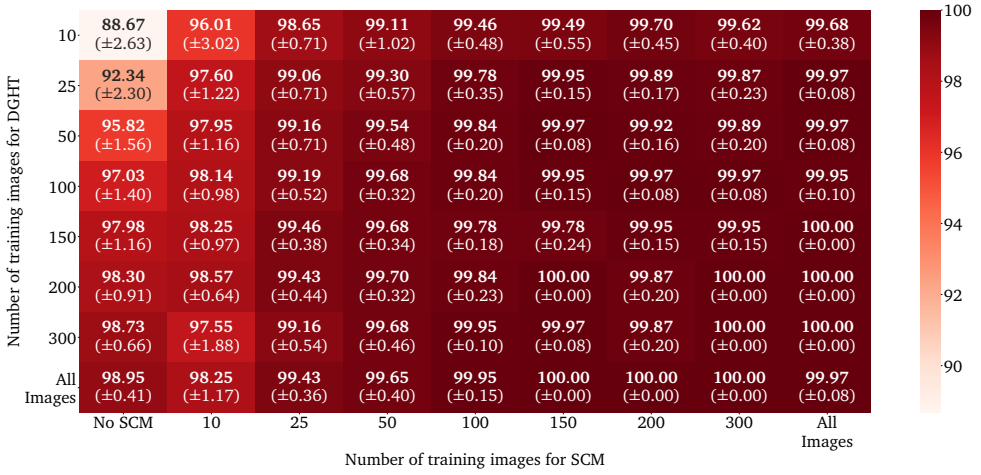
Figure A.10: Mean localization accuracy and standard deviation for RWTH Hand Database ($\Xi(\frac{6}{256})$) for different landmarks depending on the number of DGHT and SCM training images. Continued on next page.

Appendix



(h) Landmark 9D

(i) Landmark 10D

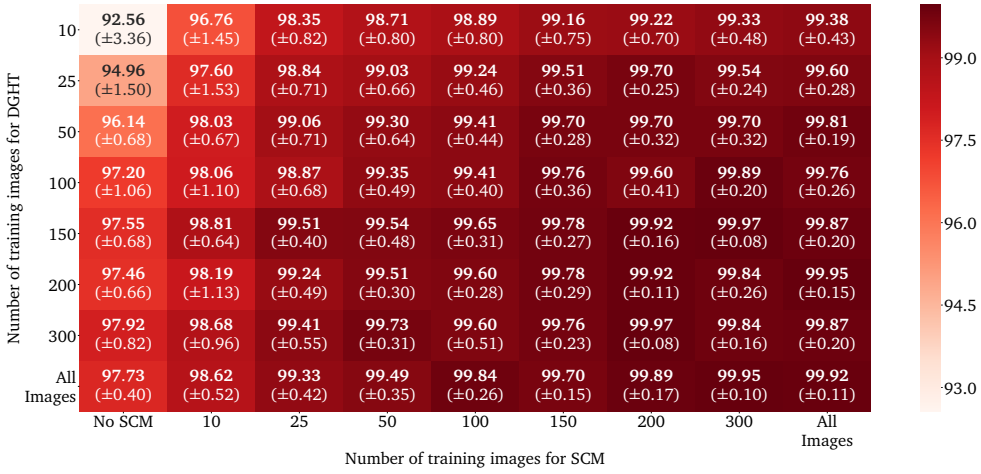


(j) Landmark 11D

Figure A.10: Mean localization accuracy and standard deviation for RWTH Hand Database ($\Xi(\frac{6}{256})$) for different landmarks depending on the number of DGHT and SCM training images. Continued on next page.

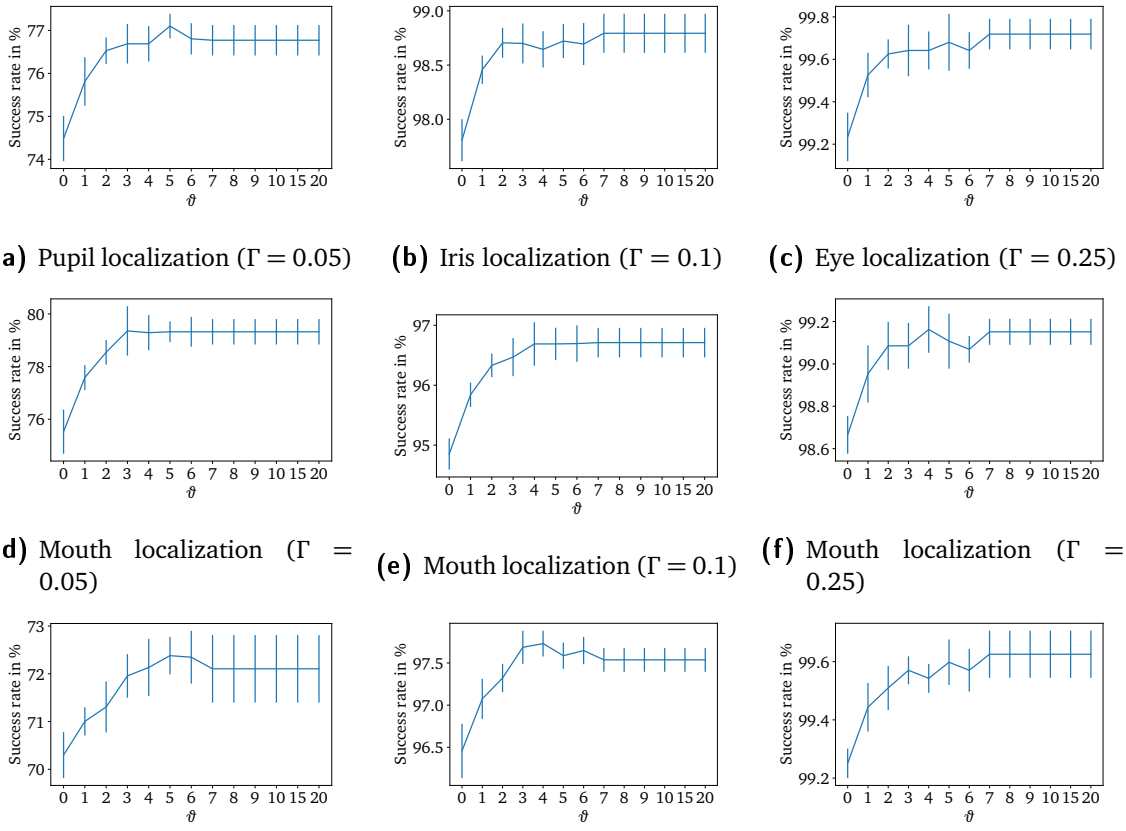
(k) Landmark 13D

(l) Landmark 14D



(m) Landmark 15D

Figure A.10: Mean localization accuracy and standard deviation for RWTH Hand Database ($\Xi(\frac{6}{256})$) for different landmarks depending on the number of DGHT and SCM training images.



(a) Pupil localization ($\Gamma = 0.05$) **(b)** Iris localization ($\Gamma = 0.1$) **(c)** Eye localization ($\Gamma = 0.25$)

(d) Mouth localization ($\Gamma = 0.05$) **(e)** Mouth localization ($\Gamma = 0.1$) **(f)** Mouth localization ($\Gamma = 0.25$)

(g) Nose localization ($\Gamma = 0.05$) **(h)** Nose localization ($\Gamma = 0.1$) **(i)** Nose localization ($\Gamma = 0.25$)

Figure A.11: Success rates for FERET Face Database and different neighborhood sizes (ϑ)

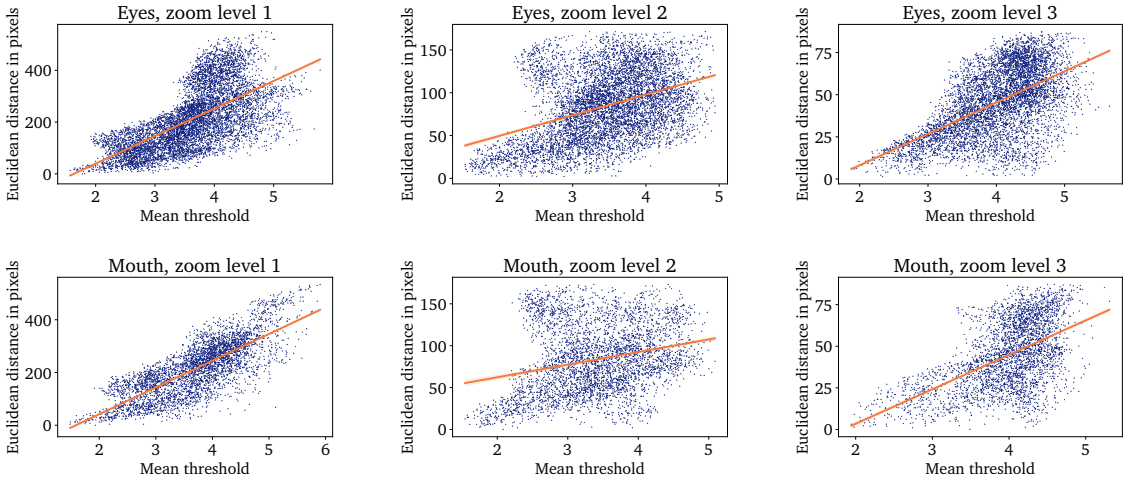


Figure A.12: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for FERET Face Database. Continued on next page.

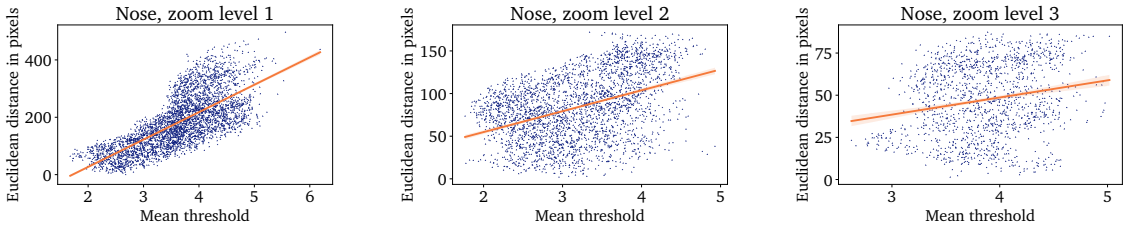


Figure A.12: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for FERET Face Database.

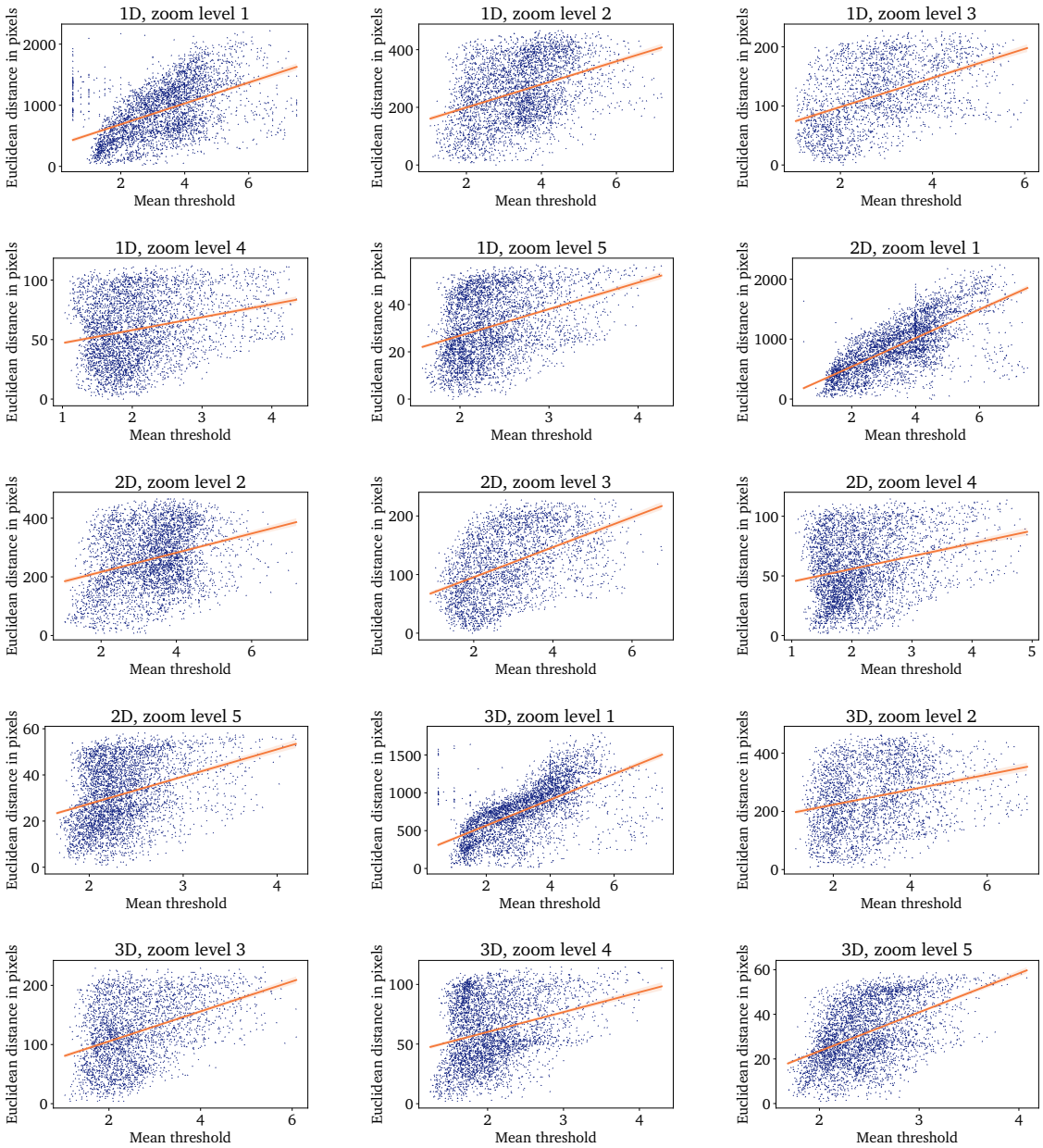


Figure A.13: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for RWTH Hand Database. Continued on next page.

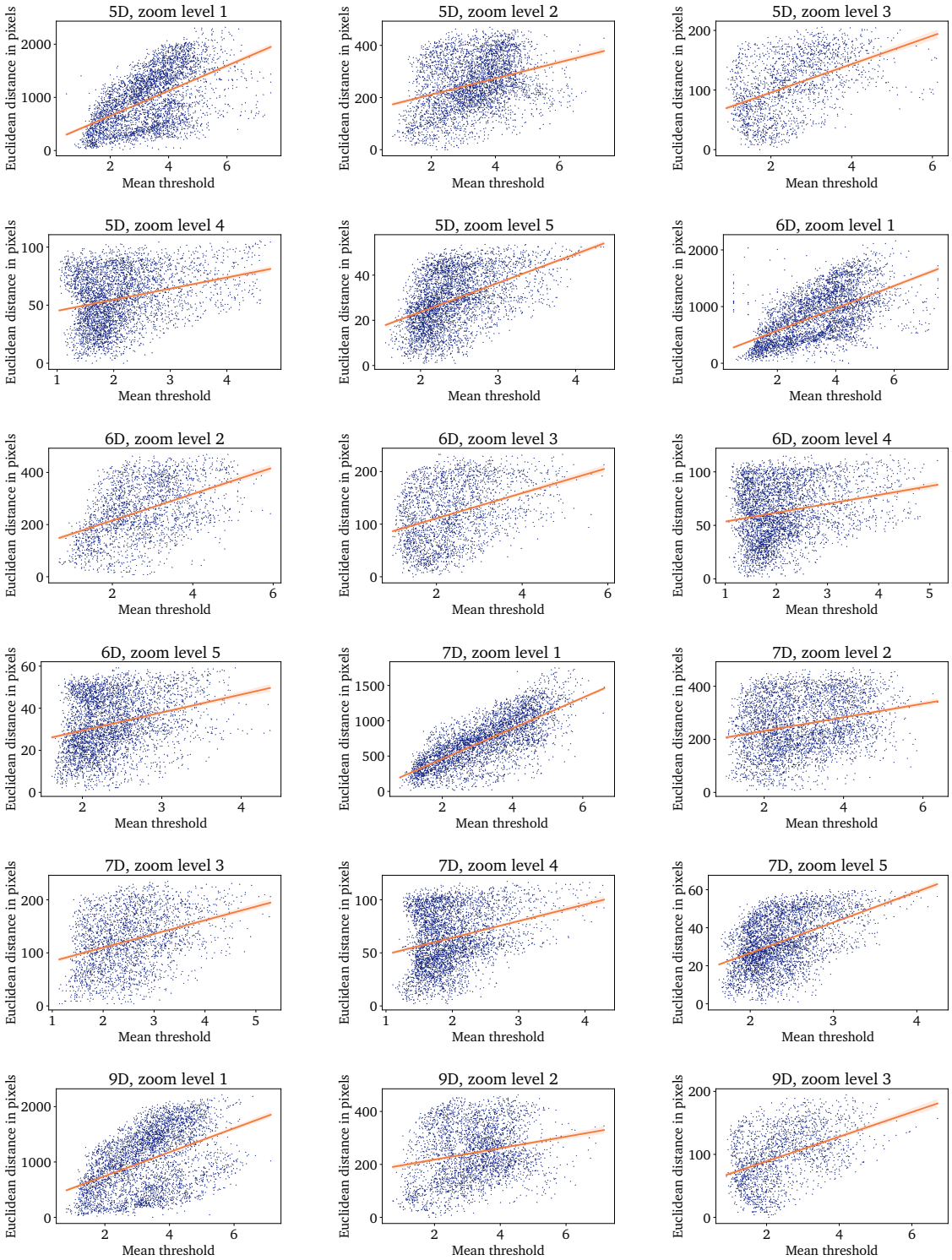


Figure A.13: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for RWTH Hand Database. Continued on next page.

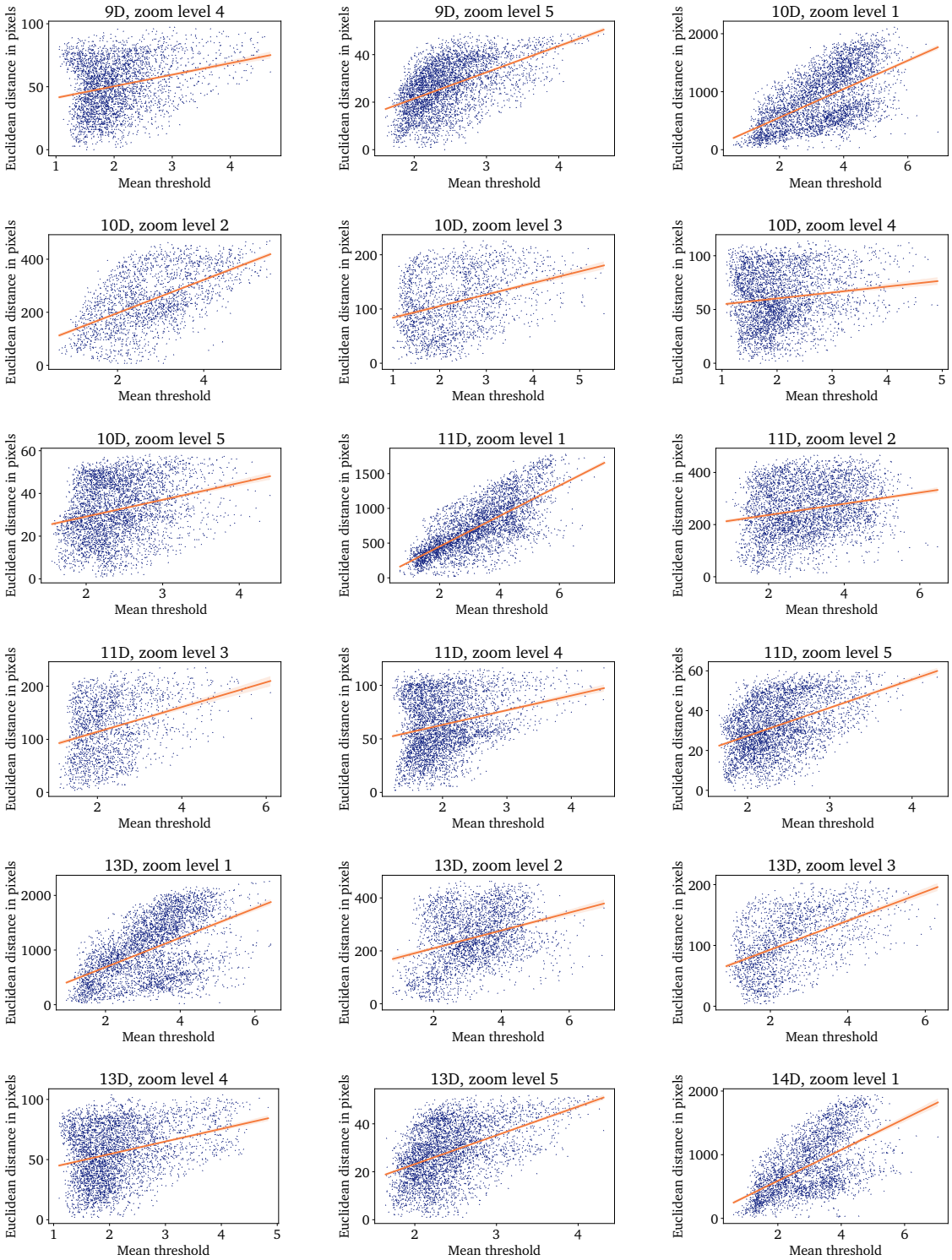


Figure A.13: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for RWTH Hand Database. Continued on next page.

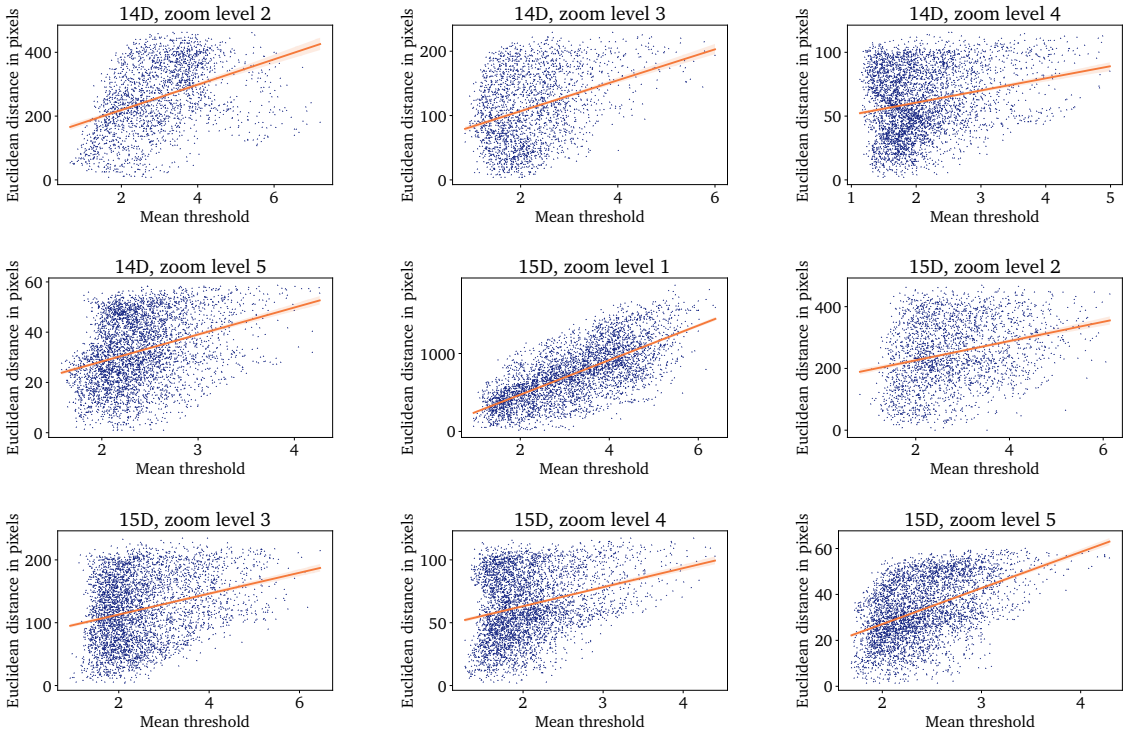


Figure A.13: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for RWTH Hand Database

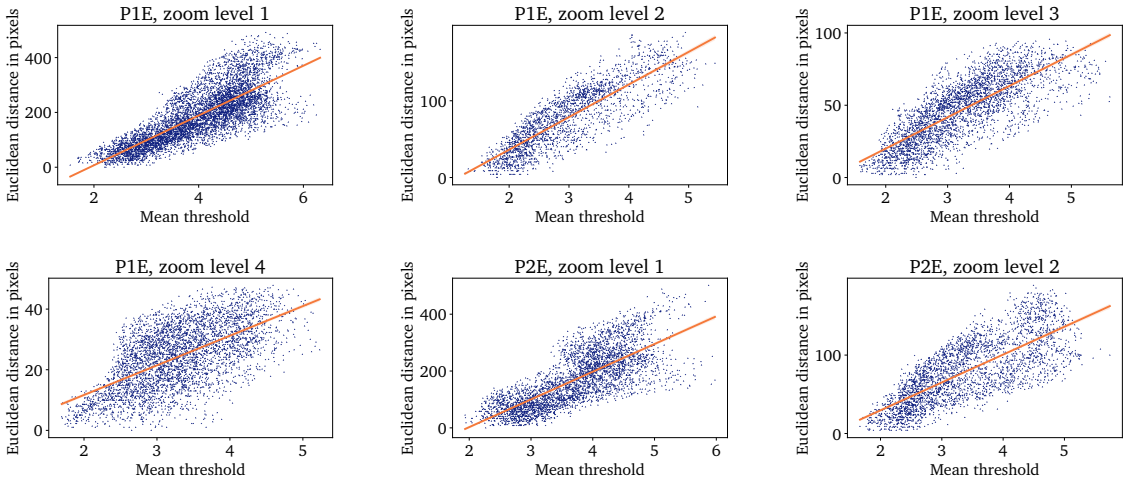


Figure A.14: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for Chokepoint Dataset. Continued on next page.

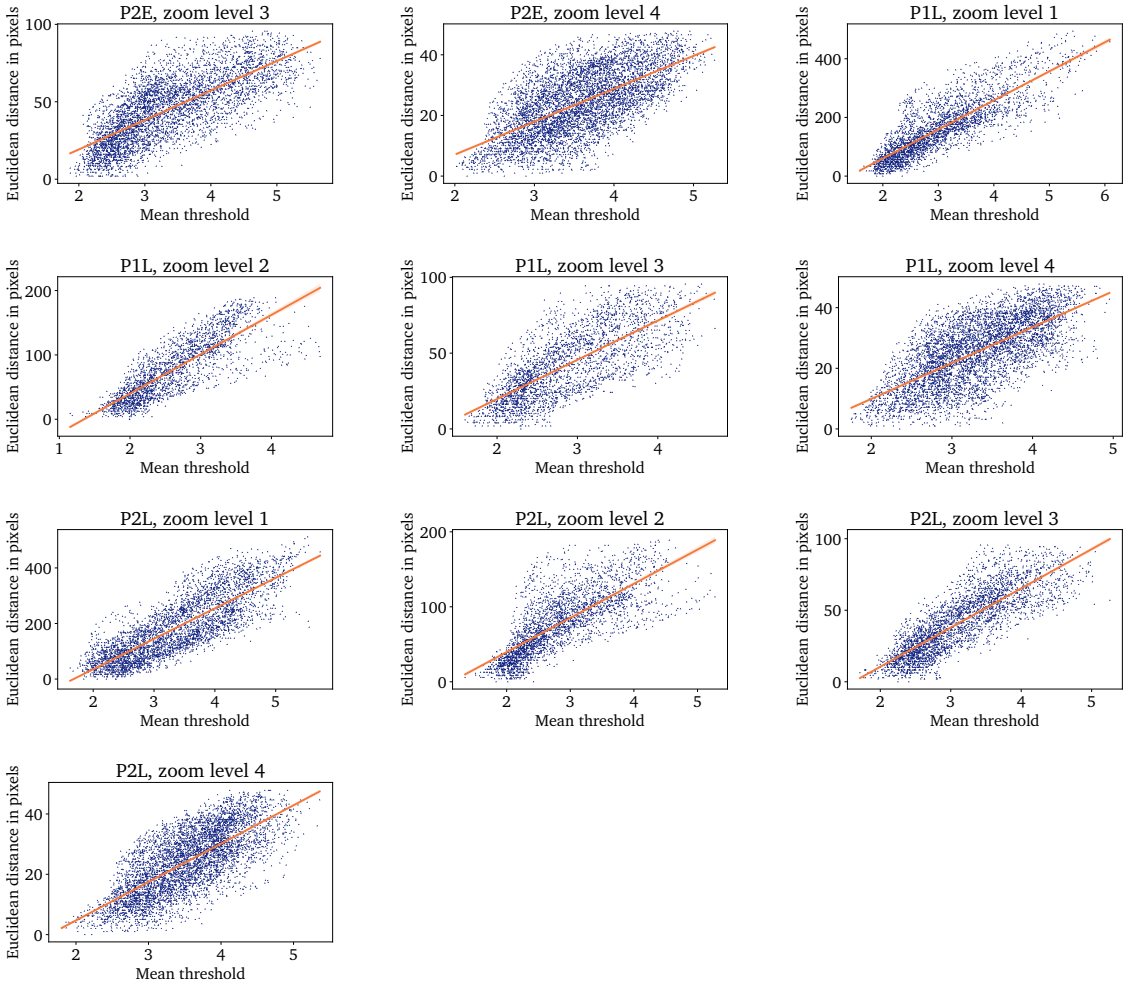


Figure A.14: Euclidean distance between model and reference point vs. average threshold for split function in the SCM for Chokepoint Dataset.