Deep Learning-based Head Orientation and Gender Estimation from Face Image

Thesis approved by the
Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)
to

Mohamed Selim

Date of Defense:	27.10.2022
Dean:	Prof. Dr. Jens Schmitt
Reviewer:	Prof. Dr. Didier Stricker
Reviewer:	Prof. Dr. Karsten Berns

DE-386

Abstract

Faces deliver invaluable information about people. Machine-based perception can be of a great benefit in extracting that underlying information in face images if the problem is properly modeled. Classical image processing algorithms may fail to handle the diverse data available today due to several challenges related to varying capturing locations, and conditions. Advanced machine learning methods and algorithms are now highly beneficial due to the rapid development of powerful hardware, enabling feasible advanced solutions based on data learning and summarization into powerful models. In this thesis, novel solutions are provided to the problems of head orientation estimation and gender prediction. Initially, classical machine learning algorithms were used to address head orientation estimation but were limited by their inability to handle large datasets and poor generalization. To overcome these challenges, a new highly accurate head pose dataset was acquired to tackle the identified problems. Novel trained deep neural networks have been exploited, that use the acquired data and provide novel architectures. The information about head pose is then represented in the network weights, thus, allowing predicting the head orientation angles given a new unseen face. The acquired dataset, named AutoPOSE opens the door for further studies in the field of computer vision and especially, face analysis. The problem of gender prediction has also been explored, but unlike humans who can easily identify gender from a face, computers face difficulties due to facial similarities. Therefore, hand-crafted features are not effective for generalization. To address this, a new deep learning method was developed and evaluated on multiple public datasets, with identified challenges in both still images and videos addressed. Finally, the effect of facial appearance changes due to head orientation variation has been investigated on gender prediction accuracy. A novel orientation-guided feature maps recalibration method is presented, that significantly increased the accuracy of gender prediction.

In conclusion, two problems have been addressed in this thesis, independently and joined together. Existing methods have been enhanced with intelligent pre-processing methods and new approaches have been introduced to tackle existing challenges, that arise from pose, illumination, and occlusion variations. The proposed methods have been extensively evaluated, showing that head orientation and gender prediction can be estimated with high accuracy using machine learning-based methods. Also, the evaluations showed that the use of head orientation information consistently improved the gender prediction accuracy. Scientific contributions have been presented, and the new acquired highly accurate dataset motivates the research community to push the state-of-the-art forward.

Acknowledgments

I am deeply grateful for the support of several persons, without whom this thesis would not have been possible. First, I would like to thank my PhD supervisor, Prof. Dr. Didier Stricker for his support, patience and guidance throughout my PhD journey. His feedback was indeed essential in completing this thesis. I would like to thank the members of my PhD committee, Prof. Dr. Karsten Berns, and Prof. Dr. Stefan Deßloch for their time and feedback. I would like to express my gratitude to Ms. Sabine Owens for her assistance with the administrative tasks during my PhD.

I would like to express my sincerest gratitude for my PhD mentor Dr. Alain Pagani. He guided me, with utmost dedication, technically and mentally throughout my PhD. His constructive feedback, continuous support, and encouragement has made the completion of this thesis possible. I am honoured that he was my PhD mentor.

I also would like to sincerely thank my friends and colleagues at DFKI, especially Chrsitiano Gava, Prof. Dr. Oliver Wassenmüller, Tewodros Amberbir Habtegebrial, and Stephan Krauß, for their humble support and guidance throughout my PhD. Their help was invaluable in pushing the boundaries of my technical knowledge. I would like to express my thanks Leivy Michelly Kaul for her invaluable support throughout my PhD journey. Her professionalism and positive attitude, were constant sources of encouragement. I am also thankful for the beautiful handmade PhD scarf she made me. I also give special thanks to Dr. Gerd Reis, Dr. Sarvenaz Salehi, Dr. Jason Rambach, and Jillam Diaz Barros, Maria Sanchez, Mathias Musahl, Rashed Al Koutayni, Ahmed Tawfik, Mahdi Chamseddine, Ramy Battrawy, for their support, and fruitful discussions. Special thanks goes to David Haase for his technical support. My thanks goes to Iuliia Brishtel for taking pictures on my defence day. I also would like to thank the AV secretary's team, Keonna Cunnigham and Simon Lüdicke, for facilitating the preparations for my PhD defence and the afterparty.

I would like to sincerely thank my friends, to whom I am grateful for their invaluable support and constant motivation. They were with me throughout this journey. Their unwavering support, and encouragement, helped me overcome the moments of doubt and uncertainty. My thanks goes to Bahaa El-Desouky, Dr. Mohammad Al-Naser, Ramy Hussein, and Winfried Kaul, for their continuous support. I am also thankful for Dr. Mohamed Khamis, Dr. Passant El-Agroudy, and Dr. Ahmed Mosaad, who traveled long distances to be with me on my defence.

I would like to thank my whole family in Egypt, and especially my mother, sister, my late grandmother (Teta) and father for their unconditional support, and love they provided me throughout my life. Their help and support contributed to the person I am today, and I am deeply grateful for each and everyone of them.

Finally, I would like to express my sincerest gratitude and appreciation to my wife, and my son. She accompanied and supported me through my PhD journey. My son showed me how to be positive, even through the hardest times. I am thankful and grateful to my family.

Mohamed Selim

April, 2023

Contents

1 Introduction 11
- 1.1 Motivation . . . 12
  - 1.1.1 Problem of Face Analysis in Computer Vision . . . 12
  - 1.1.2 Existing Challenges . . . 12
- 1.2 Problem Formulation and Approach . . . 13
- 1.3 Contributions . . . 15
- 1.4 Organization of the Thesis . . . 16
- 1.5 Publications . . . 17

2 Background 19
- 2.1 Deep Learning . . . 20
  - 2.1.1 Convolutional Neural Network . . . 20
  - 2.1.2 Residual Neural Network . . . 21
  - 2.1.3 Network Training . . . 22
- 2.2 Rigid Body Motion Representation in 3D Space . . . 22
  - 2.2.1 Rotatory Motion . . . 23
  - 2.2.2 Full Motion, Rotation and Translation . . . 25
- 2.3 Face Detection . . . 27
  - 2.3.1 Viola and Jones Face Detector . . . 27
  - 2.3.2 DPM-Based Face Detector . . . 28
  - 2.3.3 Face Detection with R-CNN . . . 29
  - 2.3.4 Face Detection algorithms used in the thesis . . . 30

3 Head Orientation Estimation with Relevance Vector Machines 31
- 3.1 Head Orientation Estimation - Input Modalities . . . 34
- 3.2 Datasets . . . 35
- 3.3 Learning Head Orientation with MVRVM . . . 39
  - 3.3.1 The Multi-Variate Relevance Vector Machine . . . 40
  - 3.3.2 Parameter Optimization . . . 42
  - 3.3.3 Results with single MVRVM . . . 45
  - 3.3.4 Discussion . . . 46
- 3.4 Tree of Relevance Vector Machines . . . 47
  - 3.4.1 Tree structure . . . 47
  - 3.4.2 Evaluation and results . . . 49
- 3.5 Conclusion . . . 51

4 Large-Scale Head Orientation and Eye Gaze Dataset 53
- 4.1 System Components 56
  - 4.1.1 Car Cabin Simulator 56
  - 4.1.2 Motion Capturing System - OptiTrack 58
  - 4.1.3 Acquisition Cameras and Camera Positioning 61
- 4.2 System Calibration 64
  - 4.2.1 OptiTrack Calibration 64
  - 4.2.2 Acquisition Cameras Calibration 66
- 4.3 Head Coordinate Frame 69
  - 4.3.1 Definition 70
  - 4.3.2 Head Calibration to OptiTrack 71
- 4.4 AutoPOSE dataset 72
- 4.5 Summary 78

5 Deep Learning for Head Orientation Estimation 81
- 5.1 Baseline Network Architecture 83
- 5.2 Network Training 84
- 5.3 Head Orientation Baseline Evaluation 85
- 5.4 Face Resolution and Network Depth 85
  - 5.4.1 Deep Neural Networks Architectures 85
  - 5.4.2 Evaluations 87
- 5.5 Summary 90

6 Gender Recognition 91
- 6.1 Deep Learning for Gender Recognition 95
- 6.2 Image Enhancement 96
- 6.3 Face Images Datasets 97
  - 6.3.1 Still Images Datasets 98
  - 6.3.2 Videos Datasets 100
- 6.4 Proposed GenderCNN 102
  - 6.4.1 GenderCNN for Still Images 102
  - 6.4.2 GenderCNN for Videos 104
- 6.5 Evaluation 105
- 6.6 Summary 108

7 Orientation-Guided Deep Gender Recognition 109
- 7.1 Introduction 110
- 7.2 Orientation-Guided Gender Prediction Models 113
  - 7.2.1 Orientation Adapter 113
  - 7.2.2 GenderCNN with Orientation Adapter 114
  - 7.2.3 ResNet-18 with Orientation Adapter 115
- 7.3 Evaluation 116
  - 7.3.1 Models Training 116

7.3.2 Models Evaluation 118
7.4 Conclusion 121

8 Conclusion 123
8.1 Summary of Thesis Contributions 124
8.2 Future Work 125

List of Figures 129

List of Tables 131

1 Introduction

This chapter motivates the thesis, gives an overview of the studied topics. The thesis outline, list of scientific contributions, and published work are provided.

Contents

1.1 Motivation **12**
1.2 Problem Formulation and Approach **13**
1.3 Contributions **15**
1.4 Organization of the Thesis **16**
1.5 Publications **17**

The objective of this thesis is the study of head orientation of faces as they appear in two dimensional images, and the gender prediction from face image. Most face analysis algorithms depend on detected faces in the images. The problem of face detection has been studied for decades. One of the early and famous methods is using Haar-cascades to detect faces in images [138]. Recently, the problem of face detection has been solved using deep learning methods, which proved to be robust under varying conditions of face appearance, head orientation, and lighting conditions. In this thesis, we use face detection methods to find faces in images, and focus on the head orientation problem. Besides, we study the problem of gender prediction from face image in still images and videos and the associated challenges under harsh conditions. Finally, we study the effect of orientation changes on the gender prediction performance.

1.1 Motivation

1.1.1 Problem of Face Analysis in Computer Vision

Vision-based human machine interaction implies human understanding. Faces deliver invaluable information about people. Artificial perception can be a very helpful tool in extracting that information, given the proper modeling of the underlying problem(s). The classical image processing algorithms fail to extract essential information accurately, because the model of the problem is difficult to describe. On the contrary, machine learning methods are capable of automatic modeling of the problem and therefore should be exploited for face image analysis and information extraction. The face analysis and understanding is a critical step in various applications, like for example, age estimation, and gender recognition. Moreover, in the field of autonomous driving, driver's attention monitoring is crucial for safety. The driver's attention monitoring can be implemented using different cues, one of them is the driver's head orientation (also sometimes referred to as head pose). Early head orientation estimation methods modeled the problem as a classification one. The aim was to classify the face appearance in the image for example into frontal, right-profile, and left-profile view. Nowadays, head pose estimation methods are expected to provide rotation angles and translation vectors that represent the orientation of the head in 3D space as a rigid object.

1.1.2 Existing Challenges

Analyzing faces in still images or videos can be challenging due to several factors. In still images, faces can be affected by variations in resolution, pose, illumination , and quality. The amount of information present in a face image can be affected if the face size is small, to the limit that the information can no longer be extracted even by humans. Large deviations in pose affect the appearance of the face in the image, thus, algorithms that are not robust against big changes in pose can fail in performing the underlying task. Consequently, variations in pose must be considered in the applications that analyze faces, in cases where pose invariance is required. Illumination changes is a challenge that can be visible in driver monitoring application. The driver's face as seen from a camera installed in the car, undergoes visible changes in illumination. For example, a driver's face might look very differently between shade and sunlight while driving. Moreover, the quality of the face image can affect the analysis systems' performance. A face image taken in a controlled environment, for example, taking picture of a sitting person, using a camera on a tripod, can result in high quality images, given proper illumination of the scene.

Nowadays, handheld cameras or handheld devices with cameras, for example smartphones, may result in blurry pictures, especially if the person and the camera are moving. Same effect of blur can be seen in videos, where the subject is moving and is being recorded by a person holding the imaging device and also moving. Stabilization systems can be of good use in such scenarios, for example lenses with built-in Optical Image Stabilization (OIS). Nevertheless, the effect of blur, can be reduced, but not eliminated. Moreover, the effect of image blur can be seen even in steady shots, in case of a low frame rate in videos, or long exposure time in still images. That being mentioned, the quality of the face image is worth considering when building applications that employ face analysis.

1.2 Problem Formulation and Approach

Most-vision based face analysis systems start with a *face detection* step. The problem of face detection has been thoroughly studied by the research community.

In this thesis, the objective is to extract, represent, store and retrieve useful information from the image of a face. In order to achieve that objective, various machine learning methods are employed in processing and extracting information from detected faces in images. In order to be able to extract information from the detected face, feature descriptors are applied to the detected faces. Different methods for feature descriptions were investigated to be able to describe the detected faces in images and use them with various machine learning techniques, ranging from classical methods to state-of-the-art deep learning methods like Convolutional Neural Networks.

The encoding or modeling of the useful information can be achieved using one of the following representations:

- **Classical feature patterns** where the face is processed and represented using feature descriptors. That could be done using for example but not limited to: Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG). [118, 114]
- **Classical machine learning** approaches, that can learn some representation of the face and endcode the information in some trained model. For example: the relevance vectors (RVs) of a Relevance Vector Machine (RVM). [116, 117]

- **Trained deep learning networks**, where the information of the dataset is represented in the weights of the network model itself. Convolutional Neural Networks is an example for such representation. [119, 114, 113, 36]

An image is a projection of the scene on the image plane, typically through a device such as a camera. Cameras can vary in sensor technology, resolution, frame rate, and many other factors. There are different sensor types, for example, greyscale cameras, color cameras, Infrared cameras, and depth cameras. The greyscale cameras (and sometimes referred to as monochrome cameras) use CMOS type sensor that captures the light intensity, producing a greyscale image of the scene. Color cameras are sometimes referred to as RGB cameras (referring to the Red, Green, and Blue channels of the camera). RGB cameras produce color images, mimicking the human vision. Infrared (IR) cameras use sensors that produce one channel images similar to the greyscale cameras, but the main difference to greyscale cameras is that they are sensitive to the IR light. Finally, depth cameras, for example Time Of Flight cameras, produces an image where each pixel represents the depth of that scene point to the camera sensor. Depth cameras usually project patterns of IR light, that is not visible to the human eye, but visible to the camera. Based on the time elapsed by the light to travel and hit the camera sensor, the camera can measure the depth of the scene point.

Each camera type has its advantages and disadvantages. Depending on the situation and application, one camera technology can be more suitable than another. In case of using a camera in a car for monitoring the driver, an IR camera is more suitable than the RGB camera or a greyscale camera because the RGB camera can be affected by light changes, for example, driving into and out of a tunnel during the day. The illumination changes are minimal in the IR camera, thus, enabling a robust tracking of the driver, which is of utmost importance in case of driver monitoring for autonomous driving.

Approach

Depending on the application in hand, a suitable camera modality should be used. In the case of driver monitoring, an IR camera would be more suitable than an RGB or monochrome camera. In other applications, that require texture analysis on the face, a monochrome or RGB camera would be suitable for such application. In order to solve the problem in hand, machine learning techniques are powerful tools that are capable of information extraction and task learning. The tasks could be a **classification** task like in *gender classification*, or a **regression** task like in *head orientation angles estimation*, or *aesthetics score prediction*.

Supervised learning The different works presented in this thesis use supervised learning to model the problems. For such machine learning method, labeled data is a crucial required input. Public datasets were used to properly evaluate and assess the proposed algorithms and pipelines. Moreover, in this thesis, a new large scale dataset that can be used for head pose and gaze estimation has been introduced. The dataset is publicly available for the scientific community [113]. In summary, labeled data is used to train different models using machine learning techniques to solve the application in hand.

1.3 Contributions

This thesis focuses on two specific problems, the head orientation estimation and the gender prediction from face image. Combining both problems, the effect of head orientation on the accuracy of gender prediction was studied. The technical contributions are summarized as follows:

- **Head Orientation Estimation:**
  - A fast and accurate head orientation estimation method. We provided an elegant learning-based pipeline that uses the appearance of the face as the input to learn the 3 rotation angles of the face. A tree-structure set of multi variate relevance vector machines (MVRVM) has been employed to learn the orientation angles of the face starting from a coarse estimation at the root node of the tree, to a fine estimation of the angles at the leaf nodes. The method proved to be light weight and accurate to estimate a rotation angles of the face [116, 117].
  - A large-scale, accurate driver head pose and eye gaze dataset, named AutoPOSE, with a deep learning-based baseline method for head rotation estimation. We used two possible camera positions in cars, the dashboard, and the center mirror. For the dashboard dataset, we used an infrared camera, which provided in total 1.1 million infrared(IR) images. At the center mirror, we used a Microsoft Kinect V2, which provided in total 315,000 images for each modality of the camera (RGB, Depth, IR). The groundtruth of the head pose was acquired using a sub-millimeter accurate motion capturing system [113, 34]. AutoPOSE Dataset is publicly available at `autopose.dfki.de`
  - A deep learning-based baseline model for estimating the head orientation using the face image on the AutoPOSE dataset [113]. Further more, in joint work, the face resolution and its effect on the head orientation estimation was studied [36].
- **Gender Prediction:**
  - A quality-aware deep learning-based method for estimating the gender under harsh conditions in still images and videos. A pipeline was proposed that employs classical image processing to estimate the quality of the face image, and based on the quality, choose a deep learning network that was trained to estimate the gender for faces with a specific quality measure. Combining classical methods and new deep learning Convolution Neural Networks (CNNs),

we propose a solution to estimate the gender of the person in blurred, bad illuminated images of the face [119].

- **Orientation-Guided Gender Prediction:**
  - A novel deep learning-based method was introduced for orientation-guided gender prediction from the face image. A new unit, called orientation adapter was introduced to encode the head orientation features. The encoded features were used as weights to recalibrate the feature maps of the convolutional layers in the image feature space. Orientation guidance consistently improved the gender prediction accuracy. [115]

1.4 Organization of the Thesis

This thesis is organized in the following way: **Chapter 2** presents background information about deep learning in general, along with information about Convolutional Neural Networks (CNN) and residual learning. It also presents the background information about rigid body pose representation in 3D space. Finally the face detection problem is discussed, and the algorithms used in this thesis are presented. It also presents background information

Chapter 3 addresses the head orientation estimation problem. After reviewing existing public datasets, different algorithms and approaches on modeling the head orientation problem, novel machine learning-based algorithms are presented to solve the head orientation problem.

Chapter 4 introduces the AutoPOSE dataset. Details of the acquisition system, and cameras and head calibration are thoroughly discussed. A car cabin simulator was used in the acquisition of the dataset. The two subsets of the dataset are presented where cameras were mounted at different positions. **Chapter 5** introduces the baseline deep learning model for the AutoPOSE dataset. Also, further study about the input face resolution for head orientation estimation is studied.

Chapter 6 is dedicated to the gender estimation problem. After reviewing existing methods and challenges in estimating gender in still images and videos, a novel deep learning-based pipeline is presented and evaluated on large datasets.

Chapter 7 merges the two main tracks in the thesis by presenting a novel deep learning-based method for orientation-guided gender prediction.

Finally, **Chapter 8** concludes the thesis and presents ideas for future work.

1.5 Publications

The work presented in this thesis has been accepted and presented in peer-reviewed conferences. In the following, we provide a list of the papers published during the time of the PhD:

1. Mohamed Selim, Stephan Krauß, Tewodros Amberbir Habtegebrial, Alain Pagani, and Didier Stricker.
   Deep Orientation-Guided Gender Recognition from Face Images. In *International Conference on Pattern Recognition Systems (ICPRS)*, 2022.

2. Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker.
   AutoPOSE: Large-Scale Automotive Driver Head Pose and Gaze Dataset with Deep Head Orientation Baseline. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

3. Mohamed Selim, Suraj Sundararajan, Alain Pagani, and Didier Stricker.
   Image Quality-Aware Deep Networks Ensemble for Efficient Gender Recognition in the Wild. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2018.

4. Mohamed Selim, Tewodros Amberbir Habtegebrial, and Didier Stricker.
   Facial Image Aesthetics Prediction with Visual and Deep CNN Features. In *Irish Machine Vision and Image Processing Conference Irish Machine Vision and Image Processing Conference (IMVIP)*, 2017.

5. Mohamed Selim, Alain Pagani, and Didier Stricker.
   Sparse-MVRVMs Tree for Fast and Accurate Head Pose Estimation in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2017.

6. Mohamed Selim, Alain Pagani, and Didier Stricker.
   Real-Time Head Pose Estimation Using Multi-Variate RVM on Faces in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015.

7. Mohamed Selim, Shekhar Raheja, and Didier Stricker.
   Real-time Human Age Estimation based on Facial Images using Uniform Local Binary Patterns. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2015.

8. Ahmet Firintepe, Mohamed Selim, Alain Pagani, and Didier Stricker.
   The More, the Merrier? A Study on In-Car IR-Based Head Pose Estimation . In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

9. Hartmut Feld, Bruno Mirbach, Jigyasa Katrolia, Mohamed Selim; Oliver Wasenmüller, and Didier Stricker.
   DFKI Cabin Simulator: A Test Platform for Visual In-Cabin Monitoring Functions. In *International Commercial Vehicle Technology Symposium CVT*, 2020.

10. Xiaohai Lin, Daniel Görges, Sebastian Schöffel, Johannes Schwank, Pascal Stahl, Achim Ebert, Mohamed Selim, and Didier Stricker.
    Eco-Driving Assistance Systems for Commercial Vehicles. In *International Commercial Vehicle Technology Symposium CVT*, 2016.

2 Background

This chapter provides background information about deep learning in general and specially Convolutional Neural Networks (CNNs), and residual networks. Rigid body motion representation in 3D space is presented with the conventions used in this thesis. Besides various face detection algorithms that were utilized through out the thesis are presented.

Contents

2.1 Deep Learning 20
2.2 Rigid Body Motion Representation in 3D Space 22
2.3 Face Detection 27

2.1 Deep Learning

In the recent years, deep learning has gained an increasing interest by the research community. Moreover deep learning have been gaining attention by the data analysis community, and in the field of natural language processing [106, 75, 127, 148]. This is mainly due to the huge success achieved by deep learning-based methods in solving various problems related to object classification, and object detection [157]. Deep learning helped in modeling and solving problems that are quite straight forward to solve by humans, but rather very difficult to model by computers. One of the advantages of deep learning is the ability of the deep neural network to learn, and model the data that would best achieve the desired output, in a supervised learning framework.

Deep learning approaches are generally based on the idea of constructing structures that learn complex representations by integrating simpler ones. The name deep learning originated from the depth of such multi-layer representations. Hence, also one layer in that representation is usually referred to as a network layer. The concept of deep learning originated from the artificial neural networks [110]. Supervised learning methods require sufficient amount of labeled data. Nowadays, the data sizes are booming due to the existence of large storage devices, and fast internet access, that helped in spreading of data all over the globe. Also the wide spread of smartphones, that are equipped with cameras, was a factor in the data explosion era. However, the crucial factor that helped in the booming of deep learning is the availability of computational capabilities. By combining the computational power, and the existence of large scale data, the deep neural networks were the proper tools for such combination.

2.1.1 Convolutional Neural Network

In 1989, Yann LeCun introduced the concept of Convolutional Neural Networks (CNNs) [67]. The proposed network performs convolution instead of the matrix multiplication in at least on of its layers. The CNNs were basically an exciting concept, but it was computationally very expensive to the existing computational capabilities available in 1989. The convolution operation of interest is the convolution of the input image with 2D kernels, which are usually referred to as feature map. When processing the convolutional filter with the input images, the filter can detect features in the image, that are most dominant to activating the filters.

In a typical design of CNNs, the convolution operations are followed by detector stage and a pooling stage, so that all the stages together form what is called a convolution layer. The detector stage is a non-linear activation function, for example sigmoid, or Rectified Linear Unit (ReLu), which introduces a non-linearity property to the learned model. The pooling layer's aim is to summarize the different outputs of convolution operator and combine them into a single output. Consequently, it propagates the output of the convolutional layer to the next layer. One example is max pooling, that selects the maximum value in a give window size, and propagate that max value, when compared to its neighbors, to the next layer. In other words, it propagates the most dominant features to the next layer.

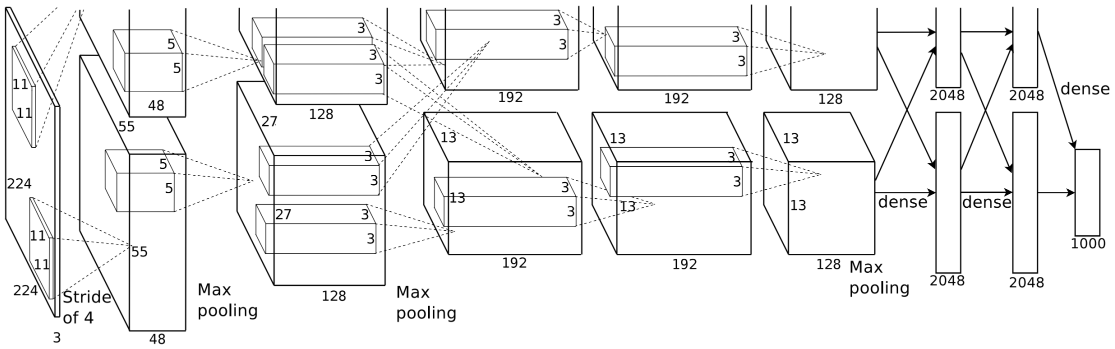


Figure 2.1: **AlexNet Architecture.** *The AlexNet CNN consists of convolutional layers, followed by max pooling layers, and at the a series of fully connected layers. The last layer contains the classification of the input image. Figure from [65].*

In 2012, Krizhevsky et. al. [65], introduced a CNN model, called AlexNet, that won the ImageNet object recognition challenge. It was the first CNN to win that challenge. Briefly, the ImageNet challenge is a 1000 class classification problem for about 1 million images dataset. The architecture of AlexNet is shown in figure 2.1. The model consists of several convolutional layers which are followed by max pooling layers. Near the end, the output of the layers is changed to a fully connected one dimensional layer. The final layer is a vector of size 1000 corresponding to 1000 classes. Most of the values shall be near zero except one which corresponds to the detected class.

2.1.2 Residual Neural Network

In 2016, He et. al [48] compared 20-layer CNN to a 56-layer CNN on CIFAR-10 dataset. They noticed that the training error of the 56-layer network was higher that the 20-layer network. The authors solved the issue by introducing a new neural network layer, called the Residual Block. This is implemented by adding a skip connection, as shown in figure 2.2. The skip connections between layers add the input of the layer to its output. Thus, allowing the input features to flow through the residual blocks which enables training deeper networks.

$$y = F(x) + x$$

, where F is the function that represents the whole learning of the block, and y is the output of the block that is passed to the following layers in the network.

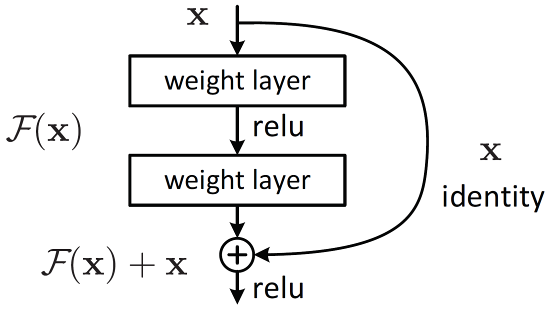


Figure 2.2: **Residual Learning.** *Residual Networks Building Block. Figure from [48].*

2.1.3 Network Training

Neural network training generally has the purpose of approximating a function f that can map the input data x to the output data y, where $y = f(x)$. In general, the training aims to learn a set of internal network weighting parameters such that the function f best approximates the input x to its corresponding output. In order to achieve that, training data is required to pass it through the network, and compare the output using a cost function. Followed by a forward pass, the weights are back-propagated through the network to modify the weights, such that the cost function is minimized, and the new weights would approximate the network output more to the desired output. In an ideal case, the network shall see training data that covers all the possible inputs that would be required to predict later, however, this is sometimes simply not available. A solution to increase the size of the training set, once can use data augmentation. Data augmentation aims at increasing the amount of the training data, so that the CNN would better approximate the learned function and adjust the internal weights. Some useful methods for data augmentation usually includes: applying some geometrical operations like for example, scaling, rotating, and cropping the image, while of course associating the same desired output value. By augmenting the data, the network learns to provide the same output for different possible forms of the input image. Consequently reaching network generalization and avoiding the problem of over-fitting.

2.2 Rigid Body Motion Representation in 3D Space

In this section, the background knowledge required for understanding the representation of a rigid object in some coordinate frame are discussed. The information is presented along with the transformations between coordinate frames. Such knowledge would help in understanding the representation of the head as a rigid object in 3D space. The head orientation is represented in 3D space by a rotation matrix R on and a translation vector T. First, the rotation part is discussed followed by the translation part.

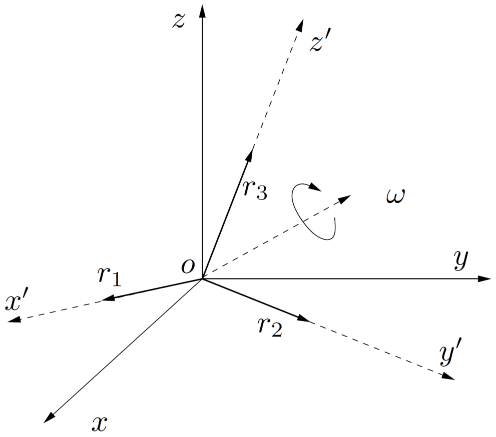


Figure 2.3: **Rotation of a rigid body about a fixed point.** *The solid coordinate frame W (World) is* ***fixed*** *and dashed coordinate frame C is attached to the rotating rigid body. Figure from [76].*

2.2.1 Rotatory Motion

Following the notations and definitions from [76], we describe our definitions as follows. The position of any point p in a coordinate frame can be described using a 3-dimensional vector $v = [x_p, y_p, z_p]^T$, where $x_p, y_p, z_p \in \mathbb{R}$. Considering an object moving in front of the camera, describing the object at any time instance, one shall describe every particle of the object using vectors described in the camera coordinate frame. However, for rigid objects, it is sufficient to describe one specific point on the object, and along with it, three coordinate axes attached to the object at that specific point. In other words, one can say, a coordinate frame is attached to the object. It is important to note that, what describes the *rigidity* property to a given rigid object, is that the relative distance between any two points on the object remains constant at any point in time. In other words, the individual points on the object cannot translate relative to each other. However, they can rotate relative to each other, but that rotation must be applied collectively to all points so that the relative distance between two points remains constant before and after the rotation.

Typically, attaching a coordinate frame to an object, one needs to define a specific point o, named origin, and three mutually orthogonal vectors e_1, e_2, e_3, all have the base at the origin o. The vectors are ordered so that they satisfy the right-hand rule, where $e_1 \times e_2 = e_3$. The attached coordinate frame is usually named the object's coordinate frame, named C, and formed by the three axis x', y', z' .

Rotation Matrix

Assuming we have attached a coordinate frame to some object, and that object can be rotating about a fixed point $o \in \mathbb{E}^3$, where $\mathbb{E}^3$ is the standard euclidean 3 dimensional space. As shown in figure 2.3,o is the origin of some coordinate frame W, formed by the principle axis x, y, z. The orientation of the rigid object is needed to be described relative to the coordinate frame W. Since the coordinate frame C, is attached to the object, so the orientation of the object in frame W is described using the orientation of the frame C in W. As shown in the figure, the three mutually orthogonal unit vectors r_1, r_2, r_3 are described in coordinate frame W, and they lie along the three principle axis x', y', z' of the coordinate frame C respectively. The orientation of the object in coordinate frame W is completely described using the following 3×3 matrix:

$$R_{wc} = [r_1, r_2, r_3] \in \mathbb{R}^{3\times3}$$

where r_1, r_2, r_3 are placed in the matrix as 3 columns.

This can be written in a matrix as:

$$R_{wc}^T R_{wc} = R_{wc} R_{wc}^T = I$$

A matrix that satisfies the above property is named an orthogonal matrix. Thus, the inverse of the matrix is also its transpose, $R_{wc}^{-1} = R_{wc}^T$. As the three vectors form a right hand rule frame, the determinant of the matrix must be $+1$. This matrix is also called a rotation matrix. Any possible rotation matrix $R_{wc} \in SO(3)$ represents a possible orientation of the object it is attached to about the point o, the point the object is rotating about. Moreover, it represents the *coordinate transformation* from coordinate frame C to coordinate frame W. By applying matrix-vector multiplication as follows:

$$P_w = R_{wc} P_c$$

the matrix R_{wc} is the matrix that transforms the coordinate P_c of the point p to the coordinates P_w, which represents the same point p but the coordinate frame W. By applying the inverse or the transpose operations to the matrix R_{wc}, we get a rotation matrix that transforms the coordinates of any point the other way around, from the coordinate frame W to the coordinate frame C.

$$P_c = R_{wc}^{-1} P_w = R_{wc}^T P_w = R_{cw} P_w$$

The convention of reading the subscripts from right to left is applied. Please note the inversion of the letters in the subscript of last term of the previous equation, R_{cw}. It is read as follows, R_{cw} is the rotation matrix that transforms the coordinates of a given point in the coordinate frame W to coordinates of the same point in the coordinate frame C. It is possible to read the same term in a shorter form as follows, R_{cw} is the rotation matrix that goes from W to C.

Quaternions

Rotation matrices represent the orientation of a rigid body in some coordinate frame, and it also represents the coordinate system transformation between coordinate frames. It is also common to represent the same orientation or transformation using another form, which is the quaternion representation. Quaternions can be used to represent the rotation matrix in a more compact form. Given a unit quaternion $q_{wc} = [q_w, q_x, q_y, q_z]^T$ along with $||q_{wc}|| = 1$. The transformation from q_{wc} to the rotation matrix R_{wc}, is given by the equation:

$$R_{wc} = \text{rot}(\mathbf{q_{wc}}) = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}$$

2.2.2 Full Motion, Rotation and Translation

As presented, the rotation matrix represents the orientation of a rigid body in 3D space. In order to generally describe the pose of a rigid body, the position of the rigid body can be described using a translation vector. As shown in figure 2.4, a coordinate frame is attached to some object at the point o, which is referred to as the origin of the object. The origin of the object is described in the coordinate frame W by the translation vector T_{wc}. The translation vector T_{wc} is read as follows: it is the position of the rigid body -described by coordinate frame C- in the coordinate frame W. The rotation matrix R_{wc} is the rotation matrix representing the rotational motion of the rigid body -described by coordinate frame C- in the coordinate frame W. In short, it is the rotation from C to W. In order to fully represent the orientation of a rigid body -described by coordinate frame C-, the full rigid motion $g_{wc} = (R_{wc}, T_{wc})$.

Given a point p in 3D space, the position of the point is described by the vector X_w in the coordinate frame W, and by the vector X_c in the coordinate frame C. The vector X_w can be seen from the figure as the sum of is the translation vector T_{wc}, and the vector X_c, but expressed relative to the coordinate frame W. Since the vector X_c is representing the point p in coordinate frame C, so to describe it in the coordinate frame W, it becomes $R_{wc}X_c$, where R_{wc}, is the rotation matrix that transforms the coordinates of the vectors in coordinate frame C to the vectors that represent the same points but in coordinate frame W. Consequently the coordinates of the vector X_w are given by the equation:

$$X_w = R_{wc} X_c + T_{wc} \tag{2.1}$$

Using the rigid body motion notation, a compact form of the transformation between the coordinate frames can be written as:

$$X_w = g_{wc}(X_c), \quad \text{where} \quad g = (R, t) \mid R \in SO(3), T \in \mathbb{R}^3$$

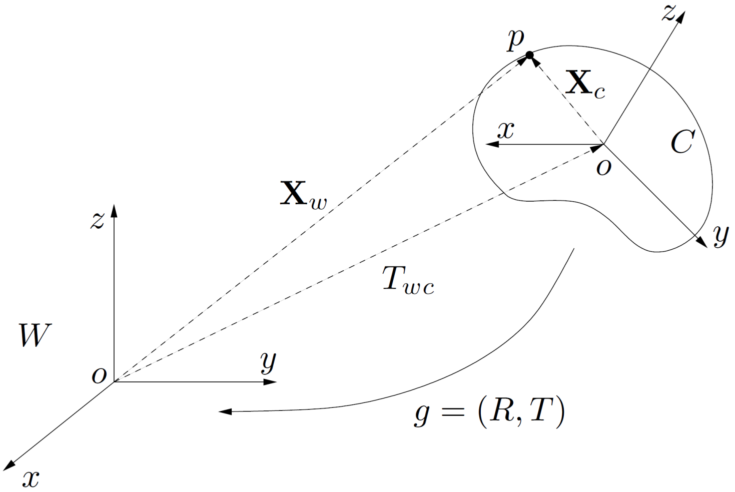


Figure 2.4: **Rigid body motion** *A rigid body motion between a moving frame C and a world frame W. Figure from [76]*

It can be noticed from equation 2.1, that the transformation is not linear, but rather affine. Linear transformations relate two vectors u, v if $u = Av$ for some matrix A. If $u = Av + b$ for some matrix A, and some vector b, then the transformation is affine. In order to describe the rigid motion g as a linear transformations, the *homogeneous coordinates* representation must be used.

Appending 1 to the coordinates of the vector $X = [X_1, X_2, X_3] \in \mathbb{R}^3$ gives a vector in $\mathbb{R}^4$ denoted by $\bar{X}$, written as:

$$\bar{X} = \begin{bmatrix} X \\ 1 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \in \mathbb{R}^4$$

In order to re-write the equation 2.1 in a linear form, using the new notation, the equation is written as follows:

$$\bar{X}_w = \begin{bmatrix} X_w \\ 1 \end{bmatrix} = \begin{bmatrix} R_{wc} & T_{wc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ 1 \end{bmatrix} = \bar{g}_{wc}\bar{X}_c \tag{2.2}$$

where the matrix $\bar{g}_{wc}$ is named the homogeneous representation of the rigid motion $g_{wc} = (R_{wc}, T_{wc})$. Generally, the homogeneous represenation of any rigid motion $g = (R, T)$ is:

$$\bar{g} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{2.3}$$

2.3 Face Detection

According to the comprehensive survey by Zafeiriou [153], face detection is one of the most studied topics in computer vision. This is not only due to the fact that it is a very challenging task on its own, but also since it is a crucial step in many applications related to face image understanding. Usually, it is the very first step in any pipeline or method that studies the face. The first algorithms for face detection have been proposed more than two decades ago. Some example uses of face detection is face recognition [156, 56], face tracking [59], face shape reconstruction [11], organizing photo albums according to persons [61], or in medical human machine interaction, where a robot is powered with face tracking for use in autism therapy [13].

Face detection is an automatic process of finding all the candidate faces in an image and localizing them using bounding boxes. In real time detection, it can be a challenging task due to many variations in the image because of scale, location, orientation, poses, expression and lightning conditions [151]. Occlusion is another big challenge to handle for this task for real time detection. Orientation and pose related issues still remains an open issue to be taken care of as these are very natural issues and very hard to resolve. Ethnicity related variations such as skin color, hair color and face size are another set of problems that complicate face detection due to cultural influences on a face. Over the last decade many promising face detection methods have been proposed, most of them are either based on template based matching, feature extraction based approaches, part based approaches, neural network based or an ensemble of aforementioned different approaches[50, 151].

The success of any method is very tightly bound with the type of features they work on. For cases of constrained environments, hand-crafted features based on Local Binary Patterns (LBP) [91] and Local Phase Quantisation (LPQ) [92] work decently well, but they don't perform equally good in unconstrained environments where face images contain large inter-personal variations like skin color and hair color. It still remains an unanswered question of how to find optimal face features that could work even in unconstrained setup. In the last few years, Convolutional Neural Networks(CNN) have been very successful in computing the optimal features for solving the problem. CNNs have been widely explored by combining them with other template based or 3D model based approaches for this task. In the following, we study and present various face detection methods.

2.3.1 Viola and Jones Face Detector

Face Detection has been a popular area of interest and extensively researched in literature of computer vision. Major works in this domain can be classified into three categories as shown in figure 2.5. The seminal work from Viola-Jones [138] sets a landmark in gaining

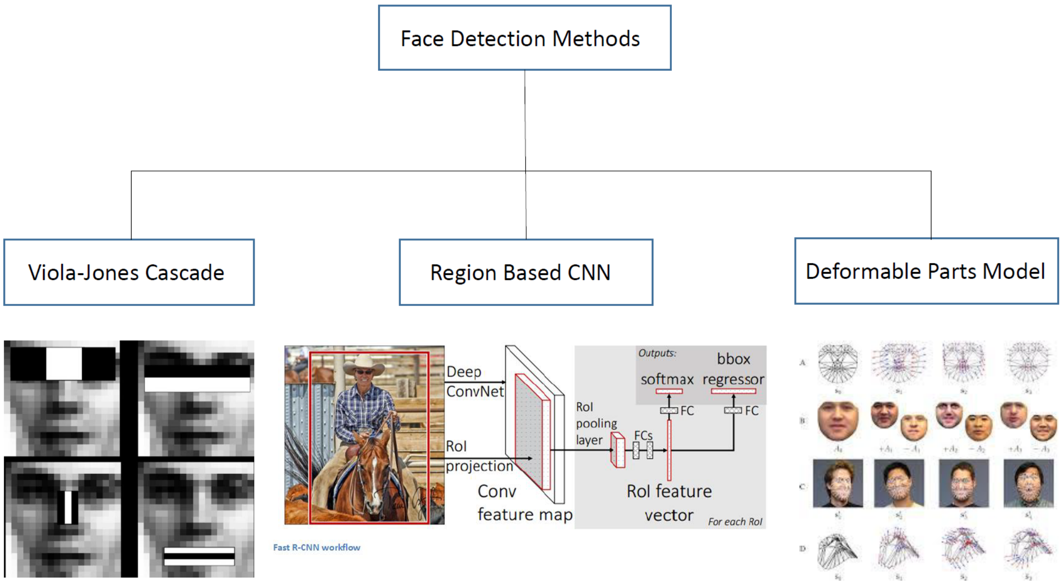


Figure 2.5: **Face Detection Methods.** *A proposed categorization of face detection methods based on the features used.*

satisfactory practical performance in face detection in real time. This framework was the first method to apply rectangular Haar-Like features along with cascaded Adaboost classifier to achieve face detection in real time with optimum performance. However, it came along with several problems as well. First, the feature size is too large and secondly, it works well for images with frontal faces but fails in cases of occluded and rotated faces i.e faces in the wild.

2.3.2 DPM-Based Face Detector

A lot of recent work has been focused on using parts model as well, also known as Deformable Parts Model (DPM). In 2010, Deformable Parts Model (DPM) gained popularity in face detection [35]. Ghiasi and Fowles [40] have successfully shown that using hierarchical DPMs gives good face detection results even in the presence of occlusion. Mathias et al. [80] show that both DPM models along with rigid template based detectors similar to Viola-Jones framework are potentially very good alternatives for solving this problem and have not been explored enough. By using retrained models with controlled training data, they managed to model face detectors that performed similarly to top performing face detectors. All the aforementioned approaches are based on feature extraction as a pre-processing step. However, lately CNNs have been popular to learn the optimal image features by itself automatically. They work as function approximates and learn the

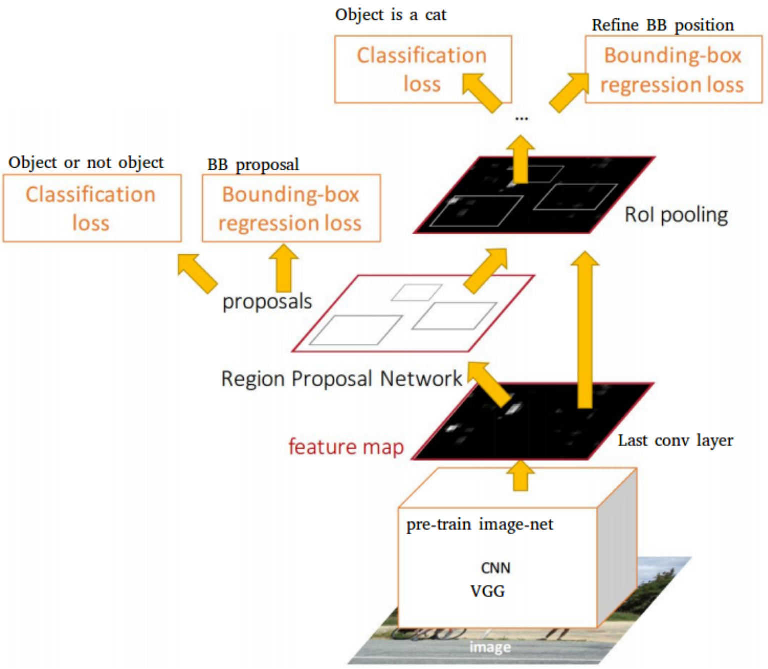


Figure 2.6: **Region-Based CNN for Face Detection.** *Region based CNN works by first computing candidate regions within an image for faces with different scales using anchors and applying classification and bounding box regression over detected candidates as face vs non-face and their location within the image, Figure from [29].*

relevant features using an iterative back propagation procedure. Li et. al. [70] utilized a cascade of CNNs for detecting faces. Using a cascaded approach allowed them to do face detection at different scales of faces at different cascade levels.

2.3.3 Face Detection with R-CNN

The key aspect of the approach used by Jiang and Miller [57], is a Region Proposal Network(RPN) to use a small CNN within a parent CNN. In the RPN, a pretrained CNN trained on Imagenet dataset is used and its convolution layers are followed by a 3×3 convolution layer corresponding to a large receptive field in the input image and produce a low-dimensional feature vector.

The output is then forwarded to separate branches classification and regression to train for both classifying an image based on the face proposals from RPN and also localizing the face using bounding boxes using the regression head as shown following a general object detection architecture 2.6.

In order to deal with faces of different scales, sizes and aspect ratio, anchors are included in a RPN. From experimental results, few canonical anchors have been identified with specific scale and aspect ratio and these are associated with each sliding location in a convolution feature map. The default setting for training included 3 scales (1282, 2562 and 5122 pixels) and 3 aspect ratios (1:1, 1:2 and 2:1) thereby giving total k=9 anchors for each sliding window. The face proposals relative to each anchor are trainable and for a $W \times H$ convolution feature map we get $W \times H \times k$ trainable proposals. Training of RPN is done using Stochastic Gradient Descent along both the classification and regression heads. The RPN and the Region CNN are trained in an end-to-end manner using softmax loss, as they both are independent of each other.

2.3.4 Face Detection algorithms used in the thesis

In this thesis, face detection was used to crop faces to be used for further analysis. Several face detectors have been used to process the datasets and extract face regions. Regarding the faces datasets that contained mainly frontal faces, the face detector from Viola and Jones was used. However, as mentioned, it cannot handle *in the wild* images. Consequently, other more advanced face detectors were tested. One of the used face detectors was the off-the-shelf face detector based on the work from Mathias et. al. [80], which was named Face detection without bells and whistles. Their face detector is based on DPM models, it give highly accurate detection on *in the wild* datasets, however, it is very slow and is not suitable for real-time applications. Nevertheless, that drawback was not an issue as it was mainly run once for the datasets and the detection results were saved.

3 Head Orientation Estimation with Relevance Vector Machines

In this chapter, the problem of head rotation angles estimation is addressed. We start by giving an introduction about the problem and discussing the importance of head rotation angles estimation in different applications. A light-weight method that uses a cascade of Multi Variate Relevance Vector Machines is introduced. An analysis on publicly available, large scale, and challenging dataset is presented. At the end of the chapter, the advantages and limitations of the method are discussed.

The work presented in this chapter has been published in the following articles:

Mohamed Selim, Alain Pagani, and Didier Stricker.
Sparse-MVRVMs Tree for Fast and Accurate Head Pose Estimation in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2017 [117]

Mohamed Selim, Alain Pagani, and Didier Stricker.
Real-Time Head Pose Estimation Using Multi-Variate RVM on Faces in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015 [116]

Contents

3.1 Head Orientation Estimation - Input Modalities 34
3.2 Datasets 35
3.3 Learning Head Orientation with MVRVM 39
3.4 Tree of Relevance Vector Machines 47
3.5 Conclusion 51

Head orientation estimation is an important topic in computer vision. It gained attention over the last years by the scientific community [62, 105, 14]. The problem of head orientation estimation can be seen as an individual problem on its own that contributes to the solution of the problem in hand, or as an important module in proposed solution for other problems. Let us look at a couple of examples to clarify these two specific points.

Considering the head orientation estimation as a problem on its own or part of the solution to a given problem can be seen in the gaze estimation problem for example. In gaze estimation, the head orientation is an important cue in the proposed solutions. The head orientation gives an important piece of information about the users' gaze. Gaze estimation is a problem that is usually addressed in Human-Computer Interaction systems as seen in [134, 155, 154]. In [134], the authors combined head orientation with eye localization for solving the problem of gaze estimation. If the problem addressed is required to be robust against users' head orientation variation, the problem of head orientation estimation is considered then an important pre-processing step in the solution proposed. Many computer vision tasks and image understanding techniques rely on a reliable, fast, and accurate head orientation estimator. In the following, we present some examples of such computer vision problems: Elharrouss et. al [31, 85] proposed a method that estimates the head orientation first, then tries to recognize the person to have a pose-invariant face recognition system. Ng et. al [90] proposed a method that estimates the head orientation first before estimating the gender of the person, this led them to a pose-invariant gender recognition system. Han et. al. [46] proposed a method to classify the age of the person or subject, but first they estimate the head orientation first in their proposed pipeline in order to have a pose-invariant age classification system. In general, head orientation estimation has a variety of uses in real world applications.

Due to the importance of the head orientation estimation problem, either as a goal itself or as part of other more complex systems, considerable attraction appeared in literature [86], and considerable effort has been put in solving the head orientation estimation problem. Another important aspect is that in some situations, a fast algorithm is required to allow the integration of the head orientation estimator module in other systems without adding a considerable overhead to the whole proposed solution. Solving the problem of the head orientation estimation can be carried out in one of several ways, that ranges from coarse classification to fine rotation angles estimation. In the case of using a classifier, a rough head rotation estimation can be carried out, and the output of such system is either, the head has a frontal pose, right profile or left profile. However, the problem can be viewed a multi-class classification problem, where the data can be classified according to the main head rotation angles, the yaw angle. The yaw angle is the rotation of the head around a virtual axis that goes along with the neck in a vertical direction. For example, the classes can be according to the yaw rotation angle ranging from left to right profile $+90$ to -90. The problem can be solved by for example using a sufficient amount of training dataset, and a Support Vector Machine - (SVM). One of the datasets that can be used in addressing the problem as a multi-class classification problem is the FERET dataset [100]. The dataset provides face images with annotations given as frontal, left profile, and right profile face image.

In case that a finer estimation is required by the application, then modeling the problem as a classification one will not yield a solution. The model that can be employed to allow angles estimation shall be modeling the problem as a regression one. The reason behind this is that modeling the head orientation estimation as a classification problem, would require defining many classes for all possible combinations of head rotation angles, and that is unfeasible. On the other hand, a regressor can have the ability of estimating an angle that was not presented in the training data, because it is trying to find the best fit for some learnt function $f(x)$, where x is a some representation of the input image using the available training data. Modeling the head orientation estimation using a regression model, the output of a trained regressor is a value in a probabilistic range of values that are detected by the range of motion of the head, given a dataset with head rotation angles annotations.

3.1 Head Orientation Estimation - Input Modalities

We can categorize existing approaches according to the input data used. In literature, one can find approaches that use 2D images as input. Also, other approaches exist that use 3D depth data.

Considering the works that use 2D images data as their input, they can be sub-categorized according to the type of data they are modelling from the input 2D image. One category would be the methods that use the appearance of the image directly as the input. Another sub-category would be the methods that search for specific features on the face, try to detect them, and use their location in the 2D image as the input to their proposed method. For example, methods that would be looking for eyes, nose, mouth,..etc in the face, and use the location information of those features to create input for their method. However, many existing methods use 2D view-based models [22, 99, 58]. Others would be using 3D models like in the works [12, 44]. Regarding the approaches that need facial features detection, they rely on the visibility of the features in the different poses they need to estimate. Previously, many works were introduced that rely on Active Appearance Models (AAMs)[22]. They rely on feature detection, tracking, and model fitting, which can lose the tracking or be error prone if the detection of the points or landmarks was not correct. Fanelli et. al. [33] uses 3D data in their method. They capture the data using depth cameras. It is not possible to apply their method on video streams, as they need a special camera that is capable of capturing 3D data.

The problem of head orientation estimation can be solved in different resulting spaces. Either discrete angles, or a continuous range of angles. The work done by Zhu et. al. [159], accepts results with error tolerance of $\pm 15°$. The work done by [5] estimates the head orientation by detecting and tracking facial landmarks. Relying on the tracking facial landmarks limits the head orientation estimation to the visibility of the landmarks, and is error prone to interpolating locations of occluded facial landmarks. Based on this limitation, the detected head orientation in the yaw angle is limited to roughly angles between -60 and +60 degrees. The method proposed in this chapter does not rely on landmark localization or tracking, it rather uses the face appearance as the input feature to the pipeline.

In the upcoming section, we discuss the chosen regressor in detail. As the proposed method is learning-based, data is crucial to this work. In the next section, the public datasets used in this work are presented. Afterwards, the free parameters are identified and optimized to the problem of head orientation angles estimation based on 2D images as input.

3.2 Datasets

Data is a key input to different Machine Learning (ML) algorithms. The aim of ML algorithm is to build a prediction model using the information stored in the data, in order to allow future predictions. Consequently, data annotations are required for learning. Data acquired must be annotated, thus, giving the data a specific organization that can be learnt by the machine learning algorithms.

In order to work with Machine Learning algorithms, a specific input with annotations or tags related to the underlying problem must be available. By training, the Machine Learning algorithm will be able to gain important information about the data, that enables decision taking, like classifying an object in a picture. Regarding the problem of head angles estimation from face images, public datasets are investigated. The standard datasets like FERET [100] has discrete specific values for head orientation angles. In the dataset, annotations exist for a specific one head rotation angle, the yaw angle. In short, if the person is looking into the camera, that is considered yaw angle of zero value. From this reference, looking right is increasing the yaw angle till it reaches 90 degrees, and -90 by looking in the other direction. Such a dataset is not suitable for the task in hand for different reasons. The values are discreet, with no data between frontal and left or right profile images for each subject. A continuous angles variation is an important feature that the dataset must have to perform a proper evaluation of our regression-based approach. Moreover, using real data captured in the wild is an important feature to assure the validity of our algorithm on real-life scenarios. The Labeled faces in the wild [54] is a challenging dataset in terms of occlusion, image quality, varying poses, different illumination, etc. However, it does not provide sufficient samples for each subject in different poses. In conclusion, possible candidates are the YouTube faces dataset [147] and the Point and Shoot Challenge dataset [10], which both are discussed in details as follows.

YouTube Faces Dataset

The YouTube faces dataset [147] is a challenging dataset that was introduced by Wolf et. al. in 2011. The authors followed the example of the Labeled Faces in the Wild dataset [54]. The authors of the dataset started the collection of videos from YouTube by using the names of the subjects existing in the Labeled Faces in the Wild dataset, which is a total of 5749 names. The top six results of the query on YouTube are downloaded. They later applied some filtering methods that includes rejecting duplicate videos. Sample images of the dataset are shown in figure 3.1. The authors used Viola and Jones face detector [138] to detect faces in the videos. Automatic screening was performed to eliminate detection of less than 48 consecutive frames, where detection were considered consecutive if the Euclidean distance between their detected centers was less than 10 pixels. The downloaded videos are processed into image frames, by extracting frames at 24 fps. Finally, the authors manually verified the videos to ensure that the videos are correctly labeled, not semi-static, not still images or slide-shows, and no identical videos are in the

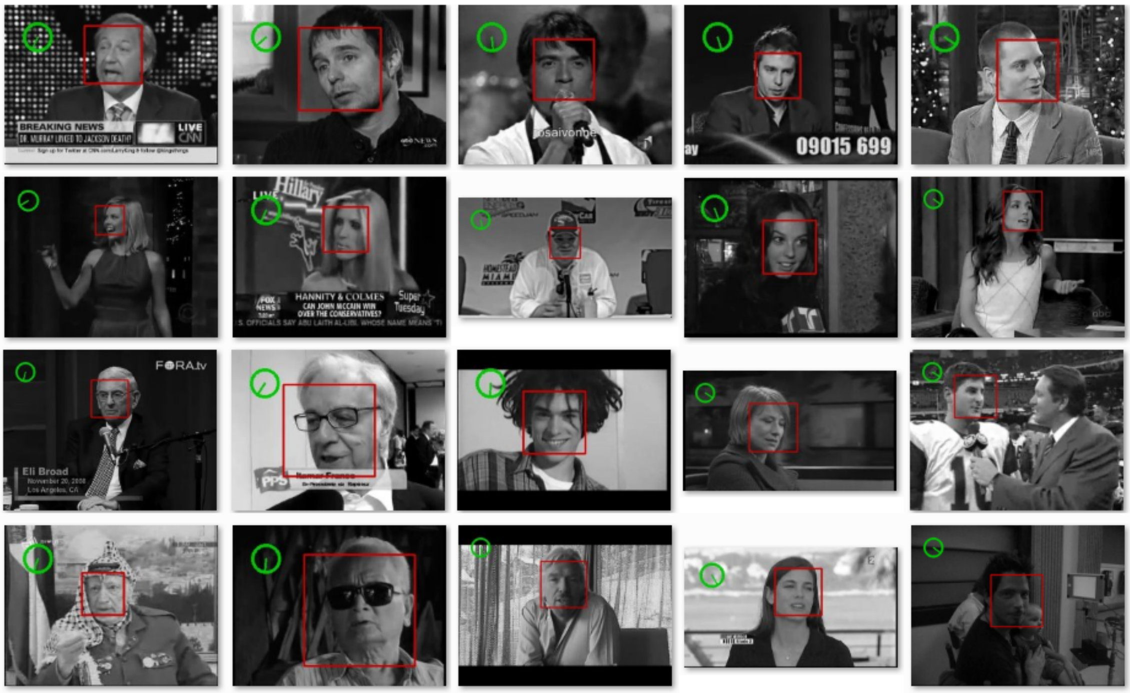


Figure 3.1: **Samples: Yaw angle estimation on YouTube face dataset.** *The red rectangle depicts the detected face, and the estimated yaw angle is indicated inside the green circle at the top-left corner. The circle represents a top view of the face, and the green line inside it shows the detected yaw angle. A detected angle of of 0 °is indicated by a line pointing downward. Despite the images having different backgrounds, presence of eye glasses or not, or some occlusion on the face, our method can predict the head orientation correctly*

dataset. The final number of videos in the dataset is 3425 videos of 1595 different people. The authors of the dataset provided the rotation angles of each frame in the dataset. They used face.com API [32] to provide the head rotation angles. The face.com API was the state-of-the-art in head orientation angles estimation at the time of the dataset creation, as mentioned by the authors. The provided angles are of sufficient quality to test the proposed pipeline.

The dataset was introduced to study the problem of face recognition across videos. However, it is also suitable for head rotation angles estimation problem. The subjects are collected from videos in unconstrained setup, and the authors used state of the art method to compute the head rotation angles. One important property of the dataset is that it has continuous angles of the head, this can be found in videos where the subject's head is moving freely. Each subject in the dataset has an average of 2.15 videos. The shortest video sequence contains 48 frames, the longest one contains 6070 frames, and the average number of frames for the videos is nearly 181 frames. The authors of the dataset followed the example of the Labeled Faces in the Wild image collection [54], which resulted in a large collection of videos. The dataset was used by [9] in video to video comparisons. Also, it was used by Best-Rowden et. al. [60] in face verification in the wild from videos. It was also used by [145, 73] for performing face recognition using deep learning methods. An important feature of the YouTube faces dataset that made it a candidate for the underlying problem is that the three rotation angles (our main interest) of the head are available for each frame in the dataset.

Point and Shoot Face Recognition Challenge (PaSC)

In 2013, Beveridge *et. al* produced the PaSC dataset [10]. They used inexpensive "point-and-shoot" cameras. They collected 9376 still images and 2802 videos of 293 persons. The videos were recorded in different locations. The locations are both outdoors and indoors, with varying illumination and backgrounds. The authors provided meta-data with the dataset that contains the face detection in the video frames. The authors used a commercial, state of the art method to provide head rotation angles, the PittPatt detector, which was acquired by Google, Inc [101]. The scenarios they had in the videos shows the face from the right profile to the left profile in continuous motion, where the yaw angles changes widely along the videos. Two video types were provided in the dataset, hand-held and controlled subsets. In the hand-held videos, the frames are very shaky and challenging. The controlled videos, have a stable background. Both video types are challenging. Figure 3.2 shows sample images from the dataset.

The rotation angle available in the metadata of the PaSC dataset was the yaw angle only. It was produced using the Pittsburgh Pattern Recognition (PittPatt) detector [101]. Pittsburgh Pattern Recognition was acquired by Google, Inc [42]. Having the yaw angle only is not enough for our problem. Consequently, the work by Asthana et. al. [5] has been used to compute the head orientation angles as it was the state of the art for

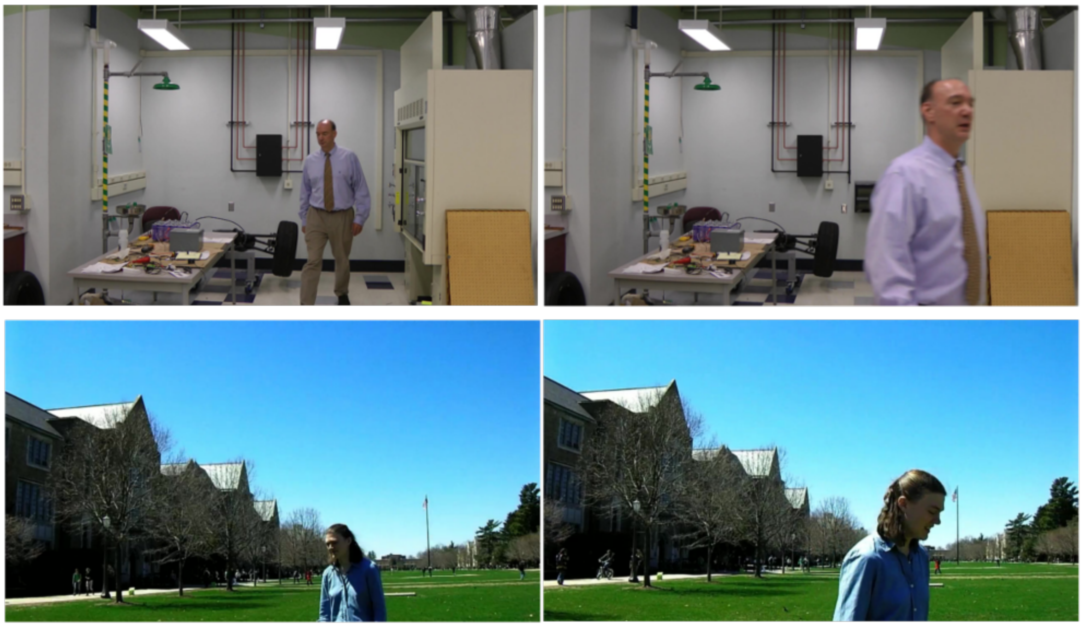


Figure 3.2: **Sample frames from the PaSC dataset [10].** *The top images are from the control subset videos (steady camera). The bottom frames are from the hand-held videos. Hand-held have lower quality and resolution. The dataset have videos captured indoor and outdoor. The persons walk during the video, thus we have different, continuous head poses*

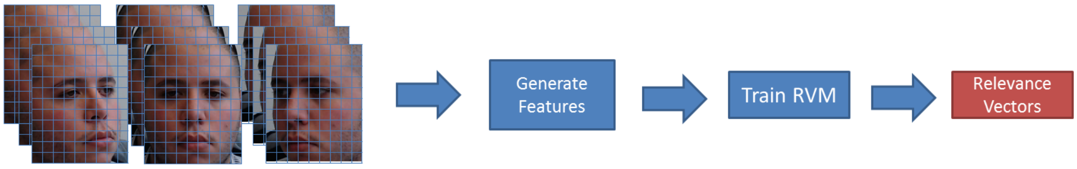


Figure 3.3: **Overview: Learning head orientation with RVM.** *Input images (for clarification purpose) are divided into patches. Feature vectors are generated for each image. The RVM is trained with the input feature vectors. The results of the training are the relevance vectors that can be used later in prediction*

facial landmark detection at the time of this work. The method proposed by Asthana et. al. [5] focuses on facial landmarks detection, and can estimate the head pose. The authors published a software library of their approach called Chehra [19]. Chehra was used to generate estimations of the head rotation angles. However, even this approach was challenged by the dataset, as it was unable to detect and track the landmarks in some hard frames of the detected faces in the PaSC dataset. The Chehra face and landmark tracker and detector was able to detect faces for about 72% of the faces provided by the meta-data in the dataset, otherwise it failed. However, all the frames were not needed, as the proposed approach learns the head orientation from appearance and can estimate it for detected face.

3.3 Learning Head Orientation with MVRVM

In this section, a proposed solution to the problem of head orientation estimation in real-time is presented. The proposed solution correlates between the face appearance and the head orientation angles. One of the advantages of the proposed method is its low computational complexity. Instead of using hand-crafted features, the face appearance is described using the information stored in the image pixels. An overview of the proposed method training is shown in figure 3.3. The method is proven to achieve results with good accuracy as presented in section 3.4.2.

The MVRVMs were used by Pagani et. al. [95] in estimating the local perspective transformation of a single point of interest in a 2D image. They trained sparse regressors by using image synthesis, generating random views of a specific reference image with known modified poses. The MVRVMs correlated between the appearance of the patch and the orientation of the patch. Likewise, the Multi-Variate Relevance Vector Machines (MVRVM) can also model the problem of head orientation estimation as a regression problem. It is shown by the proposed methods that they can be employed to learn the head rotations using the appearance of the face. The proposed approach does not rely on high quality images, but it is supposed to work on facial images taken “in the wild”, where no conditions apply while capturing the image of the face. An abstract overview of the proposed pipeline is shown in figure 3.3. The input image is partitioned into patches, followed by feature extraction. The training images are passed to the Relevance Vector Machine. In iterative learning, the relevance vectors are learnt by the RVM. The following subsections describe the approach in more details.

3.3.1 The Multi-Variate Relevance Vector Machine

The RVM, short for Relevance Vector Machine, proposed by Tipping [130], adapts the main ideas of Support Vector Machines (SVM) to a Bayesian context. Results appeared to be as precise and sparse as the SVMs, moreover, yielded a full probability distribution as output of the prediction unlike the SVM which yields non probabilistic predictions [130]. The RVMs fit the problem in hand, head orientation angles estimation, as the required output is the three angles of the head, which are floating point values in a probabilistic range. RVMs learn a mapping between input vector **y** and output vector **x** of the form:

$$x = W\phi(y) + \xi, \tag{3.1}$$

where ξ is a Gaussian noise vector with 0 mean and diagonal covariance matrix. ϕ is a vector of basis function of the form

$$\phi(y) = (1, k(y, y_1), k(y, y_2), ..., k(y, y_n))^T \tag{3.2}$$

where k is a kernel function. and y_1 to y_n are the input vectors from the training dataset. In this work, a Gaussian kernel was chosen, as shown in figure 3.4. The weights of the basis functions are written in the matrix W. In the RVM framework, the weights of each input example are governed by a set of hyperparameters, which describe the posterior distribution of the weights.

During training, a set of input-output pairs (x_i, y_i) is used to learn the optimal function from equation 3.1. To achieve this, the hyperparameters are estimated iteratively. Most hyperparameters go to infinity, causing the posterior distributions to effectively set the corresponding weights to zero. This means that the matrix W only has few non-zero columns. The remaining examples with non-zero weights are called *relevance vectors*. Tipping’s original formulation only allows regression from multivariate input to a single output value. In the head orientation estimation problem, the input vector is generated from the face image, and the output vector is a vector that has the three rotation angles of the head. Therefore, an extension of the RVM is used, called MVRVM, short for Multi Variate Relevance Vector Machine, which was proposed by Thayananthan et. al. [129]. The MVRVM uses an Expectation-maximization type algorithm for the training.

Input and Output Vectors

In the proposed approach, the face orientation shall be estimated based on the appearance of the face, without looking for specific features of the face. The idea is to learn the correlation between the appearance and the orientation angles using machine learning. Consequently, the input vector is defined as the average pixel intensities inside a grid division on the face. In details, the face image is divided into patches by a grid of size $a \times b$ blocks, where a is the number of columns in the X direction and b is the number of rows in the Y direction. For each patch, the mean value of the pixel intensity is calculated. All the mean values are concatenated together, resulting in the feature vector for the input image. The feature vector of the image is normalized as follows. The target vector is defined as a three-dimensional vector consisting of the three rotation angles of the face, yaw, pitch, and roll angles.

Normalization

Dealing with face images taken in unconstrained environment and setup can be challenging due to many factors. These factor include for example but not limited to image blur, low quality, and illumination changes. One possible method to be robust against strong illumination changes or bad illumination is to normalize the input feature vector. So, in order to prepare the data for regression training by the RVM, the feature vector is normalized such that the vector has a zero mean and unit standard deviation. By normalizing the input feature vector as described, it makes the proposed method robust to light changes that might occur among different input images. First, the standard deviation σ is calculated for the feature vector as follows:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{3.3}$$

, where N is the number of elements in the vector, and $\overline{x}$ is the mean value of the feature vector.

Later, for each element in the feature vector, a normalized value is calculated as in equation 3.4. The simple features that are used in the training of the Relevance Vector Machine do not require any facial landmark localization [159], or any complex tracking algorithms. This makes the computation of the orientation very fast.

$$\forall x_i \in X, \quad v_i = \frac{x_i - \mu}{\sigma} \tag{3.4}$$

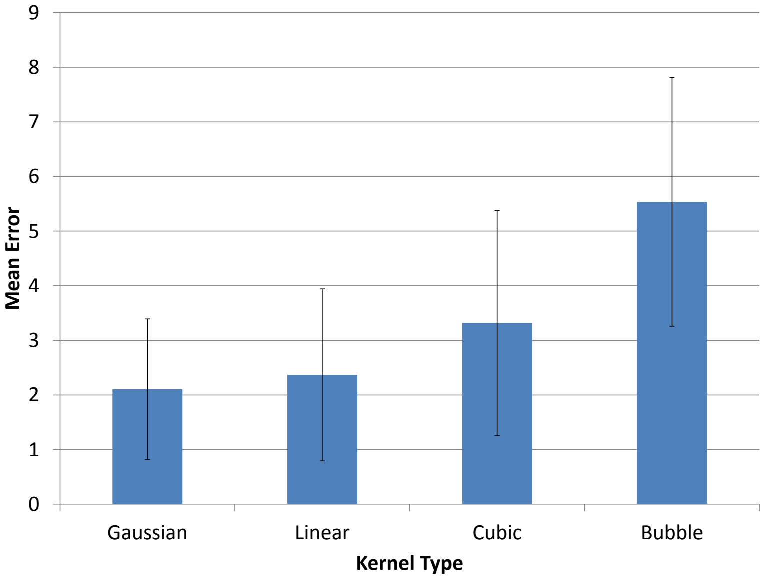


Figure 3.4: **Comparing different kernel types.** *Average errors in the yaw angle estimation are shown with the standard deviation among all dataset videos. Gaussian kernel yields the least error*

3.3.2 Parameter Optimization

In order to optimize the Relevance Vector Machine for head orientation estimation problem, the free parameters included in the process need to be optimized. One of the parameters is the kernel width of the relevance vector machine, see equation 3.2. It controls the sparsity of the RVM. Varying the kernel width, affects the number of relevance vectors, hence, it has be to optimized so that we avoid the over-fitting problem. Also, the size of the grid used in feature generation has to be investigated. The partitioning of the face incorporates the orientation varying information based on the face appearance. We would like to find the optimal grid size in both horizontal and vertical directions, such that we get the least error possible by the RVM. In this section, the parameter optimization results are presented and discussed. Cross validation training and prediction was applied on the dataset used in the optimization, the YouTube faces dataset.

Kernel type affects the accuracy of the training as it is the metric mapping the input to the output of the RVM. Four kernel types (Gaussian, Linear, Bubble, and Cubic) were evaluated on the YouTube faces dataset. Afterwards, all the average of all the errors in the yaw angle is computed. Figure 3.4 depicts the results. The Gaussian kernel yields the least error in the yaw angle estimation, hence it is the kernel that we used in the next evaluations. Figure 3.4 depicts the chosen kernel function.

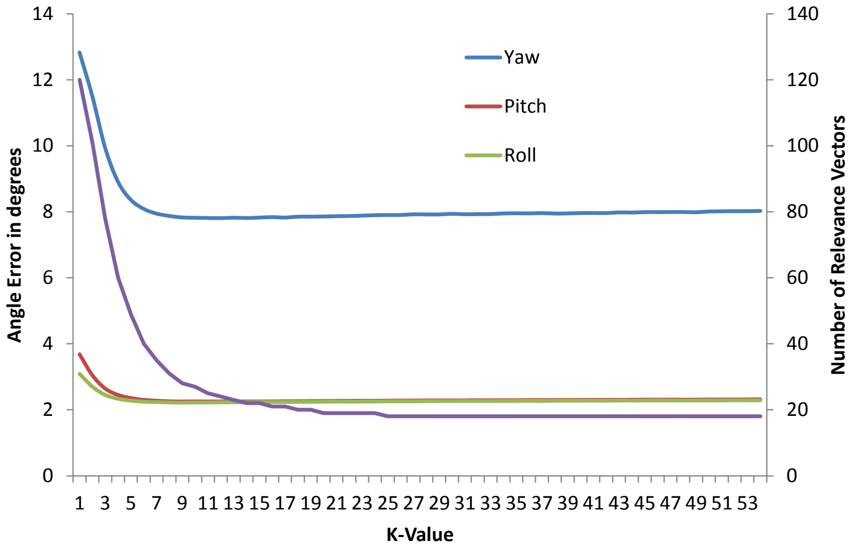


Figure 3.5: **Optimizing the kernel width k.** *The effect of varying the kernel width k on the average error (in degrees) in the three head rotation angles (yaw, pitch, and roll). On the secondary axis, the number of relevance vector is shown (violet curve). Initial grid size is* 15×15

The kernel width k has to be optimized as it controls the kernel width, and the sparsity of the RVM and it has to be taken care of in order to avoid over-fitting. As the kernel width increases, the number of relevance vectors decreases. If the kernel parameter is small, the RVM will use all the input feature vectors as relevance vectors, and that means it is not learning anything from the data and cannot differentiate between them. Figure 3.5 shows the average error in the three head rotation angles while varying the kernel width k from 1 to 55 with a different value of increments in the iterations. In each iteration, 75% of the data was used for training and the remaining 25% part of the data was used for testing, while assuring that the test set is not included in the training set. This gives us an estimate of the optimal value for the kernel parameter. It can be noticed that the average error is decreasing as we increase the kernel width. Also, the number of relevance vectors is decreasing too. The error roughly stayed nearly constant starting from the kernel width 7.5. We did not want to minimize the number of relevance vectors while maintaining the error as low as possible. As shown in the figure, the error starts increasing again at high kernel parameter values. We decided to proceed with kernel width of size 13 as it yields low error in the rotation angles and also not too small number of relevance vectors.

Now considering the optimization of grid size used in feature generation as described before, the grid size controls the number of patches on the input image. Using small number of patches (divisions) on the image of the face, reduces the size of the feature vector, which increases the prediction speed. Nevertheless, using small feature vector size

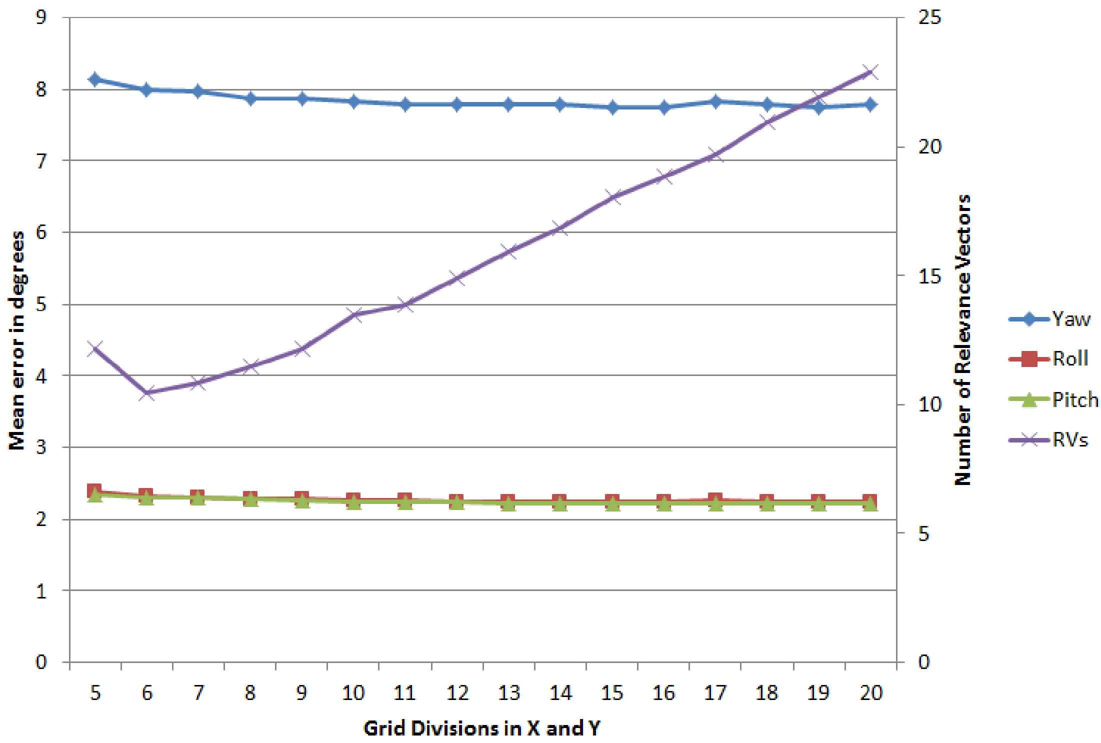


Figure 3.6: **Effect of grid size on the average error and the RVs.** *The effect of varying the grid size used in feature generation, on the average mean error in the three head rotation angles (yaw, pitch, and roll). The grid of size* 15×15 *yields the best results*

reduces the regressor's precision, because the input feature vector doesn't enclose enough information for the head orientation among different samples. In order to precisely detect the number of divisions that yields the minimum error, different grid divisions were evaluated using the YouTube faces dataset. The grid size varied from 5×5, up to 20×20. During the optimization, the optimized kernel width k was maintained at value 13. The results of that evaluation are shown in figure 3.6. It can be noticed that the size of the grid that yielded the least error in the three rotation angles was 15×15. After this value, the number of relevance vectors kept increasing. Also the error kept increasing as the difference in appearance got lost in the mean values of the pixels of the grids with higher number of divisions.

To summarize, two main free parameters were optimized. The kernel width controls the sparsity of the RVM. The grid divisions controls the feature vector size used in the training process. Based on the experimental evaluation presented, the optimal value of the kernel width is 13, and the optimal value for the grid size is 15. After performing the optimization on the YouTube faces dataset, the whole dataset videos were tested with the optimized parameter values. we evaluated the whole dataset using the tuned parameters. We ran a 4-fold cross validation tests on the datasets using all the videos provided for each subject.

3.3.3 Results with single MVRVM

In order to evaluate our approach on real data, a dataset that has a set of images with continuous degrees varying in the head orientation is required. The standard datasets like FERET [100] has discrete specific values for head pose. Most importantly, the presented approach needs to be evaluated on a dataset that has continuous angles. The Labeled faces in the wild [54] is a challenging dataset in terms of occlusion, image quality, varying poses, different illumination, etc. However, it does not provide sufficient samples for each subject in different poses. On the other hand, the YouTube faces dataset satisfies the requirements. The dataset consists of videos of different subjects, and such meets the main requirement of having faces with head rotation angles for different subjects. Also, the range of rotation of most of the subjects in the dataset is wide, for example some videos have yaw rotation from -88 degress up to 80 degrees. Moreover, it is a challenging dataset that was not captured in a controlled environment, nor was it captured using high quality cameras.

The free parameters in the proposed method were tuned on the YouTube faces dataset to find the values that yield results with the least error in the rotation angles. In this subsection, the results on the YouTube faces dataset are presented and discussed. The results as shown in figure 3.7, show that for more than 75% of the dataset the mean error was less than 10°in the main rotation angle of the head, the yaw angle. Also, for about 20% of the dataset, the error is below 6° in the same angle. So, using MVRVM in the problem of head orientation angles estimation can achieve good performance on a very challenging dataset. The MVRVM can correctly estimate the head orientation for more than 75% of the dataset with error tolerance of $\pm 3 - 4^\circ$ using simple features that does not require complex features detection on the face of the subject.

Finally, the proposed approach was tested on unseen videos of the same subject. Subject who has more than 2 videos in the dataset were used. The MVRVM was trained with two videos, and then tested on an unseen video of the subject. The results of that approach are as expected showed less accuracy. The number of subjects included in that test were 533 subjects. The average mean error in the rotation angles were 21, 6, and 5 degrees in the yaw, pitch, and roll rotation respectively. Keeping into consideration the range of the yaw angles in the training was not limited. All frames in the videos were used, from left profile to right profile appearances. Results are promising for an error of 20°in that challenging test. The architecture of the machine used in the evaluations is a 6-core Intel Xeon CPU with hyper-threading technology, and 64 GB of RAM. Predicting the head orientation angles as presented using one single MVRVM is suitable for real-time

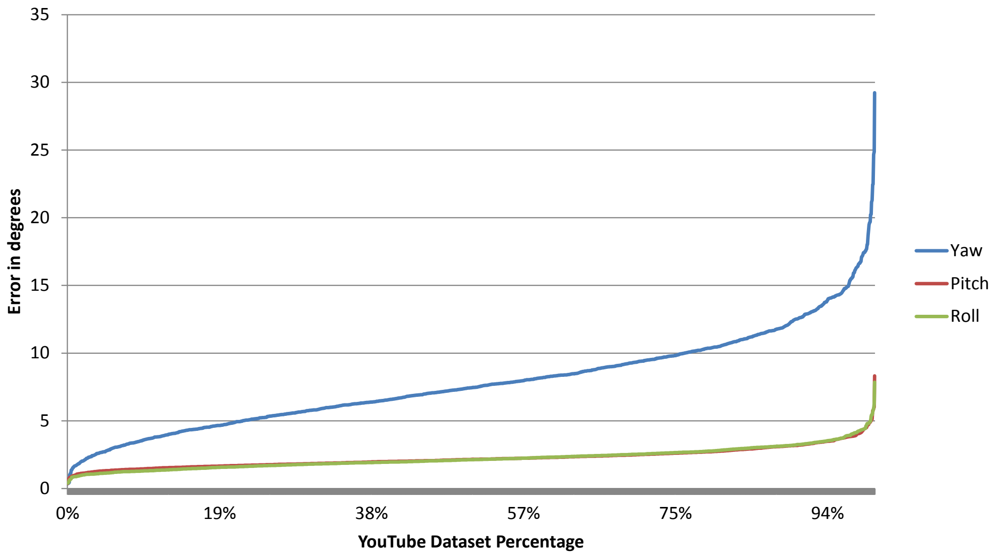


Figure 3.7: **Single MVRVM Results on YouTube faces dataset** *The* $k = 13$*, and the grid size for feature generation is* 15×15*. The X-axis represents a percentage of the dataset which consists of 1595 subjects. The Y-axis represents the mean error of the 4-fold cross validation evaluation.*

applications as the time taken by the computation is only 2-3 milliseconds, with no need of complex landmark detection or model fitting or tracking.

3.3.4 Discussion

We present a regression scheme for head orientation estimation using the appearance of the facial image. The output of our approach is an estimation of the three rotation angles in the full range of the angles, with floating point values. The MVRVM neither relies on complex features generation, nor does rely on special landmark localization, but rather relies on the appearance of the facial image to estimate the head orientation angles. The facial image is divided into patches using a grid of size $a \times b$. The grid size and the MVRVM kernel width were optimized. The online prediction of the three head rotation angles is very fast, it takes around 2-3 milliseconds, hence it is suitable for real-time use. This allows the use of the proposed method as a pre-processing step in other applications that rely on the head pose.

The proposed method was evaluated on a challenging dataset, the YouTube faces dataset. It has images from videos that were taken in uncontrolled environments, with varying face sizes, illumination, some occlusions, etc. We showed that our approach can learn the three head rotation angles using simple features. This approach doesn't rely on depth images, nor 3D information beside the 2D image. The approach doesn't need landmark detections on the face and can predict full range of motion of the face. MVRVM can learn faces with extreme rotation angles. The results of the evaluation on the YouTube faces dataset show that MVRVM can achieve an estimation with error tolerance of $\pm 6.5°$ in the yaw rotation angle, and less than $\pm 2.5°$ on the pitch and roll angles on the whole dataset. For more than 80% of the dataset, the MVRVM estimates the angles with tolerance error of $\pm 10°$. The final evaluation on the YouTube faces dataset was run in 4-fold cross validation. The results on unseen videos were promising, taking into account the different conditions from the training videos.

In order to improve the results and get more fine head orientation estimations, a cascade of RVMs can be built in a way that the first RVM can give a rough estimate on the head rotation angles. Following the first regressor, a set of RVMs are to be trained on a smaller range of angles. The set of RVMs can be one level after the main one, or different number of levels of RVMs can be trained on data with smaller range of angles. The number of levels in the cascade tree and the number of RVMs in each level must be investigated. This is presented and discussed in detail in the upcoming section.

3.4 Tree of Relevance Vector Machines

In the previous sections, it was shown that MVRVM can be a good head orientation estimator, that does not rely on landmark detection or tracking, with good accuracy. The MVRVM was trained on a single subject, and the head rotation angles were estimated for the same subject even on unseen videos. However, to generalize the solution, the accuracy is not good enough and one single MVRVM is not capable of generalizing the solution. Consequently, in this section, a more complex solution is presented. The proposed algorithm builds upon the hyperparameter optimization performed on the single MVRVM. A tree structure is presented to learn and predict head orientation angles of unseen faces. The input vector size is kept the same used in the single MVRVM method, which is the normalized average pixel intensities extracted from the face image.

Figure 3.8 shows an abstract overview of the proposed method. The detected faces are fed into the feature extraction step, then the features and corresponding angles are fed into the Root node of the cascade tree. Based on the estimated yaw angle at the root node, a child node is chosen for the next prediction, this repeats till a leaf node is reached. The cascade is discussed in details in the next subsection.

3.4.1 Tree structure

The cascade of the regressors is built in a tree structure. The yaw angle is used in branching the tree, as it is the rotation angle of the head that has the widest range. In other words, it is the most distinctive angle of the head rotation angles. The yaw angle

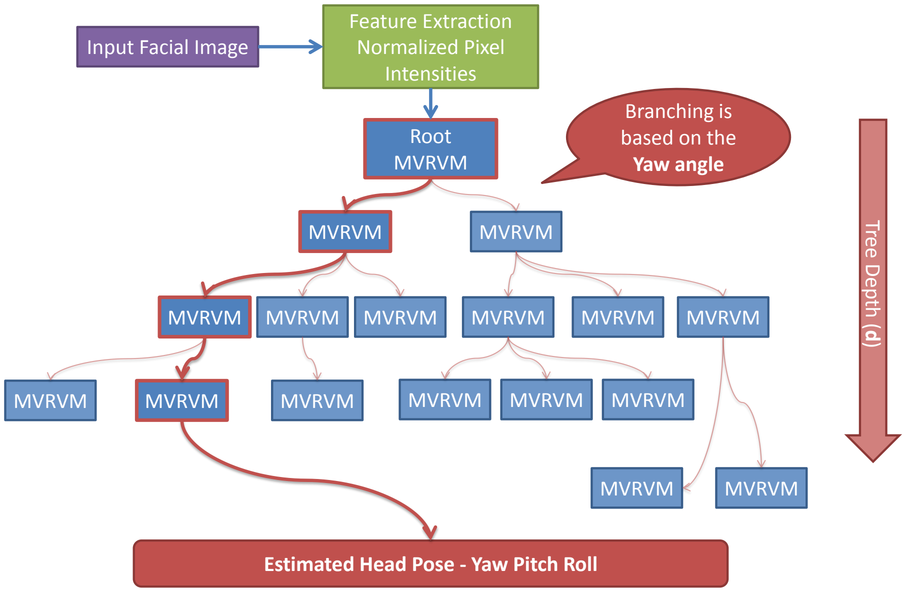

Input Facial Image
Feature Extraction Normalized Pixel Intensities
Root MVRVM
Branching is based on the **Yaw angle**
MVRVM
Tree Depth (**d**)
Estimated Head Pose - Yaw Pitch Roll

Figure 3.8: **Overview of the cascaded approach** *For a set of input images along with the groundtruth, feature vectors are extracted. All vectors are used to train the cascade of MVRVM tree.*

can start from -90°(left profile face) to +90°(right profile face). At the root node in the tree, the MVRVM is trained on the input samples, which consists of the features of each face and its corresponding three rotation angles. Going to the next level of the tree, the number of children of the node is determined by the branching factor s. If $s = 3$, the yaw angle range is split into three ranges, and the data is filtered such that each node has the samples that lie in the corresponding yaw angle range. The branching goes on until we reach the maximum depth of the cascade, or a MVRVM node does not have sufficient data to be learnt. The resulting tree of MVRVMs, is used in the prediction process. The prediction process starts from the root node. The root node is designed to give a rough estimation of the head pose. Based on the predicted yaw angle, the child node is chosen to be the next node used in the path while traversing the tree of the cascade. The longest prediction path is predicting and improving the estimation by d predictions, where d is the maximum depth of the cascade. The branching factor of the tree of MVRVMs was varied from 2 splits up to 5 splits. Having more than 5 splits makes the range very small in the child MVRVM nodes. The optimization started at branching factor of value 2, which is the minimum number of splits possible. When the branching factor was more than 4, the number of input samples decreased quickly in the tree, hence, resulting in a shallow tree. The branching factor with the least error was 3 on the YouTube faces dataset as shown in figure 3.9. We set the root MVRVM node to have only two children, thus classifying right or left profile faces. Further child MVRVM nodes in the tree have branching factor of 3. Based on that setup, the leaf nodes of the tree get very small range of angles after depth

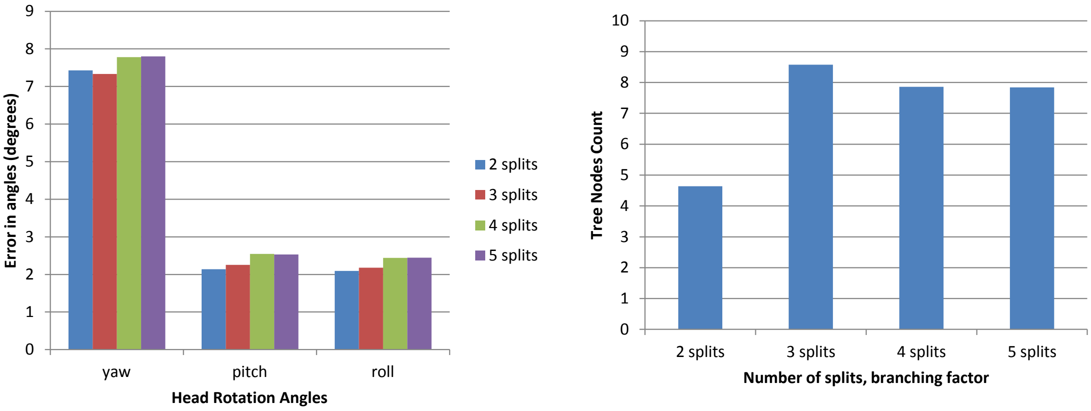


Figure 3.9: **Effect of tree branching factor on average error in head rotation estimation, and nodes count of the tree.** *On the left, the effect of the branching factor -varying from 2 to 5- can be seen on the average error in degrees. On the right, the effect of the branching factor is shown on the total number of MVRVM nodes in the tree. This optimization was performed using the YouTube faces dataset.*

of 3. Consequently, we set the maximum depth to be 3 levels.

Figure 3.9 shows the effect of varying the branching factor on the YouTube dataset. The results are the average of 4-fold cross validation on all the 1595 subjects in the dataset. In figure 3.9, we see that the 3 splits has the most number of nodes, which means a better representation of the data in the cascade tree. It also follows that the least average error on the main head rotation angle, the yaw, is at 3 splits. Considering the presented evaluations, we optimized the free parameters in our approach by 4-fold cross validation experiments which considered all the videos of one of the subjects. Thus, the next step is validating the approach with as many samples from the dataset as possible, which is discussed in subsection 3.4.2.

3.4.2 Evaluation and results

In this section, the evaluation of the cascade tree is presented and discussed. The evaluation was performed using challenging datasets of persons captured in different conditions. These datasets have continuous head orientation variation. They vary in the background, illumination, indoor and outdoor locations, resolution, etc. The results of the experiments on the datasets show the validation of the proposed method for generalization purposes on large subsets of the dataset.

Single MVRVM vs. Cascade Tree of MVRVMs

It is important to compare the single MVRVM to the Cascade tree of MVRVMs. Table 3.1 shows the mean accuracy of 4-fold cross validation test on the PaSC dataset. All the frames of the dataset were mixed, and split into chunks of 5000 frames. The reason for choosing the number 5000 is that by experimental evaluation, 5000 images were the maximum amount of data that can be handled by a MVRVM. For each chunk, 75% of the data was used for training and the remaining 25% was used for testing. The average errors with standard deviation is shown in table 3.1. The cascade tree approach yields smaller errors in all head rotation angles. The error is reduced by roughly 20% in the yaw rotation angle.

Method	Yaw	Pitch	Roll
Single MVRVM	5.4 ±4	5.4 ±4	3 ±2.5
Cascade Tree	**4.6 ±3.32**	**5 ±4**	**2.3 ±2.1**

Table 3.1: **Comparing single MVRVM to the cascade tree of MVRVMs.** *The Cascade shows smaller mean error - PaSC. (Mean error in degrees ±std). Validation experiment of chunks of 5000 samples*

Generalization

Based on the findings so far, the kernel width optimal value is 13, and the optimal number of grid divisions is 15. The final step is approach validation. For the PaSC dataset, better head orientation estimations were generated using the Chehra library as it was the state of the art at the time of this work, and better than the PittPatt face detector used which was used by the dataset authors. The method proposed by [5] deals with landmark localization and tracking, and it can be used in head orientation estimation. To validate the use of MVRVMs for head orientation estimation, all video frames from all subjects were shuffled. Then we divided the frames into sets of 5000 frames each, where as mentioned before, this is the practical maximum amount of data that can be handled by the MVRVM. Afterwards, 4-fold cross-validation was performed on each set. The number of validation sets in the PaSC is 25, each having 5000 random frames from different subjects.

The MVRVMs can learn the head orientation by the appearance of the face with acceptable accuracy. Less the 4.6 °error in the yaw angle in the validation tests are reported on very challenging uncontrolled datasets. Regarding the pitch and roll angles, the MVRVM reported errors less than 5°on PaSC. Fig. 3.10 shows the distribution of the errors in the angles on the dataset frames. We can see that the error is below 5° in the yaw angle for about 66% of the data. and below 10° for about 98% of the data. The errors in the pitch and roll are higher which could be due to the fact that the video frames did not have as

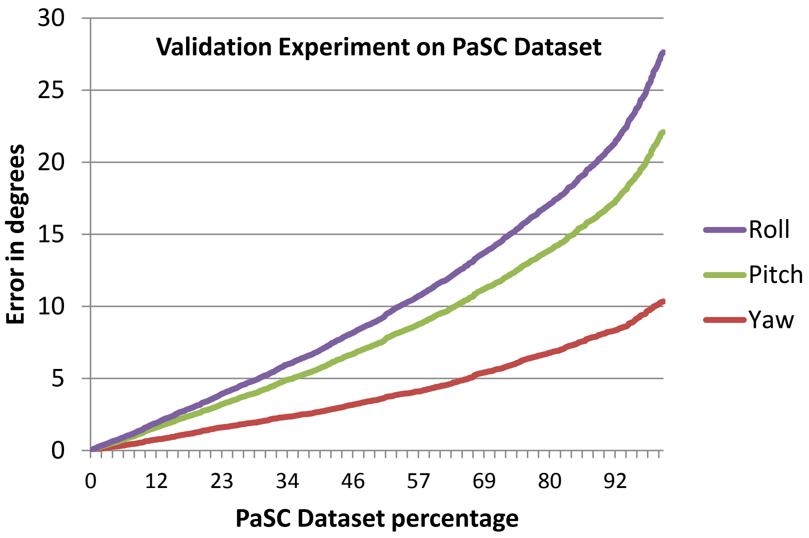


Figure 3.10: **Validation results on PaSC dataset** *Results of the cascaded approach in the validation experiment on the PaSC dataset. Yaw error is below 5° in the yaw angle for about 66% of the data. and below 10° for about 98% of the data.*

big variations as in the yaw angle. Finally, the proposed method is suitable for real-time applications as the time taken by the computation of one single prediction of the three head rotation angles is only 6 milliseconds, with no need of complex landmark detection or model fitting or tracking.

3.5 Conclusion

In this chapter, a new method for estimating the head orientation angles was introduced. Using machine learning, the head orientation angles were correlated to the face appearance. The idea of using Multi Variate Relevance Vector Machines was studied in such problem. At the beggining of the chapter, the problem was solved using just one single MVRVM to estimate the head pose. The YouTube faces large dataset was used to learn and optimize the use of RVMs in estimating the head orientation angles. The main input to the RVM was normalized pixel intensities of the face grid. Normalization was an important pre-processing step to add robustness to strong illumination variations. The YouTube faces dataset contains 3425 videos of 1595 different persons. Good initial results were achieved on such a challenging dataset. However, when testing for unseen faces, the results were not good enough and required further improvements.

In order to solve the problem of generalization, a more complex pipeline was introduced.

The pipeline builds a cascade tree of MVRVMs. The hyperparameters of the cascade tree were optimized. Using more challenging datasets, the Point and Shoot Challenging dataset was used. The dataset provided variations in different aspects. The videos varied in quality, as all the sequences were taken with two cameras. One full high definition camera that is stable on a tripod, and another lower resolution, lower quality camera that was handheld by a person. The dataset sequences were also captured indoors and outdoors, and the subjects were both males and females. The depth of the tree and the tree structure was optimized on the PaSC dataset and the YouTube faces dataset. Extensive cross-validation experiments were performed on the large scale datasets. The new introduced pipeline reduced the error by 25% on the YouTube faces dataset and the PaSC dataset. One single estimation using the cascade tree requires 5-6 milli-seconds, this is due to the fact that the features required are inexpensive to compute, and quickly predictable using MVRVMs. Consequently, the proposed method is usable in real-time applications.

Although the MVRVMs can estimate the head orientation angles accurately, they don't generalize with the same accuracy. In the presented evaluation, the MVRVMs were pushed to their limits. One single MVRVM can handle at most 5000 input samples. This is due to the very large covariance matrices built during the training phase. They lack the ability of handling larger amounts of data. This noted issue could be solved by using Convolutional Neural Networks, as they can handle large amounts of data without an issue. The use of CNNs for solving the head orientation estimation is studied in the following chapters.

One important point to note is that the available public datasets lack accurate and reliable head orientation angles groundtruth. The public datasets authors used state of the art methods to track facial landmarks, and from that compute the head orientation angles. YouTube faces dataset authors generated the head orientation angles using face.com. For the PaSC dataset, the authors used PittPatt face detector, and that provided only the yaw angle. State of the art method for landmark localization and tracking was employed to generate the angles for the dataset. The angles are acceptable as a proof of concept for using the MVRVMs to learn the provided angles. However, as mentioned, these angles can lack accuracy. Therefore, in the next chapter we introduce a new large scale, accurate, head pose dataset that was acquired using sub-millimeter accurate motion capturing system. All details about system calibration, synchronization, and acquired data are presented in details in the next chapter.

4 Large-Scale Head Orientation and Eye Gaze Dataset

In this chapter, a new large scale head orientation and eye gaze dataset is presented. The chapter explains the system calibration, and defines the head coordinate frame. The proposed dataset covers a wide range of specific tasks that where performed by the subjects. A sub-millimeter accurate motion capturing system was used to acquire the groundtruth. A deep learning-based baseline algorithm for estimating the head orientation is presented. The dataset is publicly available at `autopose.dfki.de`

The work presented in this chapter has been published in the following article:

Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker. AutoPOSE: Large-Scale Automotive Driver Head Pose and Gaze Dataset with Deep Head Orientation Baseline. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

Contents

4.1 System Components **56**
4.2 System Calibration **64**
4.3 Head Coordinate Frame **69**
4.4 AutoPOSE dataset **72**
4.5 Summary **78**

Public datasets have tremendously pushed computer vision research forward in the recent years. Data existence is crucial to various Machine Learning algorithms as without reliable training data, algorithms won't be able to properly learn the problem in hand. Importance of data is not only crucial for learning the problem described by the data itself, but it is also critical as a valid assessment of new algorithms solving the same problem. In other words, the data is the mean used for running a common evaluation of different algorithms, allowing an objective comparison. Thus, data is used as a mean for assessing new contributions. In addition, since the rise of deep learning methods, large-scale datasets have become crucial to realize research and development. This is due to the fact that enough data must exist to allow proper generalization of the deep learning models, and besides also, avoid the problem of over fitting.

There is a large interest in car interior human-centered applications during the last 10 years. It is gaining more attention nowadays, and that is related to the active development of autonomous vehicles. Examples of such applications related to the driver in the studies of autonomous vehicles are driver attention monitoring, driver intention prediction, and driver-car interaction [71, 96]. All these technologies require as basis the head pose and/or the eye gaze of the driver. The head pose describes the head position and orientation in the car, whereas the eye gaze is the direction of the driver's view. Head pose and eye gaze are important cues that can imply important information about the driver's attention, and intention. Consequently, the autonomous driving research community has put a lot of attention and effort to be able to understand the driver's behavior inside the car cabin. Thus, high quality annotated data related to such tasks are now of crucial importance, not only to the machine learning community, but also to the autonomous driving research community.

Recent datasets provide either head pose or gaze or have an automotive context. However, none of them contains the combination of all these features. Thus, in this chapter, we introduce the AutoPOSE dataset, which is the first dataset providing combined driver head pose and gaze for in-car analysis tasks, and acquired with sub-millimeter accurate motion capturing system.

In 2017, two new head pose datasets were introduced, the DriveAHead [112] and Pandora [16]. The DriveAHead proposed a novel head reference system (or head coordinate system), defining where the head center is, and how the x, y, and z-axis of the head are defined related to specific facial landmarks. We also use the same head reference system in our AutoPOSE dataset. The DriveAHead provided IR and depth frames from a Kinect camera in a real driving scenario. The authors did not provide accuracy measures of the motion capturing system while driving, although the motion of the car will affect the tracking system calibration accuracy. The dataset is suitable for deep learning frameworks.

Pandora [16] is a large scale dataset that is also suitable for deep learning frameworks.

However, the authors did not specify a head reference system (head center and rotation axis). In addition, the subjects were acting to be driving on a normal chair in front of a wall. In our AutoPOSE, we provide data captured in a real car cockpit with cameras placed at the dashboard and the center mirror location. Moreover, we use a well-defined head reference coordinate system. In 2015, the MPIIGaze [155] dataset was introduced containing RGB images only. The subjects were gazing at known points at a computer screen. As RGB cameras are highly affected by sunlight, they are not suitable for driving scenarios [112]. In AutoPOSE, IR images from two perspectives (dashboard, center mirror) are provided with 3D gaze target groundtruth in a driving environment. The center mirror set also provides color, and depth images.

In 2019, Roth et. al. presented a new head pose dataset called, DD-Pose [108]. The dataset consists of 330K bincoular stereo images of size 2048×2048. The data set was recorded in a real driving scenario. It contains natural movements of the subjects while driving a car on the street. The car was equipped with couple of motion capturing tracking camera to track the driver's head movements. Although the dataset contains natural movements, however, the accuracy of the tracking system can be affected by the motion of the car, due to vibrations. In Conclusion, all existing datasets have specific drawbacks. AutoPOSE provides groundtruth in a controlled environment, that ensures groundtruth correctness and quality. Moreover, frame annotations are provided where the subjects performed the required task while having no glasses on, with clear glasses on and with sunglasses on. All frames were manually annotated. The dataset provides two camera views (dashboard, and center mirror) in a car cockpit with gaze target groundtruth and occlusion annotations.

The contributions in this chapter are:

- A new, large-scale, accurate, driver head pose and eye gaze dataset is provided, named AutoPOSE.
- The dataset contains images acquired from two different camera positions in our car simulator and provides different image types: dashboard (IR, $\sim$ 1.1M) and center mirror (RGB, Depth, IR, $\sim$ 315K each).
- All frames of the dataset are annotated with information about driver's head pose, activity, accessories (glasses) and face occlusion.
- A baseline for head orientation estimation task using deep learning is provided.

This chapter is organized as follows: first the components of the acquisition system are presented. Then, the different necessary calibrations are discussed, including cameras intrinsics calibration, acquisition system cameras calibration, and cameras and head calibration to the acquisition system. In section 4.3, the head coordinate frame is explained. Then, the acquired data in the AutoPOSE dataset is presented. In order to avoid confusions, the mathematical representation of rigid bodies in 3D space is disucssed in the Chapter 2, section 2.2.

4.1 System Components

In order to acquire a new dataset of drivers head pose, and eye gaze targets, various components are required. One needs to have a driver, a camera, and a mean of acquiring the head pose. The driver can be sitting in a real car, and actually driving on a road while performing different tasks. On the other hand, the driver can be sitting on a chair and simulating the driving movements. Each scenario has its own advantages and disadvantages. That specific point will be discussed later in more details. The head movements and tracking can be acquired in a real driving environment. Such environment, will lead to realistic driving from the recorded subjects. However, the acquired groundtruth will be inaccurate. This is due to the fact that the moving vehicle will affect the position of the cameras tracking the driver (in some way) to acquire the head pose in 3D space. The pose of the driver's head can be acquired using one of different available technologies. The head pose acquisition could be done using motion capturing systems, magnetic systems, marker-based systems. Motion capturing systems are the most accurate systems that are suitable for such application. The tracking accuracy is in sub-millimeter values for motion capturing systems. However, having a motion capturing system in a moving car can introduce error in the tracking accuracy which is even hard to measure due to the fact that the moving vehicle can be subject to different amounts of vibrations. Consequently, in AutoPOSE, it was decided to acquire the dataset in a controlled environment, but using a car cabin simulator.

In short, the different components in the AutoPOSE dataset are:

- Car cabin simulator
- Motion capturing system: IR cameras, reflective markers, tracking software
- IR Camera placed at the dashboard
- Kinect V2 (IR, RGB, Depth) Camera placed at center-mirror

In this section, the systems components are described, starting with the car cabin simulator. Afterwards, the used motion capturing system used is described. Then, the two camera devices used in the acquisition are presented. It is important to note that both cameras, and the motion capturing system are connected to the same computer, allowing synchronization of the data using the machine hardware timestamp.

4.1.1 Car Cabin Simulator

In-cabin monitoring of vehicle occupants is a topic of increasing interest induced by the ongoing development of advanced driver assistant systems (ADAS) and automated driving, up to driver-less vehicles and shared mobility [93]. The requirements towards the monitoring functions are changing with the level of driving automation. The demand for

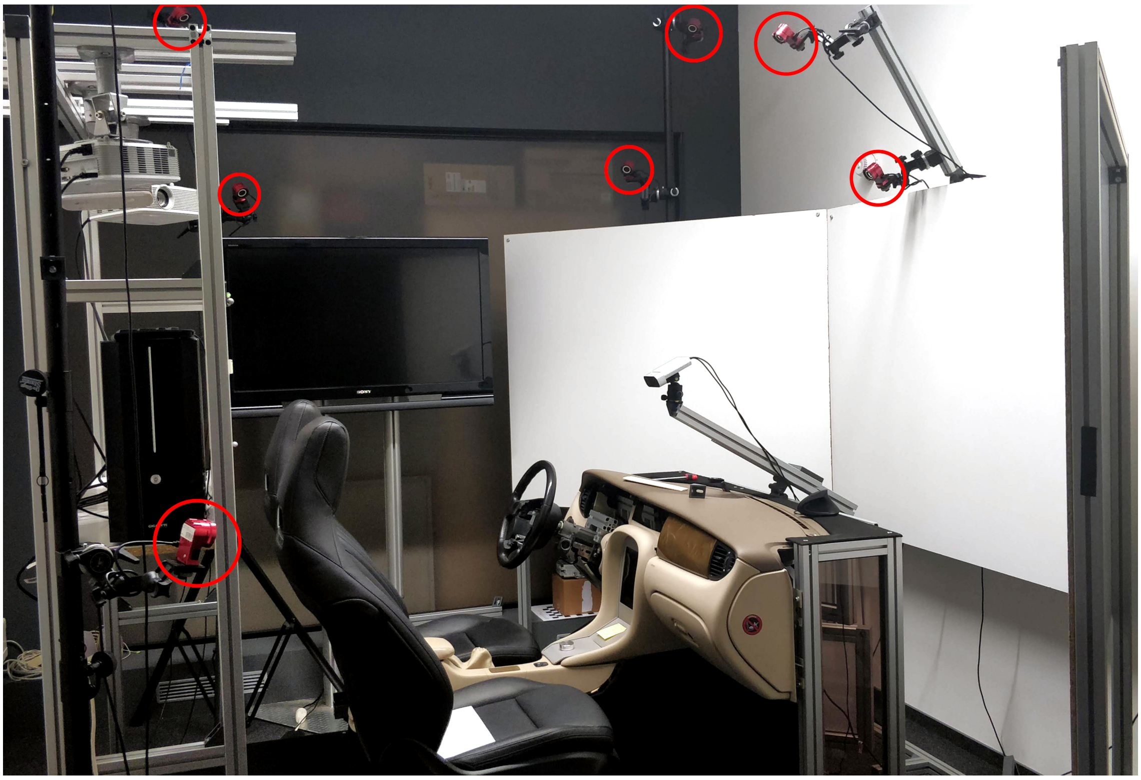


Figure 4.1: **Car Cabin Simulator** *The red circles highlights some of the motion capture system cameras.*

these novel monitoring functions comes moreover from new requirements for safety and comfort function, as well as from novel human-vehicle interfaces. The full monitoring and understanding of the scene in the vehicle cabin comprises not only the automatic detection, classification and recognition of all occupants and objects, but also the estimation of the occupants pose and state, as well as the recognition of their activities, interactions, and intentions.

Of particular interest is thereby the monitoring of the drivers state and intention. Many research activities have recently concentrated on the development of camera systems monitoring the drivers face and inferring on his/her state, such as awareness, and focus of attention, in order to realize novel warning functions and as support for ADAS functions and future automated driving. Although full body pose tracking of humans has become an intensive research domain since the launch of the first Kinect camera [149, 82] the full body pose detection in vehicles is rarely investigated. They range from the recognition of arm gestures and intentions for comfort and advanced human-vehicle interfaces, to a robust analysis of the drivers activities and availability. The later function is crucial for automated driving of levels 3 and 4, in which the hand-over of the vehicle control to the driver has to be managed. Existing benchmark datasets for in-cabin monitoring functions, as, e.g. the VIVA challenge [79], provide often videos from confined areas inside the vehicle, where either the drivers head or hand is located. Moreover, the annotations do not contain 3D groundtruth data. Recently, several in-cabin benchmark datasets with groundtruth 3D-measurements have been published [112, 108], but they are restricted to the detection of the head pose. There exists full body pose datasets, as, e.g. the MPII dataset [3], where it contains a very wide range of scenes. However it contains only a few are for driving cars. Although it is not the main focus of this thesis, but the newly introduced dataset, especially the Kinect subset, contains the driver's upper body in the images. Thus, providing the data that can be used for the analysis of the body pose in the driving context for the scientific community.

The in-cabin test platform is based on a driving simulator, consisting of a realistic in-cabin mock-up (shown in figure 4.1) and a wide-angle projection system for a realistic driving experience. The test platform has been equipped with a wide-angle 2D/3D camera system for monitoring the entire interior of the vehicle mock-up and an optical groundtruth reference sensor system that allows to track and record the occupants body movements synchronously with the 2D and 3D video streams of the camera.

4.1.2 Motion Capturing System - OptiTrack

Motion capturing systems are systems that rely on vision to track usually reflective markers in 3D space. Usually, a set of IR cameras are used together which are placed to cover the capture volume from different positions. Depending on the size of the capture volume, the number of cameras, and resolution can be decided to fit the required purpose. Also, the speed of the cameras needs to be suitable for the intended application. For example, tracking high speed body movements like practicing sport would need cameras running at at least 120 frames per second up to 500 frames per second if supported by the cameras. The vision-based motion capturing systems nowadays are extremely accurate to a sub-millimeter level. Optical motion capturing system are basically stationary infrastructure of high-speed and high-resolution infrared cameras. The cameras are calibrated together and synchronized (using hardware and software triggers) with respect to each other. Reflective markers are attached to humans in order to track the human movement, or attached to objects to track the object's movement or pose in 3D space. In order to track human movements, the markers are attached to specific human anatomical positions on the body using customized or specific marker protocol. The reflective markers appear in different cameras of the optical motion capturing system, and their 3D position is estimated in 3D space.

Motion capturing systems have been used for different applications. They have been used in movies industries, where the motion of the actors are tracked, and recorded. Afterwards, they recorded motion can be used for rendering and animating 3D models. Such technology also allowed realistic rendering of facial features by tracking 3D positions of the face of the actor and use the tracked positions to realistically animate the face of the model. Besides the use in the movies industry, motion capturing systems have been widely used in research. Applications in research can be seen in movement analysis and in biomechanics to record and analyze movements of either humans [23] or animals [107]. Different companies make vision-based motion capturing systems such as OptiTrack [94], Vicon [137], and Qualisys [152], as there is a demand for such systems. One of the manufacturers is a US company called OptiTrack [94]. In [37], the authors compared different cameras from OptiTrack. They showed the proper camera to be used according to the capture volume -whether it is a small volume on a table or a large room- and applications. OptiTrack produces different cameras, the Flex 3, Flex 13, Prime 13, and Prime 17W. Please refer to the paper [37] for more technical details and analysis about the cameras. In this dataset, it was decided to work with the Flex 13 cameras. You can see the cameras in figure 4.1. The systems used in this work contains 12 Flex 13 cameras. The resolution of the Flex 13 camera is 1280×1024, which makes it suitable for large capture volumes as the one of the car cabin simulator. The Flex 13 camera runs at 120 Hz natively, thus making it suitable for head pose acquisition application.

One important characteristic of such systems that makes them suitable for scientific research is their high accuracy. The optical motion capturing systems infrared are calibrated to one another using very precise calibration tools. For example, in the case of OptiTrack, there are calibration tools that are manufactured to utmost precision. The calibration tool of known exact length is used to calibrate the capture volume, with reflective markers installed on them at specific locations. The tool gets moved in the 3D space of the capture volume, and the markers on the calibration tool are detected in each camera, enabling intrinsic and extrinsic calibration of each infrared camera in the system. Also, since the exact size of the calibration tool is known, the error in the marker tracking can be computed by comparing the known distance between the markers with the distance between the detected markers in 3D space. Usually, the optical motion capturing system, given a proper setup and a sufficient number of cameras, the error is in sub-millimeter. Besides the highly accurate detection of reflective markers in 3D space by the infrared cameras, also the cameras run at high speeds that can reach up to 250 Hz, making them suitable for motion analysis and groundtruth data acquisition.

In [143], the authors used motion capturing system to combine the camera position for SLAM applications. In their work, they provide the combination of real depth and color data together with a groundtruth trajectory of the camera and a groundtruth 3D model of the scene. They obtained the groundtruth for the trajectory using a motion capture system (OptiTrack with Flex 13 cameras) and for the scene geometry via an external 3D scanner, each with sub-millimeter precision. In the context of body tracking and movements analysis, the research community benefited from the accuracy of the optical motion capturing system. In [18], the authors proposed a new adaptive covariance-based MIMU sensor fusion and calibration approach which addressed the degradation arising in inertial motion capture due to body accelerations and magnetic disturbances. The authors used an OptiTrack motion capturing system to acquire the groundtruth orientation of some rigid bodies in 3D space, that also had the IMU sensors attached to. That enabled them to properly perform an assessment to their algorithms, as the absolute groundtruth pose of the rigid body is provided by the OptiTrack. They calibrated the IMU system to the OptiTrack system in a way similar to the one performed here in the AutoPOSE dataset, and that will be discussed in the next section in details. Gail et. al. [38] and Hoffmann et. al. [51, 52], worked on bridging the gap between motion capturing and bio-mechanical optimal control simulations of a human steering motion on a car steering wheel. Their goal was to increase the realism of simulated human motion through measurements obtained from optical motion capturing systems. They used a mock-up steering wheel, reflective markers, and OptiTrack system to capture trajectory measurements of human motion. Their preliminary results show that a fusion of physical laws, bio-mechanical simulation and real data from OptiTrack within an optimal control simulation framework indeed have the potential to improve motion capture and synthesis.

In motion analysis the optical motion capturing systems are considered the gold standard. In the field of medical rehabilitation, the authors of [39] utilized motion capturing systems from Vicon and OptiTrack in order to aid the design of a hand rehabilitation robot. A neurological disorder like a stroke, or spinal cord injuries can lead to disability in body parts. A therapeutic training, especially at early stages, can help the patient to restore the damage and avoid permanent disability. Gezgin et. al. [39] presented a hand rehabilitation robot. In order to properly design the robot's motion, they acquired kinematic data with the motion capturing systems. They attached several markers on a person's finger, and captured the motion using the infrared cameras. The accurate trajectory of the markers helped in designing the motion that should be provided by the rehabilitation robot.

Still considering the field of medicine, Roemhildt et. al. [83] from the Department of Orthopedics and Rehabilitation, University of Vermont utilized a motion capturing system from OptiTrack in their study. Their objective was to analyze the effect of Basic calcium phosphate (BCP) on the friction in the knee joint of a rat. This type of medical studies uses animals at the beginning before human trials. The authors used a pendulum apparatus to measure the friction in the knee of rats postmortem. In order to get a precise measurement of the pendulum arm, they attached reflective markers and tracked the trajectory of the markers using motion capturing system. They were able to precisely measure the difference between their different samples of rats using the optical motion capturing system, thus allowing them to draw conclusions based on accurate measurements. In the work presented by Teufl et. al. [128], the aim of the study was validate IMU-based 3D joint kinematics of the lower extremities during different movements. They used optical motion capturing system as a reference groundtruth of the movements performed by their twenty eight participants. By applying their algorithms on the IMU data, the authors were able to validate the measurements drawn from the IMU sensors, as they compared them to the optical motion capturing system data in three functional movements, static sport, dynamic sport and physiotherapy specific movements.

4.1.3 Acquisition Cameras and Camera Positioning

In AutoPOSE, data was acquired from two different perspectives in the car cabin simulator. The location of the camera plays an important role in the appearance of the face in the camera frame. It also affects the size of the face in the camera. The appearance of the face in the camera could be of crucial importance to some applications related to face analysis. In case of eye tracking, and gaze estimation applications, it is important to have the eyes of the driver in the camera frame in a way that would allow proper tracking of both eyes, and with the minimum required resolution to have enough information about the eye in the image. One of the aims is to provide the research community with data that is annotated with high quality groundtruth information of the head pose from different position in the car. Thus, enabling the study of driver monitoring problem and even many other applications from two view perspectives, allowing the researchers to properly investigate the possibilities and limitations of installing cameras in different locations in the car. Besides, generally computer vision algorithms perform better if the the images are pre-processed or normalized [133]. Since driving can be during sunny daylight or dark at night time or cloudy days, images captured by color cameras can suffer from extreme variation in illumination. Therefore, infrared cameras are more suitable for driving scenarios as they are less subordinate on global ambient light in the scene. IR images will provide more consistent looking scenes for driving tasks compared to RGB images. In this section, the cameras used in the acquisition of the AutoPOSE are presented.

Infrared Camera

An infrared camera has been fixed at the dashboard location of the car model. The camera is small in size, that is suitable for use inside a car cabin. The camera runs at 60 frames per second, thus making it suitable for smooth motion capturing. The camera resolution is 752×480. A picture of the camera can be seen in figure 4.2. For the AutoPOSE dataset, the Blackbird camera was mounted at the dashboard location of the driving simulator. The camera was directed at the driver's face. Sample pictures taken from the camera can be seen in fig 4.8. In such a position, the face of the subject can be clearly seen through the steering wheel. The face gets occluded by the steering wheel for a short time only during action. The camera has infrared emitters that are built in the camera body, which illumines the scene at high frequency, and the camera captures images at 60 fps. The synchronization of the infrared emitters and the image acquisition is done by the camera producers. The camera is connected to hardware device which is responsible for triggering the camera, and for the data transfer to the acquisition computer. The connection interface is through a Gigabit Ethernet.

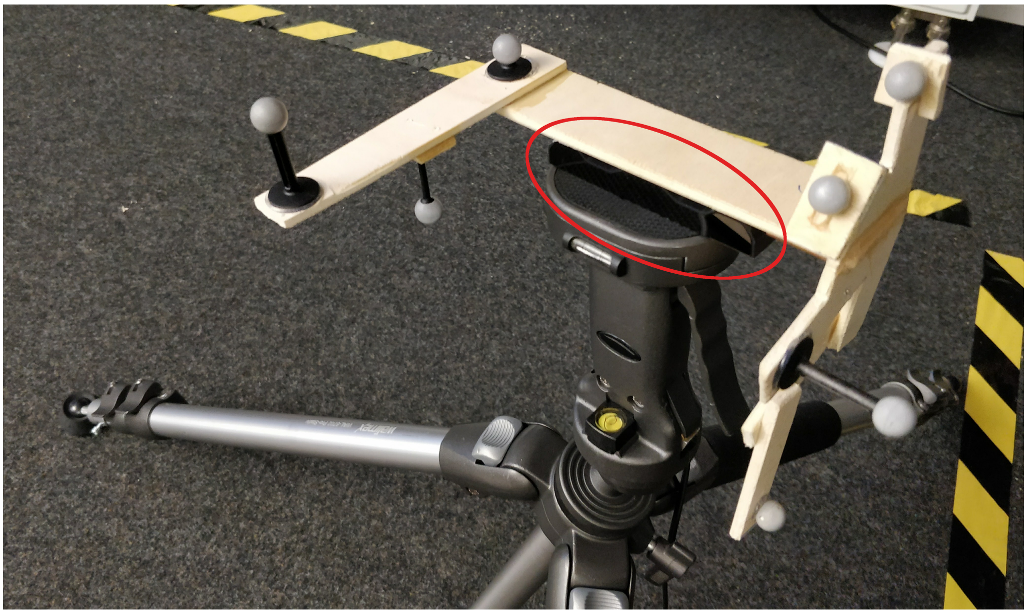


Figure 4.2: **Dashboard camera with Markers** *Spherical markers are rigidly attached to a structure on the camera. The markers are used to form a rigid body that gets tracked by the motion capturing system (OptiTrack). Picture was taken during the data acquisition for hand-eye calibration.*

Microsoft Kinect v2

In AutoPOSE, it was decided to acquire data from the center mirror location, besides acquiring data from the dashboard location. The Microsoft Kinect v2 was mounted at the center mirror location on the car cabin simulator. A picture of the Kinect in the mounting position can be seen in figure 4.3. The Kinect was directed at the driver where the driver's face and upper body were visible in both the color image and the infrared image. The face and upper body were mainly in the center of the camera frames. Wassenmüller et. al. [144] systematically evaluated and analyzed Kinect v1 and Kinect v2, with a focus on the depth images. They concluded that the all center pixels shows a similar accuracy in the Kinect v2. They recommended to use Kinect v2 in the context of 3D reconstruction, SLAM applications or visual odometry. They suggested pre-processing depth images using, for example, bilateral filters [136, 142] before using them. In AutoPOSE, the raw depth data is provided, however, it is recommended to pre-process the depth images according to the intended application or use. The Microsoft Kinect v2 has one depth camera, one color camera inside. The depth image stores in each pixel the distance from the camera to the seen object. The Kinect v2 contains a Time-of-Flight (ToF) camera that measures the depth by the time taken by the emitted light from the camera to the object and back. Therefor, it constantly emits infrared light with modulated waves and the camera detects the shifted phase of the light when it returns [120, 68]. The combination of light emitters and ToF camera is referred to as the depth camera. Due to the fact that the Kinect

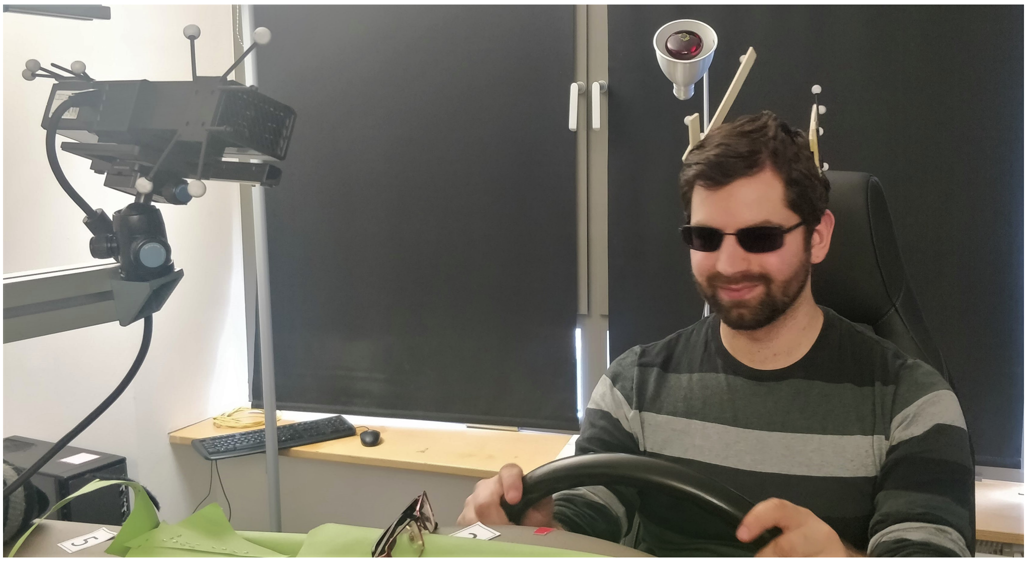


Figure 4.3: **Center mirror camera with markers.** *Spherical markers are rigidly attached to a structure on the camera (Kinect v2). The markers are used to form a rigid body that gets tracked by the motion capturing system (OptiTrack). Also a subject is shown wearing sunglasses and the head target with markers to be tracked by the OptiTrack.*

is constantly emitting infrared light, it was causing a lot of disturbance in the dashboard camera. Consequently, it was not possible to record with both cameras at the same time during the acquisition session of each subject. Consequently, the data was acquired using the dashboard camera at first, while the Kinect was switched off. Then, the data was acquired again using the Kinect camera, and the dashboard camera was switched off. The data was acquired from the Kinect v2 using the official Kinect for Windows SDK (version 2.0).

	X Resolution Pixel	Y Resolution Pixel	Frame Rate Hz
Color - RGB	1920	1080	30
Depth	512	424	30
Infrared	512	424	30

Table 4.1: **Microsoft Kinect v2 resolutions.** *Resolution [in pixels] and frame rate [in Hz] of the images acquired by Microsoft Kinect v2. Frame rate is the same for all color-RGB, infrared and depth sensors, 30 Hz. The color-RGB camera inside the Microsoft Kinect v2 provides a Full HD image.*

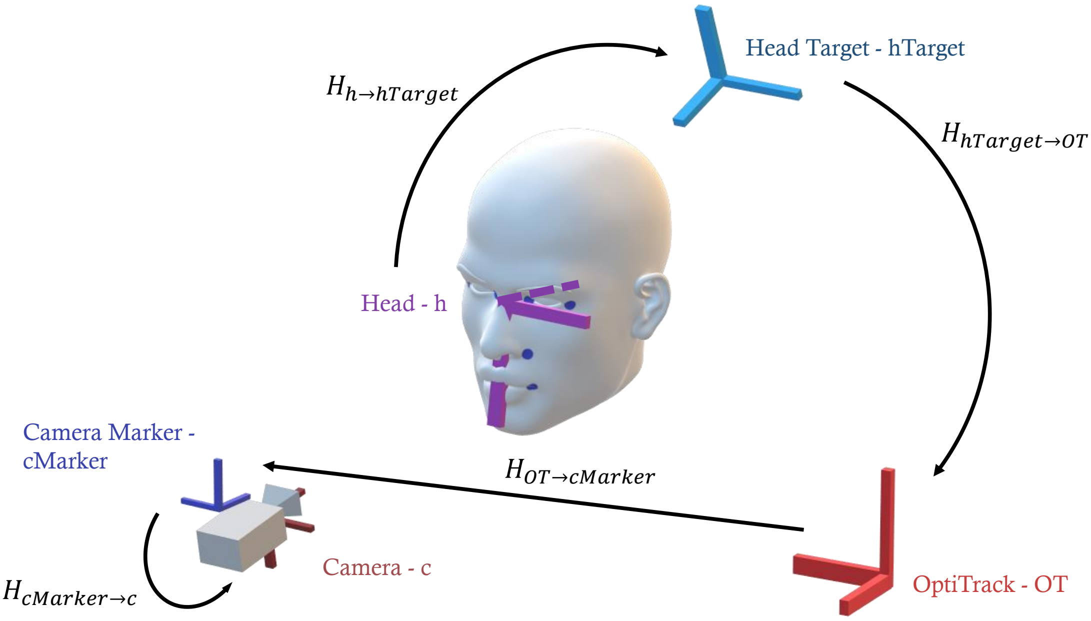

Head Target - hTarget
$H_{h \rightarrow hTarget}$
$H_{hTarget \rightarrow OT}$
Head - h
Camera Marker - cMarker
$H_{OT \rightarrow cMarker}$
Camera - c
$H_{cMarker \rightarrow c}$
OptiTrack - OT

Figure 4.4: **AutoPOSE coordinate systems.** *Different coordinate systems are illustrated, showing the relative transformations from one coordinate system to the other. The aim is to represent the head in the camera coordinate frame.*

4.2 System Calibration

In this section, the different required calibrations are discussed in details. Calibration is a crucial step in this work as it affects directly the quality of the data. There are different calibrations required to be performed accurately in order to achieve the aim of the dataset. The aim is to have the orientation of the subject's head in the camera coordinate frame. The head orientation is acquired using the motion capturing system - OptiTrack. Consequently, the camera coordinate must be calibrated to the motion capturing system, allowing describing the head in the camera frame. This method is called hand-eye calibration, which is discussed in detail in the following subsections. Another required calibration is the one of the motion capturing system. Each camera pose must be known in 3D space, in order to allow proper marker tracking and accurate measurement of the markers' 3D position in space. One final required calibration is the acquisition camera intrinsic calibration. The intrinsic calibration is required to allow proper image rectification, and is also required for back-projecting 3D points onto the image, for example the head center, and the reflective markers.

4.2.1 OptiTrack Calibration

The optical motion capturing system used in this dataset was the OptiTrack. The system consisted of 12 Flex-13 infrared cameras, each running at 120 Hz. The cameras calibration was computed using the software provided by the system producer, called Motive [84]. The camera resolution is 1280×1024, which makes it suitable for capturing large volumes. The cameras were distributed around the car cabin simulator, such that the subject's head, acquisition cameras at the dashboard and center mirror location, along with the car cabin simulator markers are well tracked by at least more than 4 cameras. A special arrangement of 3 Flex-13 cameras was setup to track facial landmarks at the beginning of the subject's recording session, but that point is discussed later in sub-section 4.3.2. At the beginning of each recording session, the calibration procedure as required by Motive is performed. Before performing the calibration, all 12 Flex-13 cameras are detected, but with no information about the pose of each one. The calibration procedure uses a specific tool called "calibration wand", which has 3 reflective markers. The distances between the markers are known by the Motive software. The wanding "as called by Motive" means moving and waving the calibration tool in the capture volume, in other works covering the visible cone by each camera with enough markers. That would allow the software to calibrate each camera intrinsics and extrinsics. The wanding procedure takes around 2-3 minutes. The calibration computation takes around 3 minutes. Followed by the calibration, a special tool from OptiTrack is used to define the ground plane and origin of the OptiTrack coordinate, which is the *World* coordinate frame of the AutoPOSE dataset. As a result of the calibration the 3D position of each camera is known in the OptiTrack frame.

The calibration procedure reports directly the average error in detecting the position of the reflective markers in the 3D space. The error is usually in sub-millimeter accuracy scale. There is also an error analysis software tool for the capture volume. The capture volume error analysis tool uses the calibration tool of the known size of exactly $500mm$ between the two outer reflective markers, and compare it with the detected distance between the markers. The result of this test was always below 1 mm error, so the detected distance was more than 499 mm at the driver's head location, and around the cameras and car cabin simulator markers. Consequently, the accuracy of the marker tracking is very high and is suitable for groundtruth acquisition. The markers are used to create user-defined rigid bodies to be tracked. Rigid bodies were created for the acquisition cameras, the head target, and the car cabin simulator. The Motive software has the ability to interpolate the 3D position of markers that are not visible for a given rigid body. For example, if the rigid body consists of 8 markers, and at a specific point in time, only 6 were visible by the OptiTrack cameras, the software can interpolate the 3D position of the occluded markers, and considers the rigid body to be tracked. In AutoPOSE, since superior accuracy is one of the aims, the Motive software was configured to consider rigid bodies as *"tracked"*, if and only if all reflective markers of the rigid body are visible, and being detected, otherwise, the rigid body is considered not tracked.

4.2.2 Acquisition Cameras Calibration

Intrinsic calibration

The acquisition cameras needs to be calibrated in order to be able to describe their properties in a mathermatical form. The information obtained from the intrinsic calibration are mainly the focal lengths in both x and y directions (f_x, f_y), and the camera center position in the image (c_x, c_y). The distortion coefficients r_1, r_2, r_3, t_1, t_2 were also estimated, but not for the depth camera as rectifying the depth image is not trivial, and is also out of the scope of the dataset acquisition. The dashboard infrared camera, the color camera of the kinect and the depth camera were all calibrated. In literature, there are different methods [8, 49, 123] that could be used especially for RGB-D cameras. In this work, the toolbox presented by Bouguet [17] was used to calibrate the cameras. Images of a checkerboard were captured using the cameras from different orientations and distances to the board. The intrinsic calibration was computed using the images of the known size checkerboard. Sample images from the dashboard camera can be seen in figure 4.5.

Acquisition cameras - Hand-Eye calibration

It is of utmost importance to describe the coordinate frame of the acquisition cameras in the OptiTrack-World coordinate frame. This relation is required so that we can achieve the aim of describing the head coordinate frame in the camera coordinate frame, which can be called the head pose with respect to the camera. In AutoPOSE, there are basically three cameras, the dashboard camera, the Kinect v2 color camera and Kinect v2 depth camera. One way to track the cameras in OptiTrack is to attach reflective markers to the body of the cameras, and define a rigid body, then track the rigid body in 3D space. However, the center and orientation of the rigid body attached to the camera does not necessary co-onside with the virtual camera center and axis. If the markers are rigidly attached to the camera body, then there is a rigid transformation between the camera frame and the frame of the rigid body. The process of computing that rigid transformation is called Hand-Eye calibration. Hand-Eye calibration is a well-known method in the area of robotics [121, 124]. In this work, the method presented by Tsai and Lenz [132] was employed.

Spherical reflective markers were placed rigidly on the camera body , thus in each frame we get the position and orientation of the camera body in our reference coordinate system. The reflective markers on the dashboard camera can be seen in figure 4.2. Since the Blackbird camera body is small, attaching the markers directly to the camera will not be suitable. The resulting rigid body will be detected by the OptiTrack but will not be stable, shaking will occur in the tracking. Therefore, a rigid wooden structure was firmly attached to the camera so that a wide baseline between the outermost markers in the three main axis can be achieved. In figure 4.3, reflective markers on the Kinect v2 can be seen. The reflective markers on the cameras were not changed, and a rigid bodies were

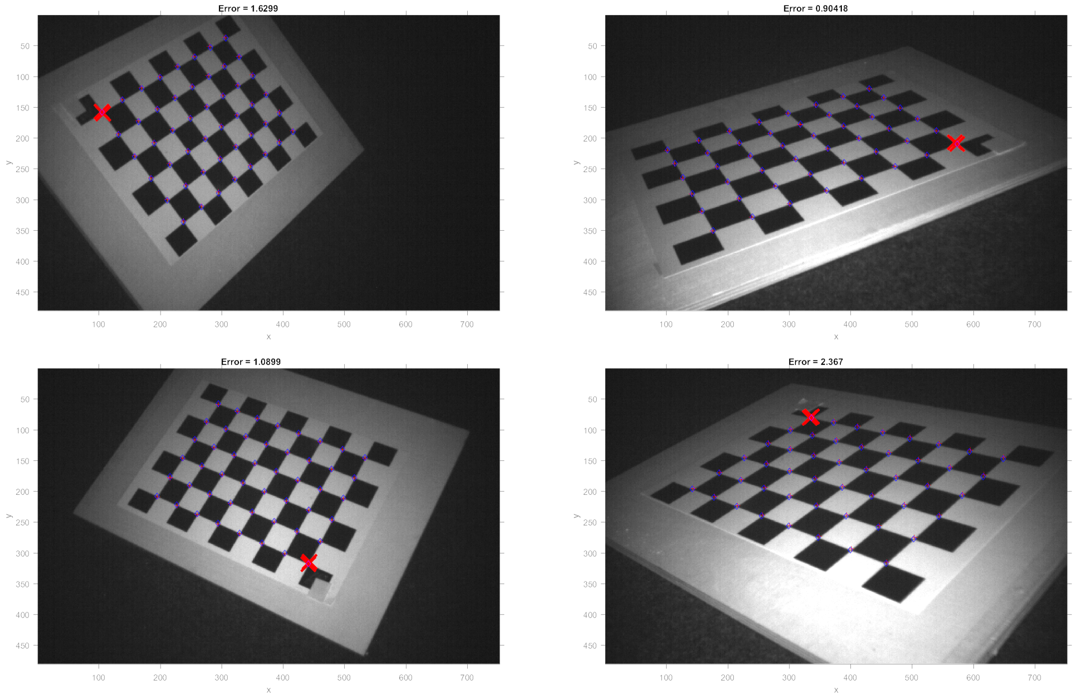


Figure 4.5: **Handeye calibration - error analysis.** *The images shown are some of the images acquired to perform the handeye calibration. The aim is to find the rigid transformation between the rigid body defined by the markers attached to the camera and the camera coordinate frame. The computed transformation is used to transform the checkerboard known structure onto the image, and compute the average error in pixels at each corner.*
Note: Intensity was improved for visibility and printing purposes.

defined once by the OptiTrack system, and used in all recording sessions. During the Hand-Eye calibration sessions, the defined rigid bodies were tracked by the OptiTrack. Two calibration sessions were recorded for each camera. The procedure is the same for all cameras, as discribed by Tsai and Lenz [132]. A fixed pattern of a checkerboard was captured by the camera providing images $C_{1:n}$, the board has to be fixed in place, and the camera was being moved around the checkerboard. Along with each image, the pose of the rigid body was detected by the OptiTrack $T_{world2marker,1:n}$, where $marker$ is the rigid body defined by the camera markers. The camera poses $T_{camera2pattern,1:n}$ with respect to the pattern can be detected with the result of the extrinsic calibration from the toolbox [17] using the images $C_{1:n}$. Since the 3D position of the pattern is fixed during the calibration session, then the term $T_{world2pattern}$ is constant. A system of equation can be built as follows

$$T_{camera2pattern,1:n} \times T_{marker2camera} \times T_{world2marker,1:n} = T_{world2pattern} \tag{4.1}$$

where $\times$ is matrix multiplication and T is a 4×4 transformation matrix in homogeneous coordinate. The system of equations is built for all n images and poses that were captured during the calibration session, by the cameras, and by the OptiTrack. The calibration solves for the wanted transformation $T_{marker2camera}$ (depicted in figure 4.4), that describes the transformation from the marker rigid body coordinate frame to the actual camera coordinate frame. In AutoPOSE, 50 images and poses, thus building and over-saturated system of equations, and solving for the needed transformation with high accuracy. The error was analyzed using the computed transformation and the captured images. Since the checkerboard patter is known, it get projected onto the image using the camera pose and the computed Hand-Eye transformation $T_{marker2camera}$. A pairwise corner difference is computed with the projected checkerboard corners and the detected corners in the image. The re-projection error is 2.19 pixels in average for the Blackbird infrared camera. The computed error for the Kinect v2 color camera is 3 pixels, and for the depth camera is 2.3 pixels in average. The error in the Kinect v2 color camera is slightly higher than the depth camera of the Kinect v2 and the Blackbird dashboard camera since the color camera has a much higher resolution, thus a slightly higher error is still acceptable taking the resolution difference into account.

System synchronization

During the recording session of a subject, the data acquired by the AutoPOSE setup is flowing from various sources. The OptiTrack provides the 3D position and orientation of the defined rigid bodies to be tracked, which are the subject's head (explained in the next section), the acquisition cameras, and the car cabin. Besides the OptiTrack data, the acquisition cameras provide image data in a continuous manner during the session. Revising the aim, which is to provide the head pose in the camera coordinate frame, requires the synchronization between the data of the OptiTrack and the captured frames from the cameras. The synchronization can be achieved by several methods, for example in [143], the authors started recording with the Kinect on a tripod in a stand still position (which is also tracked by the motion capturing system), then performed a strong impulsive push on the tripod. By detecting the frame were the sudden movement occurred, and aligning that frame with the motion capturing system data, the authors synchronized both systems. They measured an error of average $8.77ms$, however, that did not impose a problem due to the fact that the capturing system is 4 times faster that the Kinect. In this work, the dashboard camera is running at $60Hz$, and the OptiTrack is running at $120Hz$, and a strong impulsive movement or a sudden movement that appears in the camera could be used to synchronize both systems. However, some drawbacks can be noted in that method, the methods assumes a perfect frequency from all systems during the whole recording session, however, practically this might not be the case due to data transfer issues. Moreover, moving the camera suddenly is not possible since the cameras are rigidly fixed to the car cabin simulator, and moving reflective markers suddenly could damage the markers on the camera, leading to calibration issues.

In AutoPOSE, the time was synchronized using the hardware time of the PC used to acquire data from both systems. The OptiTrack and the cameras were connected to the same PC. The data from OptiTrack was acquired using the NatNet C++ SDK [87], which captures the data from a local stream on the PC. The Motive software streams the data live to a local server on the PC, and NatNet captures the data from it. Along with the saved tracking data, the machine hardware timestamp was stored with every frame of data. Also the camera acquisition software stored the same hardware timestamp of the same PC along with the image acquired from the camera at the time of frame acquisition. This time synchronization methods allows continuous time synchronization between the two systems, the camera and the OptiTrack. The frequencies for the OptiTrack, dashboard camera, and the Kinect v2 are $120Hz$, $60Hz$, $30Hz$ respectively. It was possible to select the closest data frame from the OptiTrack data to each image data, which was for most of the cases with $0ms$ difference, and at most $5ms$ for a very small fraction of the data.

4.3 Head Coordinate Frame

As mentioned in the previous sections, one of the rigid bodies to be tracked by the OptiTrack is the head target. The head target is depicted in the coordinate systems diagram shown in figure 4.4. The *head target* is composed of 8 reflective markers and are glued to a structure. The tracking of the markers is performed with error as small as $0.32mm$ as reported by the OptiTrack calibration software. The structure is worn by each subject during the recording session, and fixed on the subject's head. The rigid body defined by the head target has its own coordinate frame (as show in figure 4.4). The frame of the rigid body does not necessarily represent the coordinate frame of the subjects head. Every subject will put the on the head target differently, thus leading to inconsistent data among different subjects. Moreover, making assessments for head orientation estimation algorithms will not be possible due to the fact that the angles representation will be not the same among different subjects. Consequently, it is of utmost importance to have one specific definition that can be applied to all subjects, and can also be applied by other researchers in the community if they decide to create more datasets.

In literature the head coordinate system is not always precisely defined as in [103, 5, 33], where the authors provided algorithms for head orientation estimation. The method used can be valid for a specific dataset with a specific algorithm, but it is hard to test the same algorithm on another dataset. Later in literature, more works moved toward providing real dataset, like the work presented in [4], where groundtruth information about the head orientation angles were gathered using IMUs attached to the subject's head. The zero angle position is calibrated at the beginning of the recording session where the subject looks into the camera and is roughly in the center of the image frame of the

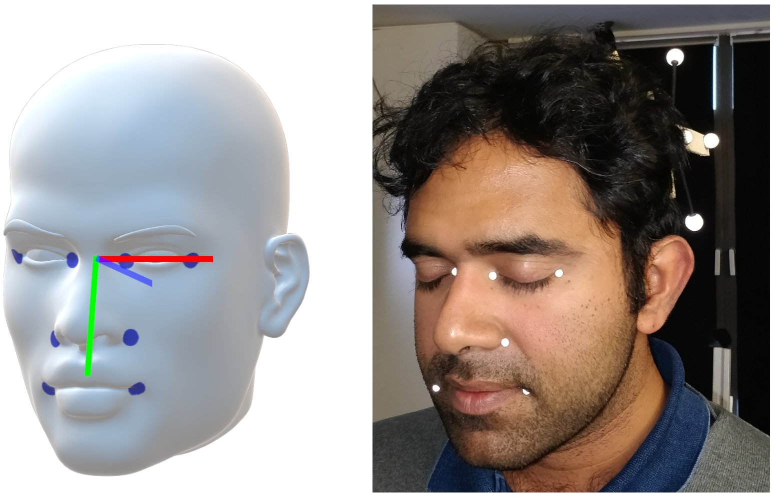

(a) (b)

Figure 4.6: **Head coordinate frame definition.** *(a) An graphical illustration, showing the x, y, and z axis of the head and the head center. The coordinate frame is defined by the 8-subject-specific points on the face. (b) A subject with the special facial markers on his face, which are required to define his own head coordinate frame.*

camera. In 2017, Schwarz et. al. [112] proposed the DriveAhead dataset with a specific head coordinate system definition. Consequently, it can be concluded that existing public datasets have different head coordinate system definition. In AutoPOSE, it was decided to use the same head coordinate system definition proposed by the authors in [112]. By applying the same definition, AutoPOSE will contribute more data to the research community that can be used directly to assess head orientation estimation algorithms without the need of head frame definition adaptation from one dataset to the other. The definition is described in detail in the next subsection.

4.3.1 Definition

In order to define a coordinate frame, it is required to define the x, y, and z axis of the frame with respect to the subject's head, and the origin of the coordinate frame, which is referred to as the head center. Practically, the head coordinate frame definition, requires 8 landmarks on the face, which are four eye corners, two nose corners, and two mouth corners. The head center is the 3D mean point of the four eye corners. The x-axis is defined to be the 3D vector that starts at the head center, and passes between the left eye corners. The y-axis is computed as follows. The 3D mean point of the two nose corners and two mouth corners is projected on the plane whose norm is the x-axis. The projected point and the head center define the y-axis of the head. Finally, the z-axis is the cross product of the x and y axis. Figure 4.6 shows a graphical diagram of a head with the 8 landmarks on the face, along with a picture of a subject with special facial reflective markers from OptiTrack.

4.3.2 Head Calibration to OptiTrack

In order to describe subject-specific head reference system in the OptiTrack coordinate frame, the subject puts on the head target, so that it rests at the back of the head, and does not occlude any part of the face. The experiment coordinator puts on the subject's 8 special facial markers from OptiTrack at the designated positions mentioned before. A calibration sequence of at least 1 minute is recorded by OptiTrack, where the head target and the facial landmarks are visible. During the calibration sequence, the subject rotates his/her head in yaw, pitch, and roll directions. A subject specific head coordinate frame is computed for each frame of the OptiTrack data. The rigid transformation from the head coordinate system to the head target is computed. Finally, we an average rigid transformation among all frames is computed. This computed rigid transformation $T_{h \rightarrow hTraget}$ defines our subject-specific head calibration. The facial markers are then removed, and the subject is now ready for recording the dataset sequence.

In order to compute the head calibration error, the calibration sequence is used. The computed head reference system at each frame of the OptiTrack data is considered the reference for that specific frame. The computed transformation is used to compute the head frame from the head target, giving the recovered head pose. The reference frame and the recovered frame are presented as quaternions, and the angle between both quaternions vectors is the error in the rotation between the actual reference head pose and the recovered one. The error in rotation is computed as follows

$$RotationError = 2 \times \arccos\left(\|reference \cdot recovered\|\right) \tag{4.2}$$

On average, for all subjects, the rotation error for the head calibration is as small as **1.6 degrees**. Fore the translation component of the computed transformation, the error is the euclidean distance between the translation components, and on average it was as small as **1.02 mm**. Consequently, the head calibration method applied yields a small acceptable error in both rotation and translation components, thus making the recovered head poses reliable for use as groundtruth for the AutoPOSE dataset.

Finally, in order to describe the head pose (translation and orientation) in the camera coordinate system, by tracking the head target and the camera rigid bodies in each frame. The following transformations computes the head pose in the camera frame, the set of transformations is depicted with arrows in figure 4.4

$$T_{h \rightarrow c} = T_{cMarker \rightarrow c} \times T_{OT \rightarrow cMarker} \times T_{hTarget \rightarrow OT} \times T_{h \rightarrow hTraget} \tag{4.3}$$

4.4 AutoPOSE dataset

After presenting the system components required to capture a head pose dataset, and explaining in details the system calibration and synchronization of all components together, this section presents the acquired dataset, AutoPOSE. As mentioned before, in the dataset, there exists two cameras for acquisition, the Blackbird infrared camera located at the dashboard, and the Microsoft Kinect v2 located at the car center mirror. The two subsets are presented in this section. Sample images from the dashboard subset can be seen in figure 4.8 and sample images from the center mirror subset can be seen in figure 4.9.

Recorded Data

The AutoPOSE dataset main aim is to provide the head orientation and position in 3D space with respect to the camera frame accurately. However, the dataset also provides information about the gaze target of the subject. The gaze target tasks were part of the tasks that are to be performed by the subjects. The groundtruth information related to the gaze is the gaze target, and the corresponding frames. The best person to annotate the frames for gazing at a specific target is the subject. The subject was provided by printed numbers labels that gets held in front of the camera before gazing at the target point. Besides, the subject was given a physical button switch, that when the gaze starts, the switch is turned on. The switch controls an IR lamp that appears in the background of the frame, and does not interfere with the light emitted by the cameras. The subject was asked to gaze at six positions in the car simulator. Six reflective markers were tracked by OptiTrack, and thus can be described in the camera frame using the calibrations described before. The markers were placed at driving-related locations, the dashboard, in front of the driver (representing looking at the road), center mirror, 2 side mirrors, and center of the car (representing media center, climate control). The subjects were asked to gaze at each marker for 5 to 10 seconds. The groundtruth gaze targets can be used to in gaze estimation algorithms assessment in automotive or other fields. To the best of our knowledge, this is the first time to provide gaze target groundtruth in automotive field using IR camera from dashboard and from center mirror views.

Along with the OptiTrack motion capturing system, two cameras were used to acquire the AutoPOSE data. The dashboard camera captured 21 sequences. The 21 subjects were 10 females and 11 males, and all of them had driving experience. The dashboard IR camera was running at 60 fps, giving in total **1,018,885** IR images. This amount of data is useful for training deep neural networks to estimate the head pose. The data acquisition was deep learning oriented, in other words, it was important to capture sufficient amount of data that would be useful for using in deep learning algorithms. The Kinect v2 was running at 30 fps, giving in total **316,497** synchronized RGB, depth, and IR images. The data from the Kinect v2 consists of 16 sequences. In each capturing sequence, the subject was asked to perform the tasks listed in Table 4.3. First, the subject was instructed about

Task ID	Task description
Task 1	Pure yaw rotation
Task 2	Pure pitch rotation
Task 3	Pure roll rotation
Task 4	Free natural motion
Task 5	Free natural motion - Hand near face actions
Task 6	Gaze point 1 - Left mirror
Task 7	Gaze point 2 - Right mirror
Task 8	Gaze point 3 - Dash board
Task 9	Gaze point 4 - looking forward at the road
Task 10	Gaze point 5 - Back mirror
Task 11	Gaze point 6 - Media center

Table 4.2: **AutoPOSE subject's tasks.** *The description of the tasks performed by the subjects in the AutoPOSE dataset.*

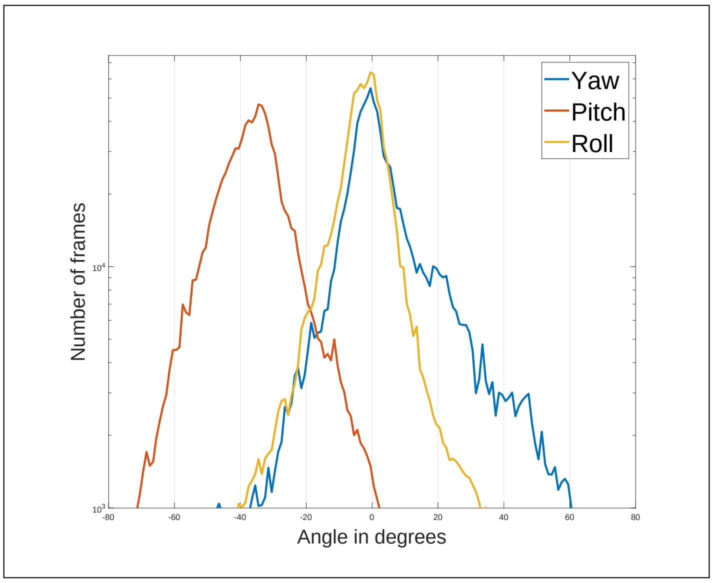


Figure 4.7: **AutoPOSE angles distribution** *AutoPOSE dashboard camera set - Acquired angles (yaw, pitch, roll) distribution histogram.*

Task ID	No glasses	Clear glasses	Sunglasses	Neck scarf	Total
Task 1	12k	12.5k	13.5k	11.7k	**50k**
Task 2	12k	11.7k	12.6k	11k	**47.5k**
Task 3	13k	12k	12.5k	11k	**48.5k**
Task 4	374k	153k	158k	22k	**705k**
Task 5	40k	-	-	-	**40k**
Task 6	7.2k	6.5k	6.7k	-	**20.5k**
Task 7	7.6k	7k	7k	-	**21.6k**
Task 8	7.5k	7.2k	7.5k	-	**22.2k**
Task 9	7.3k	6.6k	7.1k	-	**21.1k**
Task 10	6k	6.8k	7.1k	-	**19.9k**
Task 11	7.4k	6.6k	6.6k	-	**20.6k**
	495k	**230k**	**238k**	**55k**	**1M**

Table 4.3: **Frames breakdown per task.** *Each row represents the task performed, and each column shows the accessory type. Last row/column counts the sum of frames in the row/column. The total number of frames in the dashboard subset is 1 Million frames.*

all the tasks required. The subject performed pure rotations as much as possible, followed by free natural motion, with and without face occlusions using his/her hand.

All tasks were first performed without any glasses on the face of the subject. Later on, all tasks were performed again with clear glasses on, then with sunglasses on. After acquiring the data with the dashboard camera, the whole experiment was repeated again using the Kinect camera while turning the dashboard IR camera off. It was noted that the subjects were faster in performing the tasks again for the Kinect sequence, thus leading to less frames for the Kinect sequence. All tasks for all of our 21 subjects were manually annotated stating the start/end frame, along with the task performed, and glasses existence with its type. A histogram representing the yaw, pitch and roll angles distribution in all frames from the dashboard camera is shown in figure 4.7. The rotations were limited to -90 degrees to +90 degrees. As shown, the pitch angle histogram is shifted in the negative values of the rotation angles. This is due to the placement of the camera in the dashboard, where it is looking at the face from the bottom, check Figure 4.8.

Evaluation Metrics

To provide a good benchmarking foundation, meaningful metrics for the head pose estimation task are required. Consequently, 4 metrics as a basis for further benchmarking were chosen. The first metric is the angle estimation error, that is referred to as Mean Absolute

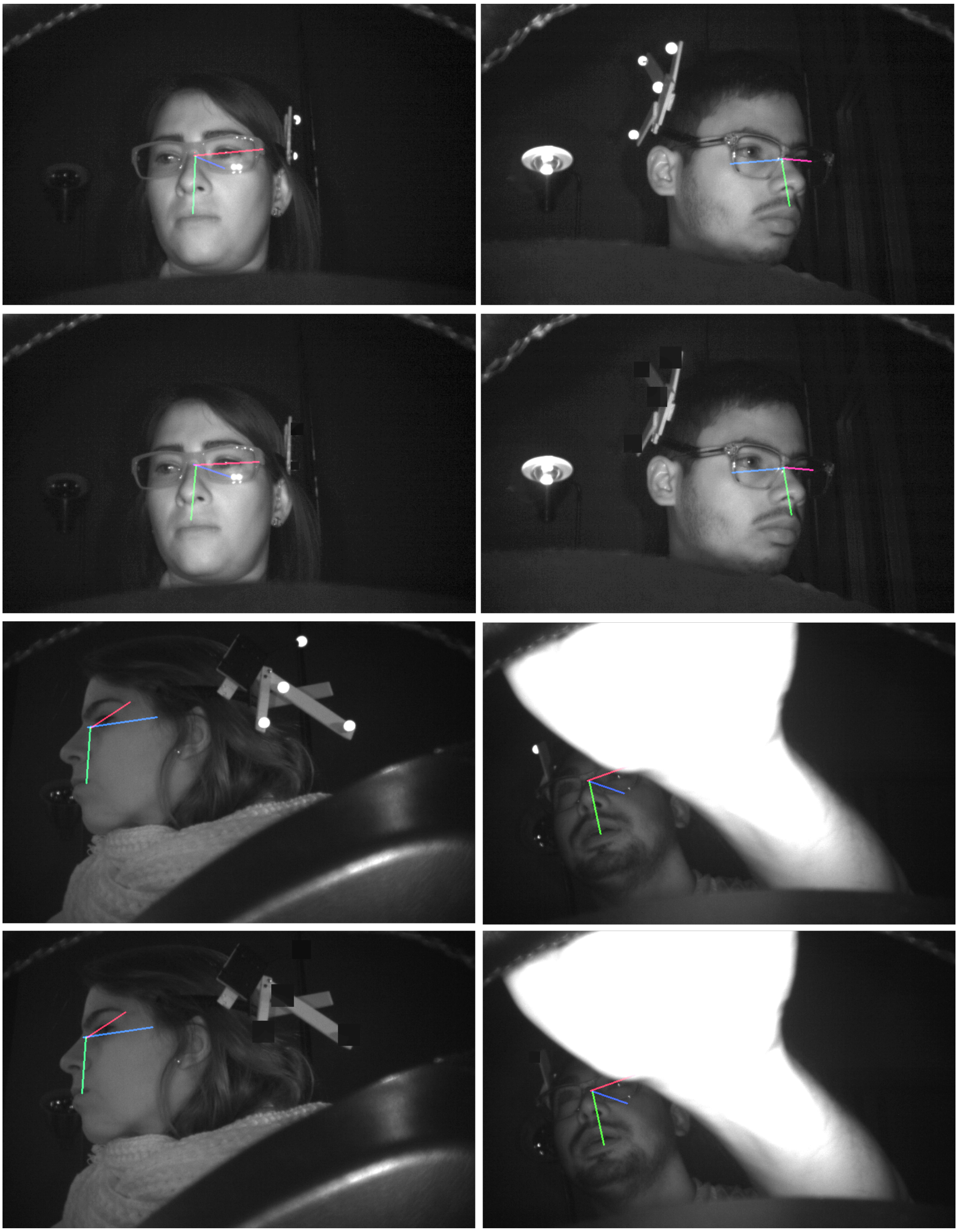


Figure 4.8: **AutoPOSE samples: Dashboard camera set.** *Row 1: RAW images with head target reflective markers visible, Row 2: post-processing - markers covered. Second column shows gaze annotation lamp. Rows 3 and 4 are similar to rows 1 and 2, but they also show the extreme rotations performed by the subjects and show sample occlusions.*
Note: Intensity was improved for visibility and printing purposes.

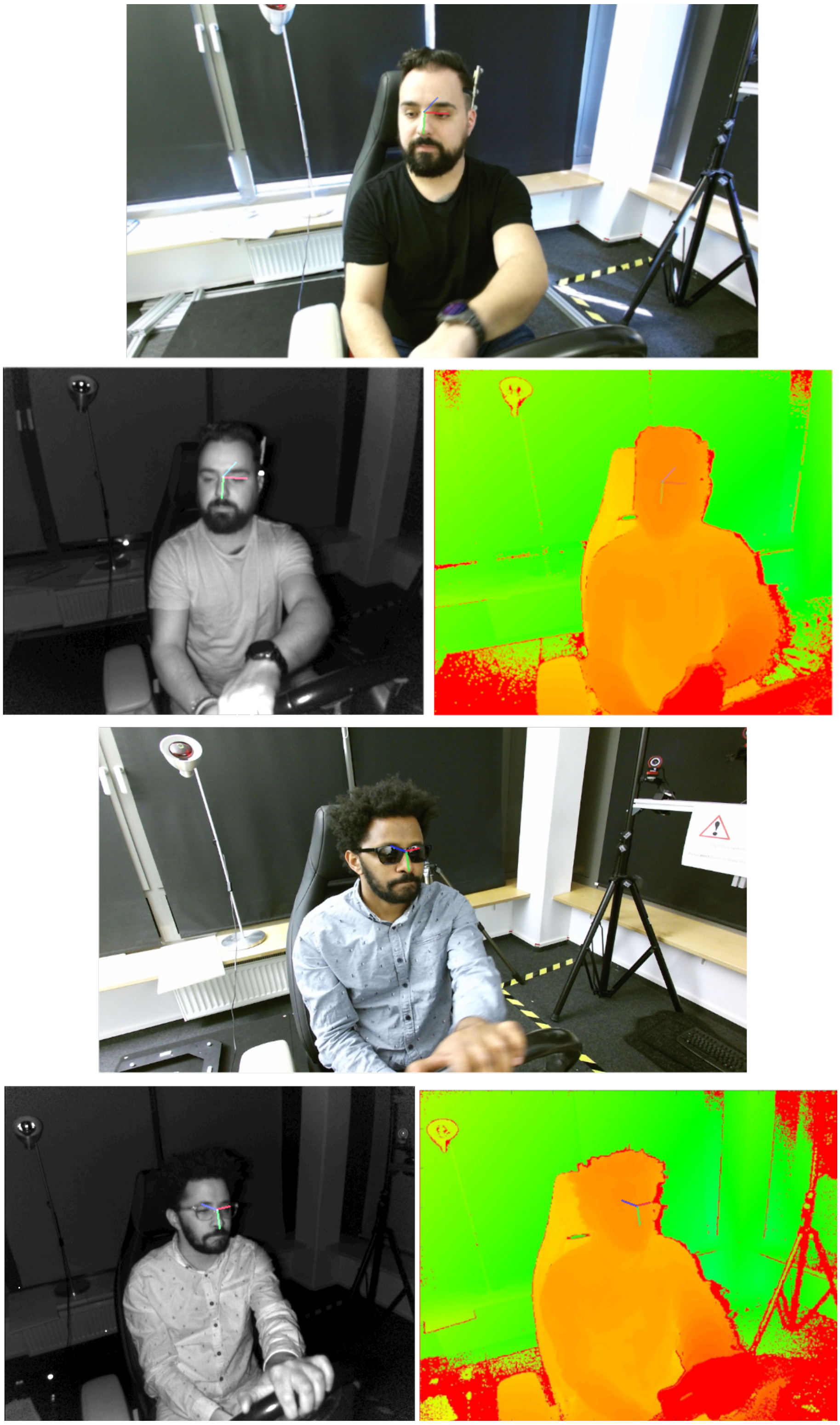


Figure 4.9: **AutoPOSE samples: Center rear mirror set** *Kinect color, IR, and depth(color mapped) images.*
Note: Intensity was improved for visibility and printing purposes.

Error (MAE).

$$MAE := \frac{1}{n}\sum_{i=1}^{n}|y - \widetilde{y}| \tag{4.4}$$

It provides an easily comprehensible metric. Computing it on one axis or all axis result in the total estimation error on the respective input. Another metric is the Standard Deviation (STD), providing further insight to the error distribution around the groundtruth.
The third metric is the Root Mean Squared Error (RMSE) to weigh larger errors higher.

$$RMSE := \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \widetilde{y})^2} \tag{4.5}$$

It takes the squared difference of the predicted value and the groundtruth value, weighing larger errors higher. Thus, high variation in predictions of an algorithm result in a higher overall error compared to the mean without squaring the values. Computing the mean over one or all axis and subsequently calculating the square root of the outcome produces the same unit as the predictions and groundtruth, thus making it more understandable.

The last metric is the Balanced Mean Angular Error (BMAE) as defined in [112], which provides further insight as it takes different ranges into consideration. The authors base their metric on the unbalanced amount of different head orientations due to driving and its bias towards frontal orientation. The BMAE addresses this:

$$BMAE := \frac{d}{k}\sum_{i}\phi_{i,i+d}, i \in d\mathbb{N} \cap [0, k] \tag{4.6}$$

$\phi_{i,i+d}$ is the average angular error. In contrast to [112] which computes the difference based on quaternions, we compute it as $|y - \widetilde{y}|$ for all labels y and predictions $\widetilde{y}$, where the absolute distance of groundtruth angle y to zero lies between i and i + d. During the evaluation, the section size d was set to 5 degrees and maximum degree k to 120. The previously presented POSEidon-model was tested on the metrics to provide a baseline for future head pose estimation benchmarking.

4.5 Summary

In this chapter, a new large-scale driver head pose and eye gaze dataset was introduced. The system components required to acquire a highly accurate head pose dataset were presented and described in details. A car cabin simulator have been used as the mockup car for the subjects to sit in and perform different tasks. A sub-millimeter motion capturing system have been used to acquire the track the subject's head, and the acquisition cameras in 3D space. An infrared camera have been used at the dashboard location in the car model, and a Microsoft Kinect v2 has been fixed at the center mirror position. Moreover, car landmarks have been tracked by the optical motion capturing system. The car landmarks were used as gaze targets by the subjects.

After introducing the system components, the system calibration was presented in section 4.2. The section covered different calibrations that are of utmost importance for achieving highly accurate groundtruth data. The optical motion capturing system was calibrated using off-shelf commercial software from OptiTrack. The intrinsic of the acquisition cameras were calibrated, and the camera frames were calibration to the motion capturing system using the hand-eye calibration method, allowing describing the pose of the head as a rigid body in the coordinate frame of the acquisition cameras. The orientation of the head in 3D space is described using a rotation matrix, which is created from 3 orthogonal axis defining the head coordinate frame. Then, the head coordinate frame was discussed in details. The model used is consistent with the model described in DriveAHead dataset, thus, adding more consistent data to the research community. Algorithms that were developed using the DriveAHead dataset can be used directly on the AutoPOSE dataset directly without head coordinate frame adaptation. The technical method to create the subject-specific frame is described in details and can be reproduced.

In section 4.4, the acquired data is presented. Data was captured from two positions in our car simulator for 21 subjects (10 females and 11 males). The amount of images collected was 1.1M images from the dashboard IR camera and 315K images for each type from Kinect v2 (RGB, Depth, IR). Groundtruth of head pose of all frames of the dataset head pose was acquired using a sub-millimeter accurate motion capturing system. Moreover, all frames of the dataset were annotated with information about driver's activity, face accessories (clear glasses, and sunglasses) and face occlusion.

The AutoPOSE dataset has a big potential for the research community. Due to the high quality of the data and the associated groundtruth, the dataset can be exploited for not only head angles estimation from cropped faces, but can be also used for evaluating the placement of the camera sensors inside the car, as the AutoPOSE provide data from center mirror perspective. Since a Kinect v2 was used, the data can be used to train and evaluate generating one modality of data from another, as the Kinect v2 provide synchronized IR, depth, and color images. Besides the head orientation data, the eye gaze frames are annotated, with the gaze target groundtruth available in the camera frame, allowing the dataset to be used for assessing gaze estimation algorithms. In conclusion, the AutoPOSE dataset opens the door for the scientific community to use high quality data for further problems and applications related to driving tasks.

5 Deep Learning for Head Orientation Estimation

In this chapter, a deep learning-based baseline algorithm for estimating the head orientation on the AutoPOSE dataset is presented. Moreover, further analysis on the deep learning network types and the face image size is presented.

The work presented in this chapter has been published in the following articles:

Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker.
AutoPOSE: Large-Scale Automotive Driver Head Pose and Gaze Dataset with Deep Head Orientation Baseline. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

Ahmet Firintepe, Mohamed Selim, Alain Pagani, and Didier Stricker.
The More, the Merrier? A Study on In-Car IR-Based Head Pose Estimation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

Contents

5.1 Baseline Network Architecture 83
5.2 Network Training 84
5.3 Head Orientation Baseline Evaluation 85
5.4 Face Resolution and Network Depth 85
5.5 Summary 90

Approaches for head pose estimation are performed either on 2D information like RGB [6, 104], or IR images [112], or on 3D information like depth maps [16, 15]. The selection of the suiting input type depends also on the category of an approach. Three main categories have been defined to classify approaches: feature-based, 3D model registration and appearance-based approaches [33, 81, 16]. Feature-based approaches need defined facial features like eye corners or mouth corners, which are then localized in frames to perform pose estimation. These approaches can work on 2D as well as 3D information.

In [7], two different feature-based approaches have been combined to regress head pose, the approaches being defined facial landmarks on the face and keypoints computed by motion. The approach requires 2D images only. 3D model registration derives a head model from the data and regresses a head pose depending on the derived 3D information. This can be done based on 2D and 3D or both. [97] uses facial point clouds and matches them with possible poses.

Appearance-based approaches take the whole information provided into consideration and try to regress a pose. They are generally learning-based methods. This can be either a raw 2D image or a depth map, as in the DriveAHead-approach [112]. The DriveAHead-approach uses both, 2D-IR-images and depth information to regress a pose. The POSEeidon-framework [16, 15] uses 3D-information only to derive other types of information like motion and grayscale image to regress the 3D orientation. The baseline method we use in this work is based on deep convolutional neural networks, which has proven to have high potential for the head pose estimation task as shown by [16, 15, 2, 1], however, requiring large amounts of data. POSEidon-CNN was used on the IR data to perform head pose estimation. Before training, the raw images were cleaned based on head visibility to obtain cropped faces of the frames. The frames with yaw rotations higher than 120 degrees were kept for training to increase robustness, but were not considered in the validation and test set. All frames were equalized and normalized.

The authors of [16] and [15] rely on the output of a neural network to regress 2D head position, which they further use for cropping. This outputs the head center in image coordinates (x_H, y_H). The groundtruth data of the head center was used to crop the face instead of a neural network, as the main focus in this baseline is the head orientation estimation. This prevents having additional error in the pose estimation part introduced through another position estimation method. A dynamic size algorithm provided the head bounding box with the acquired head center, the width w_H and the height h_H, which are used to crop the frames. They were acquired as described in [16]. With the horizontal and vertical focal lengths of the acquisition device, distance D between the head center and the acquisition device and R_x and R_y, which are the average width and height of a face. The head width R_x and height R_y in 3D were defined uniformly as $32cm$, so the head is equal in size inside the cropped images. Moreover, if more than a third of the head were not visible in the frame, the cropped frame was discarded.

5.1 Baseline Network Architecture

The part of the POSEidon-framework (recent head pose estimation framework) [15], was considered in this baseline method. The framework relied on depth data and did not perform Head Pose Estimation on IR images. The head pose estimation part in the framework is based on three different branches, which considers depth maps, grayscale images generated from depth maps and motion images. All branches were trained with the same CNN architecture. The output of the three branches is fused at the end. The model exploited *Dropout* as regularization ($\sigma = 0.5$) at the two fully connected layers. The modified CNN model was trained and tested on the data of the dashboard IR camera, providing baseline results for the dataset.

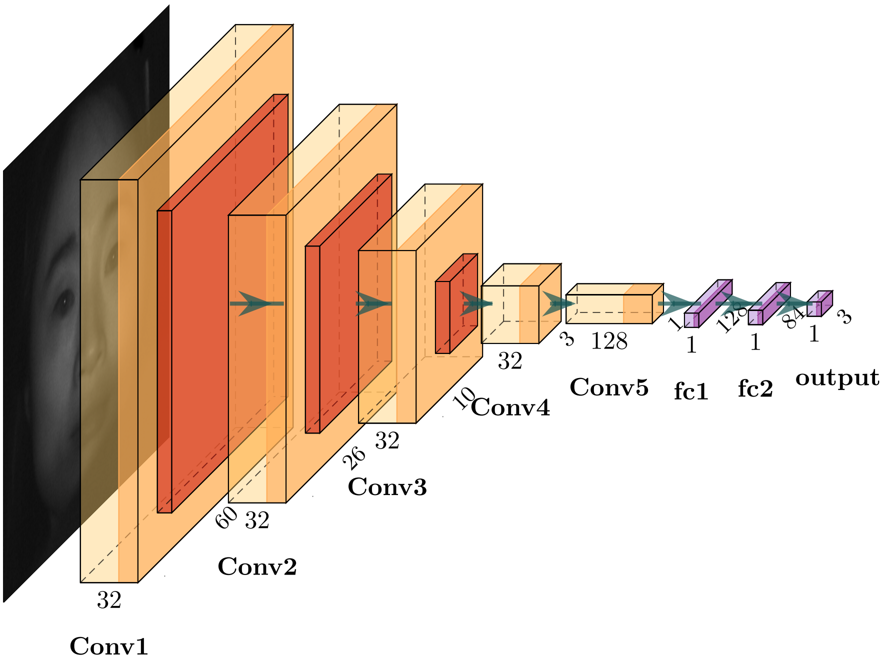


Figure 5.1: **AutoPOSE - Baseline CNN** *The deep neural network which was originally proposed by [15]. Originally the input was a depth image. In this work, the CNN's input is a* 64 × 64 *infrared image. The image goes through five convolution and pooling layers, then follows the fully connected layers. The output of the network is the regressed orientation angles.*

Metric	Model	Pitch	Roll	Yaw	Avg
MAE	Baseline	**2.96**	**3.16**	**3.99**	**3.37**
	ResNetHG18	4.02	3.32	5.20	4.18
	HPN (DriveAHead)	8.18	6.68	13.31	9.39
STD	Baseline	**4.63**	**3.93**	**7.82**	**5.46**
	ResNetHG18	6.25	4.98	11.57	7.60
	HPN (DriveAHead)	10.36	9.62	21.68	13.89
RMSE	Baseline	**4.73**	**4.55**	**7.98**	**5.97**
	ResNetHG18	6.37	5.20	11.58	8.20
	HPN (DriveAHead)	11.25	9.70	21.69	15.18
BMAE	Baseline	**7.10**	**9.42**	**19.05**	**11.86**
	ResNetHG18	12.18	13.58	35.41	20.39
	HPN (DriveAHead)	20.96	23.66	59.69	34.77

Table 5.1: **AutoPOSE - Results - Face size** 64×64**.** *Results on the* 64×64 *pixel cropped face images. The deep learning models were trained and tested using infrared dashboard set.*

5.2 Network Training

The modified Deep Neural Network was trained on the cropped images of the dataset. The selected training and test setup including loss function and training, validation and test set definition is described as follows accordingly. Also, in order to evaluate the model, benchmark metrics were defined and tested on the dataset. Regarding the training of the deep neural network, the loss function was defined as presented in [16, 15] to put more focus on the yaw, which is predominant in the automotive context. The labels are in the data range from -180 degree to 180 degree. A weighted L_2 loss between label and prediction was used, giving a different weight to each angle describing the head orientation angles. As mentioned, the yaw angle is the most dominant angle in the context of driving, consequently it was given the most weight of value 0.45 in the loss function. The pitch was given a weight of 0.35, and the least dominant angle, the roll, was weighted with value of 0.2. Thus, adding more penalty for errors in the yaw angle, relative to the other angles. This method adds the balance in the training as the variation in the yaw is the most and the least in the roll angle. Furthermore, 19 sequences out of the total of 21 sequences were used for training. One sequence was used for validation and one for testing. The training was performed in batches with a size of 128, where the batches were chosen randomly. The models were trained on 4 Nvidia Geforce GTX 1080 Ti until convergence.

5.3 Head Orientation Baseline Evaluation

The evaluations for head orientation estimation on all metrics are shown in table 5.1. The results showed the performance of the POSEidon-CNN on the 64 pixel cropped face images. It is observed that the CNN had a lower error than 3.5 degree on the MAE. The BMAE shows that the networks performed worse if more extreme poses with less examples are weighted equally as more common poses. In general, it is noted that the yaw angle is more challenging as the network performed worse on the yaw on all metrics compared to the pitch and roll. The results presented so far are the baseline estimations for the new AutoPOSE dataset. In the upcoming section, a more in depth analysis is performed using the AutoPOSe dataset is presented, that exploits the use of higher resolution cropped face images, and also presents new network architectures and exploits them on face angle estimation.

5.4 Face Resolution and Network Depth

After presenting the benchmark results on the AutoPOSE infrared images from the dashboard in [113], a follow-up study was conducted and presented in [36] to investigate other deep learning models for head pose estimation. The study investigated the effect of different resolutions of cropped faces on the accuracy of the deep learning models in estimation the head pose angles using infrared images. This work proposed novel networks like the Head Orientation Network (HON), and ResNetHG and compare them with state-of-the-art methods like Head Pose Network (HPN) from DriveAHead [112] on different input resolutions. Moreover, the IR image-based head pose estimation may have different requirements for deep learning, as they only have one channel and thus contain less information, but as mentioned before, more consistent looking scenes than RGB color images. The main scientific contributions of the work presented in [36] are:

- Novel algorithms for head pose estimation task that outperform state-of-the-art methods on IR images with HON and ResNetHG are presented.
- The effect of deeper networks and more face resolution on in-car head pose estimation is thoroughly studied.
- Performance gain is proved with fewer layers in deep neural networks on IR images.
- Higher resolution IR face images result in lower pose error.

5.4.1 Deep Neural Networks Architectures

In this subsection, the two new proposed models are presented. Both models are exploited and compared with other models using the IR face images, at different resolutions.

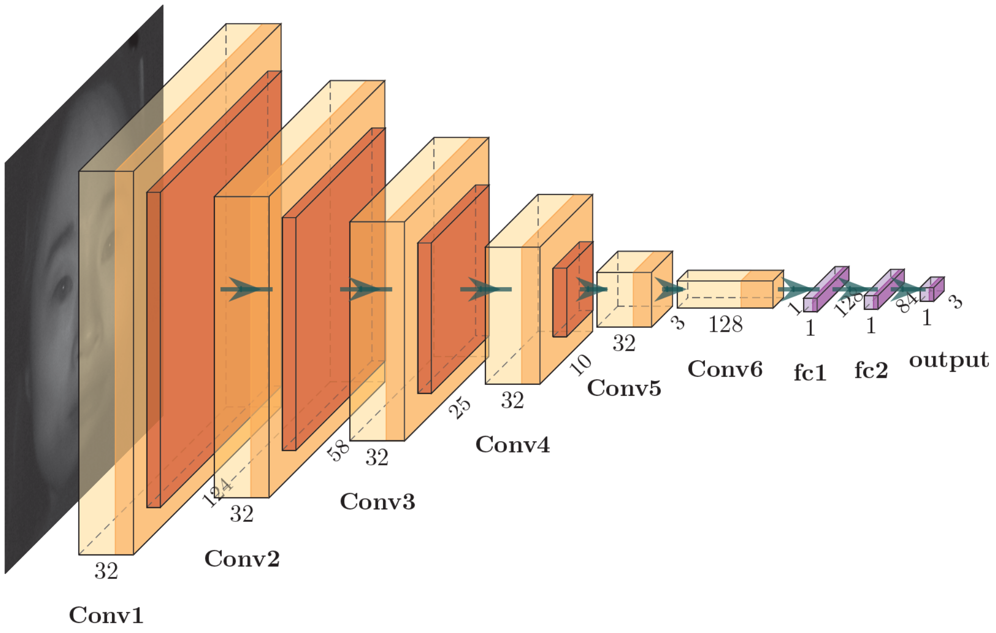


Figure 5.2: **Head Orientation Network - HON Model.** *The Head Orientation Network (HON), inspired by VGG [98] and the benchmark model. The input to the model is a* 128×128 *cropped face image. The output is the three estimated head orientation angles.*

Head Orientation Network - HON

In [36], a novel deep neural network model was introduced. It is an efficient model that is based on VGG [98] and the model benchmark model. The input image size is 128×128 pixels. The baseline input is 64×64, and has one less convolution layer compared to HON model. The HON is trained with Adam optimizer [64] with initial learning rate of value 0.00001. A drop out regularization of value 0.5 is exploited at the two fully connected layers. The HON model is shown in figure 5.2.

ResNet-based Model

One possible alternative to the previously mentioned models is a novel model based on Residual Neural Networks [47] and Hour Glass networks [88] architectures. The model maintains information and features within the building blocks of ResNet with skip connections between the blocks as described in [47]. Besides, lower level features that are available at the head of previous layers are being connected with later layers that deal with higher level features, similar to hour glass skip connections, as they are described in [88]. Consequently, coarse and fine-grained features are being exploited together to

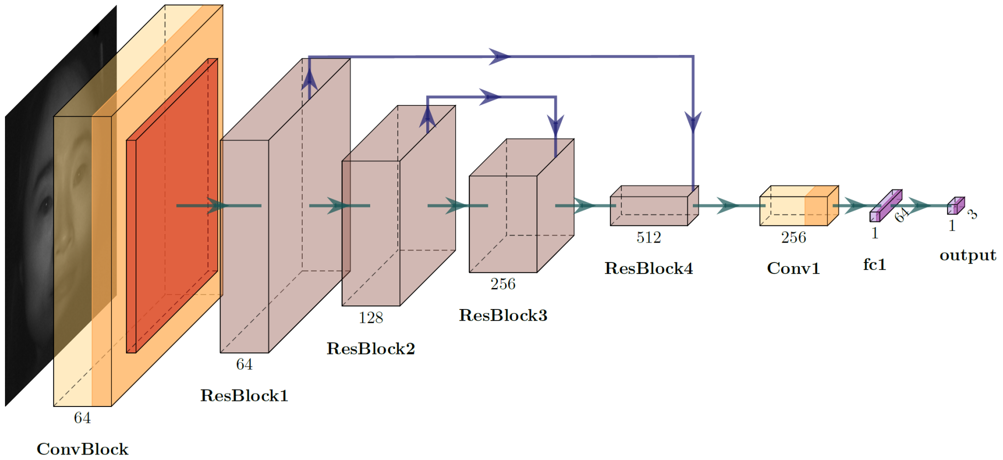


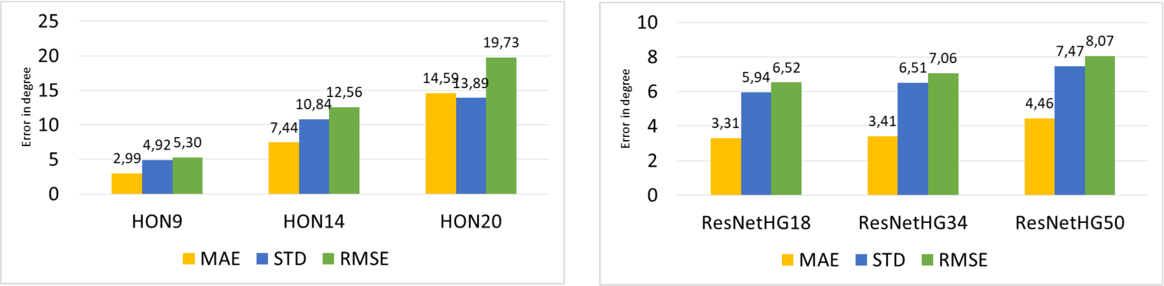
Figure 5.3: **ResNetHG Model.** *The ResNetHG-18 architecture. The ConvBlock and ResBlock are as explained in [47]. ResBlock1 is connected to ResBlock4, and ResBlock2 is connected to ResBlock3. The input to the network is the cropped face image, and the output of the network is the three head orientation angles.*

regress the head orientation angles. An overview of the model is shown in figure 5.3. Two additional skip connections are relized by first applying a Convolution with a stride of 8 or 2, respectively. Afterwards, the output of the source block is added to the output of the destination block, and applying the ReLu activation function.

5.4.2 Evaluations

The models training provided more information on the suitable architectures for the underlying problem of head pose estimation using IR images with different resolutions of 64×64 and 128×128. The HON model was implemented with different deep neural networks depths, with 9, 14, and 20 layers. Besides, architectures with different depths of residual neural networks were tested with IR cropped face images in order to deduce the amount of layers needed to solve the problem of head pose estimation with the highest possible accuracy. Results of this specific experiment are shown in figure 5.4 (a). The HON model with 9 layers yielded the best accuracy, compared to the HON model with 14, and 20 layers. Consistent results also are shown by comparing ResNet-18, ResnNet-34 and ResNet-50 as the basis including the added skip connections. ResNet-18 showed the most promising results as shown in the figure 5.4 (b).

Generally, it was found that by having more layers in the network, yields less accurate head orientation estimation results. The reason behind this could be due to the fact that



(a) (b)

Figure 5.4: **Results - Networks Depths.** *Comparing deep neural networks performances by differentiating the depth of the models (a) Head Orientation Network with 9, 14, and 20 layers (b) ResNetHG with ResNet-18, ResNet-34, ResNet-50 as a basis.*

grayscale images have only one channel. The RGB images have 24 bits of information at each pixel, but the grayscale and infrared images have only 8 bits at each pixel. This could lead to over-parametrization in the deeper architectures, thus, yielding more error in the estimations. Consequently, HON and ResNetHG models with the least layers have been exploited further in the next experiments.

Metric	Model	Pitch	Roll	Yaw	Avg
MAE	HON	**2.68**	2.73	**3.56**	**2.99**
	ResNetHG18	3.29	**2.48**	4.15	3.30
	HPN (DriveAHead)	8.32	6.87	13.81	9.67
STD	HON	**4.21**	**3.55**	**6.99**	**4.92**
	ResNetHG18	4.86	3.74	9.23	5.94
	HPN (DriveAHead)	10.36	9.62	21.68	13.89
RMSE	HON	**4.25**	**3.87**	**7.15**	**5.30**
	ResNetHG18	4.98	3.98	9.33	6.52
	HPN (DriveAHead)	11.38	9.81	21.76	15.27
BMAE	HON	**4.97**	**7.26**	**15.10**	**9.11**
	ResNetHG18	8.00	8.18	27.62	14.60
	HPN (DriveAHead)	20.93	23.96	59.4	34.81

Table 5.2: **AutoPOSE - Results - Face size 128×128.** *Results on the 128×128 pixel cropped face images. The deep learning models were trained and tested using infrared dashboard set.*

The baseline model performs better than ResNetHG18 and the DriveAHead model on the 64×64 pixels cropped face images as shown in table 5.1, being the smallest model.

By using the 128×128 cropped face images, the HON model achieves the best results in every benchmark metric, except for the roll angle on MAE, where the ResNetHG18 performed slightly better. The HPN from DriveAHead performed worse compared to all other models on both resolutions tested. Finally, in table 5.3, the average errors in all angles are shown for all models, metrics and resolutions. As shown in the table the HPN model is the model that yields the highest error. A considerable performance gain is seen using the higher resolution images in both the HON and the ResNetHG18 models. However, the ResNetHG18 model shows less accuracy when compared to HON model. This could be due to the fact that it has more layers inside the network architecture. The least error is achieved using the HON model on the 128×128 model. Consequently, the higher resolution and less layers in the deep learning network models achieve the best results on cropped faces from IR images.

Resolution	Model	MAE	STD	RMSE	BMAE
64×64	Baseline	3.37	5.46	5.97	11.86
128×128	HON	**2.99**	**4.92**	**5.30**	**9.11**
64×64	ResNetHG18	4.18	7.60	8.20	20.39
128×128		3.30	5.94	6.52	14.60
64×64	HPN (DriveAHead)	9.39	13.89	15.18	34.77
128×128		9.67	13.89	15.27	34.81

Table 5.3: **Comparing Effect of Resolution.** *Comparing using all metrics, the average error in all three head rotation angles on all models. The baseline model and HON are similar, ResNetHG18 model and HPN from DriveAHead [112] are trained and tested on the two different input resolutions,* 64×64 *and* 128×128

5.5 Summary

The AutoPOSE dataset has been presented in [113], and the dataset is publicly available to the research community for download at `https://autopose.dfki.de`. In this chapter, a deep learning-based method is introduced to provide a baseline for estimating the head orientation angles from the cropped face images on the AutoPOSE dataset. The baseline line takes the face image as input with size 64×64 pixels. The presented baseline model achives higher accuracy when compared to the DriveAHead model [112].

In a joint work presented in [36], the AutoPOSE dataset was further exploited for the problem of head orientation estimation from 2D cropped face images. A thorough study was conducted to investigate the sufficient depth of deep neural networks that would be suitable for solving the problem. Also, different face resolutions were experimented with different models. Two novel models were presented, and thoroughly evaluated. In conclusion of the study, using the AutoPOSE dataset, deep neural networks with fewer layers perform better on IR images to estimate the head angles from the face, when compared to deeper networks. Moreover, the presented models were evaluated on 64×64 and 128×128 cropped faces. The deep neural netowrks models performed consistently better on the higher resolution face images.

6 Gender Recognition

This chapter studies the problem of gender recognition from face images. It introduces the problem, and discusses the different techniques for estimating the gender from images. The method proposed solves the problem of gender recognition from face images under harsh conditions, including low quality images, and bad illuminations using an ensemble of deep Convolutional Neural Networks. The proposed method was validated on large scale datasets.

The work presented in this chapter has been published in the following article:

<u>Mohamed Selim</u>, Suraj Sundararajan, Alain Pagani and Didier Stricker. Image Quality-aware Deep Networks Ensemble for Efficient Gender Recognition in the Wild. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2018.

Contents

6.1 Deep Learning for Gender Recognition **95**
6.2 Image Enhancement **96**
6.3 Face Images Datasets **97**
6.4 Proposed GenderCNN **102**
6.5 Evaluation **105**
6.6 Summary **108**

Gender classification is an important problem in facial analysis applications. For humans, the face provides one of the most important sources for gender classification. Besides faces, also clothes, physical characteristics and gait [89] can provide information that identify the gender of a person. While this problem is a routine task for our brain, it is a challenging task for computers. Identifying gender from faces has huge potential in fields like face recognition, biometrics, advertising and surveillance. Millions of images and videos are uploaded every day to the Internet. These images and videos are captured using a variety of devices ranging from mobile phones to DSLR cameras, under varying conditions. These variations in the capture process, result in variations in headpose, illumination, resolution and noise, making gender recognition from faces challenging [89]. Gender recognition on videos is more challenging, due to the presence of blurriness in the video frames. A gender recognition learning-based method in general involves face detection, feature extraction and finding the gender from the features [89].

In the past few years, deep learning has been dominating different classification tasks [69, 98]. Convolutional Neural Network (CNN) is the most widely used neural network for visual recognition systems and natural language processing [72]. Deep CNNs came into the spotlight in 2012, when a deep CNN called AlexNet [66] outperformed traditional machine learning methods on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [111]. ImageNet is a 1000-class classification challenging competition. Deeper CNN architectures like VGG (16 or 19 layers) [98, 122] and Residual Networks that were presented with 18, 50, 100, or up to 150 layers [47], keep dominating the challenge. Gender recognition is a binary classification problem, however it is very challenging in unconstrained environments, where different variations exist in the images. It even gets more challenging in videos due to blurriness or quality when compared to still images.

Existing approaches like [77], use location of the eyes to transform faces to a canonical pose with eyes located in the same horizontal line. We propose a CNN has the ability to handle non-aligned faces as input to predict gender. The network has three convolutional layers. Deep learning methods require data to learn. The proposed approach employs several public datasets. Datasets of still images and video sequences have been exploited in this work. The proposed approach is evaluated large scale datasets. Due to the challenges available in data existing nowadays, new challenging ”in the wild datasets” have been introduced in the past few years. In this work, the still images datasets used are IMDB-WIKI dataset, which was introduced in 2015 and has more than 500k images with gender labels [109]. Also, the CelebA dataset was used in this work, and also was presented in 2015 [74, 125]. The dataset contains around 200k images with gender labels available in the meta-data of the datastet. The video datasets used in this work are the McGill-Faces dataset and the PaSC dataset. The McGill-Faces dataset consists of 60 videos, and the PaSC dataset consists of 2802 videos. The McGill dataset has gender labels available. The PaSC does not have gender labels, however, the videos were manually annotated with gender labels.

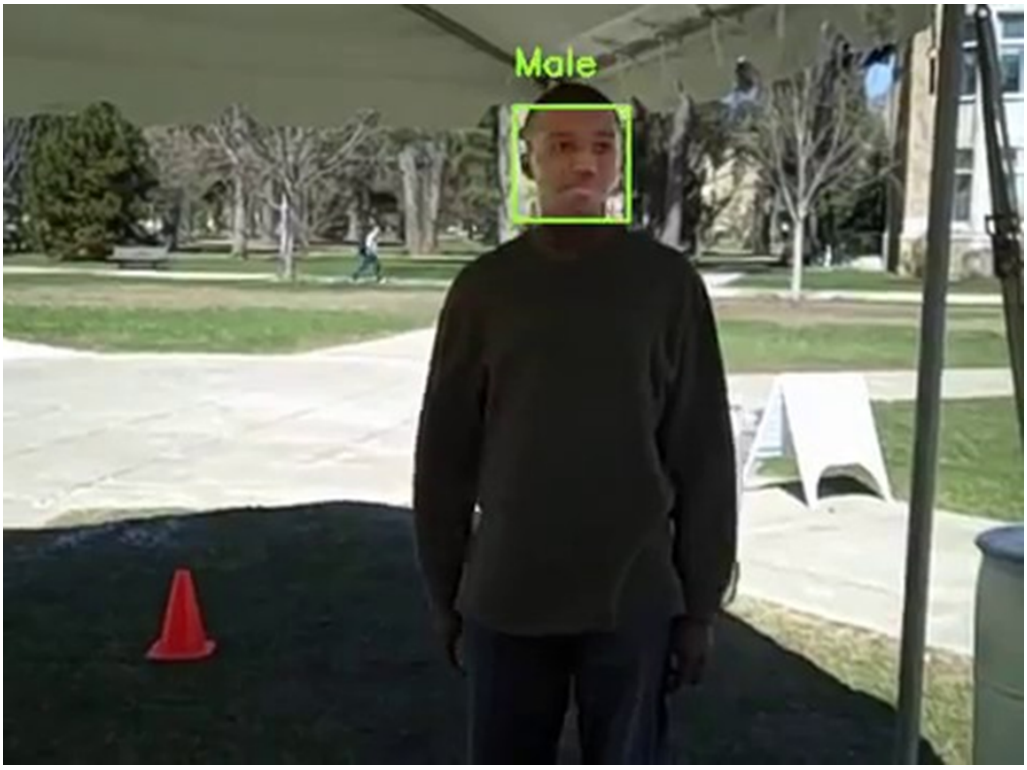


Figure 6.1: **Gender recognition under harsh conditions.** *A sample gender recognition using the proposed methods (AGC and EMBM GenderCNNs) for a frame from PaSC handheld set where the illumination not optimal, low quality frame, and low resolution image.*

Bad-illuminated faces or blurred ones exist in still images datasets and exist with higher frequency in videos datasets. In order to tackle these challenges, a novel image enhancement technique was employed in the pipeline, which is called Adaptive Gamma Correction (AGC) [102]. The AGC was used for pre-processing the faces in order to enhance the face image before the image is fed to the deep neural network. The impact of using AGC on faces as input to the network was evaluated. On the other hand, due to the movement of the subject or the camera, blurriness appears in video frames. Manually filtering the sharp and blurry faces is a time consuming process, and is not feasible on large scale datasets. Consequently, an automated method is necessary to separate the faces based on sharpness. One way of achieving this is to use a image blurriness metric. The problem of motion blur was tackled using the image blurriness metric described in [45], and called the Edge Model Blur Metric (EMBM). The metric provides a numeral value for the face image, describing how much blurriness exist in the image. The aim is to group together the faces that are similar in sharpness. The EMBM values are used to split the pre-processed faces into groups and separate CNNs are trained for each group.

Each group has faces that fall within a specific EMBM range.

The contributions presented in this chapter are:

- A compact, yet accurate deep neural network for gender recognition is presented, GenderCNN.
- The problem of illumination variations, and quality issues in still images are tackled using pre-processing algorithms.
- For videos, a blurriness-aware, based on GenderCNN, pipeline to detect gender on both sharp and blurred video frames is presented.

The presented method was thoroughly evaluated using public, challenging, and large scale datasets. The presented method outperformed the state-of-the-art in most cases, and scored second-best on some datasets. This chapter presents the proposed method, and its experimental evaluation.

The chapter is organized as follows, first, deep learning gender recognition methods are presented in the upcoming section 6.1. Afterwards, in section 6.2 image enhancement techniques are discussed. Section 6.3 presents the public datasets used in this work, and discusses the challenges available in each dataset used. In section 6.4, the proposed GenderCNN is presented, and the pipelines to solve the problem of gender recognition from faces are presented for both, still images and videos. Both pipelines use the novel network as a building block. Finally, in section 6.5, the proposed methods are evaluated and the results are presented. The chapter concludes with a summary of the proposed methods.

6.1 Deep Learning for Gender Recognition

Face has been widely used for gender detection as shown in the survey [89, 90]. Earlier works were constrained to frontal faces, and near-frontal faces, but were not able to handle profile faces. Recent methods handle varying head pose, illumination, and capturing location. Recently deep learning have been used in many applications, when the ILSVRC [111] was won by Deep Convolutional Neural Networks [66] in 2012.

Mansanet et. al. presented their work on gender classification using deep neural networks [78]. Liu et. al. [74] presented a deep neural network that predicts the gender of the person from the face image along with other facial attributes. In [74], authors introduced using two known CNNs atchitechtures, LNet and ANet. Both networks were pre-trained separately and jointly fine tuned. The LNet was used to localize the face and ANet was used to extract features for attribute prediction. One attribute was gender. They reported accuracy of 98% for CelebA and 94% for LFWA datasets. Similar to that work in [74] in terms of multi-task methods, Ranjan et. al. [104] proposed a deep neural network where they perform several predictions from the face image. They localize the face in the image, predict the location of specific facial landmarks, compute the pose, and predict the gender. They used AlexNet [65] initialized with ImageNet [111] weights. They used intermediate layers of the network for feature extraction. The features were classified using SVM. They were able to reach comparable results with [74] using much less data for training the network.

Another work that uses a deep CNN and SVM for age and gender recognition, is described in [69]. The CNN used has three convolution layers, 3 fully connected layers and 2 drop layers (for training). The dataset used for evaluation was Adience [30]. Adience dataset contains 26,000 images of 2284 subjects. The dataset is constrained with limited pose variations and no changes in location, therefore it was not used in the work presented in this chapter. Approaches with and without oversampling are described in the work. The method using oversampling achieved a gender recognition accuracy of 86.8% on Adience. Another deep neural network architecture was presented in [122]. The study indicates that the accuracy of image recognition tasks improved when the depth of the network was increased to up to 19 layers. Two deep architectures called VGG-16 and VGG-19 were proposed in the work. A face descriptor based on the VGG-16 architecture, was proposed in [98], for face recognition. The use of the descriptor can be investigated in gender recognition.

Gender recognition from still images gathered much more attention than gender recognition in videos. In [140], a temporal coherent face descriptor was introduced. Faces from the video frames are used to create one descriptor, and that descriptor is used to identify gender using a SVM. The datasets used in [140] were publicly available McGillFaces [24] and the author's private NCKU-driver database. The datasets had variations in illumination, background, head pose, facial movements and expressions. The method achieved an accuracy of 94.12% and 96.67% for McGillFaces and NCKU respectively. It is important to note that the accuracy is computed for the whole video, and not for separate frames. The method used was similar to majority voting, thus if 40% of the frames in one video were wrong predictions and the remaining 60% were correct predictions, the whole video is considered correctly predicted. In this chapter, challenges related to the acquisition process, especially in videos are tackled by using image enhancement techniques to get the most information out of the data and avoid discarding information completely. The next section discusses image enhancement techniques.

6.2 Image Enhancement

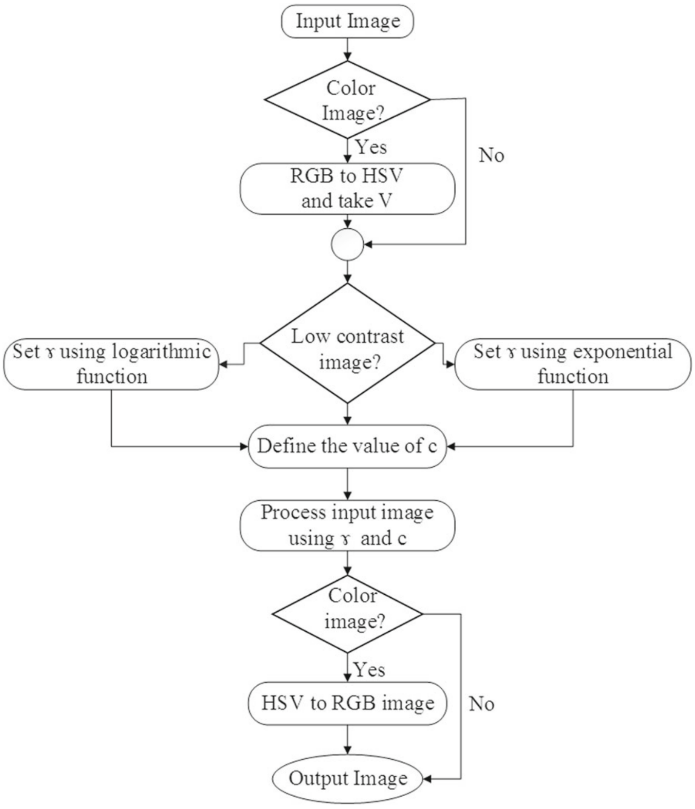


Figure 6.2: **AGC Functional Block Diagram.** *A functional block diagram for the adaptive gamma correction algorithm. Figure from [102].*

Pre-proccessing an image can refer to either image enhancement or restoration. Image enhancement can be required to get the most details available in the image. The details can be lost due to capturing conditions like illumination or the quality of the capturing device itself. The capturing conditions are not always optimal for example in images taken by amateurs using smartphones. Image enhancement is carried out to improve the image before analyzing it. Enhancement can be carried out to improve contrast, saturation, or brightness of an image. The illumination conditions affects the image contrast, consequently, important details of an image can be lost [102]. Global methods for image enhancement like histogram equalization can result in over or under-enhanced image, which also results in losing details of the image. Local methods use neighboring pixels to overcome the problems in global methods, however, they are computationally expensive. Hybrid methods use both local and global information form the image to improve the image, like Adaptive Histogram Equalization (AHE) or Contrast Limited Adaptive Histogram Equalization (CLAHE) [146], however they don't perform well on various problems, like dark or bright images.

An Adaptive Gamma Correction (AGC) method is proposed in [102], and it is able to enhance dark or bright images. In short, the AGC method classifies the image first as high or low contrast, and then classifies it as dark or bright. AGC can enhance the images that need enhancement without over or under-enhancing them. The main objective of AGC is to transform an image into a visually pleasing one through maximizing the detail information. This is done by increasing the contrast and brightness without incurring any visual artifact. To achieve this, AGC dynamically determines an intensity transformation function according to the characteristics of the input image. A functional block diagram for AGC is shown in figure 6.2. By taking several existing color models [41] into account, AGC operates on the HSV space and not the RGB space. The HSV space separates the color and brigtness information into hue (H), saturation (S), and value (V). HSV color model provides a number of advantages such as having a good capacity of representing a color for human perception and the ability of separating color information completely from the brightness information [131, 21, 55]. By altering the values of the RGB space, the color of the image will be altered. However, altering the V-channel does not modify the original color of the image. For further details about the AGC, please refer to the original paper [102].

A visual comparison of different image enhancement techniques is shown in figure 6.3. HE over enhanced the input image. Contrast stretching didn't improve the face, it is still dark. CLAHE over enhanced some face regions, however AGC improved the image without over or under-enhancing it. Facial details are more visible. It was decided to investigate the incorporation of the AGC, novel image enhancement method, in the proposed approach as a pre-processing step.

6.3 Face Images Datasets

One of the most important problems in machine learning is having the right data with the right annotations. Having the right data means gathering the data that correlates between the information available in the image data and the desired outcome. In deep learning, besides having the right data, it is important to have enough data. By enough, it is meant that it is the minimum amount of image data with annotation that would be sufficient for the deep neural networks to learn the underlying images with the associated annotations, in a way that would allow to network to correctly predict the outcome when seeing a new image. The problem in hand is basically identifying the gender of the person from the face image. Consequently, datasets with gender annotations are required. In this work, it is important to identify the gender from the face for data sourcing from still images and video frames. The faces in still image and video frames differ in appearance, due to the quality and blurriness that may appear in videos. Thus, it was noted to have both types of data in this work. The following subsections present the still images datasets and the videos datasets. Along with presenting the size and properties of the data itself, the annotations that are available along with the data will be also presented, and augmented wherever required, for example adding gender annotations.

6.3.1 Still Images Datasets

In 2015, Rothe and Gool introduced a new large scale dataset, with the name IMDB-Wiki dataset [109]. The authors crawled the internet websites of the Internet Movie Dataase (IMDB), and Wikipedia to collect about 500k images of celebrities, hence the name IMDB-Wiki for the dataset. The dataset is mainly for age estimation but it can also be used for gender classification. The dataset contains images "in the wild", which means that the images are unconstrained, they were all taken in uncontrolled environment. Sample images from the IMDB-Wiki dataset are shown in figure 6.4. The authors mentioned that the existing face datasets are either of small in size, and contain only frontal aligned faces. They tackled all the identified issues in the IMDB-Wiki dataset, where they collected half a million image, and of different poses. The authors used a list of 100,00 most popular actors as listed on the IMDB website and automatically crawled their pages to collect

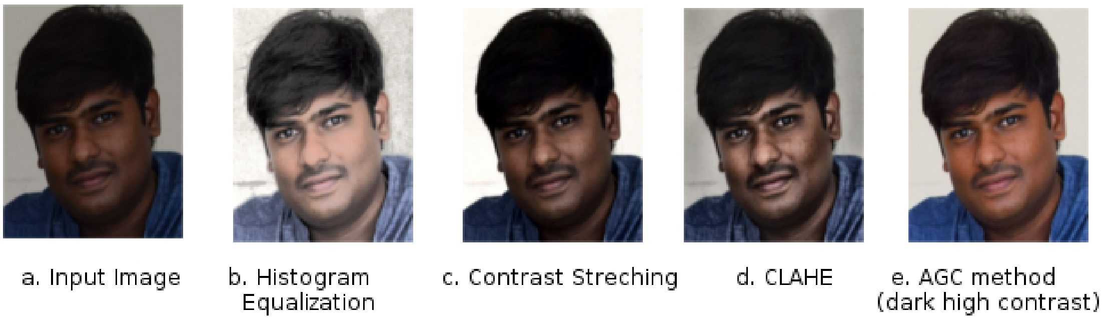


Figure 6.3: **Image Enhancement Techniques.** *(a) Input Image. (b) Histogram Equalization: result is over or under-enhanced regions. (c) Contrast Stretching. (d) CLAHE. (e) Adaptive Gamma Correction - AGC.*

(a) Wiki (b) IMDB

Figure 6.4: **IMDB-Wiki Dataset samples.** *The figure shows samples images from the IMDB-WIki dataset. (a) Wiki subset. (b) IMDB subset.*

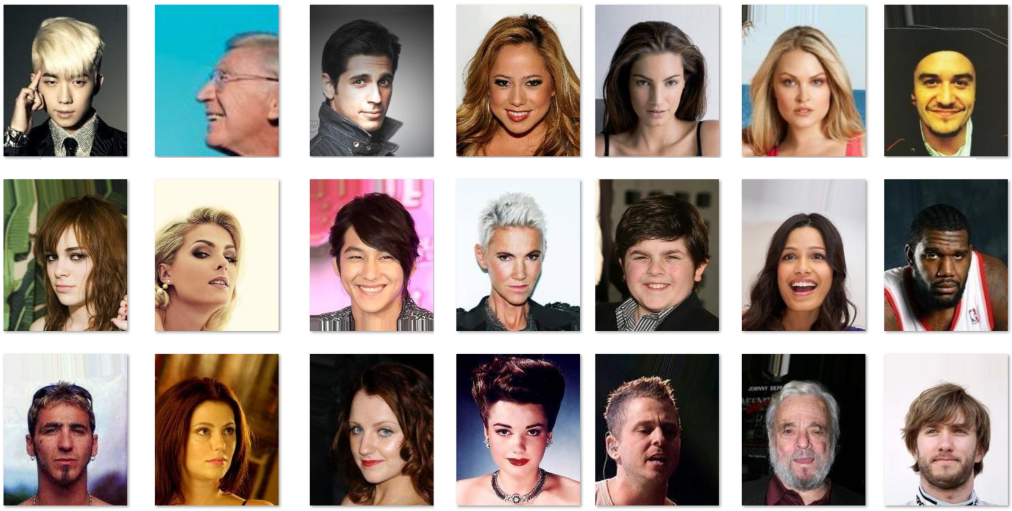


Figure 6.5: **CelebA Dataset samples.** *The figure shows sample images from the CelebA dataset. The dataset includes unconstrained images of celebrities with different backgrounds, qualities, resolution, pose, ...etc.*

their images. Besides, they also crawlled the Wikipedia articles for the same persons to collect the profile image shown for the actor or actress. In total IMDB-WIKI dataset contains 523,051 face images: 460,723 face images from 20,284 celebrities from IMDb and 62,328 from Wikipedia. Only 5% of the celebrities have more than 100 photos, and on average each celebrity has around 23 images. The authors provided the entire public image, the location of the face according to the state-of-the-art face detector they used, which was Face detector without bells and whistles [80]. The face detector is not suitable for real time applications but is accurate in the detecting faces.

In 2015, another large scale public dataset was presented by Liu et. al., called CelebA [74]. The dataset contains ten thousand identities, each of which has twenty images, exactly 202,599 images of 10,177 celebrities. There are two hundred thousand images in total. All images were annotated with forty face attributes, one of them is in our interest, the gender. The dataset was annotated -as mentioned by the authors- by a professional labelling company. The authors proposed a deep neural network to extract features from the images, and then they employed forty SVMs to predict the 40 attributes. All images in the CelebA dataset are also ”in the wild”. Recently, all publicly available datasets, especially if they are targeted to deep learning, are acquired in ”in the wild” manner. This due to the fact that the amount of data available nowadays are usually having that property, unless it is strictly aimed to a very specific context. Sampale images of the dataset are shown in figure 6.5.

6.3.2 Videos Datasets

Nowadays, with the wide spread of imaging handheld devices, and the availability of bigger storage space, videos are becoming more common and available. The videos provide a temporally dense sampling of the scene over time allowing capturing an entire activity. This is achived by storing several frames of the scene over time. Nevertheless, even with high-end devices, videos needs to be compressed aggressively to allow practical storage [126]. Beside compression, videos have a much lower resolution and higher noise levels when compared to still images [126]. Motion blur is a common issue that can be found in videos, when taking a single frame out of the video. In this chapter, image enhancement techniques are used to enhance the quality of the frame. Besides, a quality measure is computed to estimate the quality of a specific frame, since one frame may suffer from motion blur more than another, which is dependent on the scene and the frame rate of the imaging device. In order to train the proposed deep neural networks, and properly assess the method, video datasets are required. In this work, two datasets were used, the McGill-Faces dataset and the PaSC dataset. The datasets were mainly presented for person recognition, however, they can be used for gender recognition.

The McGill-Faces dataset was presented by Demirkus et. al. with its gender labels in [24, 25, 27, 26]. The dataset contains 18000 video frames of 640×480 resolution from 60 video sequences, each of which recorded from a different subject. The subjects were 31 females and 29 males, thus providing a balanced dataset with respect to male-female distribution. Each video was collected in a different environment (indoor or outdoor) resulting arbitrary illumination conditions and background clutter. Furthermore, the subjects were completely free in their movements, leading to arbitrary face scales, arbitrary facial expressions, head pose (in yaw, pitch and roll), motion blur, and local or global occlusions. Sample frames from the dataset are depicted in figure 6.6.

The Point and Shoot Face Recognition Challenge (PaSC) [10] dataset, created in 2013, includes still images and videos, however, only videos from PaSC were used. It is a large scale dataset consisting of 2802 videos of 265 different people shot at six different locations. There are equal number of control videos and handheld videos. The control videos were shot using a high quality camera at a pixel resolution of 1920×1080 (Full HD) with a tripod. The handheld videos were shot using various devices with pixel

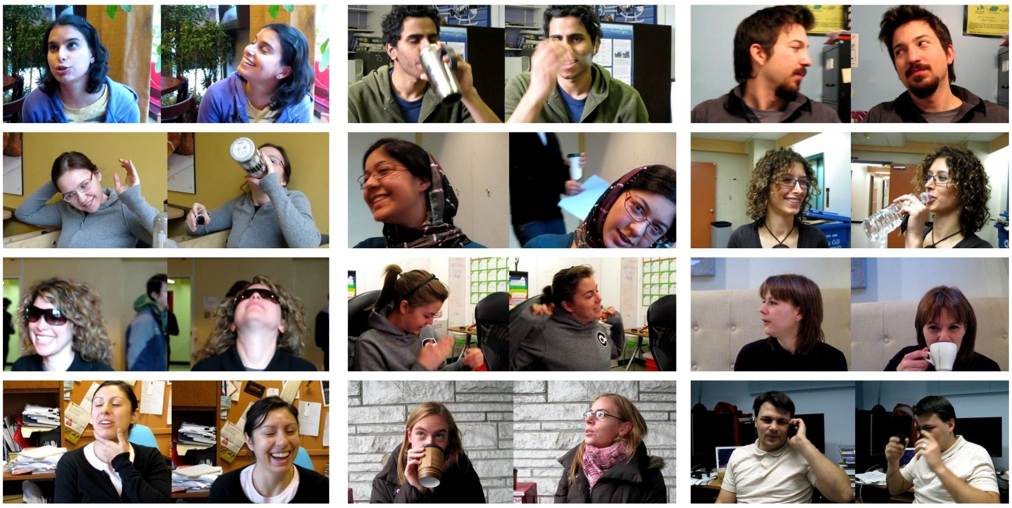


Figure 6.6: **McGill Faces dataset samples.** *This figure shows 4 frames from 2 videos from the McGill faces dataset. The dataset was acquired indoors, with various activities, like talking, free head movements with varying poses, occlusions, ...etc.*

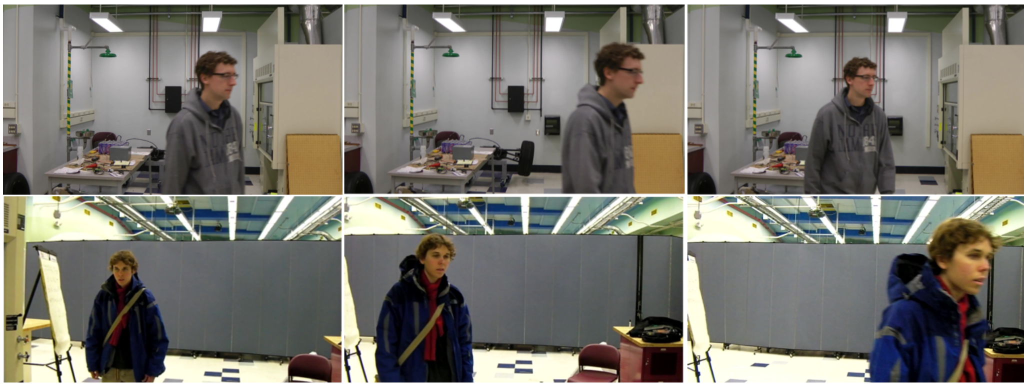


Figure 6.7: **PaSC Dataset samples.** *Sample frames from the PaSC dataset are shown in the figure. The dataset has indoor and outdoor scenes, where the subject are performing various activities, like walking for example, or answering the phone. Other samples are shown in figure 3.2*

resolution ranging from 640×480 to at most 1280×720. The dataset did not contain gender information, which is required in this work. The gender information was manually added to the meta-data. Of the 265 subjects in the dataset, 143 were males and 122 were females. The varying locations (indoor and outdoor), resolution, illuminations, head poses, makes the PaSC [10] dataset ideal for evaluating gender recognition. The provided face bounding boxes are usually the result of running a state-of-the-art face detector on the dataset by its authors. The bounding boxes can be used, however, at the time of this work, newer state-of-the-art face detectors are present, which are superior to the ones used in the datasets. Consequently, newer face detectors were employed for some datasets. The face detector presented in [80] works well in challenging conditions such as dark or blurred faces. Consequently, it was suitable for using it on the PaSC dataset.

6.4 Proposed GenderCNN

Well known deep learning networks like AlexNet, LeNet, and others were mainly designed for 1000-class classification problem in ImageNet (Large Scale Visual Recognition Challenge [111]). The problem in hand is a binary classification problem, where it is required to predict the gender of a person from the face image. It is not a trivial binary classification task, however the context of the input image is the face image only. A novel compact deep neural network is proposed, which consists of 3 convolution layers followed by two fully connected layers. The compact network can reveal the features differentiating between males and females. It is of critical importance to achieve the same or comparable results of deeper networks, otherwise, the accuracy is affected by fewer layers. Having a compact network, reduces the memory footprint of the network by reducing the number of parameters [158]. The GenderCNN architecture is depicted in figure 6.8. In the evaluation section, we show that we do not need a network with many convolution layers to recognize gender. We show the details of our proposed network in table.

The network takes a cropped face images of size 224×224 pixels. The first convolution layer conv1 has 96 filters. The second convolution layer conv2 has 256 filters. The last convolution layer conv3 has 512 filters of size 3×3. The first fully connected layer has size of 6000, followed by the second layer fc2 of size 2. It represents male and females, and is followed by a SoftMax layer to predict the gender of the input image.

6.4.1 GenderCNN for Still Images

In order to perform gender classification on still images, the proposed deep neural network model is used in combination with Adaptive Gamma Correction as a pre-processing step. The aim of pre-processing the face image with AGC is to enhance the appearance of the image before feeding it to the network. In figure 6.9, an overview of the pipeline is depicted. First, the face is detected from the input image using the face detector in [80].

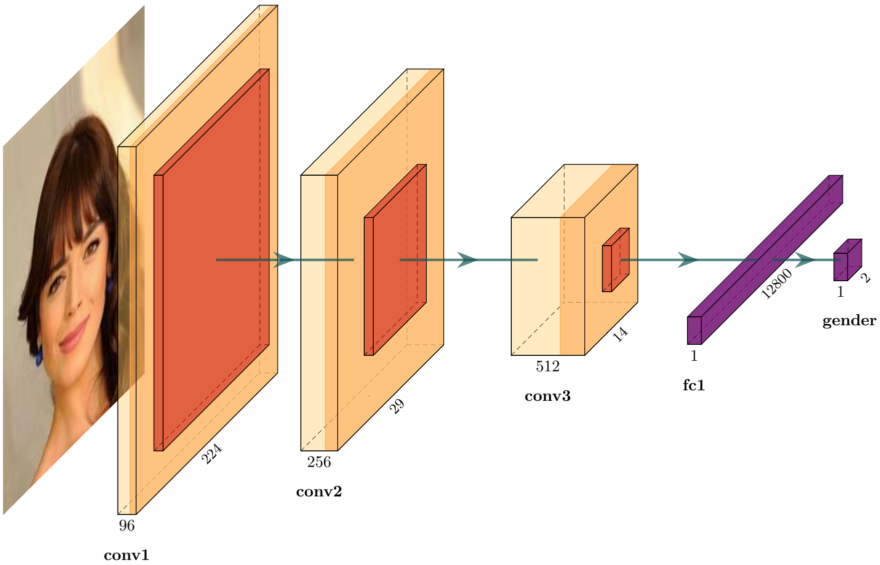


Figure 6.8: **GenderCNN.** *GenderCNN deep neural network architecture*

The detected face is cropped, and pre-processed using AGC [102]. The pre-processed face is fed to the network and the gender is predicted. It is important to mention that one advantage of AGC is that if the image is already good enough, the AGC will not distort it with over or under-enhancing. More challenges exist in video frames and a novel method to consider quality in the pipeline of gender prediction is presented in the next subsection.

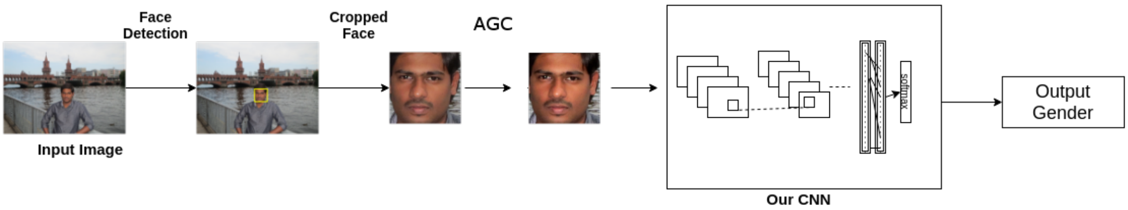


Figure 6.9: **Gender Recognition Pipeline for Still Images.** *The pipeline starts with an input image. The image is processed and the face is detected. Afterwards, the face image is cropped and then the face image is enhanced using AGC. The enhanced face image is fed to the GenderCNN for predicting the gender from the image, and outputs the predicted gender.*

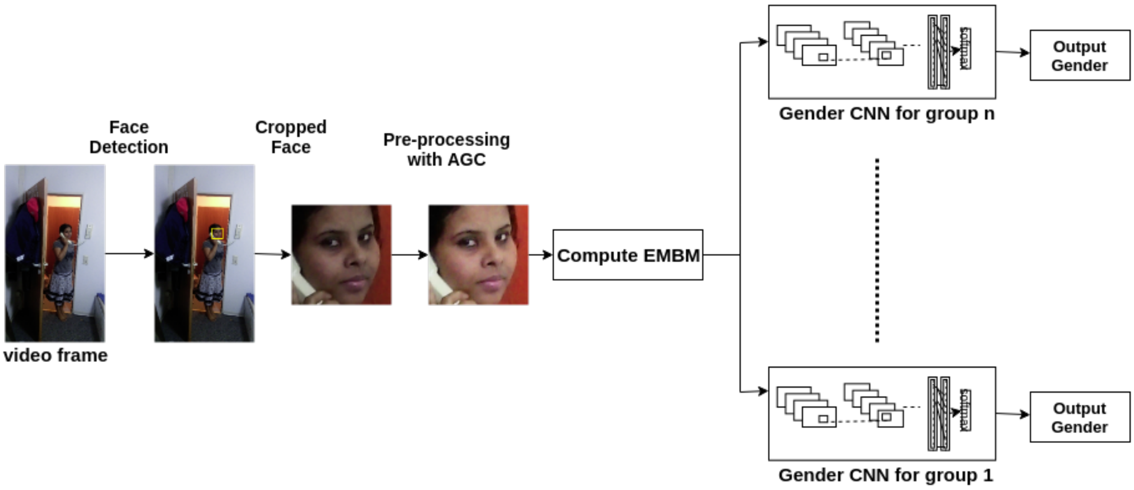


Figure 6.10: **Gender Recognition Pipeline for Videos.** *The pipeline starts with an input frame from a video. The frame image is processed and the face is detected. Afterwards, the face image is cropped and then enhanced using AGC. Using the enhanced face image, the EMBM score is computed, and according to score, the enhanced face image is fed to the GenderCNN corresponding to the EMBM score range for predicting the gender from the image. The GenderCNN outputs the predicted gender.*

6.4.2 GenderCNN for Videos

Videos captured using mobile phones, low resolution cameras or Point and Shoot cameras are challenging for gender recognition, due to the presence of varying illumination and motion blur. These challenges are not usual in still images. The presence of poorly illuminated frames, result in dark faces being fed to the machine learning algorithm. The illumination challenge is tackled in videos using the same image enhancement technique used for still images, the AGC. A sample of dark image enhanced using AGC is shown in figure 6.11. It can be noted how AGC helped in restoring details on the face of the subject. The dark image of the face is due to the the bright sky background. Detecting gender from such an image is hard even for humans, as only the silhouette of the face is visible in the image. However, the AGC was able to restore details on the face of the person in the image, thus, making it possible to identify the gender of the subject.

Motion blur is a common problem in videos captured by in an unstable manner, for example, handheld cameras, or where the subject is moving fast and the capturing device is not fast enough. The effects can be clearly seen in the PaSC dataset, especially the handheld videos. In order estimate gender in bad quality or in blurred video frames, a blurriness metric to quantify the amount of motion blur is used. The EMBM metric proposed in [45] is used to separate faces into groups according to their EMBM value. EMBM is a blur metric based on edge model (EMBM) to address the image blur assessment

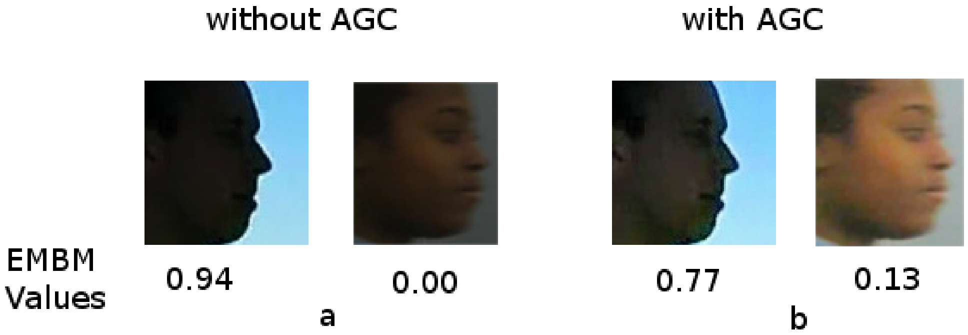


Figure 6.11: **Effect of AGC on Dark Face Images.** *(a) Original images and their EMBM [45] values. (b) AGC-enhanced [102] images and their EMBM values. Faces are extracted from PaSC [10].*

problem. A parametric edge model is incorporated to describe and detect edges, which can offer simultaneous width and contrast estimation for each edge pixel. With the pixel-adaptive width and contrast estimations, the probability of detecting blur at edge pixels can be determined. However, in case of motion blurred faces, the EMBM value is mistakenly estimated to be very blurry, which is not true as shown in figure6.11. The EMBM value can be also wrong in case of dark face with bright background, as the EMBM value estimates the sharpeness between the face and the background. Applying the AGC method first to enhance the face image, and then evaluate EMBM, yields in more realistic EMBM values. This gives a more accurate and reliable value of the EMBM as the details of the face are restored by AGC. Example images of evaluating EMBM with and without applying AGC are shown in figure 6.11.

An overview of the pipeline for estimating the gender from faces in videos is shown in figure 6.10. Following the same steps for detecting and pre-processing the face using AGC an EMBM value is computed. After computing the EMBM value, the images are split into groups, and for each group, a GenderCNN is used to estimate the gender for the face under its corresponding blurriness group. Based on the experimental results, the data is split into two groups based on EMBM threshold. The threshold value can differ based on the dataset used. The EMBM threshold for splitting the data is a hyperparameter that was optimized in the experimental work.

6.5 Evaluation

In this section, the experimental evaluation of the proposed method for gender classification are discussed. The proposed method is compared to state-of-the-art methods at the time of the presented work. Results show that the proposed method either outperforms other methods or achieves comparably close accuracy. The experiment setup is as follows, 80% of the dataset is used for training and the remaining 20% is used for testing. It is considered that no overlap between the training and testing data exists. In order to have a fair comparison with other learning-based state-of-the-art methods, the other models were fine tunes with the datasets used. This is to ensure domain adaptation to the testing sets. However, GenderCNN model is trained from scratch, and no pre-trained weights were used.

Evaluation on Still Images

Experimental results are shown in table 6.1. We evaluated our proposed method on the following still images datasets, IMDB-Wiki [109] and CelebA [74]. The proposed method results are compared to other state-of-the-art methods in table 6.1. GenderCNN outperforms other methods on the IMDB-Wiki dataset. On the CelebA dataset, the work in [74] still has the best accuracy value of 98%, however GenderCNN scores second best accuracy in the table of value 97.38%. It is important to mention that the work in [74], uses two deep neural networks and a dedicated SVM for gender estimation. GenderCNN as presented in this work has 3 convolution layers, The scores of LNet and ANet [74] method were taken from their paper for the IMDB-Wiki dataset, however, the authors did not provide trained models. It was not possible to regenerate their models as some important training parameters were not mentioned in the paper. Pre-processing the images with AGC shows slight improvement in still images datasets. The next subsection discusses the videos datasets used. The temporal coherent face descriptor [140] cannot be applied on still images datasets, as it works only with videos.

	IMDB	Wiki	CelebA
Age & Gender CNN [69]	87.2%	94.44%	97.27%
VGG face desc [122] & SVM	57.05%	81.37%	91.99%
LNet+ANet [74]	-	-	**98%**
GenderCNN wo AGC	***87.34%***	***94.75%***	97.35%
GenderCNN w AGC	**87.46%**	**94.76%**	***97.38%***

Table 6.1: **Evaluation Results on Still Images.** *Results and comparisons on still images datasets. Best result is shown in **Bold**, second best is shown in **Bold Italic**.*

Evaluation on Videos

The evaluation results are shown in table 6.2. The GenderCNN with and without AGC along with AGC and EMBM-GenderCNNs were evaluated on the McGill and PaSC datasets. On the two datasets, the proposed method outperform the Temporal coherent face desriptor used by [140] for gender detection by a big margin. On the McGill dataset,

	McGill	PaSC Control	PaSC Handheld
Age & Gender CNN [69]	84.4%	92.41%	92.62%
VGG face desc [122] & SVM	**92.13%**	90.45%	91.26%
Temporal gender detector [140]	71.42%	86.45%	88.04%
GenderCNN wo AGC	90.14%	92.1%	92.84%
GenderCNN w AGC	***90.4%***	**93.6%**	***94%***
AGC + EMBM GenderCNNs	-	***93.31%***	**94.2%**

Table 6.2: **Evaluation Results on Videos.** *Results and comparisons on still images datasets. Best result is shown in* ***Bold****, second best is shown in* ***Bold Italic****.*

using VGG face descriptor and a SVM [122] gives the best accuracy, however, the GenderCNN with AGC scored the second best accuracy value. It is important to note that EMBM values in the McGill dataset did not vary to separate the data based on blurriness. On the PaSC dataset, GenderCNN with AGC outperforms all other methods. The proposed methods give the best and second best scores. Using GenderCNN with AGC pre-processing gave the best results on the control subset of the PaSC dataset. The experimental results show that on the challenging subset, the handheld, using the blurriness metric EMBM to train different CNNs gave the best result. Figure 6.1 shows gender detection on a sample frame from a handheld video with bright background. The effect of AGC in improving the contrast of the face is visible in the experimental results.

6.6 Summary

This chapter was focused on the problem of predicting the person's gender from the face image. First, state-of-the-art methods were discussed briefly. The data handled in this work is 2D color images which could be sourcing from still images or videos. Current challengings have been identified. In summary, images can suffer from low contrast, due to bad illumination for example. Furthermore, videos suffer from aggressive compression, low resolution, and blurriness due to motion or low frame rate of the capturing device. The problem of image quality was handled using a novel pre-processing method, Adaptive Gamma Correction (AGC), which is superior to other classical contrast enhancement techniques. Regarding the blurriness, the amount of blur is estimated using a novel, no-reference, blur measurement model, EMBM. The image processing algorithms along with deep neural networks were combined together to propose a novel method to predict the gender from face image in still images and videos.

GenderCNN, a new deep nerual network is presented. GenderCNN has only 3 convolution layers. It was trained and evaluated on recent and challenging images and videos datasets. The still images used in our evaluation were IMDB-Wiki[109] dataset and CelebA [74]. The proposed method outperformed the state-of-the-art or match their accuracy values with small difference. In order to have a fair comparison, the learning-based methods were fine tuned using the different datasets. Using Adaptive Gamma Correction, AGC [102] in the gender classification pipeline as a pre-processing step was proposed in this chapter and proven that it improved the results on still images datasets.

On the videos datasets, a slightly different pipeline that tackles challenges found especially in videos was introduced. One main challenge was the image quality degradation due to blurriness and compression. Quantifying the blurriness using EMBM [45] was exploited and used in the pipeline to train different GenderCNN based on the computed metric. AGC was also included, and by using it before computing the EMBM values, misleading EMBM measurements due to bad illumination were avoided. The proposed pipeline for gender detection in videos showed improvement on the challenging handheld videos of the PaSC dataset. The work presented in this chapter opens the door for future ideas to investigate methods to tackle the blurriness found in the ”in the wild” videos. One possible idea would be investigating the usefulness of de-blurring techniques and their impact on gender classification of facial images analysis.

7 Orientation-Guided Deep Gender Recognition

This chapter presents a novel deep learning-based method to predict the gender using both the face image and the head orientation angles. We show that the use of the head orientation information consistently boosts the accuracy of gender prediction models. The proposed method was evaluated on our AutoPOSE dataset.

The work presented in this chapter has been published in the following article:

Mohamed Selim, Stephan Krauß, Tewodros Amberbir Habtegebrial, Alain Pagani, Didier Stricker. Deep Orientation-Guided Gender Recognition from Face Images. In *International Conference on Pattern Recognition Systems (ICPRS)*, 2022.

Contents

7.1 Introduction **110**
7.2 Orientation-Guided Gender Prediction Models **113**
7.3 Evaluation **116**
7.4 Conclusion **121**

7.1 Introduction

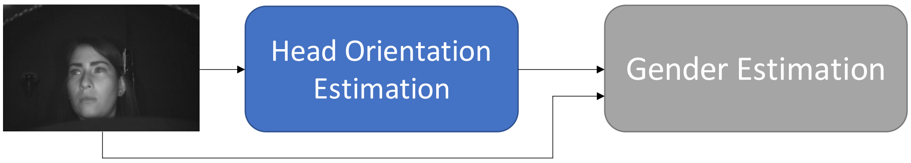


Figure 7.1: **Overview.** *The figure depicts the abstract idea of the proposed method. First, The head image is used to predict the head orientation. Afterwards, the face image, along with the predicted head orientation angles are fed into the gender estimation model.*

Gender recognition is one of the most investigated vision problems in the last decade [43]. Several contributions have been presented in constrained and unconstrained environments. Nevertheless, gender recognition is still a challenging problem. On the other hand, head orientation estimation can now achieve high accuracy, as we show in the previous chapters. We introduced the AutoPOSE dataset, and we proposed novel deep learning networks that can estimate the three head orientation angles given the face image.

In the previous chapter we presented a deep learning-based method to predict the gender from the face image. In other words, the predicted gender is a function of the face image only. The presented models learns the gender of the subjects under several changing conditions, like skin color, age, and head orientation. In this chapter, we investigate the effect of the head orientation on the gender prediction models. Our aim is to study the effect of appearance changes (due to head orientation variation), on the gender prediction's accuracy, and use the head orientation to improve the accuracy of gender prediction.

In recent years, several works aimed at improving the representational power of deep neural networks [150, 135, 63, 139]. Chen et. al. [20] presented a spatial and channel-wise attention network for the purpose of image captioning. The authors showed that employing channel-wise control over the networks feature maps outperformed the visual attention-based image captioning. Hu et. al. [53] won the first place in the ILSVRC 2017 competition with their proposed novel *Squeeze* and *Excitation* network, SENet. The authors introduced an architectural unit designed to improve the representational power of deep learning networks. This was achieved by performing channel-wise feature map recalibration. Their unit takes the output of a convolutional block as input. The n-channels of the input feature maps are squeezed into n-numerical values by applying average global pooling. The vector of numerical values is passed through the unit. The unit's output is a n-vector, which is the last fully connected layer. At the end of the module, a sigmoid function

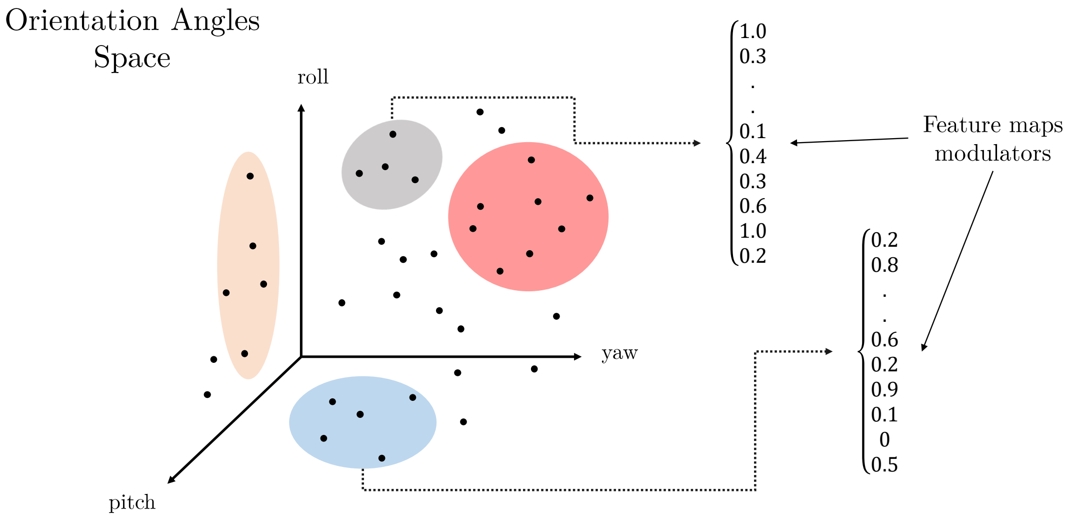


Figure 7.2: **Orientation Angles Space.** *The head orientation angles space is divided into partitions where each partition yields feature maps modulators. The feature maps modulators controls the flow of features through the deep neural network model.*

is applied, yielding weights between 0 and 1. The weights are then used to scale the channels of the input feature maps before passing them to the next layer(s) of the network. This process is referred to as excitation. In 2019, Wang et. al. [141] introduced an effective and efficient object detection system, that can work on different image domains, for example human faces, CT, and satellite images. The authors were able to achieve that by introducing a set of domain adapters to the same deep convolutional network. The aim of the domain adapters is to predict the specific image domain, and based on it, dynamically recalibrate the feature maps that are most effective in the corresponding image domain.

This chapter studies the problem of gender recognition and its relation to head orientation variations. We show that the accuracy of gender prediction can be boosted given the image and the head orientation angles as input. An overview is shown in Figure 7.1. The proposed method predicts the gender of the subject as a function of the face image and the corresponding head orientation angles. The image is passed through the deep neural network to generate image features. The head orientation angles are employed to *adapt* the network feature maps, thus boosting the accuracy in gender prediction.

The head orientation can be modeled using the three rotation angles yaw, pitch, and roll. Given a three dimensional space with axis representing the head angles. A point in the given space represents one possible head orientation. In chapter 6, all images in the datasets were fed into the deep neural networks for training and evaluation. The network learned to properly predict the gender in any of the given face images. In this chapter, we show that the angles space can be divided and separated in a way that supports and adapts the deep neural network for improving the gender prediction accuracy. Figure 7.2 depicts the head orientation space, and an abstract representation of the possible subdivisions that would yield different modulators to recalibrate the feature maps inside the deep neural network models. Detailed explanation of the proposed method is presented in the following sections.

7.2 Orientation-Guided Gender Prediction Models

This section introduces the orientation adapter unit, and shows how it can be employed in the gender prediction deep neural network models. The model proposed in chapter 6, the GenderCNN is modified to integrate the orientation adapter. Besides, deeper model, the ResNet-18 is also employed for gender prediction. Furthermore, the integration of the head orientation in the ResNet-18 is presented.

7.2.1 Orientation Adapter

The proposed head orientation unit is shown in figure 7.3. The unit is a Multi Layer Perceptron, MLP. The aim of the orientation adapter is to encode features as a function of the orientation angles. The adapter takes three rotation angles (yaw, pitch, and roll) as input. The angles are connected to three fully connected layers. The sigmoid function is applied on the last layer to encode the features as numerical values between 0 and 1. The unit's output is then employed in the gender prediction model. We present two methods to use the output of the orientation adapter are presented. The first method is a concatenation along with the fully connected layers of the deep gender models. Thus, the network would use the image features encoded in the fully connected layer that is connected to the convolution layer and the orientation features from the adapter, to predict the gender. Another method is to use the output of the orientation adapter as a weighting scale for some feature maps which are generated by the convolution layers. The unit's output layer size must match the channel dimension of the feature map of the target convolution layer, as they will be multiplied together.

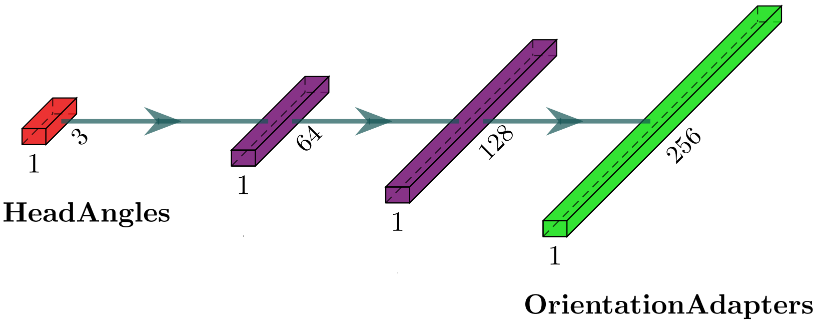


Figure 7.3: **Orientation Adapter.** *Orientation adapter network architecture. The adapter takes the 3 head orientation angles as input. They are connected to 2 fully connected layers. The last layer is the output of the network. The output is to be used in other gender prediction models.*

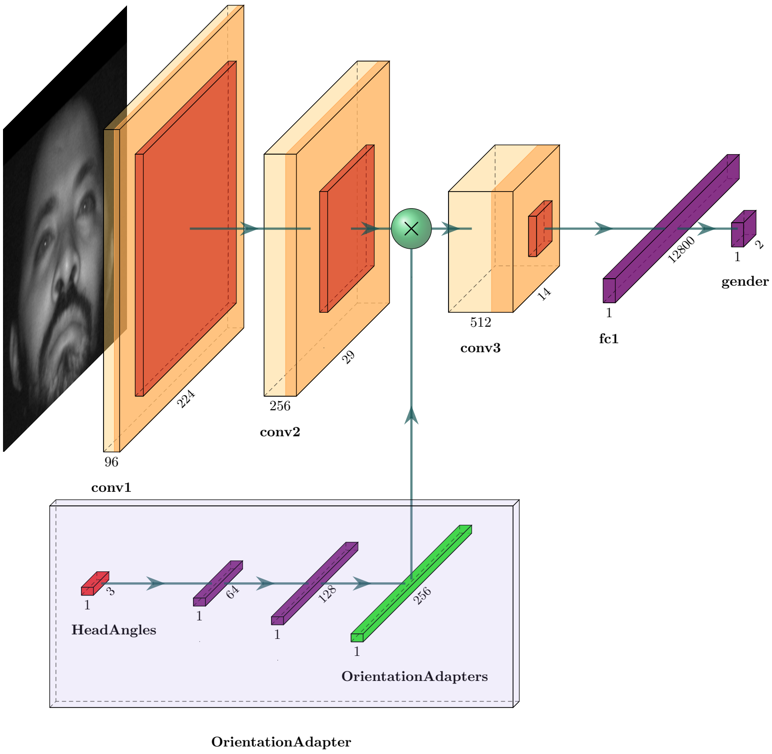


Figure 7.4: **Orientation-Guided GenderCNN.** *An overview of the orientation-guided GenderCNN. The face image is passed through convolution layers. The head orientation angles are fed into the orientation adapter. The orientation adapters are a function of the head orientation angles, and they are modulated over the output feature maps of the layer conv2. The resulting feature maps are passed to the layer conv3. The last layer contains the prediction of the gender.*

7.2.2 GenderCNN with Orientation Adapter

In chapter 6, the convolutional neural network model GenderCNN was introduced. The model was evaluated on several public still images and videos datasets for gender prediction. The model was efficient and achieved high accuracy. In this chapter, the model is modified to be employ the proposed orientation adapter unit. The modified model is shown in figure 7.4. The output of the orientation adapter is modulated with the resulting feature maps of the convolutional layer conv2. The orientation adapter takes the head angles as input, and generates weighting scales that are used to control the feature maps in the convolutional part of the model. In this work, the GenderCNN was also modified to take 1-channel images as input, since the AutoPOSE images used are infrared ones. The final prediction of the gender is dependent not only on the input image, but also on the orientation angles.

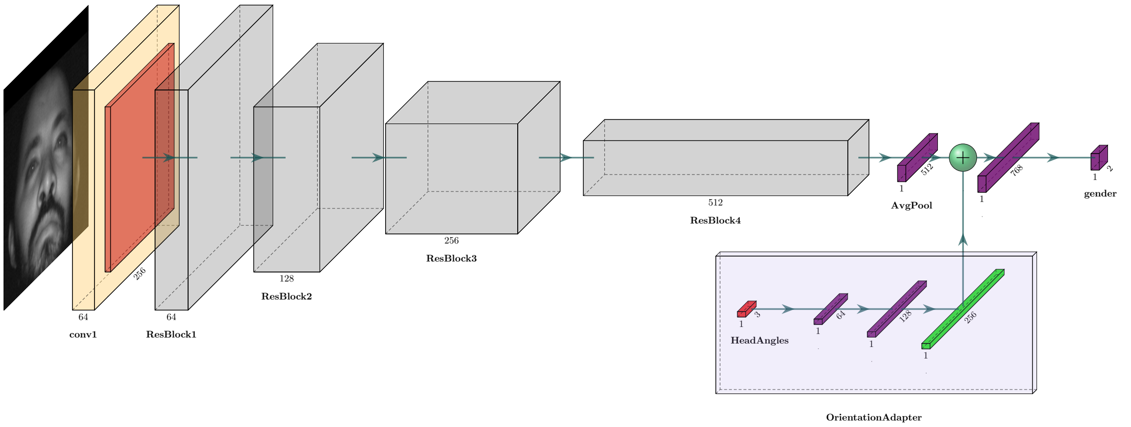


Figure 7.5: **Orientation-Guided ResNet-18 – Concatenation.** *An overview of the orientation-guided ResNet-18. The face image is passed through a convolution layer and the residual blocks. The head orientation angles are fed into the orientation adapter. The orientation features are concatenated with the image features after the global average pooling. The last layer contains the prediction of the gender.*

Detailed evaluation of the GenderCNN and orientation-guided GenderCNN are presented in the next subsection.

7.2.3 ResNet-18 with Orientation Adapter

He et. al. [47] introduced the idea of residual learning in 2015, and the authors won the first place in the ImageNet image classification challenge [28]. The authors showed the training error increases if the networks become deeper. The authors introduced the residual learning that elevates the issue and the networks can become deeper while keeping the training error low. More details about residual learning can be found in subsection 2.1.2. The ResNet-18 variant was chosen as backbone for the gender prediction problem. First, the input face image is passed through a convolution layer with 64 filters. Afterwards, the results are passed through 4 residual convolution blocks with different number of filters. After the last residual block, the feature maps are passed through a global average pooling layer, where each feature map is represented by one numeric value. The average pooling layer makes the ResNet-18 model independent of the input image dimensions.

The proposed orientation adapter is employed with ResNet-18 model by one of two methods, concatenation or modulation as briefly mentioned before. Figure 7.5 depicts the concatenation variant and figure 7.6 depicts the modulation one.

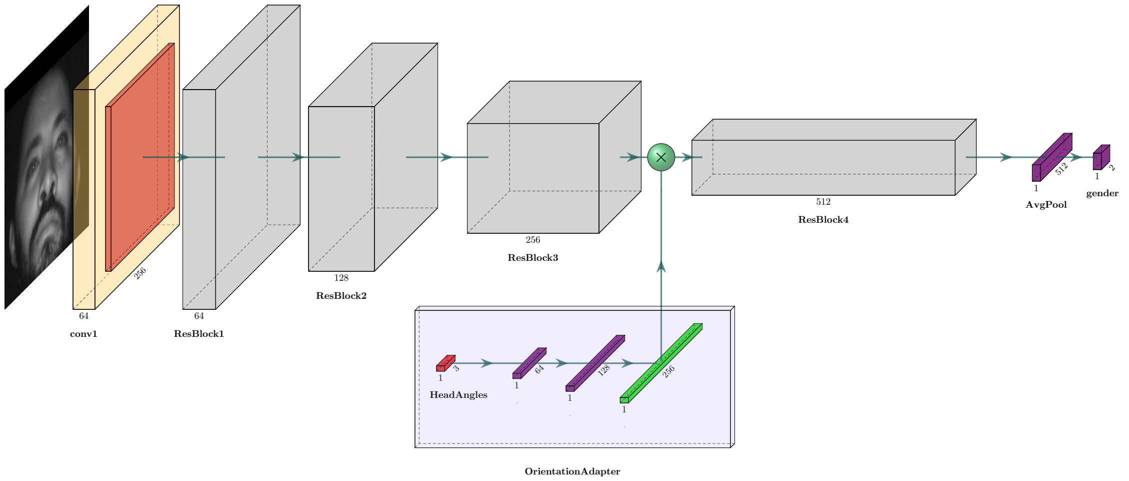


Figure 7.6: **Orientation-Guided ResNet-18 – Modulation.** *An overview of the orientation-guided ResNet-18. The face image is passed through convolution layer and the residual blocks till ResBlock3. The head orientation angles are fed into the orientation adapter. The 256 channel orientation adapters are modulated with the 256 channel feature maps resulting from ResBlock3, and are fed into ResBlock4. The last layer contains the prediction of the gender.*

7.3 Evaluation

This section presents the dataset used, along with the training setup and evaluation results. To have a baseline for the model performance on the gender prediction problem, the GenderCNN models and the ResNet-18 are tested without orientation information. Afterwards, the orientation guided-models are evaluated and compared to the baseline results.

7.3.1 Models Training

In order to evaluate the proposed methods and test the effect of the head orientation on gender prediction from face images, a dataset with faces and accurate head orientation angles is required. The AutoPOSE dataset presented in chapter 4 has high quality and accurate head orientation angles.

The participants in the dataset were 8 females and 13 males. The dataset is not perfectly balanced between males and females. Using all subjects might affect the learning of the variation between the genders. Consequently 4 male subjects were excluded from the training set. The training set consisted of 6 female subjects and 7 male subjects. The evaluation set consisted of 2 male and 2 female subjects. All face images were cropped using the groundtruth face location. The CLAHE method [146] was applied on the cropped face images. The training optimizer used was the Stochastic Gradient

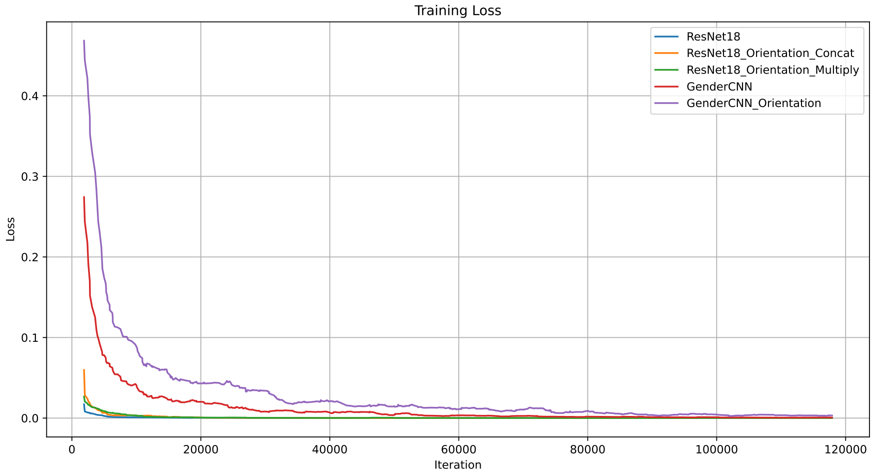


Figure 7.7: **Training loss for GenderCNN and ResNet-18 variants.** *Training loss is depicted. ResNet-18 variants can learn faster than GenderCNN variants. The training loss of ResNet-18 variants drops much quicker that of GenderCNN.*

Descent, with a fixed learning rate of value 0.001 and momentum of value 0.9. The problem of gender prediction is handled as a classification problem. Consequently, the cross entropy loss was employed in the training setup, and it is defined as follows

$$L_{CE}\left(p, y\right) = -\sum_{i=1}^{2} y_i \log\left(p_i\right)$$

where p is the prediction vector and y is the groundtruth.

Figure 7.7 shows the training loss. It can be noted that the training loss of the ResNet-18 variants drops considerably faster than the GenderCNN models. GenderCNN without orientation loss drops faster than the GenderCNN with orientation information. This could be due to the higher number of parameters that are being learnt by the orientation adapter unit. The same effect is seen in the ResNet-18 variants. Figure 7.8 shows the training set accuracy. In general, the ResNet-18 variants can reach accuracy close to 100% much faster than the GenderCNN variants. The GenderCNN variants require much more training epochs to reach convergence.

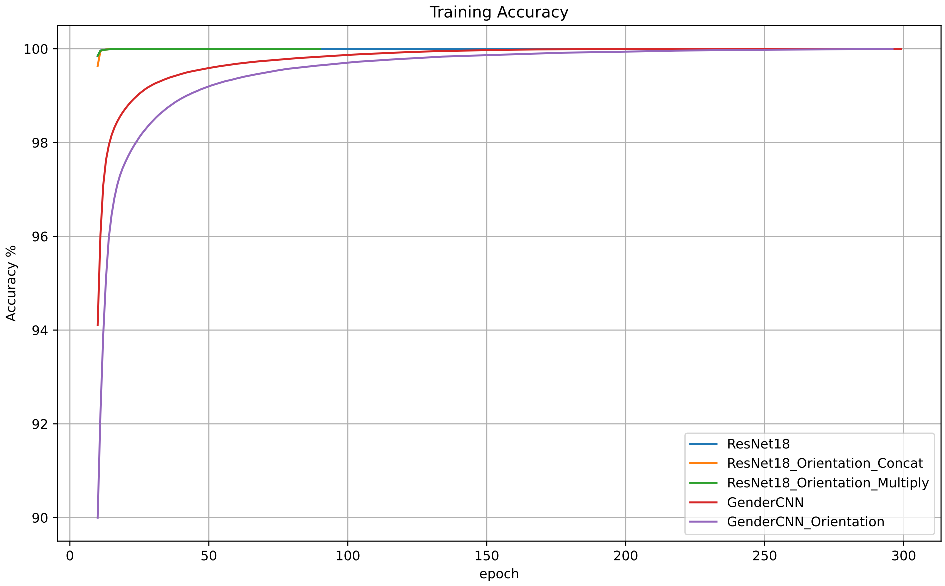


Figure 7.8: **Training accuracy for GenderCNN and ResNet-18 variants.** *Training accuracy on the AutoPOSE dataset is depicted. ResNet-18 variants converge faster than the GenderCNN variants.*

7.3.2 Models Evaluation

This subsection presents the evaluation results of the trained models on the evaluation set. The evaluation set consists of 4 subjects, two females and two males. A summary of the best accuracy results achieved by each model is shown in table 7.1.

GenderCNN with Orientation Adapter

Starting with the GenderCNN model, it is important to first check the performance of the model without the orientation adapter. As shown in figure 7.9, the baseline result for the GenderCNN is on average 82% accurate. One can notice a big difference between the best possible accuracy by the GenderCNN on the AutoPOSE dataset and the other public datasets used in chapter 6. This can be due to the difference in the data domain. The public datasets consisted of color images with mostly frontal images. On the other hand the AutoPOSE images are IR images, and have a wide variation of head orientations. However, the orientation-guided GenderCNN performs better than the baseline variant. The best accuracy achieved by the orientation-guided GenderCNN is 90.7%. Since the GenderCNN cannot reach higher accuracy on IR images, the ResNet-18 model was employed for further evaluations.

Model	Accuracy
GenderCNN	85.5%
GenderCNN with orientation	**90.7%**
ResNet18	98.0%
ResNet18 with orientation (concatenation)	98.2%
ResNet18 with orientation (multiplication)	**98.4%**

Table 7.1: **Evaluation Results – Orientation-guided gender prediction models.** *The table shows the best accuracy achieved by each model. The results show that using the orientation information consistently improved the overall accuracy.*

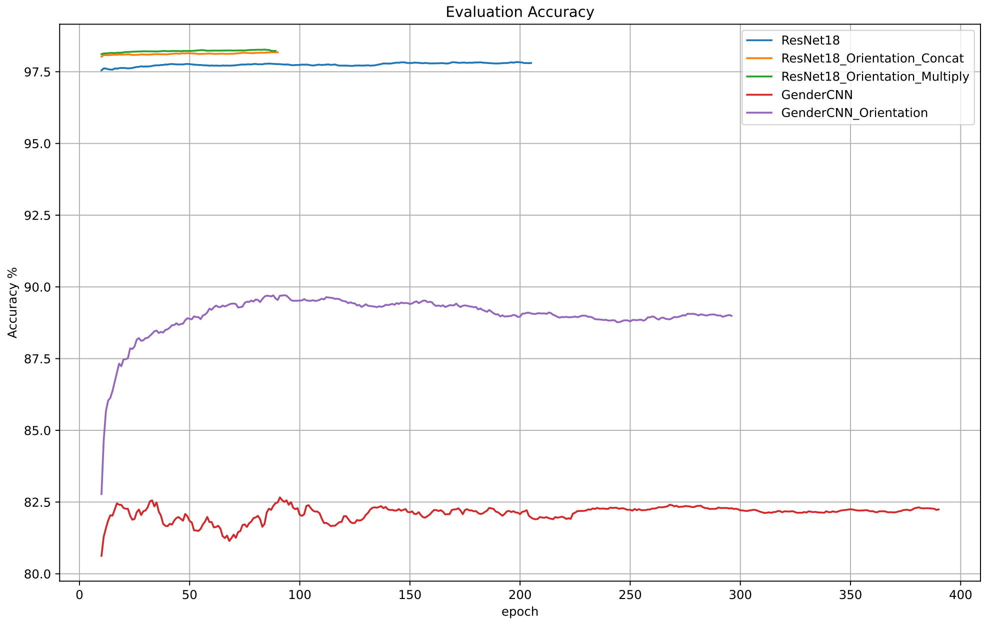


Figure 7.9: **Evaluation accuracy - Orientation-guided GenderCNN and ResNet-18 models.** *The graph shows the models accuracy on the evaluation set. GenderCNN: orientation-guided version boosts the accuracy by a big margin. ResNet-18: both orientation-guided variants are better than baseline ResNet-18. Orientation guidance boosts the accuracy on both models.*

ResNet-18 with Orientation Adapter

Since the GenderCNN could not achieve high accuracy results, ResNet-18 was used as the backbone in an improved gender prediction network. In general as shown in figure 7.9, all ResNet-18 variants performed considerably better than the GenderCNN baseline and the orientation-guided GenderCNN. The baseline result for the ResNet-18 model could reach at most 98%. Our hypothesis is that part of the 2% error in the accuracy could be related to the orientation variation, in which case the orientation adapter should improve the evaluation accuracy. Figure 7.10 shows the last part of the y-axis, where the evaluation accuracy of the baseline ResNet-18, and the two variants of orientation-guidance, concatenation and modulation are depicted. Both orientation-guided variants outperform the baseline ResNet-18 model. Over the whole training and evaluation cycles, predicting the gender using the face image and the angle is consistently better than just using the face image. The concatenation version is not as good as the modulation version. Modulating the orientation adapters vector with the feature maps of ResBlock3 achieves the best result, with an accuracy of 98.38%.

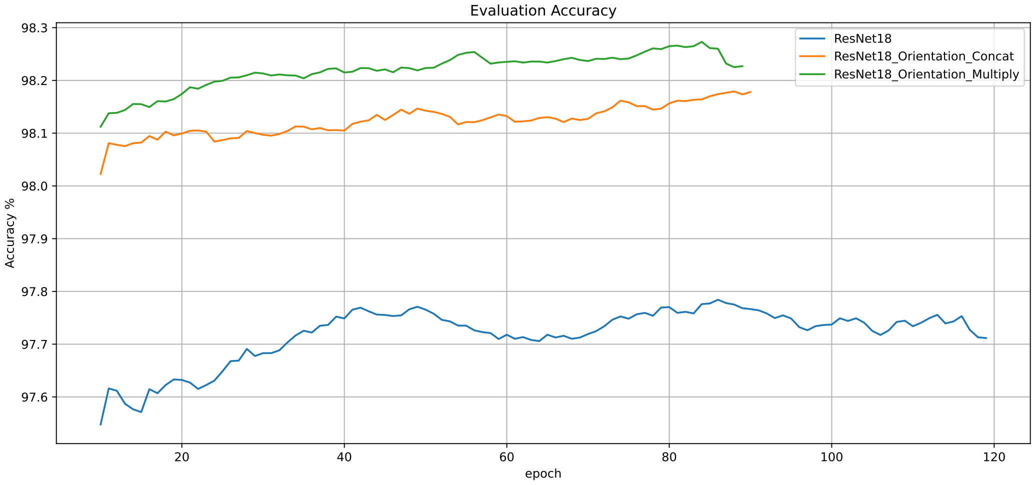


Figure 7.10: **Evaluation accuracy - Detailed ResNet-18 variants accuracy.** *The graph shows smoothed accuracy results of the ResNet-18 models. The baseline version with no orientation information achieves at most 98%. Both orientation-guided variants perform consistently better (best result is 98.38%) than the baseline version. The error is reduced by 20%.*

7.4 Conclusion

In this chapter, a novel deep learning-based orientation-guided gender prediction method from face images was introduced. A new orientation adapter unit was introduced to be employed along with deep neural networks to boost the accuracy of gender prediction. Two methods were tested for using the orientation features, concatenation with fully connected layers and modulation with feature maps inside the network flow. We show that orientation guidance consistently boosts the gender prediction accuracy on both GenderCNN and ResNet-18 models. We used our new dataset, AutoPOSE, to evaluate the use of orientation in gender prediction. Besides, we also concluded that ResNet-18 variants can predict the gender with higher accuracy compared to GenderCNN. Using the orientation information in the ResNet-18 model, the error was reduced by 20%.

8 Conclusion

This chapter concludes the thesis by providing a summary of the presented work and lists the scientific contributions. It also outlines further research directions as future work.

Contents

8.1 Summary of Thesis Contributions 124

8.2 Future Work 125

8.1 Summary of Thesis Contributions

In this thesis we present solutions to the head orientation and gender prediction problems from face images. The proposed solutions are supervised, learning-based methods. The proposed methods are all focused on representing, extracting, and retrieving information from the face image under constrained or unconstrained conditions. Novel modeling methods have been introduced, and have been exploited using classical and novel machine learning methods.
Starting with the problem of head orientation estimation, the topic have been first addressed in Chapter 3. The head orientation angles are correlated to the face appearance, and by utilizing machine learning-based methods. A novel, light-weight method that utilizes the Multi Variate Relevance Vector Machines have been presented. Existing large scale datasets have been exploited for the problem in hand. Several large scale, and challenging datasets have been used first as a proof of concept. The datasets vary in quality, resolution, and were captured at indoor and outdoor scenes. The MVRVM have been proven useful and efficient in solving the problem of head pose estimation, however, the quality of the head orientation angles that the MVRVM learn are not of enough high accuracy to be considered as *groundtruth* information. The noted challenges motivated the work presented in the Chapter 4.

The AutoPOSE dataset is presented in Chapter 4. Existing public datasets lack accurate and reliable head orientation angles. AutoPOSE is a new large scale, public dataset of head pose and eye gaze. The dataset is captured in a car cabin driving simulator. A sub-millimeter accurate motion capturing systems have been used to capture the groundtruth information of the head pose in the camera co-ordinate frame. The subject's head coordinate frame have been defined and calibration to the optical motion capturing system have been computed, resulting in high quality data that can be considered groudtruth head pose in the camera coordinate frame. In AutoPOSE, two subsets have been acquired. One subset is taken from the dashboard position of the driving simulator using an infrared camera, and the other subset was acquired from the center rear mirror position using a Microsoft Kinect V2. All frames have been annotated with the action performed by the participants. The acquired dataset along with the groundtruth information and annotations are publicly available for the research community at `https://autopose.dfki.de`.
In Chapter 5, a deep learning baseline method have been introduced with the dataset. Further experiments have been carried out in joint-work to exploit the effect of the network depth and the input face resolution on the accuracy of head orientation estimation. We show that convolutional neural networks can achieve high accuracy in head orientation estimation. Besides, we show that larger faces yield less error in the angles prediction. Moreover, we show that deep networks with fewer layers performed better when compared to deeper models.

In Chapter 6, the problem of gender estimation from face images have been studied.

The chapter presented state-of-the-art methods for estimating the gender from faces. Novel deep learning-based methods have been presented that employ a novel image enhancement technique in order to improve the images, and handle extreme challenges related to illuminations and quality. Large scale, challenging *in the wild* datasets have been used. The cropped face image can suffer from bad illumination due to low contrast, or strong background illumination as seen in outdoor scenes. Also, in videos, further challenges are present due to the aggressive compression in videos and lower resolution when compared to still images. The image quality and illumination issues have been handled in the proposed method by employing Adaptive Gamma Correction (AGC) to enhance the images as needed. AGC does not over or under enhance the images. It works on the V-channel in the HSV domain of the image, thus, does not alter the image appearance. Regarding videos, a blue metric have been used to model the amount of blur available in the image. We show that the pre-processing of the cropped faces and integrating the quality metric in the deep learning pipeline improves the gender prediction accuracy.

Finally Chapter 7 combines the work presented on gender prediction and head orientation estimation. In this chapter the effect of the head orientation variation on the accuracy of gender prediction models is studied. We introduced a new unit called orientation adapter. The adapter takes the head orientation angles as input, and outputs a vector that can be integrated in the deep learning neural networks. Best results were achieved by using the orientation adapters to recalibrate the feature maps of the convolutional neural layer by modulating them. In other words, the recalibration takes place as a function of the head orientation angles. We show that using the head orientation adapter consistently boosts the gender prediction models accuracy.

8.2 Future Work

This thesis focused on information modeling, extraction, and retrieval to solve the problems of head orientation and gender prediction. The AutoPOSE dataset has been exploited for the focus of the thesis. However, AutoPOSE opens the door for further explorations for the scientific community, as it contains valuable information such as eye gaze annotations. The dataset can be used in various applications such as gaze estimation analysis, driver's attention monitoring, and human car interaction. Also, the Kinect subset of the AutoPOSE is suitable for data generation by learning information transfer from one domain to another. Besides, the problem of gender prediction has been thoroughly exploited for still images and videos. Novel learning-based algorithms have been presented and evaluated. We show that orientation guidance helped in boosting the accuracy of the gender prediction models. This method adds invariance to extreme appearance changes. The same idea can be exploited in different applications related to face image analysis, for example, apparent age prediction.

List of Abbreviations

AGC Adaptive Gamma Correction

AHE Adaptive Histogram Equalization

API Application Programming Interface

AR Augmented Reality

BMAE Balanced Mean Angular Error

CAN Controller Area Network

CLAHE Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

DPM Deformable Parts Model

EMBM Edge Model Blur Metric

GAN Generative Adversarial Network

HE Histogram Equalization

HSV Hue-Saturation-Value

IR Infrared

MAE Mean Absolute Error

ML Machine Learning

MVRVM Multi-Variate Relevance Vector Machine

PaSC Point and Shoot Challenge

ResNet Residual Neural Network

RGB Red-Green-Blue

RMSE Root Mean Squared Error

RVM Relevance Vector Machine

SDK Software Development Kit

SIFT Scale-Invariant Feature Transform

SVD Singular Value Decomposition

SVM Support Vector Machine

ToF Time of Flight

List of Figures

2.1 AlexNet Architecture . . . 21
2.2 Residual Learning . . . 22
2.3 Rotation of a rigid body about a fixed point . . . 23
2.4 Rigid body motion . . . 26
2.5 Face Detection Methods . . . 28
2.6 Region-Based CNN for Face Detection . . . 29

3.1 Samples: Yaw angle estimation on YouTube face dataset . . . 36
3.2 Sample frames from the PaSC dataset [10] . . . 38
3.3 Overview: Learning head orientation with RVM . . . 39
3.4 Comparing different kernel types . . . 42
3.5 Optimizing the kernel width k . . . 43
3.6 Effect of grid size on the average error and the RVs . . . 44
3.7 Single MVRVM Results on YouTube faces dataset . . . 46
3.8 Overview of the cascaded approach . . . 48
3.9 Tree branching optimization . . . 49
3.10 Validation results on PaSC dataset . . . 51

4.1 Car Cabin Simulator . . . 57
4.2 Dashboard camera with Markers . . . 62
4.3 Center mirror camera with markers . . . 63
4.4 AutoPOSE coordinate systems . . . 64
4.5 Handeye calibration - error analysis . . . 67
4.6 Head coordinate frame definition . . . 70
4.7 AutoPOSE dashboard camera set - angles distribution histogram . . . 73
4.8 AutoPOSE samples: Dashboard camera set . . . 75
4.9 AutoPOSE samples: Center rear mirror set . . . 76

5.1 AutoPOSE - Baseline CNN . . . 83
5.2 Head Orientation Network - HON Model . . . 86
5.3 ResNetHG Model . . . 87
5.4 Results - Networks Depths . . . 88

6.1 Gender recognition under harsh conditions . . . 93
6.2 AGC Functional Block Diagram . . . 96
6.3 Image Enhancement Techniques . . . 98
6.4 IMDB-Wiki Dataset samples . . . 99

6.5 CelebA Dataset samples 99
6.6 McGill Faces dataset samples 101
6.7 PaSC Dataset samples 101
6.8 GenderCNN 103
6.9 Gender Recognition Pipeline for Still Images 103
6.10 Gender Recognition Pipeline for Videos 104
6.11 Effect of AGC on Dark Face Images 105

7.1 Overview 110
7.2 Orientation Angles Space 111
7.3 Orientation Adapter 113
7.4 Orientation-Guided GenderCNN 114
7.5 Orientation-Guided ResNet-18 – Concatenation 115
7.6 Orientation-Guided ResNet-18 – Modulation 116
7.7 Training loss for GenderCNN and ResNet-18 variants 117
7.8 Training accuracy for GenderCNN and ResNet-18 variants 118
7.9 Evaluation accuracy - Orientation-guided GenderCNN and ResNet-18 models 119
7.10 Evaluation accuracy - Detailed ResNet-18 variants accuracy 120

List of Tables

3.1 Comparing single MVRVM to the cascade tree of MVRVMs 50

4.1 Microsoft Kinect v2 resolutions 63
4.2 AutoPOSE subject's tasks 73
4.3 Frames breakdown per task 74

5.1 AutoPOSE - Results - Face size 64×64 84
5.2 AutoPOSE - Results - Face size 128×128 88
5.3 Comparing Effect of Resolution 89

6.1 Evaluation Results on Still Images 106
6.2 Evaluation Results on Videos 107

7.1 Evaluation Results – Orientation-guided gender prediction models 119

Bibliography

[1] Byungtae Ahn, Dong-Geol Choi, Jaesik Park, and In So Kweon. Real-time head pose estimation using multi-task deep neural network. *Robotics and Autonomous Systems*, 103:1 – 12, 2018. *82*

[2] Byungtae Ahn, Jaesik Park, and In So Kweon. Real-time head orientation from a monocular camera using deep neural network. In Daniel Cremers, Ian Reid, Hideo Saito, and Ming-Hsuan Yang, editors, *Computer Vision – ACCV 2014*, pages 82–96, Cham, 2015. Springer International Publishing. *82*

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014. *58*

[4] Stylianos Asteriadis, Dimitris Soufleros, Kostas Karpouzis, and Stefanos Kollias. A natural head pose and eye gaze dataset. In *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*. ACM, 2009. *69*

[5] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1859–1866. IEEE, 2014. *34, 37, 39, 50, 69*

[6] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 59–66, May 2018. *82*

[7] J. M. D. Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker. Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2028–2037, March 2018. *82*

[8] Stephan Beck and Bernd Froehlich. Volumetric calibration and registration of rgbd-sensors. In *2015 IEEE Virtual Reality (VR)*, pages 151–152. IEEE, 2015. *66*

[9] Lacey Best-Rowden, Brendan Klare, Joshua Klontz, and Anil K Jain. Video-to-video face matching: Establishing a baseline for unconstrained face recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013. *37*

[10] J. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, P. J. Flynn, and S. Cheng. The challenge of face recognition from digital point-and-shoot cameras. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8, Sept 2013. *35, 37, 38, 100, 102, 105, 129*

[11] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. *27*

[12] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003. *34*

[13] Laura Boccanfuso and Jason M OKane. Charlie: An adaptive robot design with hand and face tracking for use in autism therapy. *International journal of social robotics*, 3(4):337–347, 2011. *27*

[14] G. Borghi, M. Fabbri, R. Vezzani, S. Calderara, and R. Cucchiara. Face-from-depth for head pose estimation on depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3):596–609, 2020. *32*

[15] Guido Borghi, Matteo Fabbri, Roberto Vezzani, Rita Cucchiara, et al. Face-from-depth for head pose estimation on depth images. *IEEE transactions on pattern analysis and machine intelligence*, 2018. *82, 83, 84*

[16] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. *54, 82, 84*

[17] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, 2015. Online; accessed December-2020. *66, 67*

[18] H. T. Butt, M. Pancholi, M. Musahl, P. Murthy, M. A. Sanchez, and D. Stricker. Inertial motion capture using adaptive sensor fusion and joint angle drift correction. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8, 2019. *59*

[19] Chehra, fully-automatic real-time face and eyes landmark detection and tracking software. `https://sites.google.com/site/chehrahome/`, 2020. Online; accessed December-2020. *39*

[20] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional

networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017. *110*

[21] Heng-Da Cheng, X_ H_ Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001. *97*

[22] Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002. *34*

[23] Henri De Vroey, Kurt Claeys, Evie Vereecke, Jos Vanrenterghem, Jan Deklerck, Geert Van Damme, Hans Hallez, and Filip Staes. Correlation between an inertial and camera based system for the assessment of temporal parameters of gait in the knee arthroplasty population. *Gait Posture*, 57:280–281, 2017. *59*

[24] Meltem Demirkus, James Clark, and Tal Arbel. Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications*, 70, 05 2013. *95, 100*

[25] Meltem Demirkus, Doina Precup, James J. Clark, and Tal Arbel. *Probabilistic Temporal Head Pose Estimation Using a Hierarchical Graphical Model*, pages 328–344. Springer International Publishing, Cham, 2014. *100*

[26] Meltem Demirkus, Doina Precup, James J Clark, and Tal Arbel. Hierarchical spatio-temporal probabilistic graphical model with multiple feature fusion for binary facial attribute classification in real-world face videos. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1185–1203, 2015. *100*

[27] Meltem Demirkus, Doina Precup, James J. Clark, and Tal Arbel. Hierarchical temporal graphical model for head pose estimation and subsequent attribute classification in real-world videos. volume 136, pages 128 – 145, 2015. Generative Models in Computer Vision and Medical Imaging. *100*

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009. *115*

[29] Leonardo Araujo dos Santos. Object localization and detection. `https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/object_localization_and_detection`, 2017. *29*

[30] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *Trans. Info. For. Sec.*, 9(12):2170–2179, December 2014. *95*

[31] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed. Lfr face dataset:left-front-right dataset for pose-invariant face recognition in the wild. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 124–130, 2020. *32*

[32] face.com. `https://www.face.com/`, 2011. Offline; acquired by facebook in 2012. *37*

[33] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 617–624. IEEE, 2011. *34, 69, 82*

[34] Hartmut Feld, Bruno Mirbach, Jigyasa Katrolia, Mohamed Selimand Oliver Wasenmüller, and Didier Stricker. Dfki cabin simulator: A test platform for visual in-cabin monitoring functions. In *CVT 2020. 6th International Commercial Vehicle Technology (CVT) Symposium*, 01 2020. *15*

[35] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010. *28*

[36] Ahmet Firintepe, M. Selim, A. Pagani, and D. Stricker. The more, the merrier? a study on in-car ir-based head pose estimation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020. *14, 15, 85, 86, 90*

[37] Joshua S Furtado, Hugh HT Liu, Gilbert Lai, Herve Lacheray, and Jason Desouza-Coelho. Comparative analysis of optitrack motion capture systems. In *Advances in Motion Sensing and Control for Robotic Applications*, pages 15–31. Springer, 2019. *59*

[38] Tobias Gail, Ramona Hoffmann, Markus Miezal, Gabriele Bleser, and Sigrid Leyendecker. Towards bridging the gap between motion capturing and biomechanical optimal control simulations. In *ECCOMAS Thematic Conference on Multibody Dynamics*, 06 2015. *60*

[39] Erkin Gezgin, Pyung-Hun Chang, and Ahmet Faruk Akhan. Synthesis of a watt ii six-bar linkage in the design of a hand rehabilitation robot. *Mechanism and Machine Theory*, 104:177 – 189, 2016. *60*

[40] Golnaz Ghiasi and Charless C Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2385–2392, 2014. *28*

[41] RC Gonzalez and RE Woods. Digital image processing: Pearson prentice hall. *Upper Saddle River, NJ*, 1:376–376, 2008. *97*

[42] Google, inc. `https://www.google.com`, 2020. Online; accessed December-2020. *37*

[43] Antonio Greco, Alessia Saggese, Mario Vento, and Vincenzo Vigilante. A convolutional neural network for gender recognition optimizing the accuracy/speed tradeoff. *IEEE Access*, 8:130771–130781, 2020. *110*

[44] Lie Gu and Takeo Kanade. 3d alignment of face in a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1305–1312. IEEE, 2006. *34*

[45] Jingwei Guan, Wei Zhang, Jason Jianjun Gu, and Hongliang Ren. No-reference blur assessment based on edge modeling. *J. Visual Communication and Image Representation*, 29:1–7, 2015. *93, 104, 105, 108*

[46] Jian Han, Wei Wang, Sezer Karaoglu, Wei Zeng, and Theo Gevers. Pose invariant age estimation of face images in the wild. *Computer Vision and Image Understanding*, 202:103123, 2021. *32*

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. *86, 87, 92, 115*

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. *21, 22*

[49] Daniel Herrera, Juho Kannala, and Janne Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 2012. *66*

[50] Erik Hjelmås and Boon Kee Low. Face detection: A survey. *Computer vision and image understanding*, 83(3):236–274, 2001. *27*

[51] Ramona Hoffmann, Bertram Taetz, Markus Miezal, Gabriele Bleser, and Sigrid Leyendecker. On data-guided optimal control simulation of human motion. 03 2016. *60*

[52] Ramona Hoffmann, Bertram Taetz, Markus Miezal, Gabriele Bleser, and Sigrid Leyendecker. On optical data-guided optimal control simulations of human motion. *Multibody System Dynamics*, 48(1):105–126, 2020. *60*

[53] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. *110*

[54] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. *35, 37, 45*

[55] Noor A Ibraheem, Mokhtar M Hasan, Rafiqul Z Khan, and Pramod K Mishra. Understanding color models: a review. *ARPN Journal of science and technology*, 2(3):265–275, 2012. *97*

[56] Rabia Jafri and Hamid R Arabnia. A survey of face recognition techniques. *journal of information processing systems*, 5(2):41–68, 2009. *27*

[57] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster r-cnn. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 650–657. IEEE, 2017. *29*

[58] Michael Jones and Paul Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3:14, 2003. *34*

[59] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Face-tld: Tracking-learning-detection applied to faces. In *2010 IEEE International Conference on Image Processing*, pages 3789–3792. IEEE, 2010. *27*

[60] Meina Kan, Dong Xu, Shiguang Shan, Wen Li, and Xilin Chen. Learning prototype hyperplanes for face verification in the wild. *Image Processing, IEEE Transactions on*, 22(8):3310–3316, 2013. *37*

[61] Ira Kemelmacher-Shlizerman, Eli Shechtman, Rahul Garg, and Steven M Seitz. Exploring photobios. *ACM Transactions on Graphics (TOG)*, 30(4):1–10, 2011. *27*

[62] K. Khan, M. Mauro, P. Migliorati, and R. Leonardi. Head pose estimation through multi-class face segmentation. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 175–180, 2017. *32*

[63] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning*, pages 1857–1865. PMLR, 2017. *110*

[64] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. *86*

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012. *21, 95*

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. *92, 95*

[67] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989. *20*

[68] Damien Lefloch, Rahul Nair, Frank Lenzen, Henrik Schäfer, Lee Streeter, Michael J Cree, Reinhard Koch, and Andreas Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013. *62*

[69] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshops*, June 2015. *92, 95, 106, 107*

[70] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334, 2015. *29*

[71] Congcong Liu, Yuying Chen, Lei Tai, Haoyang Ye, Ming Liu, and Bertram E. Shi. A gaze model improves autonomous driving. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research and Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery. *54*

[72] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017. *92*

[73] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. *37*

[74] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. *92, 95, 99, 106, 108*

[75] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. *20*

[76] Yi Ma, Stefano Soatto, Jana Košecká, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models. Chapter 2: Representation of a*

Three-Dimensional Moving Scene, pages 15–43. Springer New York, New York, NY, 2004. *23, 26*

[77] Jordi Mansanet, Alberto Albiol, and Roberto Paredes. Local deep neural networks for gender recognition. *Pattern Recogn. Lett.*, 70(C):80–86, January 2016. *92*

[78] Jordi Mansanet, Alberto Albiol, and Roberto Paredes. Local deep neural networks for gender recognition. *Pattern Recognition Letters*, 70:80–86, 2016. *95*

[79] S. Martin, K. Yuen, and M. M. Trivedi. Vision for intelligent vehicles applications (viva): Face detection and head pose challenge. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1010–1014, 2016. *58*

[80] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014. *28, 30, 99, 102*

[81] Gregory P Meyer, Shalini Gupta, Iuri Frosio, Dikpal Reddy, and Jan Kautz. Robust model-based 3d head pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3649–3657, 2015. *82*

[82] Microsoft. Kinect. `http://www.microsoft.com/en-us/kinectforwindows/`, 2012. *57*

[83] Roemhildt ML, Gardner-Morse MG, Morgan CF, Beynnon BD, and Badger GJ. Calcium phosphate particulates increase friction in the rat knee joint. *Osteoarthritis and cartilage*, 22(5):706–709, 2014. *60*

[84] Motive - optical motion capture software. `https://optitrack.com/software/motive/`, 2020. Online; accessed December-2020. *65*

[85] Mohd. Abdul Muqeet and Raghunath S. Holambe. Local binary patterns based on directional wavelet transform for expression and pose-invariant face recognition. *Applied Computing and Informatics*, 15(2):163 – 171, 2019. *32*

[86] Erik Murphy-Chutorian and Mohan M Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):607–626, 2009. *32*

[87] Natnet sdk. `https://optitrack.com/software/natnet-sdk`, 2020. Online; accessed December-2020. *69*

[88] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. *86*

[89] Choon Boon Ng, Yong Haur Tay, and Bok-Min Goi. Vision-based human gender recognition: A survey. *CoRR*, abs/1204.1611, 2012. *92, 95*

[90] Choon-Boon Ng, Yong-Haur Tay, and Bok-Min Goi. A review of facial gender recognition. *Pattern Analysis and Applications*, 18(4):739–755, Nov 2015. *32, 95*

[91] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002. *27*

[92] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*, pages 236–243. Springer, 2008. *27*

[93] R. Okuda, Y. Kajiwara, and K. Terashima. A survey of technical trend of adas and autonomous driving. In *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, pages 1–4, 2014. *56*

[94] Optitrack. `https://optitrack.com/`, 2020. Online; accessed December-2020. *59*

[95] A. Pagani and D. Stricker. Learning local patch orientation with a cascade of sparse regressors. In *The British Machine Vision Conference (BMVC)*, 2009. *39*

[96] Anwesan Pal, Sayan Mondal, and Henrik I. Christensen. "looking at the right stuff" - guided semantic-gaze for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. *54*

[97] Chavdar Papazov, Tim K. Marks, and Michael Jones. Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. *82*

[98] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. *86, 92, 95*

[99] Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 84–91. IEEE, 1994. *34*

[100] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, October 2000. *33, 35, 45*

[101] Pittsburgh pattern recognition. `https://www.pittpatt.com`, 2020. Online; accessed December-2020. *37*

[102] Shanto Rahman, Md Mostafijur Rahman, M. Abdullah-Al-Wadud, Golam Dastegir Al-Quaderi, and Mohammad Shoyaib. An adaptive gamma correction for image enhancement. *EURASIP Journal on Image and Video Processing*, 2016(1):35, 2016. *93, 96, 97, 103, 105, 108*

[103] Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2879–2886, Washington, DC, USA, 2012. IEEE Computer Society. *69*

[104] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, Jan 2019. *82, 95*

[105] Mudassar Raza, Zonghai Chen, Saeed-Ur Rehman, Peng Wang, and Peng Bao. Appearance based pedestrians' head pose and body orientation estimation using deep learning. *Neurocomputing*, 272:647 – 659, 2018. *32*

[106] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. *20*

[107] ML Roemhildt, MG Gardner-Morse, CF Morgan, BD Beynnon, and GJ Badger. Calcium phosphate particulates increase friction in the rat knee joint. *Osteoarthritis and cartilage*, 22(5):706–709, 2014. *59*

[108] M. Roth and D. M. Gavrila. Dd-pose - a large-scale driver head pose benchmark. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 927–934, June 2019. *55, 58*

[109] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015. *92, 98, 106, 108*

[110] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. *20*

[111] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. *92, 95, 102*

[112] Anke Schwarz, Monica Haurilet, Manuel Martinez, and Rainer Stiefelhagen. Driveahead-a large-scale driver head pose dataset. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition Workshops, pages 1–10, 2017. *54, 55, 58, 70, 77, 82, 85, 89, 90*

[113] Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker. Autopose: Large-scale automotive driver head pose and gaze dataset with deep head orientation baseline. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020. *14, 15, 85, 90*

[114] Mohamed Selim, Tewodros Habtegebrial, and Didier Stricker. Facial image aesthetics prediction with visual and deep cnn features. In *IMVIP 2017, Irish Machine Vision and Image Processing Conference*, 2017. *13, 14*

[115] Mohamed Selim, Stephan Krauß, Tewodros Amberbir Habtegebrial, Alain Pagani, and Didier Stricker. Deep orientation-guided gender recognition from face images. In *12th International Conference on Pattern Recognition Systems (ICPRS)*, pages 1–6, 2022. *16*

[116] Mohamed Selim, Alain Pagani, and Didier Stricker. Real-time head pose estimation using multi-variate rvm on faces in the wild. In George Azzopardi and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, pages 254–265, Cham, 2015. Springer International Publishing. *13, 15, 31*

[117] Mohamed Selim, Alain Pagani, and Didier Stricker. Sparse-mvrvms tree for fast and accurate head pose estimation in the wild. In Michael Felsberg, Anders Heyden, and Norbert Krüger, editors, *Computer Analysis of Images and Patterns*, pages 240–250, Cham, 2017. Springer International Publishing. *13, 15, 31*

[118] Mohamed Selim, Shekhar Raheja, and Didier Stricker. Real-time human age estimation based on facial images using uniform local binary patterns. In *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, volume 2, pages 408–415, 01 2015. *13*

[119] Mohamed Selim, Suraj Sundararajan, Alain Pagani, and Didier Stricker. Image quality-aware deep networks ensemble for efficient gender recognition in the wild. In *VISAPP 2018 - 13th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, pages 351–358, 01 2018. *14, 16*

[120] John Sell and Patrick O'Connor. The xbox one system on a chip and kinect sensor. *IEEE Micro*, 34(2):44–53, 2014. *62*

[121] Mili Shah, Roger D. Eastman, and Tsai Hong. An overview of robot-sensor calibration methods for evaluation of perception systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, PerMIS '12, pages 15–20, New York, NY, USA, 2012. ACM. *66*

[122] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. *92, 95, 106, 107*

[123] Aaron Staranowicz, Garrett R Brown, Fabio Morbidi, and Gian Luca Mariottini. Easy-to-use and accurate calibration of rgb-d cameras from spheres. In *Pacific-Rim Symposium on Image and Video Technology*, pages 265–278. Springer, 2013. *66*

[124] Klaus H Strobl and Gerd Hirzinger. Optimal hand-eye calibration. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 4647–4653. IEEE, 2006. *66*

[125] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. *92*

[126] Kalyan Sunkavalli, Neel Joshi, Sing Bing Kang, Michael F Cohen, and Hanspeter Pfister. Video snapshots: Creating high-quality images from video clips. *IEEE transactions on visualization and computer graphics*, 18(11):1868–1879, 2012. *100*

[127] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014. *20*

[128] Wolfgang Teufl, Markus Miezal, Bertram Taetz, Michael Fröhlich, and Gabriele Bleser. Validity of inertial sensor based 3d joint kinematics of static and dynamic sport and physiotherapy specific movements. *PLOS ONE*, 14:e0213064, 02 2019. *60*

[129] Arasanathan Thayananthan, Ramanan Navaratnam, Bjoern Stenger, PhilipH.S. Torr, and Roberto Cipolla. Multivariate relevance vector machines for tracking. 3953:124–138, 2006. *40*

[130] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001. *40*

[131] Chun-Ming Tsai and Zong-Mu Yeh. Contrast enhancement by automatic and parameter-free piecewise linear transformation for color images. *IEEE transactions on Consumer Electronics*, 54(2):213–219, 2008. *97*

[132] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989. *66, 67*

[133] Scott E Umbaugh. *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press, 2010. *61*

[134] R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *Image Processing, IEEE Transactions on*, 21(2):802–815, Feb 2012. *32*

[135] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. *110*

[136] Alessandro Vianello, Francesco Michielin, Giancarlo Calvagno, Piergiorgio Sartor, and Oliver Erdler. Depth images super-resolution: An iterative approach. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3778–3782. IEEE, 2014. *62*

[137] Vicon. `https://www.vicon.com/`, 2020. Online; accessed December-2020. *59*

[138] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. *12, 27, 35*

[139] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. *110*

[140] W. C. Wang, R. Y. Hsu, C. R. Huang, and L. Y. Syu. Video gender recognition using temporal coherent face descriptor. In *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 1–6, June 2015. *95, 106, 107*

[141] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7289–7298, 2019. *111*

[142] Oliver Wasenmüller, Gabriele Bleser, and Didier Stricker. Combined bilateral filter for enhanced real-time upsampling of depth images. In *VISAPP (1)*, pages 5–12, 2015. *62*

[143] Oliver Wasenmüller, Marcel Meyer, and Didier Stricker. CoRBS: Comprehensive rgb-d benchmark for slam using kinect v2. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, March 2016. *59, 68*

[144] Oliver Wasenmüller and Didier Stricker. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma, editors, *Computer Vision – ACCV 2016 Workshops*, pages 34–45, Cham, 2017. Springer International Publishing. *62*

[145] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 499–515, Cham, 2016. Springer International Publishing. *37*

[146] Wikipedia, the free encyclopedia. Adaptive histogram equalization, 2016. *97, 116*

[147] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–534. IEEE, 2011. *35*

[148] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. *20*

[149] L. Xia, C. Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *CVPR 2011 WORKSHOPS*, pages 15–22, 2011. *57*

[150] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. *110*

[151] Ming-Hsuan Yang, David J Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1):34–58, 2002. *27*

[152] Qualisys. `https://www.qualisys.com/`, 2020. Online; accessed December-2020. *59*

[153] Stefanos Zafeiriou, Cha Zhang, and Zhengyou Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015. *27*

[154] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Evaluation of appearance-based methods and implications for gaze-based applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI-19, pages 1–13, New York, NY, USA, 2019. Association for Computing Machinery. *32*

[155] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015. *32, 55*

[156] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003. *27*

[157] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. *20*

[158] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016. *102*

[159] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886. IEEE, 2012. *34, 41*

Curriculum Vitae

Mohamed Selim

Education

2010 – 2012	**Master of Science in Computer Science and Engineering** German University in Cairo (GUC), Egypt *Thesis project conducted at the German Research Center for Artificial Intelligence (DFKI)* *Department Knowledge Management, Kaiserslautern, Germany*
2006 – 2010	**Bachelor of Science in Computer Science and Engineering** German University in Cairo (GUC), Egypt

Work Experience

since 03.2018	**Researcher** German Research Center for Artificial Intelligence (DFKI) Department Augmented Vision Kaiserslautern, Germany
03.2017 – 02.2018	**Research Assistant** TU Kaiserslautern Faculty of Computer Science
10.2012 – 02.2017	**Research Assistant** German Research Center for Artificial Intelligence (DFKI) Department Augmented Vision Kaiserslautern, Germany
03.2012 – 09.2012	**Teaching Assistant** German University in Cairo (GUC) Faculty of Media Engineering and Technology Cairo, Egypt

Publication list

Author's publication list as of October 2022

1. Mohamed Selim, Stephan Krauß, Tewodros Amberbir Habtegebrial, Alain Pagani, Didier Stricker.
   Deep Orientation-Guided Gender Recognition from Face Images. In *International Conference on Pattern Recognition Systems (ICPRS)*, 2022.

2. Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker
   AutoPOSE: Large-Scale Automotive Driver Head Pose and Gaze Dataset with Deep Head Orientation Baseline. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

3. Ahmet Firintepe, Mohamed Selim, Alain Pagani, and Didier Stricker.
   The More, the Merrier? A Study on In-Car IR-Based Head Pose Estimation . In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

4. Mohamed Selim, Suraj Sundararajan, Alain Pagani, and Didier Stricker.
   Image Quality-Aware Deep Networks Ensemble for Efficient Gender Recognition in the Wild. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2018.

5. Mohamed Selim, Alain Pagani, and Didier Stricker.
   Sparse-MVRVMs Tree for Fast and Accurate Head Pose Estimation in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2017.

6. Mohamed Selim, Alain Pagani, and Didier Stricker.
   Real-Time Head Pose Estimation Using Multi-Variate RVM on Faces in the Wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015.

7. Hartmut Feld, Bruno Mirbach, Jigyasa Katrolia, Mohamed Selim; Oliver Wasenmüller, and Didier Stricker.
   DFKI Cabin Simulator: A Test Platform for Visual In-Cabin Monitoring Functions. In *International Commercial Vehicle Technology Symposium CVT*, 2020.

8. Mohamed Selim, Tewodros Amberbir Habtegebrial, and Didier Stricker.
   Facial Image Aesthetics Prediction with Visual and Deep CNN Features. In *Irish Machine Vision and Image Processing Conference Irish Machine Vision and Image Processing Conference (IMVIP)*, 2017.

9. Mohamed Selim, Shekhar Raheja, and Didier Stricker.
   Real-time Human Age Estimation based on Facial Images using Uniform Local Binary Patterns. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2015.

10. Xiaohai Lin, Daniel Görges, Sebastian Schöffel, Johannes Schwank, Pascal Stahl, Achim Ebert, Mohamed Selim, and Didier Stricker.
    Eco-Driving Assistance Systems for Commercial Vehicles. In *International Commercial Vehicle Technology Symposium CVT*, 2016.