TÉCNICO LISBOA

2D RGB Head Pose Estimation in Face Occlusion Scenarios

José Carlos Faria Celestino

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Dr. Manuel Ricardo de Almeida Rodrigues Marques
Prof. João Paulo Salgado Arriscado Costeira

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Dr. Manuel Ricardo de Almeida Rodrigues Marques
Member of the Committee: Prof. Jacinto Carlos Marques Peixoto do Nascimento

November 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

A conclusão desta dissertação assinala o passo final neste longo percurso académico. Foram 5 anos verdadeiramente únicos que passei neste instituto, uma época que relembrarei para sempre. As conquistas e os momentos ficam na memória, mas as amizades que aqui fiz são para toda a vida.

Um agradecimento especial aos meus orientadores, o Dr. Manuel Marques e o professor João Paulo Costeira, por todo o apoio que me deram ao longo destes meses e por todo o entusiasmo que me transmitiram na realização deste projeto. Pelo constante incentivo e desafio para ir mais além, que me inspirou sempre a alcançar mais do que julgava que conseguiria. São um verdadeiro exemplo para mim.

Quero agradecer à minha mãe, Lígia, por ter estado sempre a meu lado nesta jornada, não só na tese mas em todo o meu percurso no Técnico, nos bons e maus momentos. Agradeço ao meu pai, José, por ter feito de tudo para garantir o meu sucesso académico e aos agradeço aos meus avós, Carlos e Elisa, o incondicionável apoio que me dão. A minha família foi o pilar de tudo o que conquistei ao longo destes anos. Devo-lhes tudo.

Um grande obrigado a todos os meus amigos que partilharam comigo as gargalhadas, as aventuras e as lutas desta vida de estudante. Qualquer que seja o momento passado ao longo destes 5 anos, foi sem dúvida melhor graças a vocês. As amizades que fiz desde o início no Taguspark até esta conclusão na Alameda foram a melhor parte de todo este trajeto. Nunca esquecerei os nossos momentos, e mal posso esperar pelos próximos!

A todos os que fizeram parte do meu percurso até aqui, o meu mais profundo agradecimento.

Abstract

Head pose estimation, the task that deals with the prediction of the orientation of human heads from images or videos, is a challenging Computer Vision problem that has been extensively researched and has a wide variety of applications. Despite several studies having been carried out to achieve the most accurate pose prediction possible, current state of the art systems still exhibit a much larger estimation error in the presence of occlusions. This inaccurate prediction makes them inadequate and unreliable for many task applications in such occlusion scenarios.

This thesis proposes to study different methodologies in order to achieve a robust head pose estimation in occlusion scenarios, for images, videos and real-world applications. The implemented methodologies are based on the development of personalized occluded training and testing sets and the adaptation of state of the art deep learning network frameworks and strategies.

We show that our models improve occluded head pose estimation and achieve state of the art non-occluded estimation results. We demonstrate the application of our best method in the real-life context of Feedbot, an autonomous feeding robotic arm. We reveal that our model performs better than a state of the art model for the occlusions of the robotic arm, while achieving similar performance for non-occluded estimation.

Keywords

Head Pose Estimation, Euler Angles, Occlusion, Neural Networks

Resumo

A estimação da pose da cabeça, a tarefa que envolve a previsão da orientação da cabeça a partir de imagens ou vídeos, é um problema desafiante que tem sido investigado extensivamente e possui uma grande variedade de aplicações. Apesar de vários estudos terem sido realizados para alcançar a estimação mais precisa possível, os sistemas estado de arte ainda apresentam um erro de estimativa considerável na presença de oclusões. Esta previsão imprecisa torna-os inadequados e pouco fiáveis para muitas aplicações nas quais ocorre a oclusão da cabeça.

Esta tese propõe o estudo de diferentes metodologias a fim de se conseguir uma estimativa robusta da pose da cabeça em cenários de oclusão, para imagens, vídeos ou mesmo aplicações em tempo real. Esta metodologia será baseada no desenvolvimento de datasets de treino e teste ocluídos e na adaptação de estratégias baseadas em redes neuronais e que apresentam resultados estado de arte.

Comprovamos que os nossos modelos melhoram a estimativa da pose da cabeça com oclusões e alcançam resultados estado da arte para imagens sem oclusão. Demonstramos a aplicação do nosso melhor método no contexto da vida real do Feedbot, um braço robótico autónomo para alimentação assistida. Verificamos que, na presença de oclusões do braço robótico, o nosso modelo apresenta melhor estimação do que um modelo de estado de arte, alcançando um desempenho semelhante para a estimação sem oclusões.

Palavras Chave

Estimação da Pose da Cabeça, Ângulos de Euler, Oclusão, Rede Neuronal

Contents

1 Introduction **1**

1.1 Motivation . . . 3

1.2 Problem and Contributions . . . 4

1.3 Outline . . . 5

2 State Of The Art **7**

2.1 Head Pose Estimation . . . 9

2.1.1 Model-Based Strategies . . . 10

2.1.2 Occlusion Related Works . . . 12

2.1.3 Learning-Based Strategies . . . 14

2.2 Summary . . . 16

3 Generating a Synthetic Occluded Dataset **19**

3.1 Synthetic Occlusion Generation Procedure . . . 21

3.2 Head Pose Datasets . . . 24

4 Methodologies For Head Pose Estimation With Occlusions **27**

4.1 End-to-end Multi-loss Approach With Latent Space Regression . . . 29

4.2 Occluded Head Pose Estimation Through Face Reconstruction . . . 32

4.2.1 Reconstruction Loss . . . 33

4.3 Multi-Loss Autoencoder For Occluded Head Pose Estimation . . . 36

5 Results and Discussion **39**

5.1 Results . . . 41

5.1.1 Multi-loss Head Pose Estimation With Latent Space Regression . . . 41

5.1.1.A Angle Regression Weight Study . . . 41

5.1.1.B Latent Space Regression Weight Study . . . 42

5.1.2 Occluded Head Pose Estimation Through Face Reconstruction . . . 44

5.1.3 Multi-Loss Autoencoder For Occluded Head Pose Estimation . . . 47

5.2 Method Results Comparison and Discussion . . . 48

5.2.1 Testing Pose Estimation in the Feedbot Scenario . . . 49

6 Conclusions and Future Work **53**

6.1 Conclusions 55

6.2 Method Limitations and Future Work 55

Bibliography **57**

A Autoencoder and U-Net Architecture **63**

A.1 Autoencoder 63

A.1.1 U-Net architecture 64

B Data Pre-Processing **67**

List of Figures

1.1 Head pose estimation in 2D image depicted by Euler rotation angles - Yaw, Pitch, Roll. . . 3
1.2 Head pose problem for occlusion scenarios 3
1.3 Head pose estimation problem. 4

2.1 Rigid transformation that defines head pose estimation [1]. 9
2.2 Head Pose Euler Angles - Yaw, Pitch and Roll. 10
2.3 Facial landmarks detection example [2]. 10
2.4 PnP problem illustration [3]. 11
2.5 Pose estimation through Keypoints and Landmarks [4]. 11
2.6 3DDFA architecture [5]. 12
2.7 Combined head pose framework under facial occlusions [6]. 13
2.8 Multi-loss deep leaning architecture [7]. 14
2.9 FSA-Net architecture [2]. 15

3.1 RGB and depth image captured by RGB-D camera. 21
3.2 Overview of our synthetic occlusion generation procedure. 22
3.3 Determining threshold depth for occlusion segmentation. 23
3.4 Synthetic occlusion generation in non-occluded images. 23
3.5 Examples of image data and ground truth pose in datasets. 25

4.1 Multi-loss head pose estimation framework with latent space regression. 29
4.2 Euler angle prediction vector through fully-connected layer. 30
4.3 Ground truth latent space and error minimization. 32
4.4 Framework for occluded head pose estimation through face reconstruction. 33
4.5 Simplified standard U-Net architecture. 34
4.6 Training framework for autoencoder. 35
4.7 Multi-Loss autoencoder high-level pipeline. 36
4.8 Training procedure stages for the pipeline described in figure 4.7. 38

5.1 Histograms of Euler angle bin frequency for each dataset. 44
5.2 Reconstruction comparison for different loss weights. 45
5.3 Examples of face reconstruction results for each dataset. 46
5.4 Feeding context of Feedbot. 50
5.5 Head pose estimation comparison between *Hopenet* and our model - non-occluded frames. 51
5.6 Head pose estimation comparison between *Hopenet* and our model - occluded frames. . 52
5.7 Comparison between *Hopenet* and our model in frames with larger occlusions. 52

A.1 Basic autoencoder structure. 64
A.2 PCA(linear) vs. autoencoder(non-linear) dimensionality reduction [8]. 65
A.3 U-Net architecture from [9]. 66

B.1 Original and adjusted bounding boxes. 68

List of Tables

5.1 Head pose estimation MAE° tests with BIWI for different angle regression weights (α). . . 42
5.2 Head pose estimation MAE° tests with AFLW2000 for different angle regression weights (α). . . 42
5.3 Head pose estimation MAE° tests with BIWI for different latent space regression weights (β). . . 43
5.4 Head pose estimation MAE° tests with AFLW2000 for different latent space regression weights (β). . . 43
5.5 Head pose estimation MAE° results for reconstructed images in both datasets. . . 46
5.6 Parameters for each ResUnet trained model. . . 47
5.7 Head pose estimation MAE° results for ResUnet. . . 48
5.8 Method comparison in BIWI. . . 48
5.9 Method comparison in AFLW2000. . . 48

Acronyms

HPE	Head Pose Estimation
CNN	Convolutional Neural Network
DL	Deep Learning
PnP	Perspective-n-Point
MSE	Mean Squared Error
MAE	Mean Absolute Error
DoF	Degree of Freedom
SSR	Soft Stagewise Regression
VDC	Vertex Distance Cost
FPN	Feature Pyramid Networks
WPDC	Weighted Parameter Distance Cost
DBSCAN	Density-based spatial clustering of applications with noise
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
SSIM	Structural Similarity Index Measure
GAN	Generative Adversarial Network

1

Introduction

Contents

1.1 Motivation 3

1.2 Problem and Contributions 4

1.3 Outline 5

1.1 Motivation

Extensively researched over the last 25 years [10], 2D Head Pose Estimation (HPE) is a challenging but compelling and relevant computer vision problem, essentially due to the wide variety of applications for which it can be used, such as driving aid systems [11], motion capture [12] and gaze estimation. Succinctly, this problem consists in approximately determining the orientation of a head in a 2D image, as exemplified in figure 1.1.

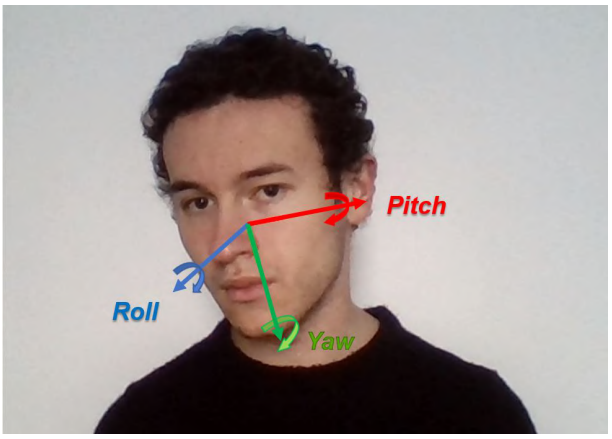
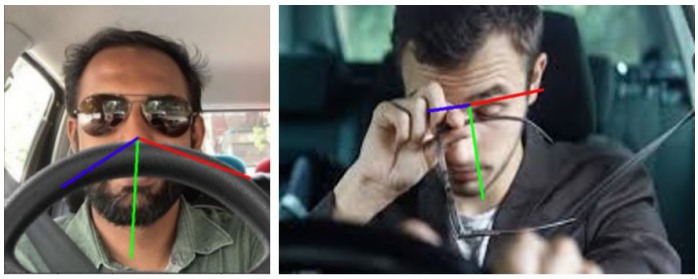
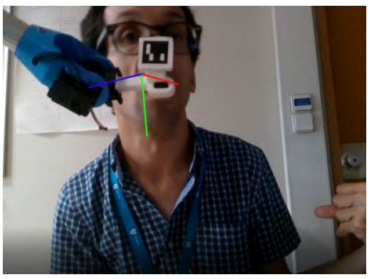


Figure 1.1: Head pose estimation in 2D image depicted by Euler rotation angles - Yaw, Pitch, Roll.

Many new methods have been introduced throughout the recent years to address the HPE problem. The recent advances in deep learning have allowed to achieve significant improvements in the accuracy and speed of estimations. Despite the constant progress, current state of the art systems scarcely approach one of the most challenging and common problems in HPE, the occurrence of facial occlusions. This issue affects several aspects common to the estimation process, such as the detection of the face or the detection of facial landmarks, and therefore often leads to highly inaccurate predictions in the presence of object occlusions or self-occlusions, as exemplified in figure 1.2.



(a) Occlusions in driver attention systems



(b) Occlusions in autonomous assisted feeding

Figure 1.2: Head pose problem for occlusion scenarios (Blue axis points towards estimated face direction).

Current state of the art HPE systems exhibit substantial estimation error when a face is partially occluded by a certain object, which makes them unreliable for many real-world applications. One example of this is the case of Feedbot [13] (figure 1.2(b)), an autonomous feeding robot arm developed for people with upper arm disabilities, capable of acquiring the food from a plate and track the user's mouth in order to feed him. A limitation of the current Feedbot implementation[1] is that the detection of the user's intention of feeding is time-based. An head pose estimator could be implemented to employ a human-robot interface that would use the estimated pose to track the gaze of the user within a known environment, and determine if the user is looking at the plate, and therefore intends to be fed. However, the trajectory of the feeding robotic arm often occludes the user's face, which means that an head pose estimator that is robust to occlusions would be required, otherwise this implementation is impossible. Lately, driving safety systems also use HPE as an evaluation parameter and tool for the detection of driver distractions [14] [15] (figure 1.2(a)). This scenario is also susceptible to several occlusions from either the wheel, arms or hands and therefore the reliability of this kind of system would also benefit from an head pose estimation that is robust to occlusions.

To address the issue, this thesis aims to study different methods and ways of approaching the occluded HPE challenge, all based on the use of deep learning solutions and with the aid of synthetic occluded datasets. We seek to achieve robust 2D head pose estimation for occluded faces and extend on current works that achieve state of the art estimation in non-occluded benchmark datasets. The intent is that the developed framework can be implemented in images, videos and real-world applications where head occlusion occurs often and/or for long periods of time. In order to improve the robustness of the estimation, our synthetic training dataset will include diverse occlusions for several different facial regions and head orientations. With this work, we propose ways of accurately estimating the user's head pose, regardless of the part of the face that is occluded, and present a procedure to generate synthetic face occlusions in any head pose dataset.

1.2 Problem and Contributions

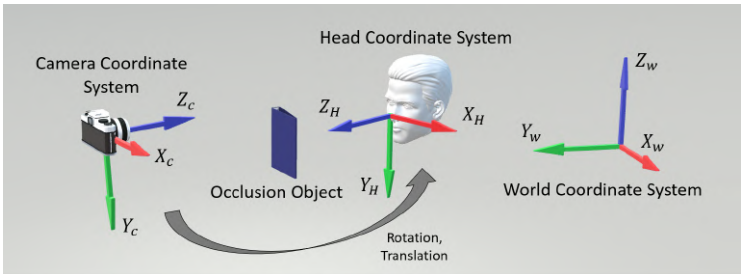


Figure 1.3: Head pose estimation problem.

[1] http://users.isr.ist.utl.pt/~manuel/FeedBot/

The head pose estimation problem is exemplified in figure 1.3. It corresponds to finding the three-dimensional geometric transformation (translation and rotation) that maps the camera coordinate system into the head coordinate system. In this dissertation we focus in approximately determining, from 2D images and even in the presence of occlusions, the orientation of an head in the reference world coordinate system with respect to a camera.

The work and study developed throughout this thesis had the following scientific contributions:

- **2D Head Pose Estimation Methods For Face Occlusion Scenarios**: We developed methods that vastly improve the results that state of the art head pose estimation methods produce for partially occluded face images, while also maintaining/improving the accuracy for non-occluded images. The proposed strategies include:

  1. A multi-loss neural network approach with latent space regression;
  2. Face reconstruction from occluded images using auto-encoders;
  3. Multi-loss auto-encoder for face reconstruction.

- **Development Of Synthetic Occlusion Generation Procedure And Occluded Datasets For Training And Testing**: We established a procedure that allows the development of new synthetically occluded datasets of face images. We apply it to some of the most commonly benchmarked datasets for head pose estimation, such as 300W-LP [16], BIWI [17] and AFLW2000 [18] and generate occluded versions of them.

1.3 Outline

This thesis is organized in the following way:

- **Chapter 1 - Introduction**: Motivation, contributions and outline of this thesis.
- **Chapter 2 - State of Art**: Overview of the state of the art for the subject, featuring the projects that display the best results in head pose estimation and how they approached the problem, featuring the advantages and disadvantages of each.
- **Chapter 3 - Generating a Synthetic Occluded Dataset**: Describing a procedure for the generation of synthetically occluded face images and datasets to be used in training and testing.
- **Chapter 4 - Methodologies For Head Pose Estimation With Occlusions**: Presentation of the methodologies and strategies studied and implemented throughout this thesis, with detailed framework for each.

- **Chapter 5 - Results and Discussion**: Display of the head pose estimation results for all implemented strategies, along with throughout discussion and comparison.
- **Chapter 6 - Conclusions and Future Work**: Overall conclusions for the developed work and discussion of possible future improvements and strategies.

2

State Of The Art

Contents

2.1 Head Pose Estimation 9
2.2 Summary 16

2.1 Head Pose Estimation

Head Pose Estimation has become a very popular research area in the past two to three decades. New approaches continue to emerge every year and set new state of the art estimation results in the main benchmark datasets [16] [17] [18]. Deep Learning (DL) has contributed to improve the performance of many Computer Vision systems, in particular, to solve the HPE task efficiently, with real-time estimation with acceptable low error.

Essentially, this task consists in approximately determining the orientation and position of a human head with respect to a camera within a 3D environment. Envisioning the head as a rigid object, it corresponds to the 3D rigid transformation (rotation and translation) that maps the camera coordinate system into the head model coordinate system (figure 2.1).

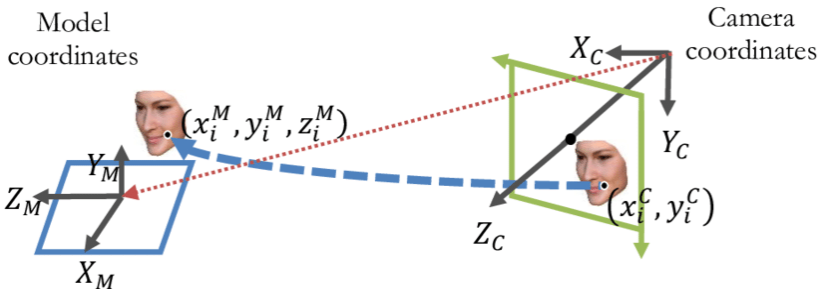


Figure 2.1: Rigid transformation that defines head pose estimation [1].

As described in the extensive study of HPE given in [19], the literature for this task is divided in methods based 2D images and methods based on depth data [20] [21] [22]. Our focus is the estimation of pose in 2D images that can be implemented with any conventional RGB camera. Since we will be dealing with 2D data, determining the position of the head mainly comes down to estimating it's location in an image rather that in the 3D coordinates, hence being a face localization task. Therefore, the focal point of HPE will be to predict the 3D orientation of the head. This orientation is commonly depicted through Euler angles - Yaw, Pitch and Roll - exemplified in figure 2.2.

Generally, there are two main distinct ways of handling this problem in 2D data: The model-based approach and the learning-based approach. Model-based methods resort to the estimation of facial landmarks and 3D computer vision techniques for pose estimation. These landmarks correspond to main facial features such as the nose, eyes, mouth and chin (figure 2.3).

Learning-based methods go for a more straightforward end-to-end approach of directly performing the estimation from 2D RGB images inputs to neural networks. We will analyze related works for both courses of action as well as some works that deal with the occlusion challenge.

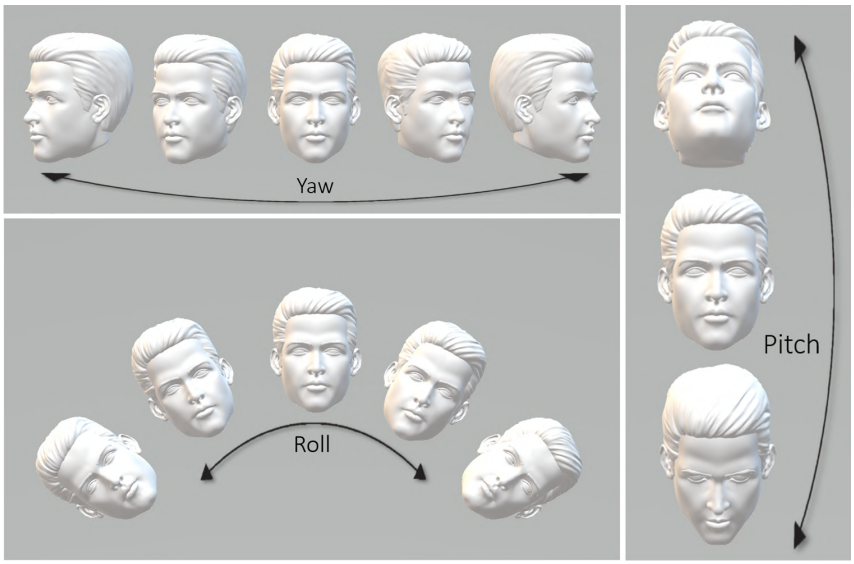


Figure 2.2: Head Pose Euler Angles - Yaw, Pitch and Roll.

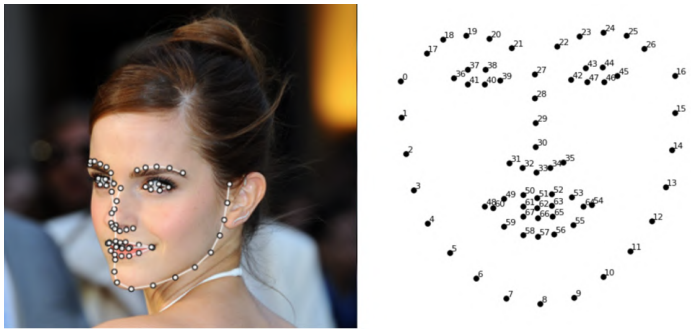


Figure 2.3: Facial landmarks detection example [2].

2.1.1 Model-Based Strategies

Yinguobing [23] provides a simple way of performing HPE through 3 steps: first, the detection of a face using a built-in face detector from OpenCV [24] which provides a face box containing a detected face. Then the cropped face box is used as input in a deep-learning facial landmark tracker implemented on TensorFlow [25] that outputs 68 facial landmarks as seen in figure 2.3. Once the image landmarks are obtained, the pose is calculated by solving a PnP problem [26], which is modeled by the pinhole camera model (figure 2.4). This problem consists in estimating the pose of an object (in this case, a head) given a set of 3D object points in the world, their corresponding 2D image projections, and the calibrated intrinsic camera parameters that define the matrix K in the figure below. The estimated pose consists in the rotation R and translation T that transforms the 3D points expressed in the world frame into the camera frame.

The authors from [4] extend on their previous work [27] [28] and propose a method developed for reli-

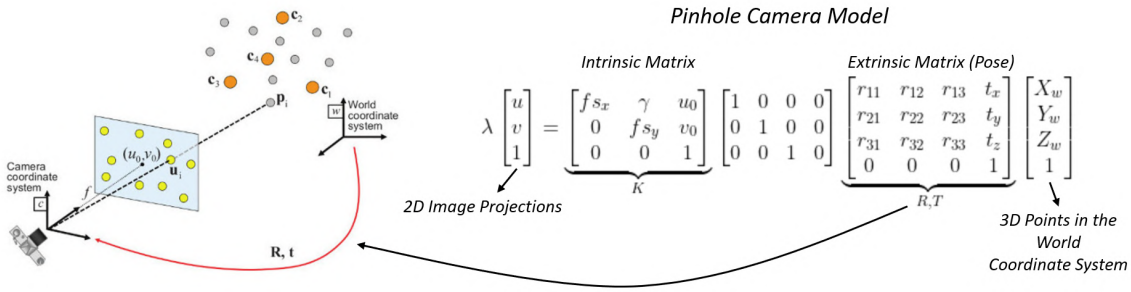


Figure 2.4: PnP problem illustration [3] and pose (R, T) in the pinhole camera model.

able real-time use that similarly uses 68 landmark prediction for head pose estimation but also computes the pose using predicted 2D keypoints of the head. These keypoints do not have fixed locations and can be located not only in facial areas but also in the back or top of the head. This characteristics make the pose estimation more robust to possible occlusions and illumination differences. They are detected using the Features from Accelerated Segment Test (FAST) [29] algorithm and a pyramidal Lucas-Kanade feature tracker [30] is implemented to find 2D keypoint correspondences between frames using optical flow [31]. A Kalman Filter [32] is used for the fusion of two strategies: a keypoint-based tracking strategy used for the prediction step of the Kalman Filter; and a facial-landmark detection strategy, used for the correction step. This pipeline is exemplified in figure 2.5.

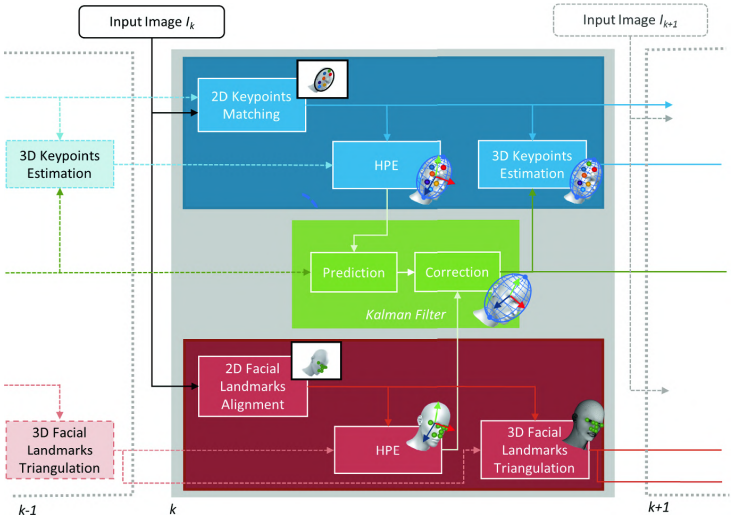


Figure 2.5: Pose estimation through Keypoints and Landmarks [4].

Ultimately, as shown in [4], this fusion method shows better results than using exclusively either standard keypoint or facial landmarks HPE methods.

The work developed in [5] focuses on the practical applications of performing 3D face alignment in

2D images, one of which HPE, and tries to achieve a balance between speed, accuracy and stability. It approaches the 3D face alignment problem as a 3D Morphable Model (3DMM) [33] parameters regression problem. These parameters include the vectorized similarity transformation matrix, face shape parameters and face expression parameters. According to [5], the similarity transformation matrix, which contains the rotation matrix and translation vector that define the 3D orientation and location of the head, is used for regression instead of Euler angles. The reason for that is to avoid the ambiguity caused by the gimbal lock [34] problem that occurs when faces get close to profile view. This ambiguity is undesired as it would degrade the performance of the implemented regressor. The architecture of this method is as exemplified in figure 2.6, with the 3DMM parameters being regressed using a fast lightweight backbone Convolutional Neural Network (CNN) such as MobileNet [35]. The authors implement a strategy called meta-joint optimization, which consists in applying a cost function with two terms, Weighted Parameter Distance Cost (WPDC) and Vertex Distance Cost (VDC). This losses minimize the vertex distances between a fitted 3D face and the ground truth. A landmark-regression regularization is also implemented to further aid the optimization problem to achieve higher accuracy. To ensure better stability on videos or real-time estimations, the authors establish a 3D aided short-video-synthesis method where one still image is transformed into a short synthetic video using in-plane and out-of-plane rotations. This helps to obtain a trained model that produces smoother results and avoid random jittering when running on videos.

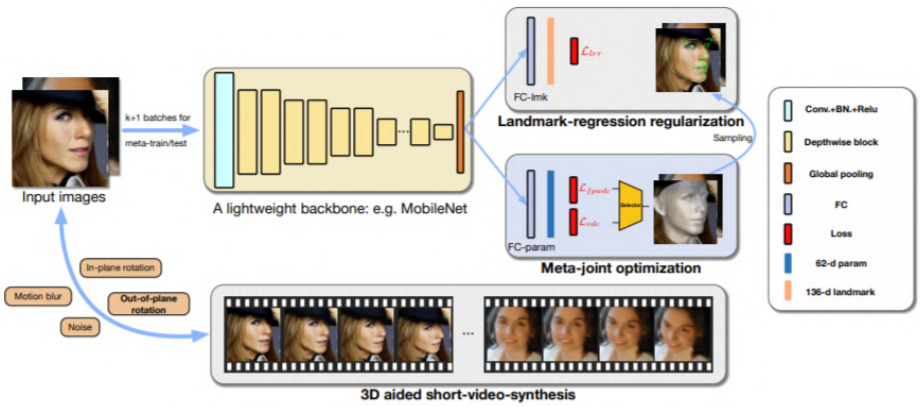


Figure 2.6: Architecture of the method in [5].

2.1.2 Occlusion Related Works

There are not many works that directly address the facial occlusion problem. The authors in [36] detail an algorithm which estimates the head pose of partially occluded faces by tracking the displacement of a face feature with respect to the center of the head. CamShift [37], an object tracking algorithm

method, is used to track the center of the head and a iterative Lucas-Kanade optical flow tracker [30] is used to track the feature face point. However, this method is specifically oriented for head-mounted displays and requires the mouth not to be occluded. Furthermore, it is based on outdated software and hardware, and does not present relevant quantitative results. The authors of [38] focus on achieving robust facial landmark detection for severe occlusions and images with large head poses. To tackle both challenges, it introduces the estimation of landmark visibility probabilities to measure if a landmark is visible, and performs occlusion prediction. One unified model is trained to be able to deal with different kinds of occlusion. A prior occlusion pattern loss is added as a constraint to aid the performance of the prediction. This work, however, does not have real-time tracking capabilities and does not focus on estimating poses.

The method of [6] improves on this and proposes a unified framework (figure 2.7) for simultaneous estimation of facial landmark locations, head pose and facial deformation under facial occlusions. This framework achieves robustness to facial occlusion by estimating landmark visibility probabilities, similarly to [38]. Only visible points are used for pose estimation, which aids to attain more accurate results.

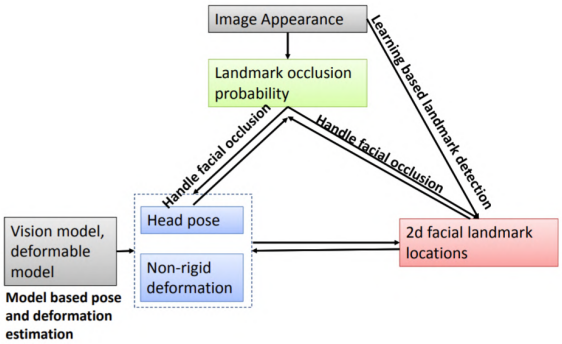


Figure 2.7: Combined framework for the method in [6].

Within this method, landmark locations are initialized using a mean face, all landmarks are presumed to be visible, and it is assumed that there is a frontal initial pose and no non-rigid facial deformation. The facial landmark locations, head pose, facial deformation coefficients and landmark visibility vector are then estimated by performing an iterative cascade method. This procedure allows to fully exploit their joint relationships and update each parameter based on the previously estimated values of the others. According to the authors, the combined framework outperforms the baseline landmark estimation results of [38] and achieves better results in head pose estimation than other methods that use all landmarks (as a rigid model) instead of only the ones that are visible. Despite pose estimation being one of it's goals, this work only evaluates yaw angles and has low accuracy for larger yaws.

2.1.3 Learning-Based Strategies

The authors from [7] claim that methods that compute the head pose by estimating keypoints or landmarks and solving the 2D to 3D correspondence Perspective-n-Point (PnP) problem with a mean human head model are more fragile methods as they rely on the head model chosen and are very sensitive to any errors that might occur in the detection and tracking of landmarks/keypoints. Another mentioned disadvantage is that landmark-to-pose methods can't output poses if there aren't sufficient detected landmarks. To avoid these drawbacks, they propose another way of approaching this problem by skipping the landmark detection step and using a deep neural network to predict the Euler angles directly from 2D RGB images intensities. The method developed in [7], follows the framework exemplified in figure 2.8.

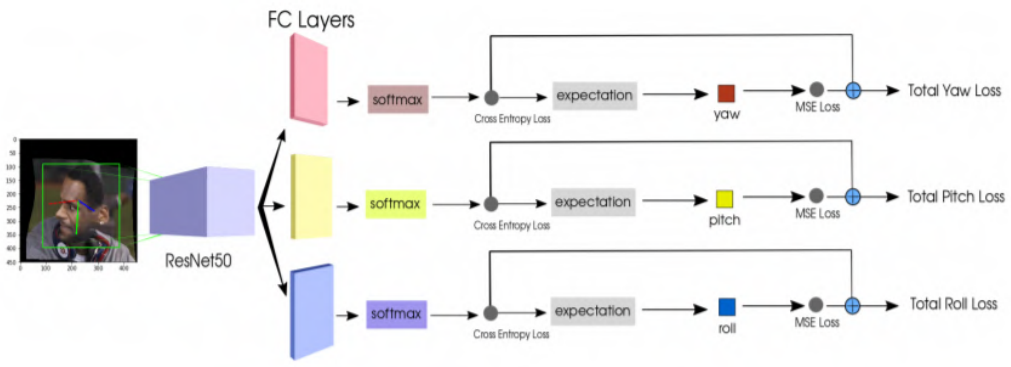


Figure 2.8: Multi-loss deep leaning architecture [7].

This architecture involves the input of normalized 2D RGB face images into a backbone network that might consist of any CNN, even though the authors highlight the performance of ResNet50 [39] and AlexNet [40]. The network is augmented with three fully-connected layers that share the previous convolutional layers, each one predicting a different Euler angle, establishing a three-branch neural network framework that implements a multi-loss approach for each angle. The fully-connected layers output logits, vectors of n raw (non-normalized) binned predictions which are passed to a softmax function that generates a vector of normalized probabilities for each possible class/bin. For this classification, the authors propose using vectors of 66 bins with a width of $3°$. This vectors cover the range $[-99°,99°]$ for all angles. The multi-loss approach defines that the overall loss in every branch is the sum of a bin classification cross-entropy loss that uses predicted probability distribution vectors (the output of softmax layers) and a regression fine-grained Mean Squared Error (MSE) loss for the computed expectation of each output angle from the binned output. The use of separate layers for each Euler angle is advantageous since it makes possible to optimise losses individually, given that the regression loss is weighted. The classification component aids the model to predict the vicinity of the pose and the

regression component helps it to achieve fine-grained estimation.

Another solution, FSA-Net [2], defines a more complex method for pose estimation from a single RGB image based on regression and feature agreggation, applying the Soft Stagewise Regression (SSR) problem defined in [41] to the HPE challenge. Figure 2.9 exemplifies the overall architecture for the method, in which input face images go through two streams where feature maps are extracted and fused together across K stages. The fused feature maps are spatial grids in which each cell contains a feature representation of a particular spatial location. Afterwards, they are fed into a module that performs fine-grained structure mapping to extract a set of representative features, which are then aggregated to obtain a final smaller set of features, one for each stage, and used for soft-stage wise regression. The pose estimation problem in this method is interpreted as a pose bin/class classification, where stage outputs are probabilities distributions for the angle interval classes. Each successive stage refines the decision within an angle group assigned by the previous stage. The estimated pose is given by the SSR function, which corresponds to the summation of the product between probability distribution and the values of pose groups at each stage.

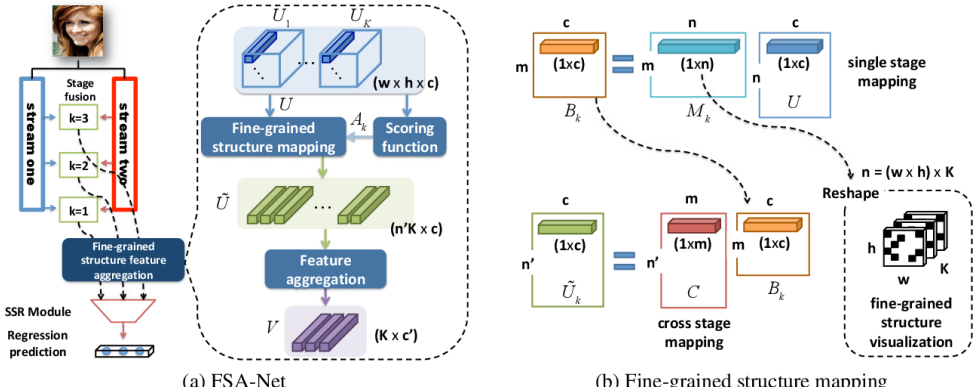


Figure 2.9: FSA-Net architecture [2].

The method *img2pose* presented in [42] proposes 3D face pose estimation based on Faster R-CNN [43] with Feature Pyramid Networks (FPN) [44], without facial landmarks or the use of face detectors. They point out disadvantages in HPE through landmark detection. Landmark detectors are optimized to the bounding boxes produced by specific face detectors, which means updating the face detector requires a new optimization of the landmark detector. Furthermore, the 68 facial landmarks are hard to detect in tiny faces in images, which makes it complicated to estimate their poses. To address these limitations they introduce a novel real-time capable solution to perform 6 Degree of Freedom (DoF) head pose estimation (3D rotation and translation vectors) in an image without requiring a prior face detection step. This approach is easier to estimate that landmark-based approaches since it is a 6-dimensional regression problem, while 5-point 2D landmark detection is 10-dimensional (5x2D), and 68 landmark implies the regression of 68x2D=136 elements. Moreover, 6 DoF pose can be transcribed to a 3D-2D

projection matrix which means that, with known intrinsic camera parameters, it possible to align the 3D face with its location in an image. For this reason, [42] performs simultaneous face detection and head pose estimation by regressing the 6 DoF which further improves the speed and robustness of the estimation.

The authors of Wide Headpose Estimation Network (*WHENet*) [10] builds on the end-to-end multi-loss approach of *Hopenet* [7] and extends it for full 360° yaw estimation, being the first fine-grained modern method applicable to the full-range of head yaws. The overall architecture is identical to the one of figure 2.8. However, there are four main differences in order to improve the method and adapt it to work with a yaw range of 360°. Firstly, the authors used binary-cross entropy with sigmoid activation as the classification loss, since this loss improved the estimation accuracy for the full-range task. The regression loss was also adapted to work with a full-range of yaws, since the MSE loss behaved erratically for bigger absolute yaw values. To replace the MSE loss, the authors introduced a wrapped loss that penalizes the minimal rotation angle required to align each yaw prediction with its corresponding dataset annotation:

$$L_{wrap} = \frac{1}{N_{batch}} \sum_{i}^{N_{batch}} min[|\Theta_{pred} - \Theta_{true}|^2, (360 - |\Theta_{pred} - \Theta_{true}|)^2] \tag{2.1}$$

This wrapped loss decreased errors in large yaws in approximately 50% when compared to MSE. Since their focus was developing a lighter mobile network that can be used in low-power systems, they opted to use a lighter backbone network, EfficientNet-B0 [45]. Another innovation is the introduction and generation of a new training dataset. Since no head pose dataset provides examples with (absolute) yaws larger than 100°, the authors generated a new dataset combining 300W-LP with computed Euler angle data from the CMU Panoptic Dataset [46]. The modifications made to *Hopenet* achieved state-of-the art of performance for full-range HPE.

2.2 Summary

Throughout the previous section, several works related to head pose estimation were analyzed as to explore different procedures and learn their respective advantages and disadvantages in regard to the objectives of this thesis. We saw that the literature on the challenge of occlusion in head pose estimation is scarce. The system in [36] presents a solution specifically intended for head-mounted display occlusion, using outdated software and hardware. The procedure in [38] addresses occlusions but focuses on landmark detection and is not extended for real-time tracking, and while the method in [6] includes pose estimation, it only evaluates yaw angles and displays low accuracy for large yaw values.

We also saw that model-based methods [23] [4] [33] have some crucial drawbacks in comparison to

end-to-end learning-based methods [7] [2] [42] [10]. They rely on the chosen head/face model (mainly those that do not account for facial deformation) and the estimation is very sensitive to landmark detection and tracking errors. They are also more susceptible both to self-occlusions (extreme poses for example) and object occlusions, since having a smaller number of detected landmarks greatly reduces the precision of the estimation of the pose.

Learning-based methods do not require the detection of landmarks and therefore avoid the occlusion problem mentioned above, a crucial drawback to the objective of this thesis. Furthermore, recent learning-based approaches have outperformed model-based methods and currently hold state of the art results [47]. For these reasons, our work will follow a learning-based approach and develop strategies based on some of the studied end-to-end deep learning frameworks while adapting them to be able to deal with the challenge of facial occlusions in head pose estimation.

3

Generating a Synthetic Occluded Dataset

Contents

3.1 Synthetic Occlusion Generation Procedure 21

3.2 Head Pose Datasets 24

In this work, we will follow a learning-based approach and resort to deep neural networks, which are multi-layered neural networks that learn to map specific sets of input to the desired outputs. To train these networks for the problem at hand, it is required to use adequate training data that comprises a wide variety of input facial occlusions images and respective labeled output head poses in order to allow the network mapping to work properly not only for the given examples, but also on new examples that were not seen by the model during the training, and therefore gain the ability to generalize.

Despite the fact that most head pose datasets include some partial occlusion examples, there aren't, to the best of our knowledge, datasets that are specifically oriented towards the occlusion challenge. For that reason, it is necessary to develop our own exclusively occluded datasets. Within this chapter, we will describe a procedure to generate synthetic occlusions in images and explain how we used it to create the datasets required to train and test our models.

3.1 Synthetic Occlusion Generation Procedure

Instead of recording our own images and manually labelling the respective head poses, a lengthy and difficult process since it is hard to capture sufficient distinct examples and to define the ground truth poses with precision, we opted for a different course of action. We use current existing head pose datasets that contain thousands of images and respective ground truth pose annotations in order to generate synthetic occlusions for all images and thus develop the new occluded datasets required for the training and testing of the deep neural networks.

Our procedure to generate synthetic occlusions in images is based on the use of 2D RGB color data and depth data (camera distance to an object). To that end, we require RGB-D sensors such as Microsoft's Kinect [48] or Intel's RealSense [49], which are depth sensing devices that work in association with a conventional RGB sensor camera and are capable of simultaneously recording 2D coloured images and the respective map of per-pixel depth information (figure 3.1).

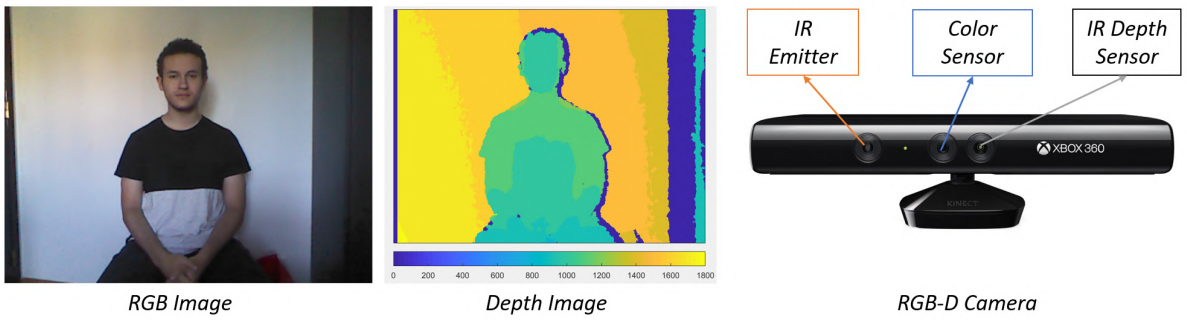


Figure 3.1: RGB image, depth Image (with scaled colors, representing depth in millimeters) and RGB-D camera (Kinect example).

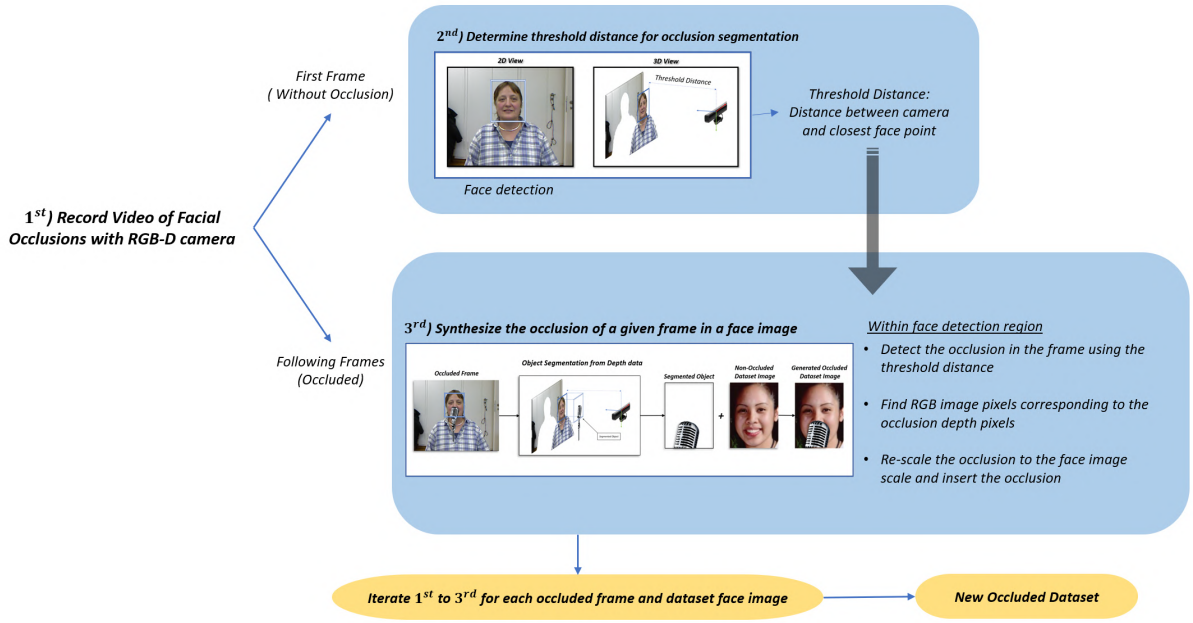


Figure 3.2: Overview of our synthetic occlusion generation procedure.

The main idea of this methodology is to perform depth-based object segmentation in videos recorded with RGB-D cameras in order to be able to insert any desired occlusion object in non-occluded face images. Figure 3.2 displays an overview of this procedure.

A first step in this procedure is to use the RGB-D sensor to record a video of RGB and depth data where different regions of the face of a person are occluded by an object. An essential aspect of this video is that, in the first frame, the face is not occluded. The reason for this is that initially we need to detect the face in the image in order to find the depth points that correspond to the image pixels within the face detection box [50]. We use the depth information determine the face point at minimum distance from the camera. This distance will serve as a threshold to separate the depth points that correspond to the head of the person from those that will correspond to occlusion objects. This procedure is exemplified in figure 3.3.

The depth data captured by RGB-D cameras might contain outliers that may be caused by the reflective nature of an object, by light intensities or sensor limitations [51]. When determining the threshold distance it is important to make sure that the depth points within the face detection region are actually face points and not sensor outliers, as that could lead to a wrong threshold. For that reason, we implement the Density-based spatial clustering of applications with noise (DBSCAN) algorithm [52], a density-based clustering algorithm that groups together points based on a distance measurement (commonly Euclidean distance, the length of the line that connects two points) and a minimum number of points for the clusters. With DBSCAN, points that exist in low-density regions and are not reachable from any

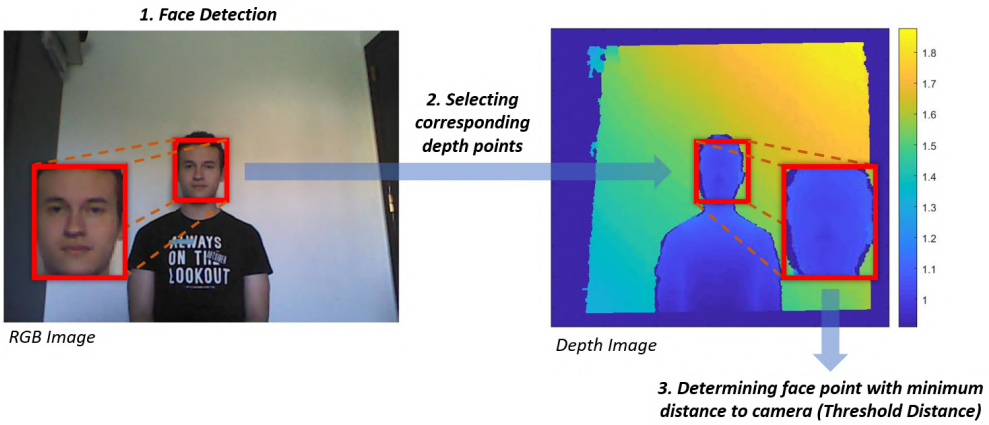


Figure 3.3: Determining threshold depth for occlusion segmentation.

other point are classified as outliers and are discarded in the clustering process. The face/head points can therefore be identified as the biggest cluster detected by the algorithm among the points contained in the face detection box and, free of outliers, the threshold distance can be correctly determined.

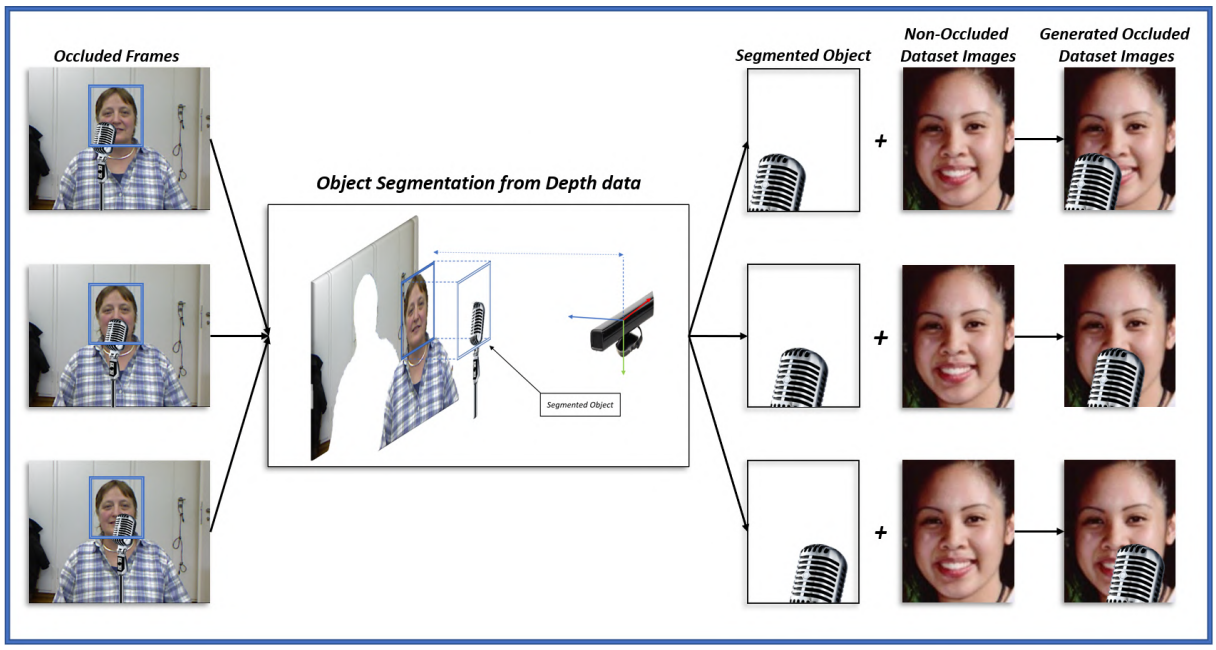


Figure 3.4: Synthetic occlusion generation in non-occluded images.

When this threshold is determined we can start to reproduce a synthetic occlusion in a given following frames. Figure 3.4 exemplifies the procedure with different video frames. For a given occluded frame of the video, the depth points corresponding to the RGB image points within the face detection box are extracted. From these depth points we select those that are at a shorter distance than the threshold, as they will correspond to face occlusions. By carrying out the inverse process and determining the

RGB data for the selected depth points, we obtain the RGB image pixels where the occlusion object is represented.

Once the RGB pixels are determined, the next step is to insert the object into a face image and generate the proposed synthetic facial occlusion. Since the size of the face detection box in the video captured with the RGB-D camera is different from the size of the face in the image which we want to occlude, it is necessary to first re-scale the object image to the dimensions of the non-occluded face image, and only then we superimpose the object in the original image.

Following establishment of the threshold distance that allows to detect occlusions from depth information, and given a set of non-occluded face images, it is possible to iterate the process exemplified in figure 3.4 using an occlusion from an occluded frame for each individual face image in a dataset, and therefore generate a new occluded dataset.

3.2 Head Pose Datasets

As mentioned before, the purpose of synthesizing occlusion in images is the application of that method to datasets that are commonly used in the training and/or benchmark for head pose estimation as to generate new occluded versions of them to be used in the implementation of this dissertation methodologies, which are based in neural networks. The datasets in question are the 300W-LP [16], BIWI [17] and AFLW2000 [18].

The 300W-LP dataset [16] is a synthetic augmented expansion of the 300W [53] dataset consisting of 61225 unconstrained generated face samples and respective vertically flipped versions of them for a total 122452 examples. It covers a large variation of identity, expression, illumination conditions, pose, occlusion and face size, and since it was initially developed for facial landmark localization, it provides annotations for both 2D 68 landmarks and the 2D projections of 3D landmarks from which it is possible to extract the pose of the head. Despite the original purpose, it is commonly used in the training process of head pose estimation works [7] [10].

The BIWI dataset [17] contains over 15000 images of 20 people recorded sitting in front of a Kinect turning their head around, in order to span the most varied poses possible, covering about ± 75 degrees yaw and ± 60 degrees pitch. It was developed specifically for head pose estimation and is one of the most commonly benchmarked datasets [7] [10] [42]. For each frame, it provides a depth image, the corresponding RGB image (both 640x480 pixels), and the ground truth pose annotation.

AFLW2000 [18] is a dataset that contains 2000 images of diverse head poses under challenging conditions (hard to detect by a face detector). It has been annotated with 68-point 3D facial landmarks from which pose can be extracted and is commonly used for both the evaluation of 3D facial landmark detection models and head pose estimation models.

We will use the occluded 300W-LP as the training dataset due to the larger number of samples, variety of poses, and the fact that it was used as training dataset in some of the state of the art works [7] [10] from which we will base our methodologies. Since the benchmarked datasets for head pose estimation in [47] are BIWI and AFLW2000, and due to their also varied pose span and challenging characteristics, we will use occluded and non-occluded versions of those datasets for testing.

Figure 3.5 displays examples of the data used in all methodologies, which includes synthetically occluded and original non-occluded images, as well as ground truth poses for each (here represented in images with respective head pose coordinate systems).

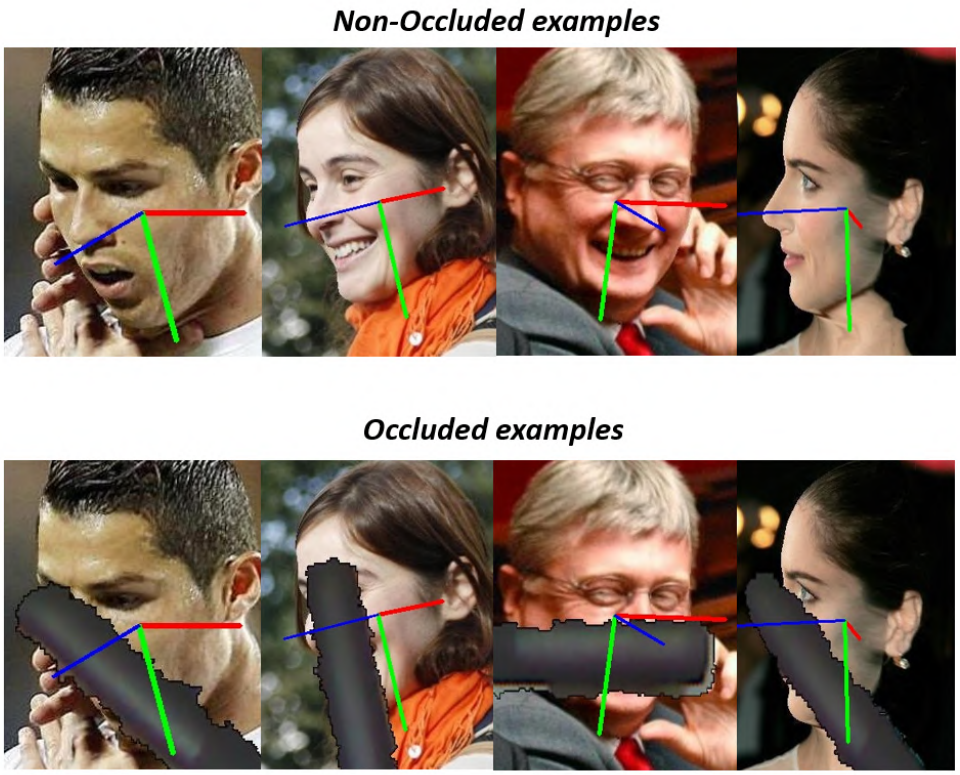


Figure 3.5: Examples of image data and ground truth pose in datasets.

4

Methodologies For Head Pose Estimation With Occlusions

Contents

4.1 End-to-end Multi-loss Approach With Latent Space Regression 29

4.2 Occluded Head Pose Estimation Through Face Reconstruction 32

4.3 Multi-Loss Autoencoder For Occluded Head Pose Estimation 36

4.1 End-to-end Multi-loss Approach With Latent Space Regression

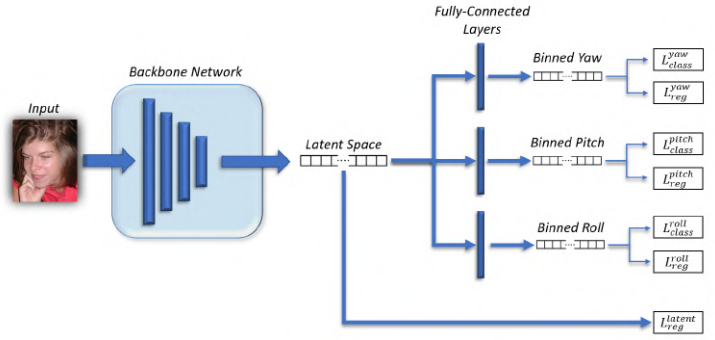


Figure 4.1: Multi-loss HPE framework with latent space regression.

This approach follows a similar framework to the work developed in [7]. Unlike landmark-to-pose methods, which require the estimation of 68 2D or 3D landmarks to perform face alignment and estimate the pose of the head, we use deep learning networks to train a model to predict pose directly from 2D RGB image intensities and avoid the extra landmark step, along with its complications (dependency on chosen head model and inability to estimate pose if landmark detection fails). The framework is as exemplified in figure 4.1: A 2D RGB image is input to any backbone network of choice, such as Resnet-50 and Alexnet mentioned in [7] or EfficientNet used in [10]. This backbone network is expanded with three extra fully-connected layers which will be used to output the predictions for each Euler angle (figure 4.2). A fully-connected layer connects every neuron in the input layer to every neuron in the output layer, which mathematically corresponds to a linear transformation function from $\mathbb{R}^m$ to $\mathbb{R}^n$, $y(x) = Wx + B$, where W is the weight matrix ($\mathbb{R}^{n\times m}$), B is the bias vector ($\mathbb{R}^{n\times 1}$), x is the input vector ($\mathbb{R}^{m\times 1}$) and y is the output vector ($\mathbb{R}^{n\times 1}$). Here, the output of the final layer in the backbone network, a multi-dimensional matrix that encloses the latent space representation, is flattened to unfold all its values into a vector which will be the input x to the fully-connected layer. The output y for each one of this layers will be a vector of logits, which are raw prediction scores (real numbers ranging from $[-\infty, +\infty]$) for the predicted angle belonging to a certain w degree bin. The size of these vectors depends on both the angle interval/span for each bin, and the full prediction range for the given Euler angle.

Figure 4.2 displays the input and output of each fully-connected layer. The flattened latent space vector has 2048 elements. We adopt the binned vector dimensions of [7] with 66 classification bins each with a width of $3°$ (e.g. [$0°$,$3°$], [$3°$,$6°$], ...) and covering a range from $-99°$ to $99°$ for all yaw, pitch and roll predictions. This range therefore covers extreme profile poses.

Henceforward, the output of each fully-connected layers is used in a multi-loss scheme that com-

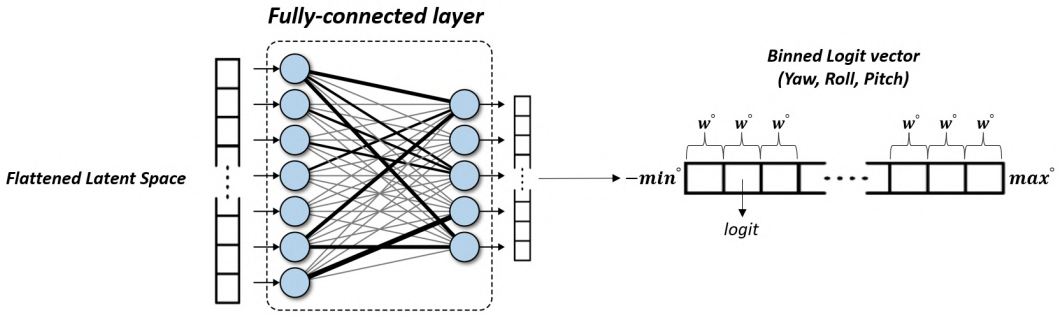


Figure 4.2: Euler angle prediction vector through fully-connected layer.

prises the combination of a classification component and a regression component to provide an overall loss for a given Euler angle. For the classification task, a softmax activation function plus a cross-entropy loss (also known as categorical cross-entropy loss or softmax loss) is applied to the n-dimensional vector output of the fully-connected layer. The softmax function turns logits into probabilities by computing the exponents of each bin output and normalizing it by the sum of those exponents so that all probabilities in the activated vector add up to 1:

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}} \tag{4.1}$$

Afterwards the cross-entropy loss result is computed by equation 4.2, where t_i and $S(y_i)$ are the ground-truth (0 or 1) and the activation result of the score for each of the C angle classes/bins, respectively.

$$L_{class} = CE = -\sum_{i}^{C} t_i log(S(y_i)) \tag{4.2}$$

In addition to the classification loss, the regression component is introduced to determine and regress the error between the predicted angle and the ground truth in degrees. It is possible to determine the predicted angle in degrees by using the bin probabilities obtained from softmax activation to calculate the expectation of the given angle:

$$\theta_{pred} = w \sum_{i=1}^{N} p_i (i - \frac{1+N}{2}) \tag{4.3}$$

Where θ_{pred} is the predicted angle in degrees, w is the width of the bin in degrees (3, in our case), N is the number of bins for classification (66, in our case), and p_i is the probability of the angle belonging

to bin i. The offset $\frac{1+N}{2}$ shifts the bin indices to the respective bin centres, as mentioned in [10]. The loss used for the regression component is the MSE between the predicted angle θ_{pred} and the ground truth angle θ_{gt}, for N predictions.

$$L_{reg} = MSE = \frac{1}{N}\sum_{i=1}^{N}(\theta_{pred} - \theta_{gt})^2 \tag{4.4}$$

The classification component aims to help the model predict the vicinity of each pose angle by classifying it in a angle interval bin and the regression error is introduced to aid the model in achieving fine-grained angle predictions. While adopting this multi-loss system for the estimation of yaw, roll and pitch, we introduce an extra regression loss for the latent space of the backbone network, specifically added to aid the model deal with the occlusion challenge. Convolutional neural networks reduce the dimensionality of an input across each layer, compressing its information and learning distinct features at each layer, given that deeper layers in the network encode higher-level, more general features. The latent space is the abstract multi-dimensional space that contains the highest-level feature values. This values encode the most relevant inner representation of the observed input data. This concept is at the core of deep learning since it is, essentially, how the model interprets real word images provided and finds the patterns and features that better and most generally characterize them. To understand how to use the latent space to help deal with the occlusion challenge, let's introduce an overview of the procedure carried out in this end-to-end multi-loss approach with latent space regression.

Firstly, we need to train or use a pre-trained model for head pose estimation in non-occluded images, with the same framework as figure 4.1 apart from the latent space loss. Then, using this model, we perform inference for each non-occluded image and store the flattened output of the final layer in the backbone network, which corresponds to the model's latent space representation for that given image. Finally, we use the occluded dataset and train the full framework of figure 4.1, where L^a_{class} and L^a_{reg} are the cross-entropy classification loss and MSE regression loss for Euler angle a (yaw, pitch or roll), and L^{latent}_{reg} is the MSE regression loss for the latent space.

The classification and regression loss for the Euler angles are combined with a parameter α that allows to vary the weight of each regression loss. Overall, four losses are used to train the model:

$$\begin{aligned} L_{yaw} &= L^{yaw}_{class}(y,\hat{y}) + \alpha\, L^{yaw}_{reg}(y,\hat{y}) \\ L_{pitch} &= L^{pitch}_{class}(y,\hat{y}) + \alpha\, L^{pitch}_{reg}(y,\hat{y}) \\ L_{roll} &= L^{roll}_{class}(y,\hat{y}) + \alpha\, L^{roll}_{reg}(y,\hat{y}) \\ L_{latent} &= L^{latent}_{reg}(y,\hat{y}) \end{aligned} \tag{4.5}$$

where y is the predicted value and $\hat{y}$ is the ground truth for the respective loss. The ground truth for

all Euler angles is provided in the training dataset, and the ground truth for the latent space regression corresponds to the stored latent space output of the inference for the respective non-occluded dataset images (figure 4.3).

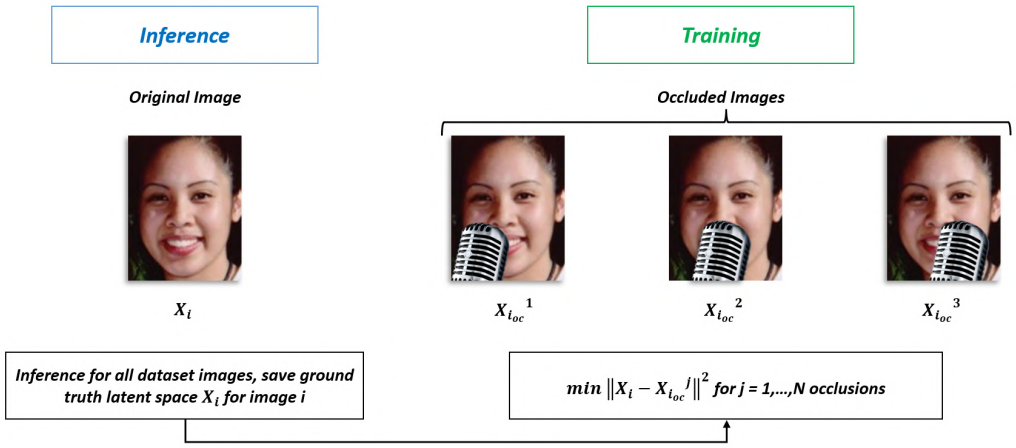


Figure 4.3: Ground truth latent space and error minimization.

By introducing this latent loss we ensure that the latent space of the re-trained model for occlusions does not deviate much from the one produced by the previous training for non-occluded images, which leads to good estimation results. This allows the model to better identify and distinguish poses regardless of the varying occlusions that may occur, while at the same time maintaining the correct estimation for images without occlusion. The total loss is given by function that combines the four losses:

$$L_{total} = (1 - \beta)(L_{yaw} + L_{pitch} + L_{roll}) + \beta L_{latent} \tag{4.6}$$

The influence of both the latent space regression loss and the angle losses is balanced by a weight parameter β.

4.2 Occluded Head Pose Estimation Through Face Reconstruction

The backbone network from the previous approach is a feedforward neural network that compresses the input to produce a more compact, low dimensional and higher-level code known as the latent space representation. Feedforward networks such as these are called encoders, since they encode the input. They are frequently part of a larger kind of unsupervised neural networks named autoencoders, which

learn to copy its input to its output. They are composed of a encoder that maps the input into the code and a decoder that maps the code into a reconstruction of the input. Appendix A conveys a more detailed description of this framework. Instead of directly adapting the head pose estimation model to deal with occlusions as we did previously, we benefit from the reconstruction capabilities of an autoencoder and train it to output reconstructed occlusion-free faces from facial occluded inputs. The idea is that the outputs of the autoencoder with removed occlusions and reconstructed faces are then input to a trained head pose estimation network. We use the HPE multi-loss pipeline defined in [7]. The framework is as exemplified in figure 4.4.

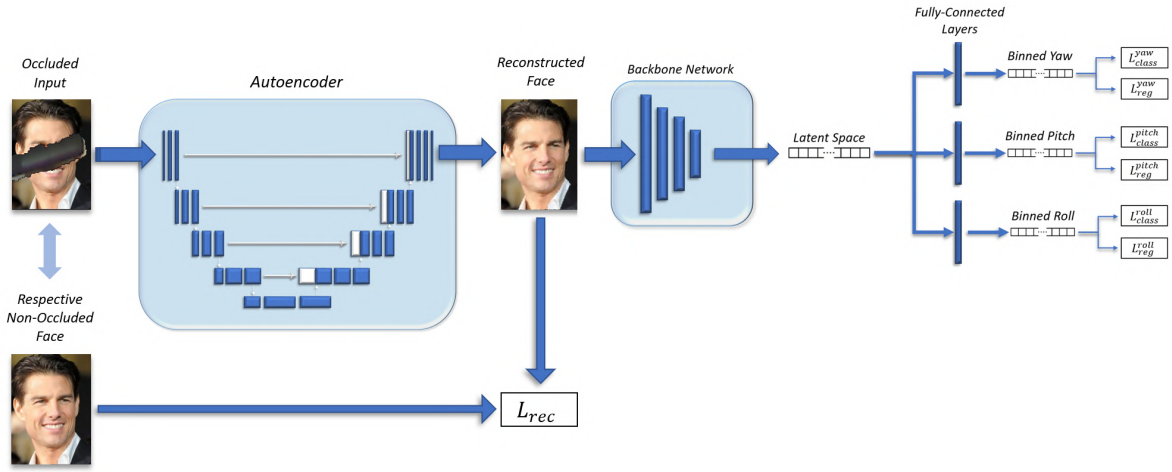


Figure 4.4: Framework for occluded head pose estimation through face reconstruction.

Standard autoencoders suffer from a loss of feature information when the input dimensions are reduced in the encoder, which affects the quality of facial reconstructions. For that we reason, we base ourselves in the approach carried out by the authors of *mask2face* [54] and employ the U-Net [9] architecture as the chosen autoencoder structure. The advantage is that U-Net adds connections between layers in the encoder and layers of identical dimension in the decoder (skip connections) to pass information directly from the encoder to the decoder. This improvement accelerates the learning process of the network and helps to reduce the loss of information and achieve more fine-grained face reconstructions to input to the head pose estimation network. A simplified representation of the U-Net architecture (further described in appendix A) is presented in figure 4.5.

4.2.1 Reconstruction Loss

The autoencoder receives occluded face inputs and is trained to minimize the reconstruction error between the predicted output and the ground truth face without occlusion. Since the inputs correspond to face images that we're synthetically occluded, we utilize the respective original non-occluded face

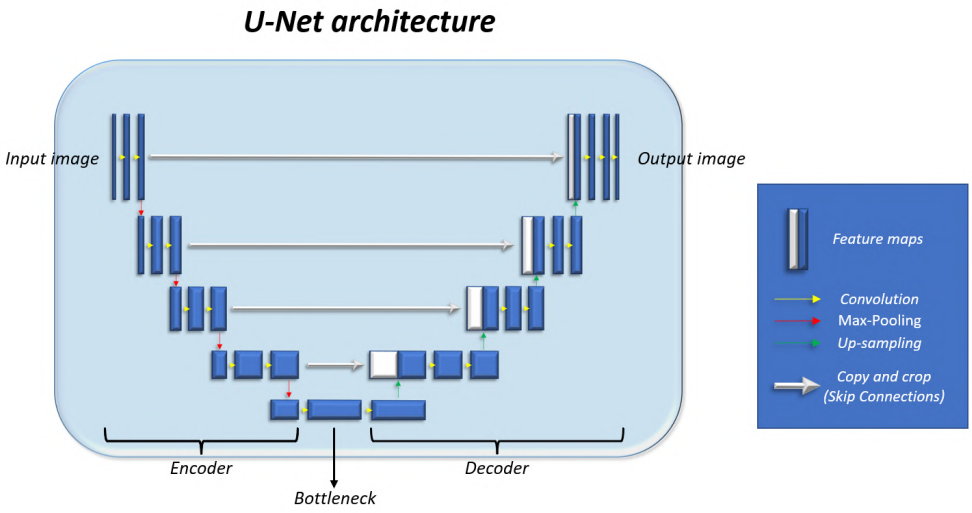


Figure 4.5: Simplified standard U-Net architecture.

images as ground truth in the reconstruction loss function, L_{rec}. This training procedure is exemplified in figure 4.6.

We define the reconstruction loss function as the combination of two losses: The l_1 the SSIM losses. The l_1 loss or Mean Absolute Error (MAE) applied to images consists in the mean of per-pixel absolute difference between the intensities in the image generated by the autoencoder and the ground truth non-occluded image. For each image this loss corresponds to:

$$L_{l_1}(I_{rec}, I_{gt}) = \frac{1}{N} \|I_{rec} - I_{gt}\|_1 = \frac{1}{N} \sum_{p \in I_{rec}, I_{gt}} |I_{rec}(p) - I_{gt}(p)| \tag{4.7}$$

where p is a pixel, N the number of pixels in the images and I_{rec},I_{gt} are the intensity values of that pixel in the reconstructed image and in the ground truth, respectively. As mentioned in [55], this loss function leads to better performance when restoring images that the l_2 (MSE) loss (similar, but based on pixel intensity squared differences), since it does not over-penalize larger errors and is therefore less prone to over-smooth the output image.

The Structural Similarity Index Measure (SSIM) [56] is a metric used for the measurement of the similarity between two images. This index is calculated on local NxN windows that move across the images, since localized quality measurement provides more information about the quality degradation of a sample image in comparison to a reference image. Unlike most metrics that quantify the difference between the intensity of corresponding pixels between sample and reference images (such as MAE and MSE), SSIM extracts and compares three different measures between images: the luminance (l) , the contrast (c) and the structure (s).

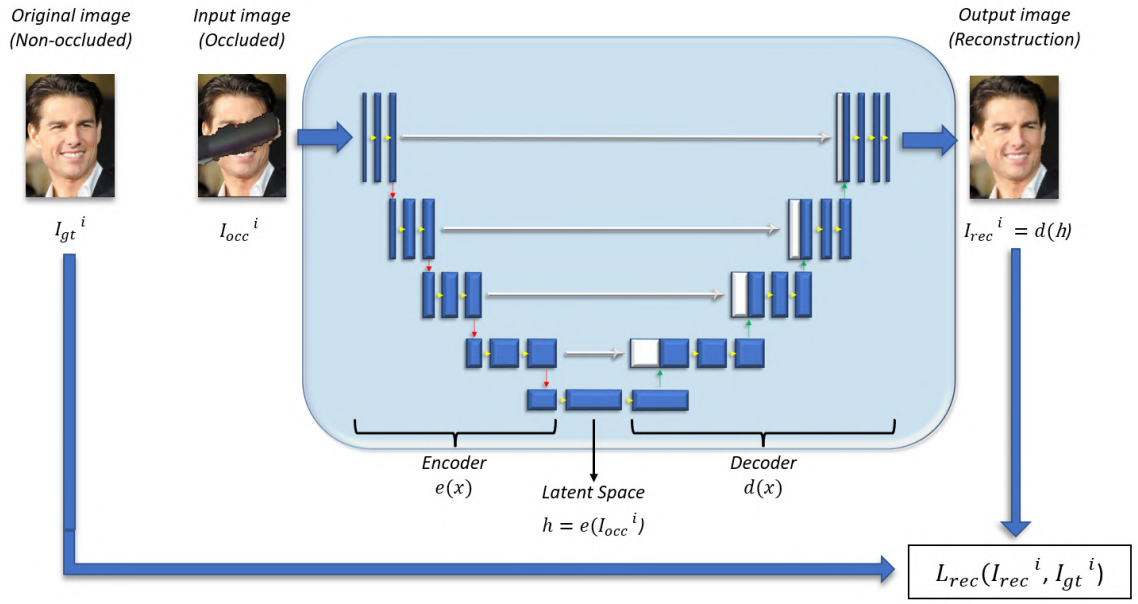


Figure 4.6: Training framework for autoencoder.

$$l(x,y) = \frac{2u_x u_y + c_1}{u_x^2 + u_y^2 + c_1} \quad c(x,y) = \frac{2\sigma_x \sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3} \tag{4.8}$$

where x, y are windows of both images; (u_x, u_y) is the average of (x, y); (σ_x, σ_y) is the standard deviation of (x, y) ;(σ_x^2, σ_y^2) is the variance of (x, y); σ_{xy} is the covariance of x and y; c_1 and c_2 are constants included to avoid instability when $\mu_x^2 + \mu_y^2$ and $\sigma_x^2 + \sigma_y^2$ are close to zero and $c_3 = \frac{c_2}{2}$.

The SSIM measure is obtained by combining the three measures of equation 4.8:

$$SSIM(x,y) = [l(x,y)]^{\mu} \cdot [c(x,y)]^{\phi} \cdot [s(x,y)]^{\psi} \quad SSIM(x,y) = \frac{(2u_x u_y + c_1)(2\sigma_{xy} + c_2)}{(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.9}$$

with $\mu > 0$, $\phi > 0$ and $\psi > 0$ as parameters used to adjust the relative importance of each component in the index are commonly set to 1.

Since the SSIM outputs a value between 0 and 1, the loss function for this metric applied to each image will actually be:

$$L_{SSIM}(I_{rec}, I_{gt}) = \frac{1}{N} \sum_{x,y \in I_{rec}, I_{gt}} 1 - SSIM(I_{rec}(x), I_{gt}(y)) \tag{4.10}$$

where x, y are windows in the image predicted by the autoencoder I_{rec} and in the ground truth image I_{gt} and N is the number of windows. By comparing the three different measures, SSIM becomes more capable of identifying the differences between the structural information of sample and reference images,

which allows to better preserve the contrast and edges from the reference image. However, as denoted in [55], it preserves the brightness and colors worse than the l_1 loss, since this loss weights errors equally regardless of the local structure. For that reason, we combine both losses for the image reconstruction loss that we implement in the autoencoder:

$$L_{rec}(I_{rec}, I_{gt}) = L_{l_1}(I_{rec}, I_{gt}) + \gamma L_{SSIM}(I_{rec}, I_{gt}) \tag{4.11}$$

where γ is the parameter that regulates the weight of the loss function for the SSIM metric.

4.3 Multi-Loss Autoencoder For Occluded Head Pose Estimation

The pipeline explained in the previous section included two different neural networks, the autoencoder and the head pose estimation network. Despite the fact that this allows for more flexibility in the choice of the network that predicts the pose, since its input is a non-occluded (reconstructed) face, it also means that the entire process is computationally heavier and slower than using a single network.

Having that in mind, this section discusses a different approach that combines the face reconstruction task with the estimation of the pose in a single neural network. The pipeline of this procedure is illustrated in figure 4.7.

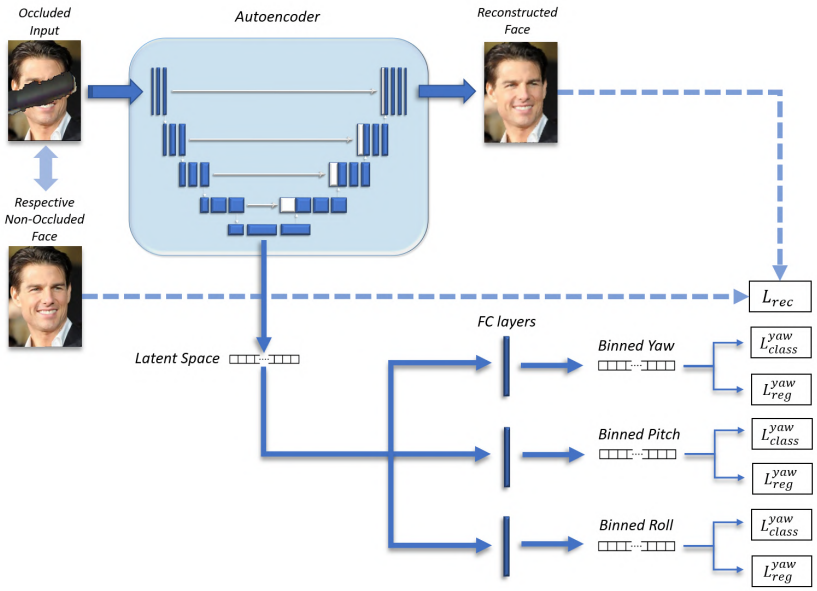


Figure 4.7: Multi-Loss autoencoder high-level pipeline.

As mentioned before, the neural network involved in the task of estimating the pose of the head is, by nature, an encoder. It receives an input image and maps it to a compressed lower-dimensional representation rich in feature information, the latent space, which then is shared by three different branches

for the estimation of each of the Euler angles. This means that we can add a decoder to the network and convert the network's structure to that of an autoencoder. The encoder is combined with the fully-connected layers to estimate the pose and combined with the decoder to reconstruct the face. This way, we can merge and adapt both tasks instead of having a different neural network for each. We use the U-Net architecture for the designed autoencoder in this approach as well. The details of this architecture (number of layers, dimension) depend on the encoder network used for head pose estimation. We base ourselves in the ResNet-Unet architecure described by the authors of [57] and use ResNet-50 [39] as the encoder for pose estimation.

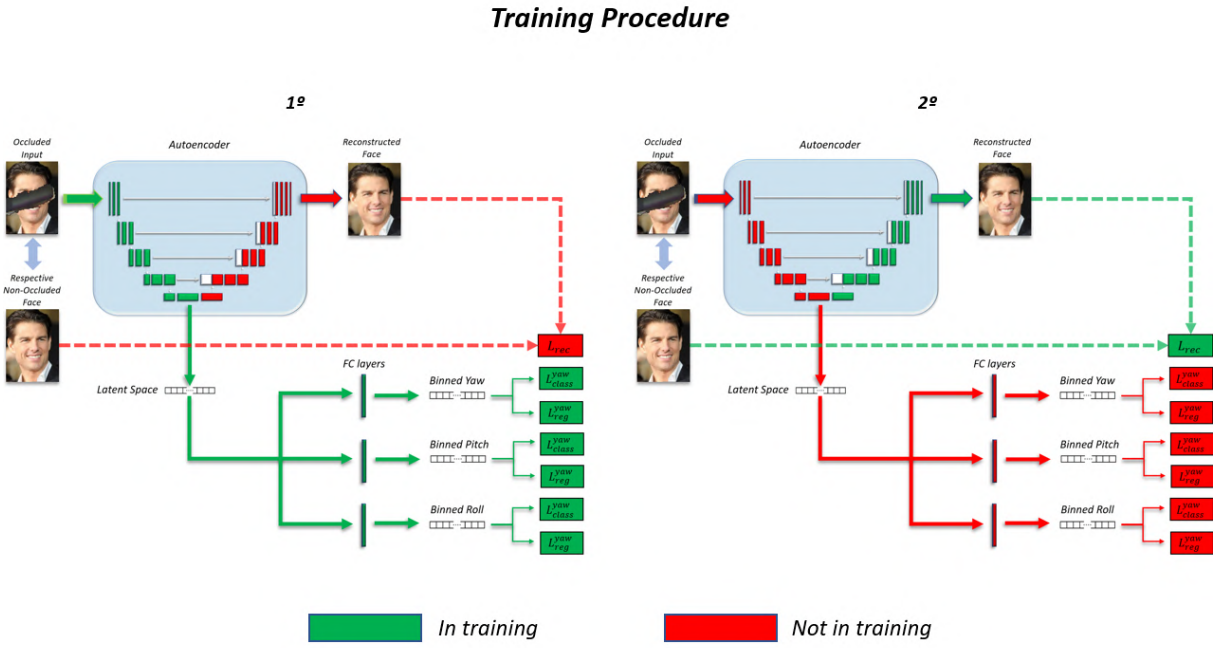
The training plan (figure 4.8) for this procedure is as follows: In the first stage the encoder and fully connected layers are trained for the estimation of all three Euler angles. This training involves the minimization of the regression and classification losses for each angle, as in section 4.1. In the second stage, the decoder is trained for the reconstruction of the face without the occlusion. This training involves the minimization of the reconstruction loss between reconstructed and non-occluded ground-truth images, as in 4.2. In the third and last stage, the the entire autoencoder and fully-connected layers are trained. This training involves the minimization of all losses involved in the first and second stage.

The purpose of the first and second stages is to provide good initialization for the entire framework in the third stage and therefore aid the convergence of the learning model. In the first stage the encoder learns the latent space representation for pose estimation. In the second stage the decoder learns to map this compressed data to generate reconstructed faces without occlusion. The third stage stems from both and combines them so that the pose estimation is adapted to the reconstruction of occluded faces. The losses involved in the training stages of the autoencoder are:

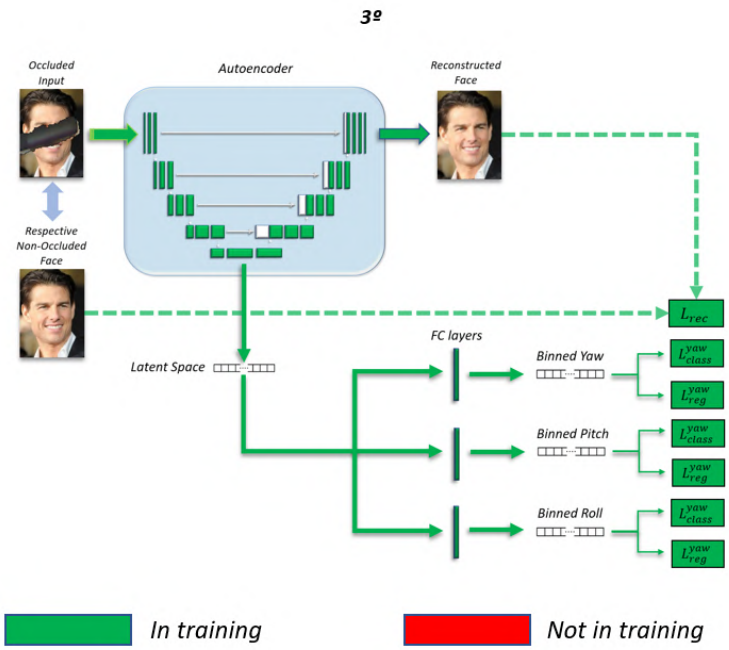
$$\begin{aligned} L_{yaw} &= L_{class}^{yaw}(y, \hat{y}) + \alpha\, L_{reg}^{yaw}(y, \hat{y}) \\ L_{pitch} &= L_{class}^{pitch}(y, \hat{y}) + \alpha\, L_{reg}^{pitch}(y, \hat{y}) \\ L_{roll} &= L_{class}^{roll}(y, \hat{y}) + \alpha\, L_{reg}^{roll}(y, \hat{y}) \\ L_{rec} &= L_{l_1}(I_{rec}, I_{gt}) + \gamma L_{SSIM}(I_{rec}, I_{gt}) \end{aligned} \tag{4.12}$$

The 1st stage involves L_{yaw}, L_{pitch}, L_{roll} (head pose estimation training), the 2nd stage involves L_{rec} (face reconstruction), and the 3rd stage (combined training) requires the minimization of all losses with the ρ parameter to define the weight of angle components:

$$L_{total} = L_{rec} + \rho(L_{yaw} + L_{pitch} + L_{roll}) \tag{4.13}$$



(a) 1st and 2nd stages



(b) 3rd stage

Figure 4.8: Training procedure stages for the pipeline described in figure 4.7.

5

Results and Discussion

Contents

5.1 Results 41
5.2 Method Results Comparison and Discussion 48

5.1 Results

5.1.1 Multi-loss Head Pose Estimation With Latent Space Regression

We evaluate this method on both the original and synthetically occluded versions of the BIWI and AFLW2000 datasets. ResNet-50, a 50 layers deep convolutional neural network with over 23 million trainable parameters, is used as the backbone network for this method. We choose ResNet-50 since it produces the HPE best results in the work developed in [7], from which this method is based on. All networks in this section are trained for 25 epochs and their parameters are initialized with a pre-trained model for 300W-LP non-occluded images provided by the authors of [7]. To optimize the parameters we use the Adam [58] optimization algorithm with a learning rate of 10^{-5}.

We train the framework defined in section 4.1 using synthetically occluded versions of 300W-LP. Since the images in this dataset include large backgrounds and elements external to the faces, we extract information from the provided 2D face landmark annotations to define a face bounding box and adjust the margins of the bounding box to capture the entire face/head. This procedure is described in appendix A. We apply the same practise to the images of AFLW2000. For BIWI, we use the face detector provided by Google's Mediapipe [59], an open source cross-platform framework. Both for training and testing, the images of all datasets are cropped to the pre-defined input dimension of the ResNet-50 network, 224x224 pixels, and the mean and standard deviation of ImageNet [60] is used to normalize the data. The annotations for ground truth pose angles are converted from radians to degrees in all datasets. We use 66 3° bins for classification, within a range from -99° to 99° for each one of the Euler angles. There are 31 images of the AFLW2000 dataset that are not used in testing since their pose angles surpass this range.

5.1.1.A Angle Regression Weight Study

In order to extrapolate the best regression weight for the estimation of each of the Euler angles, we train the framework without the latent space regression loss and with 5 different α parameter values (equation 4.5). The head pose estimation MAE results in synthetically occluded and non-occluded datasets are listed in tables 5.1 and 5.2. We can observe that, generally, $\alpha = 2$ produces the smallest average MAE errors. In particular for occluded images, the lowest MAE errors across all datasets correspond to the networks trained with that parameter value. Despite the fact that the results do not show substantial differences, we can also observe that the largest errors tend to occur for $\alpha = 1$. Since this α value corresponds to an equilibrium between the regression and classification losses, this result highlights the importance of correctly distributing their weight. For the following tests, we use $\alpha = 2$ as the weight for the head pose multi-loss framework.

BIWI	Occluded Images				Non-Occluded Images				
Reg. Weight	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
$\alpha = 0.5$	5.250	6.529	4.255	5.345	4.259	4.704	3.580	4.181	4.763
$\alpha = 1$	5.533	6.916	4.036	5.495	4.765	4.493	3.956	4.405	4.950
$\alpha = 2$	5.110	6.832	3.629	**5.190**	4.242	4.041	3.845	4.043	**4.617**
$\alpha = 5$	5.477	6.542	4.304	5.441	4.474	4.043	3.494	**4.004**	4.723
$\alpha = 10$	5.218	7.565	4.344	5.709	4.196	4.503	3.628	4.109	4.909

Table 5.1: Head pose estimation MAE$^{\circ}$ tests with BIWI for different angle regression weights (α).

AFLW2000	Occluded Images				Non-Occluded Images				
Reg. Weight	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
$\alpha = 0.5$	6.227	8.271	5.713	6.737	5.281	6.544	4.497	5.441	6.089
$\alpha = 1$	6.411	8.713	6.017	7.047	5.675	6.868	4.760	5.768	6.4075
$\alpha = 2$	5.672	8.101	5.783	**6.519**	4.886	6.636	4.643	**5.389**	**5.954**
$\alpha = 5$	6.156	8.279	5.841	6.759	5.403	6.413	4.546	5.454	6.1065
$\alpha = 10$	5.4044	8.407	5.923	6.578	4.986	6.695	4.696	5.459	6.0185

Table 5.2: Head pose estimation MAE$^{\circ}$ tests with AFLW2000 for different angle regression weights (α).

5.1.1.B Latent Space Regression Weight Study

Having verified the proper α weight parameter for the estimation of the Euler angles, we proceeded to analyze the influence of the latent space regression loss. We trained 4 different HPE networks, one for each different β (equation 4.6), the parameter that balances the weight of this loss and of Euler angle losses, and compared the results with *Hopenet* [7]. All networks were trained for 25 epochs. We use the latent space produced by the pre-trained model in non-occluded inference as ground truth. The datasets for training and testing include 25 different regions of occlusion. Tables 5.3 and 5.4 display the results for each dataset. ML LSR stands for multi-loss with latent space regression.

We observe that when the latent space loss is not used ($\beta = 0$), despite substantially decreasing the error for occluded images when compared to *Hopenet*, the head pose estimation worsens for non-occluded images. This is more evident in the BIWI dataset, where the average MAE for non-occluded images increases by nearly 1°. We can also observe that as the the influence of the latent space regression loss (β parameter) increases over the influence of Euler angle losses, the average MAE becomes lower for both occluded images and non-occluded images. In fact, for the AFLW the non-occluded estimation results are better when $\beta \geq 0.5$ than the ones of *Hopenet* which was trained for non-occluded images. This confirms that introducing this loss helps not only to achieve improved generalization for occlusions, but also to avoid detouring from accurate non-occluded pose estimation. When only the latent space regression loss is used ($\beta = 1$), we obtain the best results for pose estimation in BIWI and the pitch and roll MAE decreases in both occluded and non-occluded images. This reveals that

the pre-trained encoder (which was optimized with Euler angle losses in non-occluded images) already produces a good latent space representation for angle estimation and therefore the model is better optimized when exclusively trained to maintain that representation for occluded images.

BIWI	Occluded Images				Non-Occluded Images				
Methods	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	6.725	8.616	7.338	7.560	4.375	3.559	3.348	3.761	5.661
ML LSR ($\beta = 0$)	5.990	7.778	4.346	6.038	4.940	4.873	3.911	4.575	5.307
ML LSR ($\beta = 0.5$)	5.797	7.394	4.537	5.910	4.413	4.910	3.556	4.293	5.102
ML LSR ($\beta = 0.990$)	5.798	6.881	4.572	5.750	4.204	4.343	3.750	4.099	4.925
ML LSR ($\beta = 0.999$)	5.174	6.622	4.117	5.304	4.297	4.186	3.617	4.033	4.669
ML LSR ($\beta = 1$)	5.429	4.823	3.467	**4.573**	4.291	3.086	3.179	**3.519**	**4.046**

Table 5.3: Head pose estimation MAE$^\circ$ tests with BIWI for different latent space regression weights (β).

AFLW2000	Occluded Images				Non-Occluded Images				
Methods	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	12.438	10.277	8.586	10.434	4.965	5.250	3.956	4.724	7.579
ML LSR ($\beta = 0$)	5.057	7.120	4.961	5.713	4.114	6.002	4.061	4.726	5.220
ML LSR ($\beta = 0.5$)	4.891	6.424	4.918	5.411	3.855	5.447	3.947	4.416	4.914
ML LSR ($\beta = 0.990$)	4.714	6.360	4.906	5.327	3.709	5.517	4.083	4.436	4.882
ML LSR ($\beta = 0.999$)	4.741	6.254	4.765	**5.253**	3.813	5.420	4.003	**4.412**	**4.833**
ML LSR ($\beta = 1$)	5.117	6.075	4.590	5.261	4.258	5.272	3.888	4.473	4.867

Table 5.4: Head pose estimation MAE$^\circ$ tests with AFLW2000 for different latent space regression weights (β).

Re-optimizing the model along with the fully-connected layers and Euler angle losses deviates the encoder from the latent space representation original pre-trained model. However, we verify that the yaw MAE increases when compared to the model with $\beta = 0.999$. Figure 5.1 shows the histograms of Euler angle bin frequency for both the training and testing datasets. From this figure we observe that the fact that the yaw estimation is better when $\beta = 0.999$ is quite relevant, since the yaw is the angle that covers the biggest range of values and therefore the most important angle in head pose estimation. These results also show that the introduction of the Euler angle losses can be beneficial in occlusion training to improve the estimation of the angles with the biggest range of values and the biggest frequency of extreme values in the training dataset. We also observe that the reason why BIWI results for $\beta = 1$ are greatly improved is that it is the dataset with the largest range of varied pitch and roll poses, while being the one with the smallest range of yaw values.

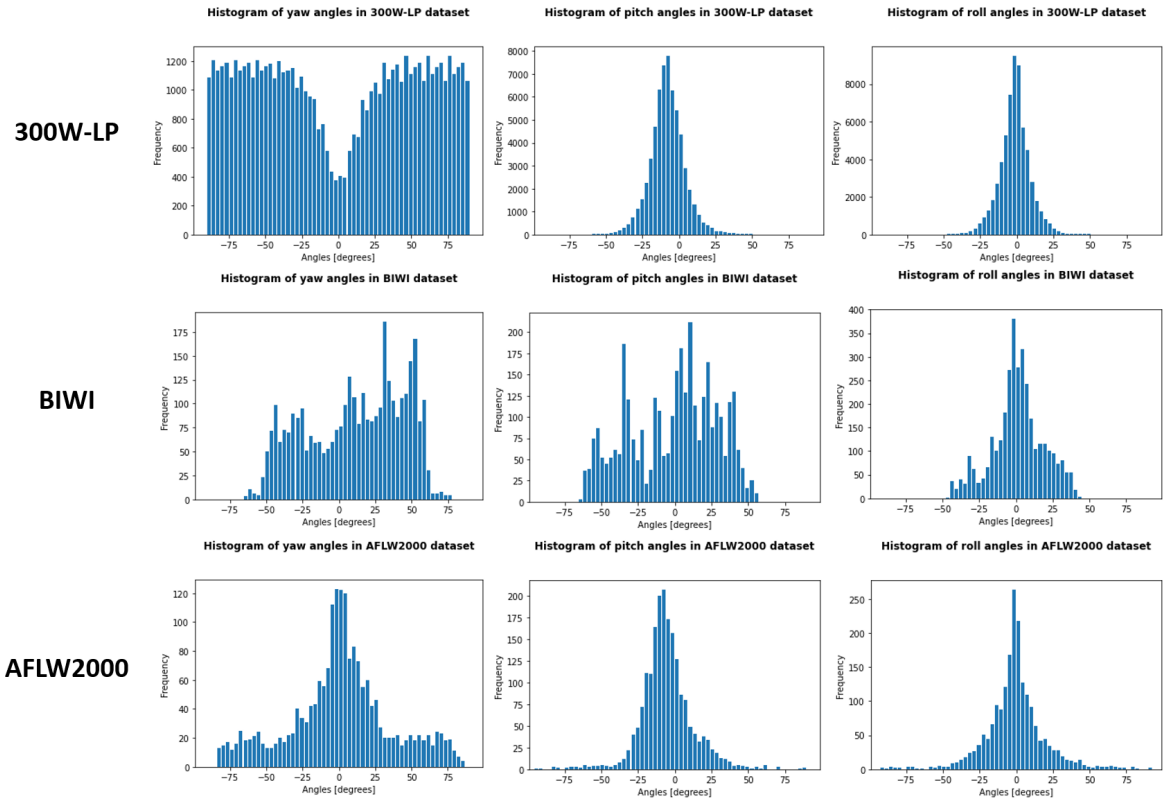


Figure 5.1: Histograms of Euler angle bin frequency for each dataset.

5.1.2 Occluded Head Pose Estimation Through Face Reconstruction

Within this section we present a study regarding the use of an U-Net autoencoder for the purpose of removing occlusions and reconstructing faces whose pose is to be estimated. The training and testing datasets are the same we used in the previous approach. We train the autoencoder for 25 epochs and use a batch size of 116. During the first epoch of training the networks are trained to reconstruct the faces with non-occluded image inputs. This is done to provide better initialization for the following epochs, for which the model is trained with batches of 100 occluded images and 16 non-occluded images. The images are cropped as in the previous approach but they are not normalized. In order to enhance the robustness of the reconstruction, we apply image transformations during training to randomly adjust the brightness, contrast, saturation and hue during training. The learning rate is set to 10^{-5} and Adam is selected as the optimization algorithm.

We train 3 models with distinct weights for the SSIM loss in the reconstruction loss function defined in equation 4.7 (section 4.2.1). The first is setting $\gamma = 0$, which amounts to using only the l_1 loss. Afterwards we set $\gamma = 1$ and include the SSIM loss in training. Lastly we increase the weight of the SSIM loss in the reconstruction loss function to $\gamma = 2$.

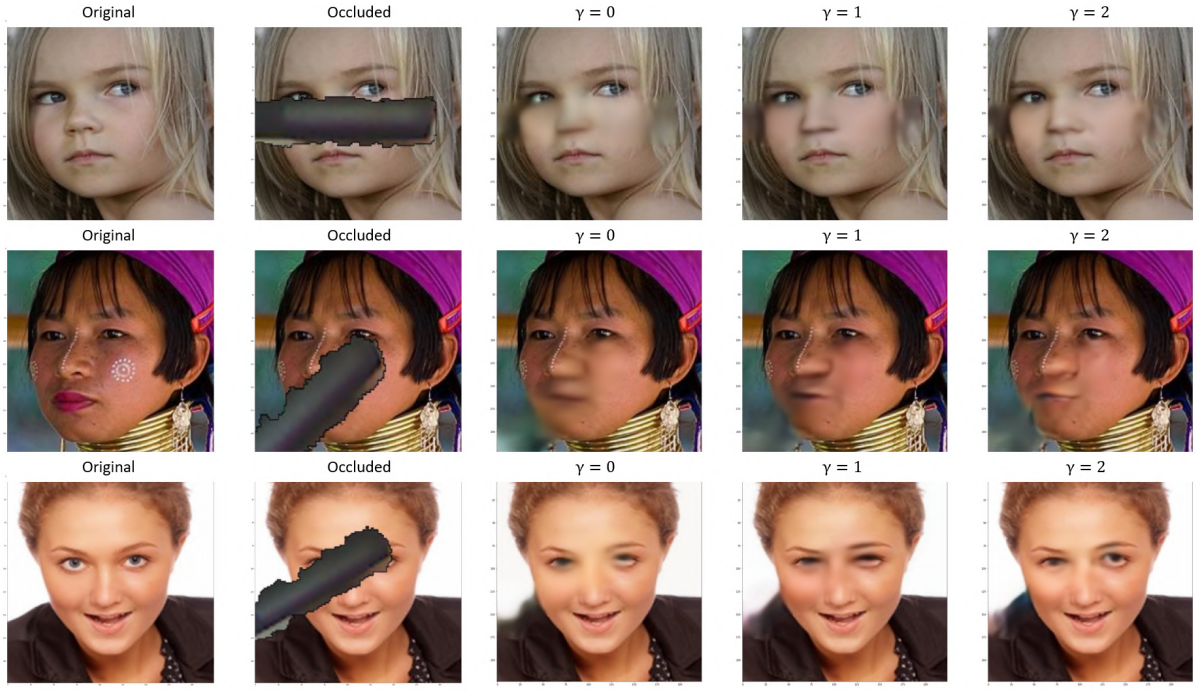


Figure 5.2: Reconstruction comparison for different loss weights.

Figure 5.2 displays examples of the different reconstructions generated by each model. The presented 300W-LP examples are not included in training. We observe that when only the l_1 loss is used ($\gamma = 0$), the reconstructed regions are blurrier and less detailed, with faded edges and lower contrast. The introduction of the SSIM metric loss improves this aspects and allows the model to better replicate the main features of a face (mouth, nose, eyes). As we increase the influence of this loss over the l_1 loss, we see even better results, with significant improvement over the first model. Lips become more defined, and complicated details such as the outline of nostrils in noses and the iris in each eye are now more visible in the reconstruction. While the l_1 loss helps to generate good results regarding the brightness and color intensities of the reconstructed area, the SSIM loss improves the structural details of faces and helps to produce more fine-grained results.

In figure 5.3 we can observe some examples of face reconstructions for each dataset. The quality of the reconstruction depends on the resolution of the original image. The models reconstruct the faces better in 300W-LP and AFLW2000, since these datasets contain higher resolution images. The original images in the BIWI dataset have lower resolution and the faces occupy a much smaller region than they do in the other datasets, which leads to worse reconstructions in this dataset.

We used the head pose estimation network and model of *Hopenet* to estimate the poses from the occluded/non-occluded images and from the images reconstructed by each autoencoder model. The results for each dataset are listed in table 5.5. We can observe that the estimation error decreases for all reconstruction models, in both occluded and non-occluded images. The improvements for the average

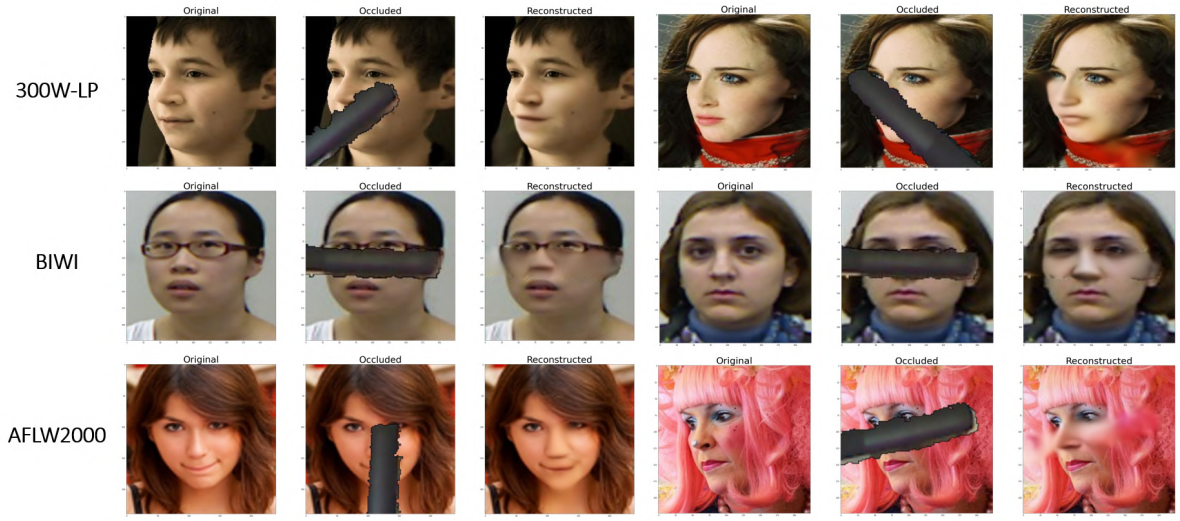


Figure 5.3: Examples of face reconstruction results for each dataset.

MAE in occluded image range from around $1.5°$ in BIWI to nearly $5°$ in AFLW2000. The fact that the models don't reduce the errors in the BIWI dataset as much they do in the remaining datasets is related to the lower resolution of the images in this dataset. It is also noticeable that the error decreases when the SSIM loss is used ($\gamma > 0$), with the best results corresponding to the highest weight used for this loss ($\gamma = 2$). For AFLW2000, the average MAE head pose estimation error for the reconstruction of occluded images with $\gamma = 2$ is less than $1°$ higher that the one produced with the original non-occluded images. This confirms that the extra structural fine-grained detail added by the SSIM loss helps to produce face reconstructions closer to the original ones and lead to the most substantial error reduction in head pose estimation.

	Occluded Images				Non-Occluded Images				
BIWI	yaw	pitch	roll	Avg. $MAE°$	yaw	pitch	roll	Avg. $MAE°$	Avg. $MAE°$
Hopenet	6.725	8.616	7.338	7.560	4.375	3.559	3.348	3.761	5.661
Rec. ($\gamma = 0$)	5.979	6.526	5.755	6.087	4.137	3.392	3.324	3.618	4.853
Rec. ($\gamma = 1$)	5.755	6.306	5.852	5.971	4.117	3.389	3.313	**3.606**	**4.789**
Rec. ($\gamma = 2$)	5.847	6.323	5.727	**5.966**	4.133	3.391	3.326	3.617	4.792
AFLW2000	yaw	pitch	roll	Avg. $MAE°$	yaw	pitch	roll	Avg. $MAE°$	Avg. $MAE°$
Hopenet	12.438	10.277	8.586	10.434	4.965	5.250	3.956	4.724	7.579
Rec.($\gamma = 0$)	5.738	6.392	4.962	5.697	4.978	5.250	3.959	4.729	5.335
Rec. ($\gamma = 1$)	5.674	6.196	4.812	5.560	4.676	5.353	3.965	4.665	5.113
Rec. ($\gamma = 2$)	5.546	6.138	4.690	**5.458**	4.675	5.345	3.956	**4.658**	**5.058**

Table 5.5: Head pose estimation MAE° results for reconstructed images in both datasets. The results of Hopenet correspond to the original inputs to the network.

5.1.3 Multi-Loss Autoencoder For Occluded Head Pose Estimation

In this set of experiments we train a single autoencoder U-Net network to remove face occlusions and estimate the pose. More specifically, we use the architecture of ResNet-UNet [57] with a ResNet-50 network as the down-sampling encoder. The data is pre-processed in the same way of the multi-loss latent space regression method, with cropped normalized images. Each training has 3 stages and 3 different models are trained. The first stage of this method implies exclusively training the encoder for head pose estimation, the second only the decoder for face reconstruction and the third the entire framework. In the first stage we use the pre-trained encoder provided by the authors of [7] for all 3 models. The distinction between the models arises from the differences in the second and third stages. For simplification, we name each model as ResUnet 1,2 and 3. In ResUnet 1, the reconstruction loss of the second stage is applied for images normalized with ImageNet's mean and standard deviation. We set the parameter $\gamma = 0$ (equation 4.12), since the SSIM loss does not work for negative values in color channels. For the third stage, these parameters are unchanged and the weight for pose estimation is $\rho = 0.1$ (equation 4.13). In both ResUnet 2 and 3 we reverse the normalization of the channels of the output image and apply the reconstruction loss for images without normalization. We set $\gamma = 2$ in the second and third stage. The difference between the two models is that $\rho = 0.1$ in ResUnet 2, while in ResUnet 3 we increase it to $\rho = 1$. The parameter for the weight for each angle regression loss is set to $\alpha = 2$ (equation 4.12) in all models. The networks were trained for 25 epochs in each stage, using Adam optimization, a learning rate of 10^{-5} and image batches of 116 images. In ResUnet 1 and 3, we use 100 occluded and 16 non-occluded images in each batch, and in ResUnet 2 we use only occluded images in each batch. Table 5.6 summarizes the different parameters for each model and 5.7 lists the results for all models in each dataset.

Method	α	γ	ρ	Normalized Images
ResUnet 1	2	0	0.1	Yes
ResUnet 2	2	2	0.1	No
ResUnet 3	2	2	1	No

Table 5.6: Parameters for each ResUnet trained model.

By comparing the results from ResUnet 1 and ResUnet 2 we observe that training the model with the reconstruction loss applied to images without normalization helps to produce better HPE results in occluded images. This may be due to the addition of the SSIM loss in ResUnet 2 which allows for better image reconstructions and leads the network to produce embeddings closer to those of an face image without occlusion. However, ResUnet 2 has worse results in regards to non-occluded pose estimation. This seems to be a consequence of using only occluded images in the second and third stages of training. ResUnet 3, which includes non-occluded images in training batches, improves non-occluded

	Occluded Images				Non-Occluded Images				
BIWI	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	6.725	8.616	7.338	7.560	4.375	3.559	3.348	**3.761**	5.661
ResUnet 1	6.874	6.462	4.994	6.110	4.586	4.193	3.654	4.144	5.127
ResUnet 2	6.192	7.218	4.355	5.922	4.798	4.944	3,712	4.845	5.384
ResUnet 3	5.377	6.318	4.564	**5.420**	4.380	4.175	3.669	4.075	**4.747**
AFLW2000	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	12.438	10.277	8.586	10.434	4.965	5.250	3.956	4.724	7.579
ResUnet 1	5.557	6.859	5.372	5.929	4.237	5.623	4.173	4.678	5.304
ResUnet 2	5.329	6.550	4.978	5.619	4.380	5.699	4.196	4.758	5.189
ResUnet 3	5.123	6.354	4.633	**5.370**	4.235	5.569	4.126	**4.643**	**5.001**

Table 5.7: Head pose estimation MAE$^{\circ}$ results for ResUnet.

results when compared to both previous models. Furthermore, setting the weight parameter for pose estimation losses to $\rho = 1$ leaded the network to produce the best results for all occluded datasets, despite using less occluded examples in batches than ResUnet 2. As a result of these improvements, we observe that ResUnet 3 has the lowest overall average MAE in both datasets.

5.2 Method Results Comparison and Discussion

BIWI	Occluded Images				Non-Occluded Images				
Methods	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	6.725	8.616	7.338	7.560	4.375	3.559	3.348	3.761	5.661
ML LSR ($\beta = 0.999$)	5.174	6.622	4.117	5.304	4.297	4.186	3.617	4.033	4.669
ML LSR ($\beta = 1$)	5.429	4.823	3.467	**4.573**	4.291	3.086	3.179	**3.519**	**4.046**
Rec. ($\gamma = 2$)	5.847	6.323	5.727	5.966	4.133	3.391	3.326	3.617	4.792
ResUnet 3	5.377	6.317	4.564	5.420	4.380	4.175	3.669	4.075	4.747

Table 5.8: Method comparison in BIWI.

AFLW2000	Occluded Images				Non-Occluded Images				
Methods	yaw	pitch	roll	Avg. MAE°	yaw	pitch	roll	Avg. MAE°	Avg. MAE°
Hopenet	12.438	10.277	8.586	10.434	4.965	5.250	3.956	4.724	7.579
ML LSR ($\beta = 0.999$)	4.741	6.254	4.765	**5.253**	3.813	5.420	4.003	**4.412**	**4.833**
ML LSR ($\beta = 1$)	5.117	6.075	4.590	5.261	4.258	5.272	3.888	4.473	4.867
Rec. ($\gamma = 2$)	5.674	6.138	4.690	5.458	4.675	5.345	3.956	4.658	5.058
ResUnet 3	5.123	6.354	4.633	5.370	4.235	5.569	4.126	4.643	5.001

Table 5.9: Method comparison in AFLW2000.

Tables 5.8 and 5.9 display the head pose estimation results in the respective dataset, for the best models of each method. We can observe that all of them substantially reduce the head pose estimation errors in occluded images when compared to *Hopenet*. The reductions in the average MAE for occluded images range from $2°$ to $5°$. Furthermore, they sustain accurate results for non-occluded images and in some cases even lower the MAE of Hopenet, despite being trained mostly or completely with occluded examples. This is important since it was also an objective of the developed methodologies to maintain the best accuracy possible in non-occluded images. By comparing the different procedures we observe that the reconstruction method produces the worst results in occluded images, specially in the BIWI dataset. This is mainly due to the sensitivity of the reconstruction autoencoder to the resolution of the input images. Images of lower resolution such as the ones from BIWI lead to worse reconstructed faces which in turn leads the separate pose estimator to produce worse estimations. However, this method improves the head pose estimation for non-occluded images in both datasets when compared to inputting the original image directly in *Hopenet*. The ResUnet method improves these results in BIWI and AFLW occluded examples, which corroborates that combining the reconstruction of faces with the estimation of the pose in one network leads to a model that generalizes better for occlusions. Ultimately, the latent space regression methodology produces the lowest occluded and global average MAE for both BIWI and AFLW2000 datasets. Moreover, this method allows to further decrease error in non-occluded images when compared to *Hopenet*. Both these factors make it the method that better fulfills the main purpose of achieving the best occluded head pose estimation, while preserving or improving on state of the art non-occluded head pose estimation. Using $\beta = 1$ achieves the best results on BIWI, the dataset with smallest range of yaw values, and largest range of pitch and roll values. On the other hand, using $\beta = 0.999$ achieves the best results on AFLW which has a bigger range in yaw values. Since in real-life applications the yaw Euler angle, which defines if a head is turned left, center or right, is the most varied and therefore most important angle in the pose estimation, we consider the model trained with $\beta = 0.999$ to be the best for such scenarios.

5.2.1 Testing Pose Estimation in the Feedbot Scenario

We also tested our best method (with $\beta = 0.999$) in the real world feeding scenario of Feedbot [13], a robotic arm for autonomous assisted feeding of people with upper-extremity disabilities, to find out how our model performs and compare it to *Hopenet*. Feedbot's framework combines the robotic arm with a camera to perceive the environment, so that it can localize and track the head and mouth of the user, as well as the end-effector (identified by a QR-code) of the arm where a spoon is located. With this cooperative scheme, it performs the feeding task by placing the end-effector in a comfortable position for the user to eat up the food. The context framework is illustrated in figure 5.4.

We used both our and *Hopenet's* models to estimate the head pose of the user from the frames

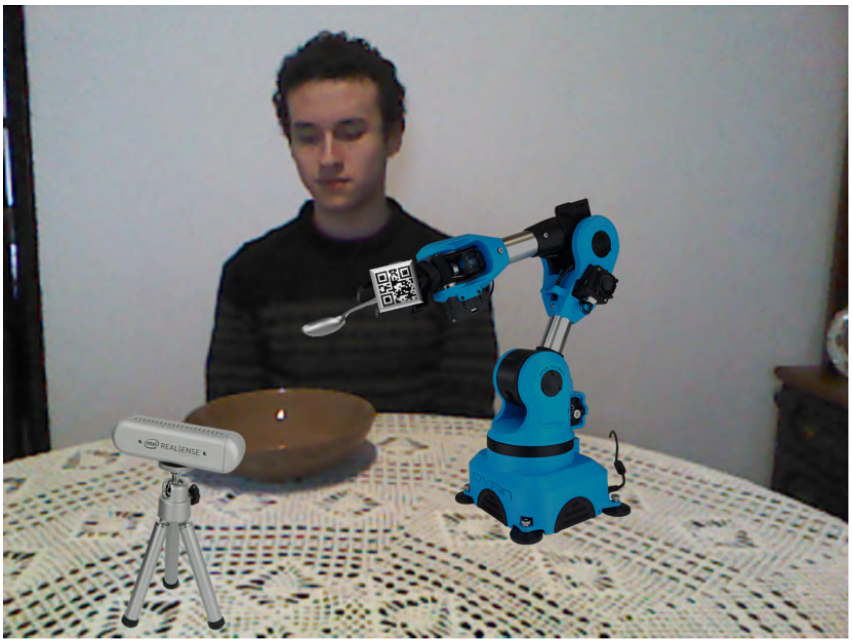


Figure 5.4: Feeding context of Feedbot.

captured by the camera that perceives the environment, as the robot performs the feeding task. Since we do not have the ground truth pose in this testing conditions, we carry out a qualitative analysis and evaluation in this section using the estimated pose coordinate systems. The frames in figure 5.5 show the perspective of that camera, where both the end-effector and the user are seen. Initially, the end-effector does not occlude the face of the user. We verify that for these non-occluded frames both *Hopenet* and our model outputs good pose estimations regarding each one of the Euler angles.

After the robot collects the food, it moves the end-effector towards the mouth and places it in the correct position for the user to eat. As the end-effector approaches the mouth, it partially occludes the face of the user. These occlusions can be seen in the frames from figure 5.6. From the head pose estimation results for those frames, we observe how *Hopenet* struggles with occlusions. The most affected pose angle is yaw (rotation around the green axis), as the model is seen to indicate head poses of opposite direction to that of which the head is turned, namely estimating the head to be rotated towards the left of the image when it is rotated towards the right. Our model, on the contrary, indicates head poses much closer to reality despite the existing occlusions. We verify that it greatly improved the yaw rotation when compared to *Hopenet*. In the far right image we also observe improvements in the roll rotation around the blue axis.

There are, however, occluded frames for which the head pose estimated by of our model is not entirely accurate. This occurs particularly when the end-effector covers several facial regions at once. Figure 5.7 displays some examples of such larger occlusions. In left and center examples, despite the user looking straight at the camera, the yaw rotation estimated by our model indicates that the head

Figure 5.5: Head pose estimation comparison between *Hopenet* and our model in frames that are not occluded by the end-effector. Pose coordinate systems represented in the center of the detected face.

is slightly tilted to the left of the images. For the example in the right, our model estimates yaw and roll angles correctly, but the pitch rotation (around the red axis) indicates the head is tilted down, when it is tilted upwards. Nonetheless, our model still performs better than *Hopenet* in these cases. While roll rotation is correctly estimated by both our and *Hopenet's* models in all examples, our model shows evident improvements in yaw and pitch angles.

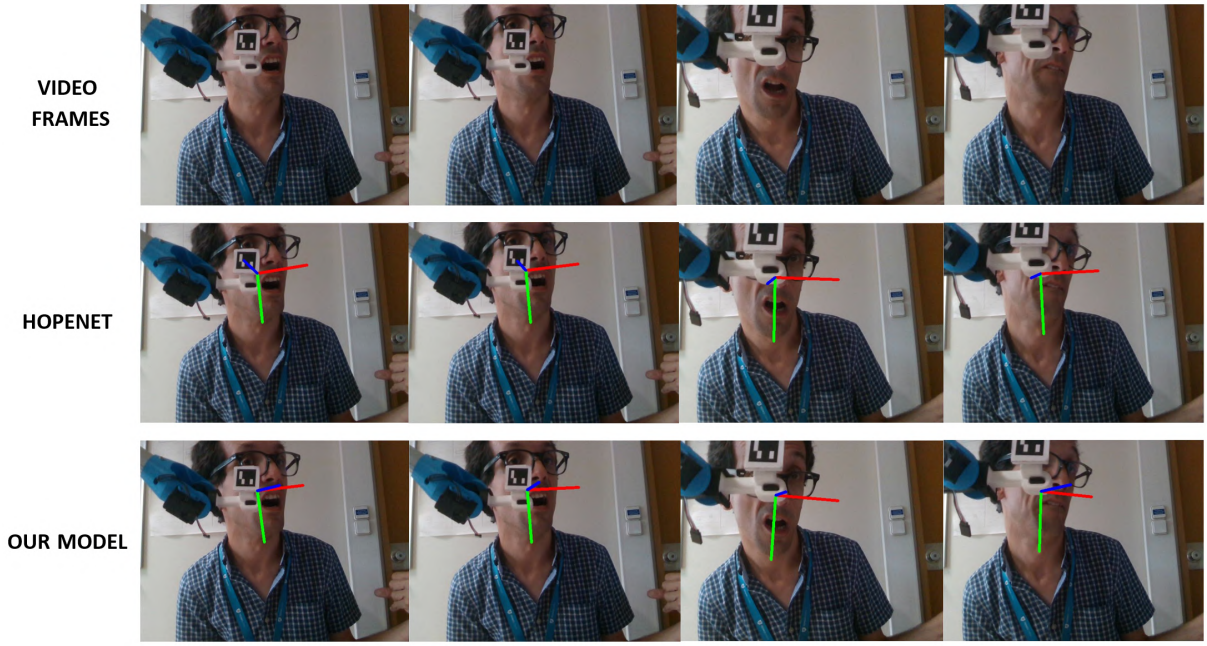


Figure 5.6: Head pose estimation comparison between *Hopenet* and our model in frames that are occluded by the end-effector. Pose coordinate systems represented in the center of the detected face.

Figure 5.7: Frames with larger occlusions for which our model is not as accurate, while still performing better than *Hopenet*. Pose coordinate systems represented in the center of the detected face.

6

Conclusions and Future Work

Contents

6.1 Conclusions 55

6.2 Method Limitations and Future Work 55

6.1 Conclusions

In this work, we developed three different learning-based methodologies to deal with the occlusion problem in head pose estimation. To be able to implement and test these approaches, we also introduced a procedure to generate synthetic occlusions in face images, using an RGB-D camera. We show how to segment occlusions based on depth data captured by the camera and how to inpaint the occlusion in any RGB face image. We applied this procedure to three datasets which are commonly used and benchmarked in head pose estimation problems, 300W-LP, BIWI and AFLW2000, and generated synthetically occluded versions for each one of them.

We conceived a new multi-loss head pose estimation framework combined with a latent space regression loss. We showed how introducing and increasing the influence of this loss improves the accuracy and generalization for occluded images and non-occluded images. We developed and experimented a different approach which resorts to the use of an autoencoder to reconstruct non-occluded faces from occluded images in order to input the reconstructions to a head pose estimation network. We demonstrate that combining the minimization of the absolute deviation of color intensities along with the minimization of perceived differences in the structural information of images leads to more fine-grained face reconstructions. We also verify that improving the quality of the face reconstructions contributes to achieve better estimation of head poses. Lastly, we combined head pose estimation with face reconstruction in a unique autoencoder which adapts both tasks through three training stages. We saw that adding the reconstruction loss metric to minimize structural differences between images, and increasing the weight of angle losses in the overall framework leads to better and more generalized pose estimation results. We also found that these results surpassed those of the previous approach, which used two different networks to solve tasks separately.

By performing ablation studies for each method we measured the influence of losses used in both face reconstruction tasks and head pose estimation frameworks and determined the best training configurations and models. We verified that all methodologies improved occluded head pose estimation and equaled or surpassed the estimation performance for the original non-occluded datasets.

We carried out qualitative tests using our best model in the real world application of the Feedbot, an autonomous assisting feeding robot. We confirmed that our model improved the head pose estimation for the occlusions of the robotic arm when compared to a state of the art estimation model, while achieving identical performance without occlusions.

6.2 Method Limitations and Future Work

Despite achieving good results, the developed methodologies have some limitations and further work could be done to improve them. The RGB-D Microsoft Kinect camera has low image resolution (640x480

pixels) and a minimum depth range of 0.8 meters. When recording occlusions for the synthetic occlusion generation procedure, this means that the faces and occlusion objects have to be farther than that distance and therefore the segmented occlusions occupy a small region in the image. When scaling occlusions to dataset face images this leads to less natural synthetic occlusions. An RGB-D sensor of higher resolution would improve this aspect.

ResNet-50, the encoder used in pose estimation frameworks, is a large network with over 23 million parameters and is therefore slower to train and requires more GPU power. A network such as EfficientNet [45] could be used to improve this, since it has only 11 million parameters and achieves superior accuracy on the ImageNet dataset, for which the pose estimation model is pre-trained.

The performance of our reconstruction models depends on the resolution of the detected face. An interesting improvement would be to explore the recent progress in face generation capabilities of a Generative Adversarial Network (GAN) to implement a more robust model that generate further fine-grained reconstructions.

Ultimately, we plan to implement and quantitatively evaluate these head pose estimation frameworks in the autonomous feeding Feedbot system in order to further assert their robustness to occlusions of the feeding robotic arm.

Bibliography

[1] K. Bertok and A. Fazekas, "Recognizing complex head movements," *Australian Journal of Intelligent Information Processing Systems*, vol. 14, pp. 3–17, 12 2016.

[2] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang, "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] OpenCV, "Camera calibration and 3d reconstruction," 2021. [Online]. Available: https://docs.opencv.org/3.4.15/d9/d0c/group__calib3d.html#ga549c2075fac14829ff4a58bc931c033d

[4] J. M. Diaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker, *Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks*, 07 2019, pp. 326–349.

[5] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3d dense face alignment," 2021.

[6] Y. Wu, C. Gou, and Q. Ji, "Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion," *CoRR*, vol. abs/1709.08130, 2017. [Online]. Available: http://arxiv.org/abs/1709.08130

[7] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[8] J. Jordan, "Introduction to autoencoders." 2018. [Online]. Available: https://www.jeremyjordan.me/autoencoders/

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[10] Y. Zhou and J. Gregson, "Whenet: Real-time fine-grained estimation for wide range head pose," *CoRR*, vol. abs/2005.10353, 2020. [Online]. Available: https://arxiv.org/abs/2005.10353

[11] A. Fernández Villán, R. Usamentiaga, J. Carús Candás, and R. Casado, “Driver distraction using visual-based sensors and algorithms,” *Sensors*, vol. 16, p. 1805, 10 2016.

[12] M. C. d. F. Macedo, A. L. Apolinário, and A. C. d. S. Souza, “A robust real-time face tracking using head pose estimation for a markerless ar system,” in *2013 XV Symposium on Virtual and Augmented Reality*, 2013, pp. 224–227.

[13] C. Silva, J. Vongkulbhisal, M. Marques, J. P. Costeira, and M. Veloso, “Feedbot - a robotic arm for autonomous assisted feeding. in: Oliveira e., gama j., vale z., lopes cardoso h. (eds) progress in artificial intelligence. epia 2017. lecture notes in computer science, vol 10423. springer, cham.” *Lecture Notes in Computer Science*, vol. 10423, 2017.

[14] Z. Zhao, S. Xia, X. Xu, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, “Driver distraction detection method based on continuous head pose estimation,” *Comput. Intell. Neurosci.*, vol. 2020, pp. 9 606 908:1–9 606 908:10, 2020. [Online]. Available: https://doi.org/10.1155/2020/9606908

[15] H. Abdi Khojasteh, A. Abbas Alipour, E. Ansari, and P. Razzaghi, “An intelligent safety system for human-centered semi-autonomous vehicles,” 12 2018.

[16] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, “Face alignment across large poses: A 3d solution,” *CoRR*, vol. abs/1511.07212, 2015. [Online]. Available: http://arxiv.org/abs/1511.07212

[17] K. S. Mawer, “Biwi Kinect Head Pose Database,” 2018. [Online]. Available: https://www.kaggle.com/kmader/biwi-kinect-head-pose-database

[18] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Towards large-pose face frontalization in the wild,” in *In Proceeding of International Conference on Computer Vision*, Venice, Italy, October 2017.

[19] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation in computer vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, 2009.

[20] A. Rekik, A. Ben-Hamadou, and W. Mahdi, “3d face pose tracking using low quality depth cameras,” vol. 2, 01 2013.

[21] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, “3d deformable face tracking with a commodity depth camera,” vol. 6313, 09 2010, pp. 229–242.

[22] T. Baltrušaitis, P. Robinson, and L.-P. Morency, “3d constrained local model for rigid and non-rigid facial tracking,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2610–2617.

[23] Y. Guobing, “Head pose estimation using tensorflow and opencv,” 2019. [Online]. Available: https://github.com/yinguobing/head-pose-estimation

[24] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[26] OpenCV, “Opencv: Solvepnp.” [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga549c2075fac14829ff4a58bc931c033d

[27] F. Garcia, J. M. Diaz Barros, B. Mirbach, K. Varanasi, and D. Stricker, “Combined framework for real-time head pose estimation using facial landmark detection and salient feature tracking,” 01 2018.

[28] 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)*, 2018, pp. 2028–2037.

[29] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[30] J. yves Bouguet, “Pyramidal implementation of the lucas kanade feature tracker,” *Intel Corporation, Microprocessor Research Labs*, 2000.

[31] B. K. Horn and B. G. Schunck, “Determining optical flow,” USA, Tech. Rep., 1980.

[32] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[33] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3d faces,” *SIGGRAPH’99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 09 2002.

[34] V. Lepetit and P. Fua, “Monocular model-based 3d tracking of rigid objects: A survey,” *Foundations and Trends in Computer Graphics and Vision*, vol. 1, 01 2005.

[35] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017.

[36] M. Wenzel and W. Schiffmann, “Head pose estimation of partially occluded faces,” in *The 2nd Canadian Conference on Computer and Robot Vision (CRV’05)*, 2005, pp. 353–360.

[37] G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” 1998.

[38] Y. Wu and Q. Ji, “Robust facial landmark detection under significant head poses and occlusion,” *CoRR*, vol. abs/1709.08127, 2017. [Online]. Available: http://arxiv.org/abs/1709.08127

[39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[41] T.-Y. Yang, Y.-H. Huang, Y.-Y. Lin, P.-C. Hsiu, and Y.-Y. Chuang, “Ssr-net: A compact soft stagewise regression network for age estimation,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 1078–1084. [Online]. Available: https://doi.org/10.24963/ijcai.2018/150

[42] V. Albiero, X. Chen, X. Yin, G. Pang, and T. Hassner, “img2pose: Face alignment and detection via 6dof, face pose estimation,” 2020.

[43] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[44] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[45] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[46] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, “Panoptic studio: A massively multiview system for social motion capture,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3334–3342.

[47] “State of the Art Benchmark: Head Pose Estimation on BIWI,” 2021. [Online]. Available: https://paperswithcode.com/sota/head-pose-estimation-on-biwi

[48] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE MultiMedia*, vol. 19, no. 2, p. 4–10, Apr. 2012. [Online]. Available: https://doi.org/10.1109/MMUL.2012.24

[49] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel realsense stereoscopic depth cameras,” *CoRR*, vol. abs/1705.05548, 2017. [Online]. Available: http://arxiv.org/abs/1705.05548

[50] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, p. 1499–1503, Oct 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603342

[51] C. Jia, T. Yang, C. Wang, B. Fan, and F. He, “A new fast filtering algorithm for a 3d point cloud based on rgb-d information,” *PLOS ONE*, vol. 14, no. 8, pp. 1–21, 08 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0220253

[52] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” AAAI Press, 1996, pp. 226–231.

[53] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[54] L. Koucky and J. Maly, “Mask2face: How we built ai that shows the face beneath the mask,” 2021. [Online]. Available: https://www.strv.com/blog/mask2face-how-we-built-ai-that-shows-face-beneath-mask-engineering

[55] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[56] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[57] J. Charng, D. Xiao, M. Mehdizadeh, M. Attia, S. Arunachalam, T. Lamey, J. Thompson, T. McLaren, J. Roach, D. Mackey, S. Frost, and F. Chen, “Deep learning segmentation of hyperautofluorescent fleck lesions in stargardt disease,” *Scientific Reports*, vol. 10, p. 16491, 10 2020.

[58] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[59] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. Chang, M. G. Yong, J. Lee, W. Chang, W. Hua, M. Georg, and M. Grundmann, “Mediapipe: A framework for building perception pipelines,” *CoRR*, vol. abs/1906.08172, 2019. [Online]. Available: http://arxiv.org/abs/1906.08172

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[61] H. Hotelling, "Analysis of a complex of statistical variables with principal components," *J. Educ. Psy.*, vol. 24, pp. 498–520, 1933.

[62] I. Jolliffe, "Principal component analysis," *Encyclopedia of statistics in behavioral science*, 2005.

[63] N. Adaloglou, "Intuitive explanation of skip connections in deep learning," *https://theaisummer.com/*, 2020. [Online]. Available: https://theaisummer.com/skip-connections/

[64] M. Shao, Z. Sun, M. Ozay, and T. Okatani, "Improving head pose estimation with a combined loss and bounding box margin adjustment," *CoRR*, vol. abs/1905.08609, 2019. [Online]. Available: http://arxiv.org/abs/1905.08609

A

Autoencoder and U-Net Architecture

A.1 Autoencoder

An autoencoder is an unsupervised artificial neural network that learns to efficiently encode unlabelled input data (hence unsupervised) and afterwards reconstruct it from the encoded representation, as close to the original input as possible. The structure of an autoencoder has three main parts (figure A.1): The encoder, which reduces the input dimensions and compresses its data into a lower dimensional encoded representation, the latent space; The bottleneck, the layer which contains this compressed information and therefore has the lowest dimensional representation of data in the entire network; and the decoder, which decompresses the encoded data and maps it to a reconstruction of the input. In order to be able to reconstruct the output as close to the original image as possible, autoencoders are trained by minimizing and a reconstruction error loss which measures the differences between the reconstructed output and the original input and allows the model to learn the most important latent attributes to recover the input information in the best way possible.

The fact they learn to encode and decode information without the need for any labels means they are an unsupervised learning technique, able to generate their own labels from the training data. This is

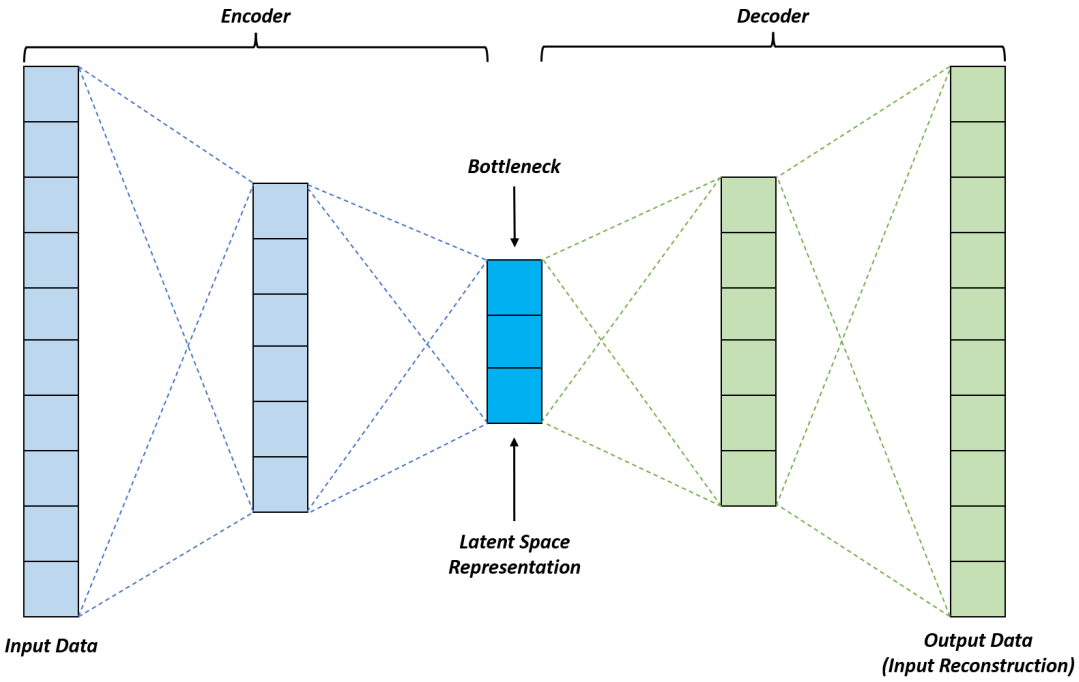


Figure A.1: Basic autoencoder structure.

an advantage as it makes them simpler to train. Another advantage is that unlike Principal Component Analysis (PCA) [61] [62], a common linear dimensionality-reduction method which aims to discover a lower dimensional hyperplane (subspace with one less dimension than that its original space) that best describes the original data, autoencoders are capable of learning more complex non-linear representations of data due to the non-linear activations of the encoder and decoder, which are neural networks, and therefore are a more powerful generalization of the PCA technique (an autoencoder without the non-linear activations acts in the same way as PCA).

Despite these important advantages, standard autoencoders also have some issues. As the input dimensions are reduced in the encoder through down-sampling, there is a loss of image information which might be relevant to the quality of the facial reconstruction. The authors of *mask2face* [54] carried out a machine-learning project that aimed to develop a model to show what a person wearing a face mask looks like without that mask. For that purpose they employed the U-Net architecture, an enhanced autoencoder pipeline which reduces the information loss of standard autoencoders and achieves more fine-grained results.

A.1.1 U-Net architecture

U-Net is a convolutional neural network architecture developed originally for biomedical image segmentation [9], but since then broadened for a wide variety of tasks, such as image reconstruction, inpainting

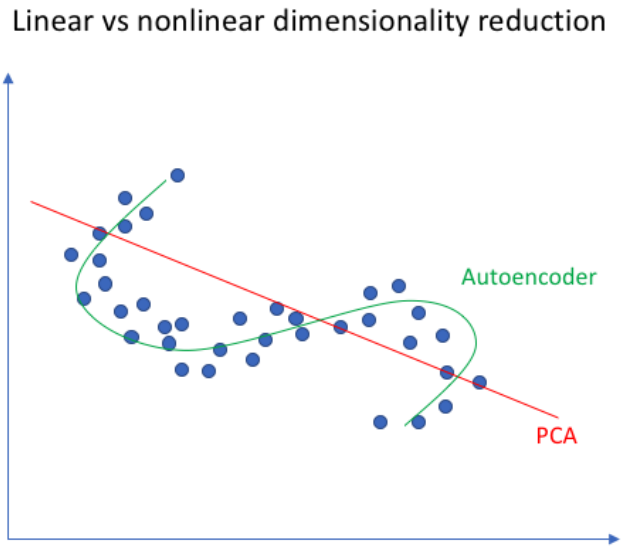


Figure A.2: PCA(linear) vs. autoencoder(non-linear) dimensionality reduction [8].

or colorization. The original U-Net pipeline is exemplified in figure A.3.

Essentially, this U-shaped architecture is that of an autoencoder. There is a contracting path (encoder) which consists of convolutions, each followed by a Rectified Linear Unit (ReLU) activation function that nullifies non-positive outputs, and max-pooling operations which calculate the maximum value for patches in a feature map to generate a down-sampled (pooled) feature map. This encoder generates an embedding with the lowest spatial information but highest feature information (bottleneck), which is afterwards expanded back to the original input dimensions through a series of up-samplings with transposed convolutions and regular convolutions (decoder). There is, however, a crucial difference in regards to the standard autoencoder. U-Net implements links called skip connections between feature maps of the encoder and feature maps of equal dimensions in the decoder. Skip connections concatenate feature maps of an encoder layer to the corresponding decoder layer that results from each up-sampling, a process that speeds up the learning process, while stabilizing training and convergence, and tackles the vanishing gradient problem [63]. Another advantage is that, since skip-connections pass the information from the encoder directly to the decoder, they ensure that the features that are learned in the encoding process are actually used in the reconstruction process and therefore help to recover from the loss of information that occurs during the down-sampling in the encoder.

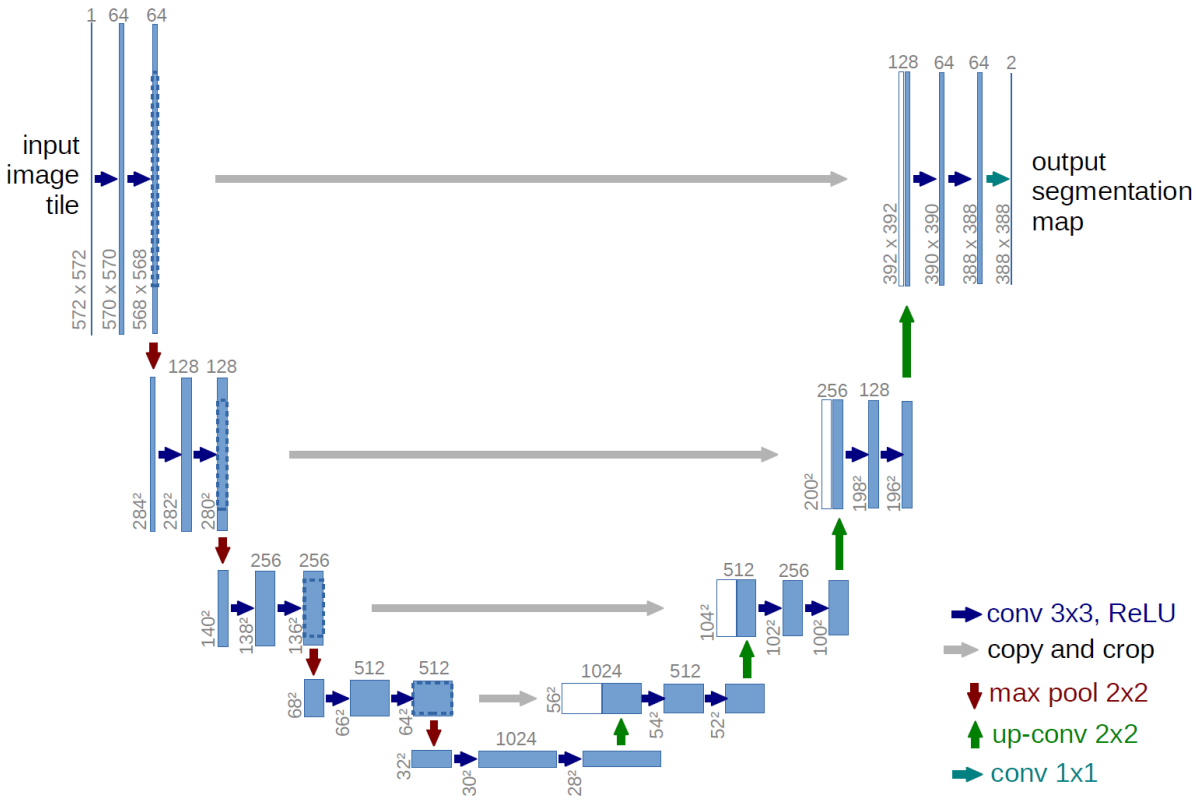


Figure A.3: U-Net architecture from [9].

B

Data Pre-Processing

For 300W-LP and AFLW2000 datasets we use provided 2D landmark annotations to extract a face bounding box:

$$\begin{aligned} x_{min} &= min(L_x^{2D}) \quad x_{max} = max(L_x^{2D}) \\ y_{min} &= min(L_y^{2D}) \quad y_{max} = max(L_y^{2D}) \end{aligned} \tag{B.1}$$

where x_{min}, x_{max}, y_{min}, y_{max} are the margins of the bounding box for x and y image coordinates, and L_x^{2D}, L_x^{2D} are x and y face landmark coordinates. Furthermore, we widen the margins of the bounding boxes so that the cropped images which will be inputted to the network better capture the face and head. This approach is based in the work developed in [64], which studies the benefits of adjusting the bounding box margins of detected faces to achieve better head pose estimation results. For that purpose, we define the adjusted box margins according to a margin control parameter K (equation B.3). The y_{min} margin is multiplied by two, since the top of the original bounding box is defined very close to the eyes. Figure B.1 shows the bounding box defined with 2D face landmarks and the bounding box with adjusted margins in an example image of the 300W-LP dataset.

$$
\begin{aligned}
x_{min} &= x_{min} - k(x_{max} - x_{min}) \\
x_{max} &= x_{min} + k(x_{max} - x_{min}) \\
y_{min} &= y_{min} - 2 \times k(y_{max} - y_{min}) \\
y_{max} &= y_{max} + k(y_{max} - y_{min})
\end{aligned}
\tag{B.2}
$$

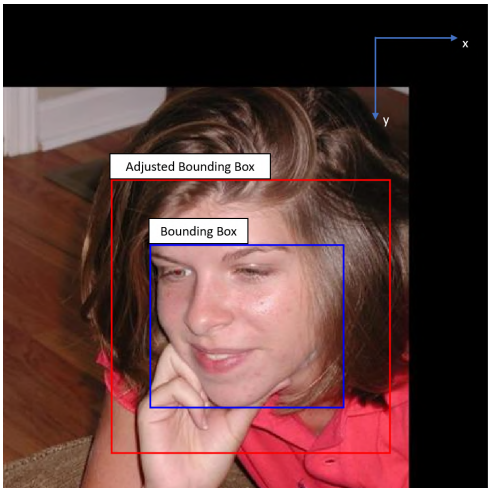


Figure B.1: Original and adjusted bounding boxes.

Pose labels are provided in radians and converted to degrees:

$$
Euler^{\circ} = \frac{Euler^{rad} * 180^{\circ}}{\pi} \tag{B.3}
$$

The continuous labels for the regression losses correspond directly to the converted angles, and the labels for classifications correspond to the indices of the bins to which each angle value belongs.