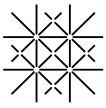


University of Basel

An Empirical Comparison of Deep Learning and 3DMM-Based Approaches for Facial Occlusion Segmentation

Bachelor-Thesis

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Graphics and Vision Research Group (Gravis)
http://gravis.dmi.unibas.ch/

Examiner: Prof. Dr. Thomas Vetter
Supervisor: Dr. Adam Kortylewski

Elias Arnold
elias.arnold@stud.unibas.ch
14-930-770

10.08.2018

Acknowledgments

I would like to thank Dr. Adam Kortylewski for his extraordinary support and guidance in this thesis process. He assisted me greatly and was always willing to help me. Thanks to Prof. Dr. Thomas Vetter for his expert advice and the opportunity to write this thesis in his research group.

This project would have been impossible without the excellent tutorials of the Gravis group of the University of Basel. These tutorials have made it much easier for me to get started with Scalismo. The results of this work have been generated on the hardware of the Gravis group.

Abstract

To make a 3D reconstruction of a 2D face, it has to be separated from the background. Often masks are used which determine the facial region. A major problem in finding these masks is partial occlusion. There are several approaches on how to recognise them. A *fitting-algorithm* depends heavily on these masks. It optimises the parameters of a *3D Morphable Model (3DMM)* so that the 3D face looks similar to the depicted face.

In this thesis, we use the occlusion-aware face segmentation network proposed by Nirkin et al [1]. In a first step, we measure the quality of the mask by applying the network to different datasets and occluded facial images whose segmentation is known. Then, multiple fits of the same 2D image are performed, each with a different mask. The resulting fits are compared by their parameters.

Table of Contents

1 Introduction

Fitting is the process of optimising the parameters of a 3DMM when a 2D image is given. There are different approaches on how to do that. Some use a mask with labels for each pixel of the image that determine whether or not the pixel represents a part of the imaged face. An additional difficulty is partial occlusion of the facial region, which must be excluded from the mask too. To achieve satisfying fits, the segmentation has to cut out the face as accurately as possible. There are different approaches to algorithmically generate such a mask. Due to the variety and diversity of such occlusions, this is not a simple problem. In this thesis, the quality of such segmentations by machine-learning-algorithms is measured. We compare the neural network of Nirkin er al [1] and the existing model-based top-down approach by Egger et al [2].

1.1 Artificial Neural Networks

The idea of artificial neural networks was somehow influenced by biology. These networks consist of a variety of neurons which are grouped in layers. The way each neuron works is very simple. It takes multiple inputs of varying strength from other neurons, sums them up, puts the sum into a non-linear function (e.g. Maxout, Sigmoid, ReLu) [Figure 1.1], and decides depending on the output if it should send a stimulus itself and if so, in which strength. Each layer is somehow connected to the next layer. Some layers are fully connected (each neuron of a layer is connected to every other neuron in the next layer) while others are convolutional. Convolutional means that a neuron only gets an input of spatially close neighbours in the previous layer. There are many different architectures which mainly differ in the number of layers, number of neurons per layer, and the interconnectivity of the neurons. A classical convolutional neural network (CNN) is depicted in [Figure 1.2].

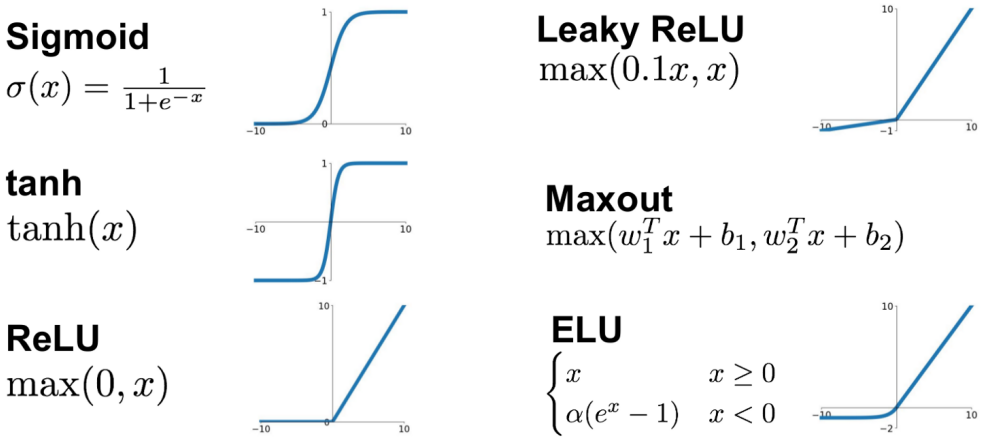


Figure 1.1: Six often used activation functions[1]. The most commonly used activation function is ReLU because of its simplicity. The disadvantage of this function is if one input has a negative sign, then all following neurons output a signal of 0. An improvement of ReLU is Leaky ReLU.

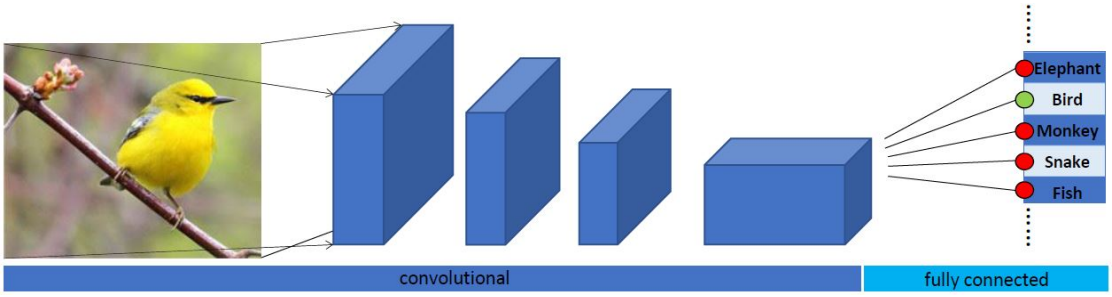


Figure 1.2: An example of a classical convolutional neural network (CNN). After a certain number of convolutions, a pooling layer extracts the most important information of the image and writes it into the next layer. That is why the picture is getting smaller and smaller until just a vector is left.

Already in 1943, Warren McCulloch and Walter Pitts [3] showed that even simple networks of this kind can simulate every possible logical formula. For this, they used a neuron model which consisted of simple logic gates and could only process binary input and output signals.

[1] Jadon S. (2015 March). Introduction to Different Activation Functions for Deep Learning. Retrieved July 19, 2018 from *https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092*

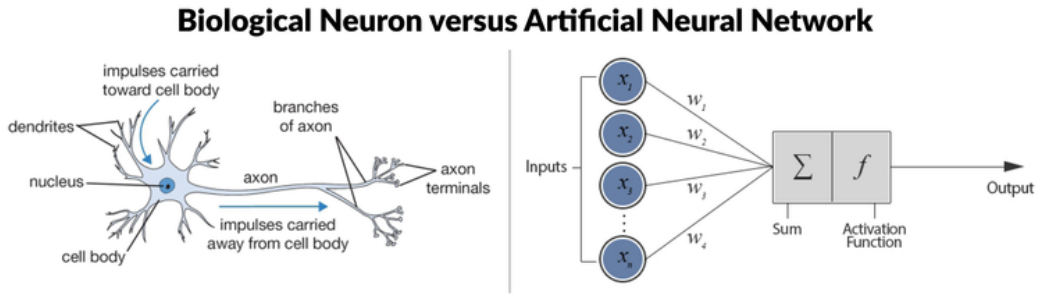


Figure 1.3: The left-hand side of the image[2] shows a biological neuron. It is a nerve cell which occurs in almost every animal. On the right-hand side, an artificial neuron is depicted. It sums up the stimuli of the previous neurons, applies an activation function to the sum, and forwards the output.

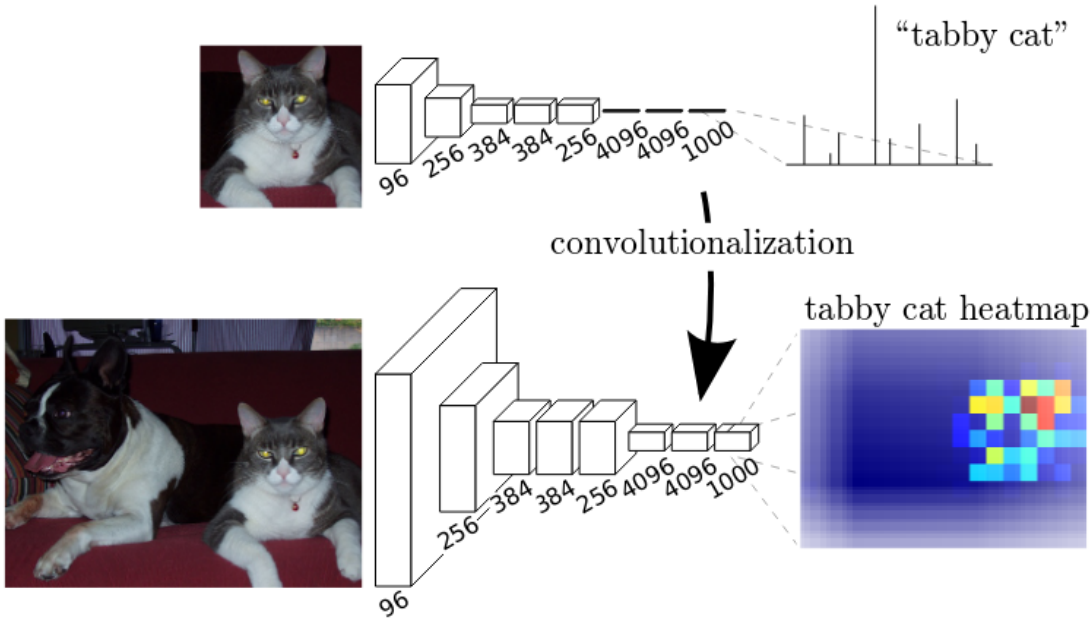


Figure 1.4: The transformation from a CNN into an FCN. This is a schematic diagram of Long et al [4], which fine-tuned the network used in this work. They call this process "convolutionalization". They turn the fully connected layers into convolution layers which produces an efficient machine for end-to-end dense learning.

[2] Willems K. (2017 May, 2nd). Keras Tutorial: Deep Learning in Python. Retrieved July 10, 2018 from *https://www.datacamp.com/community/tutorials/deep-learning-python*

1.2 The Fully Convolutional Network Used

For this thesis, a pretrained fully convolutional network from [1] is used. A fully convolutional network (often called: FCN) is basically a CNN but with a modified architecture. An FCN does not have the fully connected layers which are usually found at the end of an CNN [compare Figure 1.2 and Figure 1.4]. These layers would enable the network to make decisions based on global information. A CNN for example can be used for classification. Nevertheless, for image analysis we want local information of the input image (we do not want to know if there is a face in the image, but where the face is in the image). Therefore, a FCN uses only convolutional and pooling layers. In the whole fully convolutional network only the following structure is repeated: One or more convolution layers and a pooling layer which downsamples the picture. This constellation is recurring several times.

The assembly of the network used for this thesis follows the FCN-8s-VGG architecture with extensions of Long et al [4] [Figure 1.4]. The first part 'FCN' stands for 'fully convolutional network', '8s' means that the result gets upsampled eight times (because of the pooling layers), and 'VGG' means that the popular 16-layer network by Oxford's visual geometry group [5] is used [Figure 1.5]. The original task of the network was to find the name of an object in an input image. The network could distinguish between 1000 different objects. Each cell in the final vector (1*1000 in size) was a boolean variable for one specific item. A schematic representation of this architecture can be found in [Figure 1.2].

For our experiments a pretrained fully convolutional network (FCN) of [1] is used. It was shown that even with a widespread network, good segmentation can be made and that the network does not have to be specially tailored for the future purpose. However, the network must have been trained with a large enough data set. Nirkin et al used the FCN for intra- and inter-subject face swapping on the Labeled Faces in the Wild (LFW) dataset and showed that intra-subject swapped faces remain as recognisable as before the swap and that in the inter-subject version better face swapping leads to less perceptibility.

Nirkin et al used a semi-supervised approach to produce training data in order to train the FCN. To make large quantities of them, 2'043 face videos of the IARPA Janus CS2 dataset of Klare et al [6] were used. To avoid searching for the face in every frame of the video, they used motion queues which tracked the face given an initial segmentation based on the approach of [7] which enriched their training set to 9'818 images. To enlarge the collection of images, they rendered 3D Shapes of various objects (e.g. sunglasses, hands) into existing images. Each occlusion adds 9'500 images to their training set.

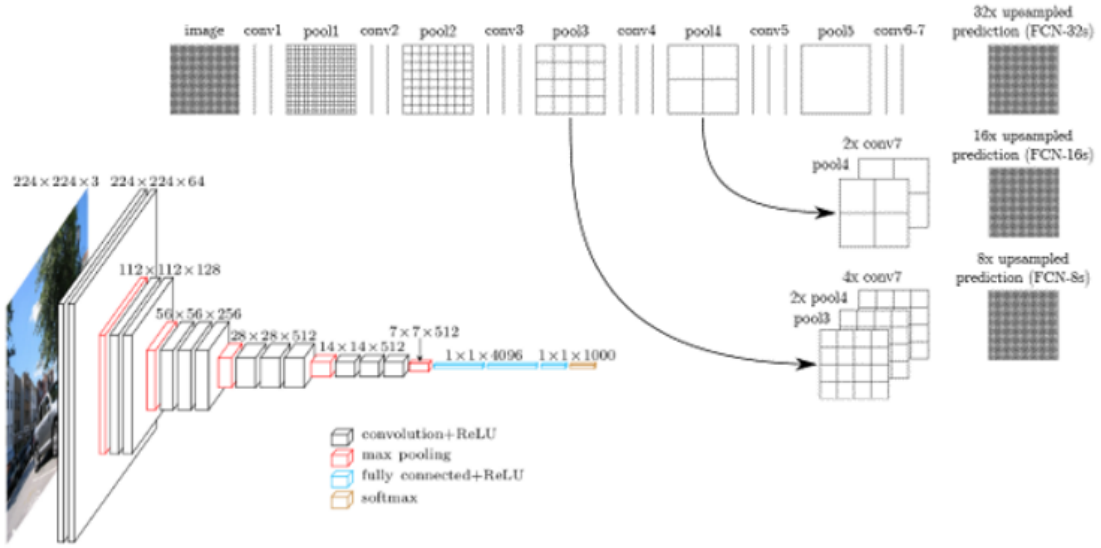


Figure 1.5: In the lower-left corner, the 16 layers of the well known VGGnet are shown. In the top-right, you can see the meaning of the '8s' term of the FCN-name. It means that the resulting image has to be 8x upsampled, to get a replica of it which is equal in size to the input image.

1.3 Occlusion aware 3DMM

Egger et al propose a fully automated, probabilistic and occlusion-aware 3D morphable face model adaptation framework [2]. These methods use an iterative approach to label each pixel whether it belongs to the face or to the background. This approach can handle multiple labels and differentiate between multiple occlusion types, for example, face specific ones (eg. beards) and background. For updating the z-labels they use an algorithm which classifies a pixel based on the probabilities for each possible label, but for our experiments, we limited ourselves to two. We only need to distinguish between the face (including skin and beard) and the background.

The algorithm does two things at the same time. In addition to creating a face mask (segmentation), it estimates the parameters of the popular Basel face Model to reconstruct the given face. It is an EM-algorithm-like method to solve two problems simultaneously. In the E-steps Egger et al update the z-labels and in the M-steps they update the face model parameters based on the current estimate of the z-labels. For face model adaptation they apply a stochastic sampling strategy based on the Metropolis–Hastings algorithm (Markov Chain Monte Carlo). The likelihood of each pixel is split up into a background model (X_{BG}) and a foreground model (X_{FG}).

$$\underbrace{p(\Theta|I) = p(I|\Theta) * p(\Theta)}_{\text{posterior probability of face model parameters } \Theta \text{ given an image } I} \qquad \text{with: } p(I|\Theta) = \prod_{X \in \text{pixels}} p(X_{BG}|\Theta)^{z-1} * p(X_{FG}|\Theta)^{z}$$

Conventional approaches for occlusion-segmentation often fail on important parts of the face such as the eyes, eyebrows or the oral region due to their strong variability in colour and shape. The segmentation of Egger et al has difficulties with these aspects too as Figure 1.6 shows. The algorithm starts with an initial guess and then alternating updates the parameters Θ and the z-labels. From the updated parameter set (M-Step) the algorithm renews the z-labels (E-Step) and vice versa (see Figure 1.7).

Figure 1.6: This picture shows the development of the labels after 0, 10 and 20 iterations. Noticeable in this sample image are not only the eyes as mentioned before but also the shadow of the nose, which is first segmented as a background. Only after a certain number of iterations these errors are partially recognised and provided with the correct label.

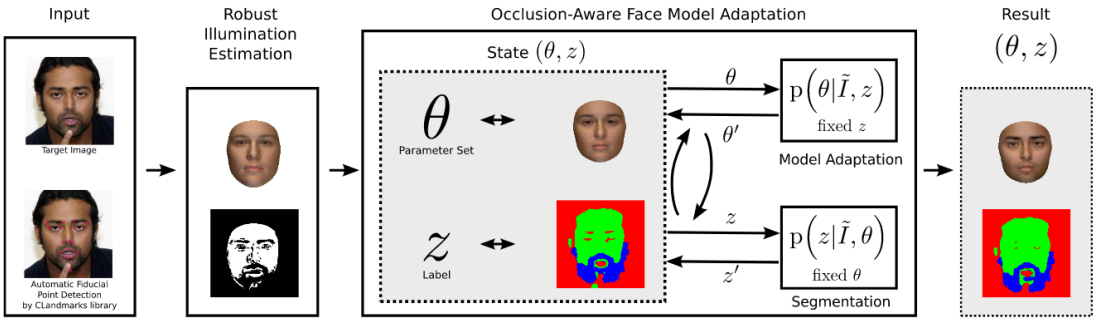


Figure 1.7: [3] Algorithm overview: As input, the algorithm takes a target image and fiducial points. The external Clandmark library for automated fiducial point detection from still images [Uřičář et al (2015) [8]] is used. The algorithm starts with an initial face model fit of our average face with a pose estimation. Then, a robust illumination estimation for initialisation of the segmentation labels z and the illumination settings is performed. For this task, a random sample consensus (RANSAC) algorithm is used. RANSAC methods estimate parameters of a mathematical model from a set of observed data that contains outliers. Then the face model and the segmentation are simultaneously adapted to the target image I. The result is a set of face model parameters Θ and a segmentation into face and non-face regions. The presented target image is from the LFW database (Huang et al (2007))

An approach using convolutional neural networks to segment occluded faces has already been described by [9]. The big difference to our approach is that multiple frames are needed for the final segmentation. Another approach of Morel-Forster et al [10] uses random forests to detect facial-occlusions by hair. These are integrated into the fitting process in order to model uncertainty. Unlike us, Morel-Forster et al integrate the occlusion in the face likelihood of the evaluator.

[3] Figure 1.7 is copied from Fig.4 of the "Occlusion-aware 3D Morphable Models and Illumination Prior for Face Image Analysis" paper of Egger et al. [2]

2
Evaluation

2.1 Qualitative Evaluation on Real-World Datasets

The face segmentation network was tested on two different datasets with real-life images: Firstly with the Caltech Occluded Faces in the Wild (COFW) dataset by [11] and secondly with the parts-labeled LFW dataset of the University of Massachusetts [12]. Both datasets are designed to present faces in real-world conditions. The COFW dataset provides 29 landmarks and a bounding box for all 507 images. The original LFW dataset contains 13'000 images of 1'680 different subjects. Each face is labeled with the name of the depicted person. This database is actually meant to test facial recognition/verification algorithms. Nevertheless, we tested the segmentation of the FCN on the 500 images of the Parts-LFW validation set. For each image, there is a ground truth segmentation, which makes it possible to measure the quality of the FCN's segmentation in numbers.

2.1.1 Evaluation on the COFW Dataset

Since on the COFW dataset only landmarks and bounding-boxes are given, the segmentation had to be evaluated qualitatively. We tried reconstructing the graphic on Nirkins [1] github-repository 'face_segmentation' [Figure 2.1]. In [Figure 2.2] the same 18 images are segmented by the iterative method of Egger et al. This algorithm outputs both, parameters for a 3DMM and a segmentation. [Figure 2.3] shows the fits of 9 images of the COFW dataset where both segmentations are used. A matrix of the other 9 fits can be found in the Appendix [Appendix A.1].

Figure 2.1: 18 images of the COFW Dataset overlaid with the FCN output (in red). The segmentation results are very similar to those on Nirkin's github page.

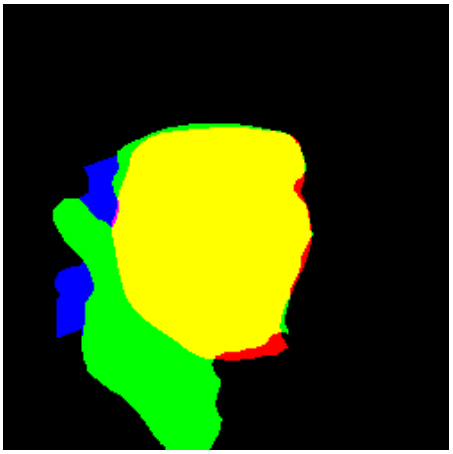
Figure 2.2: The same target images as in [Figure 2.1], but this time with the (final) segmentation of the occlusion-aware method of Egger et al [2]. Often the eyes are not segmented or the segmentation includes skin other than the face (eg. hands).

Figure 2.3: Tuples of facial images. In every tuple, the first image shows the fit with the mask of the algorithm of Egger et al itself [Figure 2.2]. The second shows the fit with the FCN mask which are depicted in [Figure 2.1].

2.1.2 Evaluation on the Parts-LFW Dataset

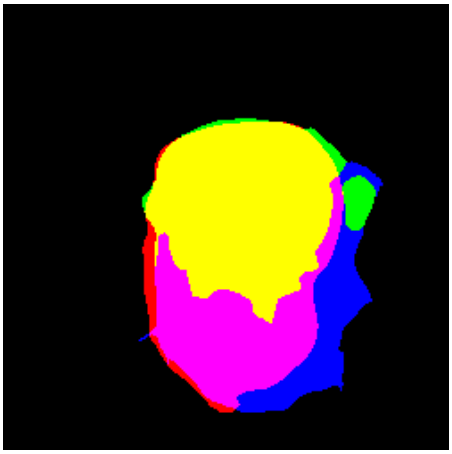
In the Parts-LFW dataset, we had a ground-truth mask for every image. The mask distinguishes between hair, skin, and background. We looped through all the provided ground-truth masks and overlaid them with the segmentations of the FCN. Now we can compare the labels of both masks. The evaluation is quite impressive. The FCN performs very well in segmenting only pixels that belong to the face. On average over the 500 images of the Parts-LFW dataset, there are 98.5% right non-segmentations (only 1.5% false positives). On the other hand, only 85.4% of all the pixels which belong to the face are segmented as face (14.6% false negatives) which is not a good but acceptable result. [Figures 2.4(a) and 2.4(b)] depict such a face image and its mask. Unfortunately, on the given mask no distinction is made between face and other parts of the body, but everything is segmented as skin [Sub-figures 2.4(a) and 2.4(b)]. However, it is more problematic that some faces have beards. The FCN of [1] segments facial hair which was excluded from the provided labels. Therefore, we had to manually remove these images [Sub-figures 2.4(c) and 2.4(d)]. To reduce the effort, we took the Parts-LFW validation set containing 500 images. After removing the ones with a beard or mustache, we were left with 447 images. The results are summarised in [Figure 2.5].

(a) A facial-image with the segmentation of the FCN highlighted in red.



(b) The provided ground-truth segmentation of the image 2.4(a) overlaid with the mask of the FCN (red).

(c) An facial-image with the segmentation of the FCN highlited in red.



(d) The provided ground-truth segmentation of the image 2.4(c) overlaid with the mask of the FCN (red).

Figure 2.4: In each row, the left-hand is an image of the Parts-LFW dataset overlaid with the segmentation of the FCN (red). The skin region is colored in green, hair regions are blue, and the background is black. The mixed colours (yellow and magenta) arise because the red mask of the FCN is also present in these plots.

false-positives (hair)	7.04%
false-positives (background)	0.68%
false-negatives (background)	14.13%
right-segmentations	85.87%
right non-sementations	99.32%

Figure 2.5: The averages over the 447 images of the Parts-LFW evaluation set of [12]. The FCN recognises (almost) only pixels which belong to the face (little false positives). By reducing from 500 to 447 images, we were able to lower this number even more. Unfortunately, there are many false negatives (skin pixels labeled as background).

2.2 Evaluation on Synthetic-Data

2.2.1 Experimental Setup

In order to evaluate the FCN on synthetic-data, we used the parametric face image generator of Kortylewski et al [13] to produce images of a random face in a given pose [Figure 2.6]. We extended the software so that it now renders occlusions over the face. Furthermore, we changed the parametric face image generator so that it now generates a ground-truth mask, which classifies every pixel either as part of the face or as non-face. In this experiment the estimated mask of the FCN is compared to the ground-truth mask of the parametric face image generator.

Figure 2.6: Five examples of the synthetic face images. The same face is shown with yaw angles -45°, -25°, 0°, 25°, and 45°. The type of the occlusion is chosen randomly and in an arbitrary orientation and position.

2.2.2 Data for the Experiment

Nirkin et al [1] claim that both the face itself and the context of the face play an important role for the outcome of the segmentation. To prevent this effect and reduce the impact of outliers, we repeat the experiment 5 times with a different face and a different background image. The number of images used in a single run of the experiment varies but is similar to the number of grid cells in the following plots that visualise the experiments: 101 images were used to measure the dependence of one rotation alone (top row of [Figure 2.7]), in the bottom row of [Figure 2.7] 441 images were used, to plot the rotations versus the degree of occlusion [Figure 2.9] we used 340 images per plot and 380 were used for the plots in [Figure 2.10].

2.2.3 Dependence of the Euler Angles

Because we are now able to create synthetic face images in any desired pose, we first want to measure the segmentation-accuracy of the FCN for the rotations: Yaw, roll, and pitch. We evaluate each rotation itself and every possible combination of two rotations in order to create a hierarchy under the rotations [Figure 2.7].

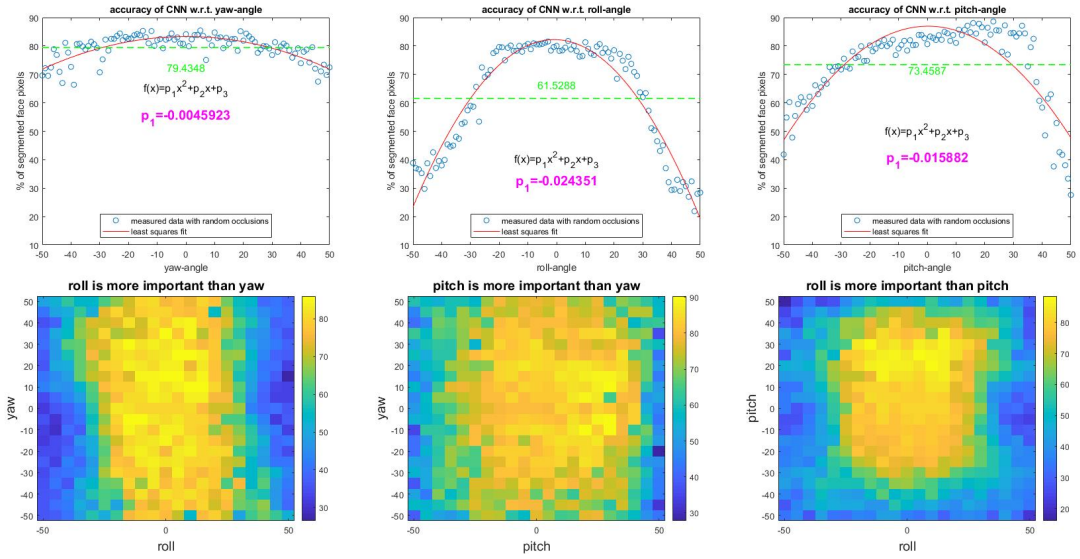


Figure 2.7: In the plots on the top row we see the segmentation accuracy in percent (on the y-axis) for every single image (with face angles from $-50°$to $50°$on the x-axis). The point cloud is approximated by a quadratic function via a least squares fit (red curve / $f(x)$). The first parameter of this function determines the opening angle (p_1). The greater the absolute value of this parameter, the more sensitive is the FCN to the respective rotation. In the bottom row, the colours indicate the segmentation accuracy. The brighter the colour, the better the segmentation. In every plot, there is a cluster of high accuracy segmentations centered in the origin. The rotation on whose axis the cluster has the lower variance is the more important of the two. A rotation is called 'important' when a small change of this rotation leads to a failure of the FCN.

From the graphs of [Figure 2.7], we can conclude that the roll rotation is the most relevant for the FCN, that the pitch rotation is less important, and that the accuracy of the FCN is still good even with high yaw angles (yaw is the least important rotation).

2.2.4 Random Boxes as Occlusions

The 340 images for the plots in [Figure 2.9] consist of pictures with 20 different occlusion levels, where one rotation is in the range from $-40°$to $40°$and the other two rotations are set to 0.

In the given table of [Figure 2.8], all provided images show a face, from which 20% of the pixels are occluded by a randomly colored box. We can optically verify, that firstly, a high yaw rotation, despite the occluding box, has not much of an effect. Secondly, that in both situations, -40°and 40°of the roll rotation, the result is not satisfying. That is interesting because no matter how big the roll rotation is, the information (the face) stays the same. Further, in the third column, the segmentation with a negative pitch angle is much worse than with a positive one. This supports our assumption that the roll rotation plays a big role, followed by the (asymmetric) pitch rotation. The yaw rotation is less important because even at large angles a big part of the face is still segmented.

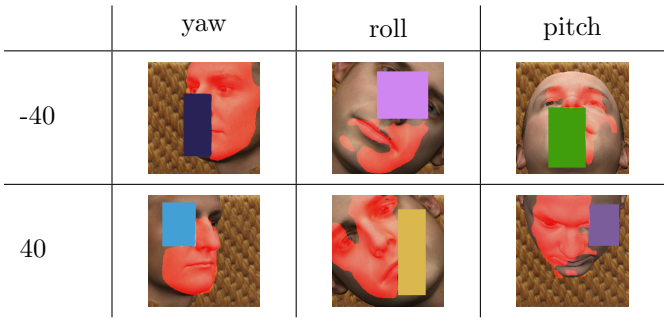


Figure 2.8: Based on these images, we can see that the roll rotation is the most sensitive, followed by the pitch rotation, where the segmentation works better on positive angles than on negative ones. The most stable detection is at the yaw rotation. It has the least influence on the segmentation.

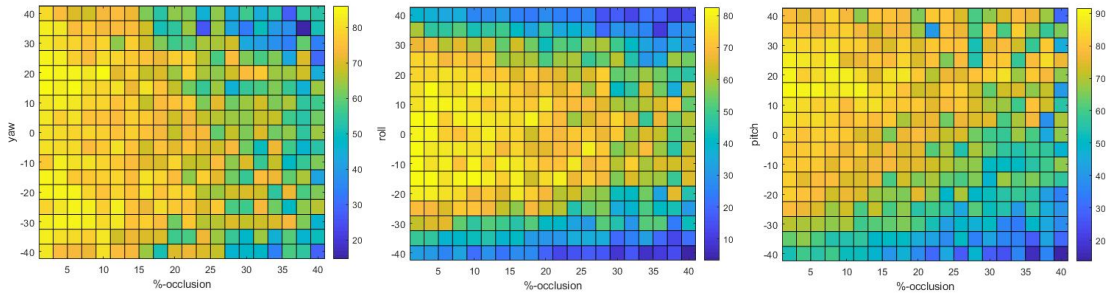


Figure 2.9: The colour of each grid cell indicates the accuracy of the segmentation of the FCN on a set of faces turned by the corresponding angle and occluded with a rectangle so that the corresponding amount (on the x-axis) of the face is hidden. The brighter the colour, the better the segmentation. On each plot, we would expect to see a triangle pointing to the right. This means that the combination of a large angle and a big occlusion make the face even more unsegmentable.

We see that the segmentation is very sensitive to the roll rotation and that the FCN is not trained to segment faces that are aslope. Surprisingly, the yaw rotation plays a subordinate role here. The rightmost plot of [Figure 2.9] tells us, that the sign of the pitch rotation plays a significant role because the plot is asymmetric. Since we are not able to determine at which angles exactly the FCN begins to fail, we repeated the experiment with a higher angle range [Figure 2.10].

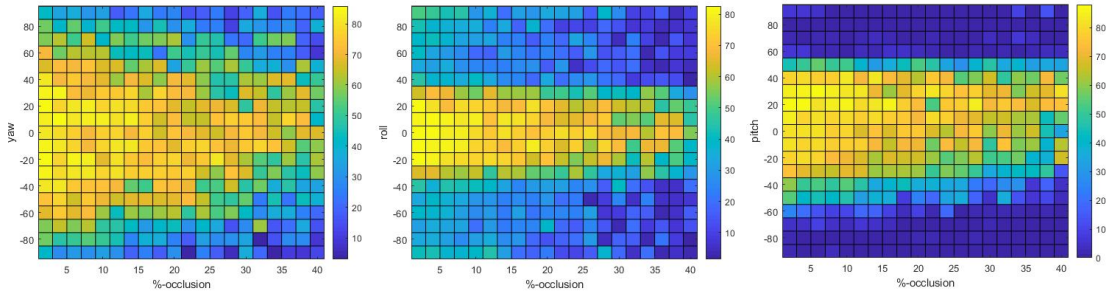


Figure 2.10: This plot shows the outcome of a similar experiment as shown in Figure 2.9 with less resolution. All the angles (yaw, pitch, and roll) range from $-90°$to 90 °in steps of 10°. The scale on the "%-occlusion" stays the same as in Figure 2.9. We can clearly see the limits of the FCN even with a occlusion of 2%. Very interesting is the hard transition from good segmentations to bad segmentations in the two right plots.

Although in practice exact rectangles are very rare, boxes as occlusions are very simple and we are in control of the rectangle's size. It is the simplest method to occlude a given amount of the face region.

3

Evaluation of the Fitting with the Segmentation of the FCN

The EM-algorithm-like method of Egger et al [2] uses a binary mask to determine whether a pixel is relevant for the fitting-process or not. We changed the method of Egger et al to not estimate a mask itself, but to use the segmentation of the FCN in each iteration.

As the Morphable Model, we use the popular Basel Face Model of 2017 [14] which was originally proposed in 1999 by Blanz and Vetter [15]. This model is used by the fitting process, but also by the parametric face image generator of Kortylewski et al [13] which outputs renderings of sample facial images, the ground truth parameters of the face, and provides a ground truth segmentation. These ground truth data gets compared to the parameters of the fit.

The experiments in this chapter will only be explained with one specific type of occlusions (hands) so as not to overwhelm the reader. Of course, we repeated the experiments on other occlusions types such as microphones, glasses, random boxes, and without any occlusion. Please find these fits and the error-plots in the appendix [Appendix A.2].

3.1 Experimental Setup

For the evaluation of the fitting, we generated faces with 50 shape parameters of the Basel Face Model, another 50 for the colour and 50 for the expression. We now render different faces with various occlusions, diverse rotations/poses, a random illumination and cover them with different occlusions.

The aim of this chapter is to reconstruct the generated synthetic faces with the EM-like algorithm of Egger et al which originally estimates fitting parameters and a segmentation itself.

But we manipulate the algorithm such that it skips the E-step and takes the segmentation of the FCN, the ground truth segmentation, or no mask in every iteration. To get acceptable results we fix the number of EM steps to 20. In each iteration, 1000 samples are drawn. We measure the fitting accuracy and speed in this experiment, without letting the fitting algorithm produce its own z-labels. The FCN is trained to cut hands, glasses, and microphones out of the picture. That's why in the following experiments we use these three classes of occlusions. With the mask of the FCN, the fitting should also be much faster, because the time-consuming job of updating the z labels is eliminated. After every iteration, we get information about the pose, the probability of the current sample, and about the 3DMM parameters of the fit. We compare those to the ground truth information from the parametric face image generator. To calculate the error, RMSE (Root Mean Squared Error) is used.

3.2 Evaluation on the Tailored Face

In this setting, we use the 'face12' version of the Basel Face Model. In contrast to the original, it has no neck, no ears, and no hairline. In every iteration the errors of the parameters of the Basel Face Model, the rotations, the illumination, and the probability of the best of the 1000 samples are measured [Figure 3.3]. The experiments are repeated 10 times with different faces.

We see that in the three parameters (shape, colour, and expression) the quality of the fit with the FCN segmentation is very close to the one of the iterative approach of Egger et al. The differences in the Euler rotations (yaw, pitch, and roll) are negligible, since all errors move in a very small order. The plot named 'EnvironmentMap' [Figure 3.3], shows the deviation of the illumination parameters. These parameters simulate different light sources around the face.

Since the evaluation data consists of occluded synthetic-facial images where the exact position and shape of the face is known, the masks used consist of the ground truth segmentation, the segmentation of the FCN, the occlusion-aware segmentation proposed by Egger et al [2], and a dummy mask where each pixel is segmented as face [Figure 3.2].

(a) original facial image (b) facial image with occlusion

Figure 3.1: These pictures show the faces that should be modeled with different masks. We fit with the method Egger et al [2] from Figure 3.1(b) a face as similar as possible to the face in Figure 3.1(a).

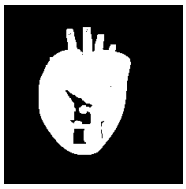
	ground truth mask	Egger	FCN	no mask
z-labels				
fits				

Figure 3.2: In the top row are the different segmentations we use for our experiments. The segmentation of Egger et al (second image from left) was calculated iteratively. Only the final mask is given. In the second row are the particular fits, when the above z-labels are used as mask. You can see that the fits without any mask (no mask) and the fits with the mask according to Egger et al (Egger) are far away from the true face depicted in [Figure 3.1(a)]. However, the fits with ground truth mask of the parametric face image generator (ground truth mask) and the one with the FCN-Segmentation (FCN) come very close to the original.

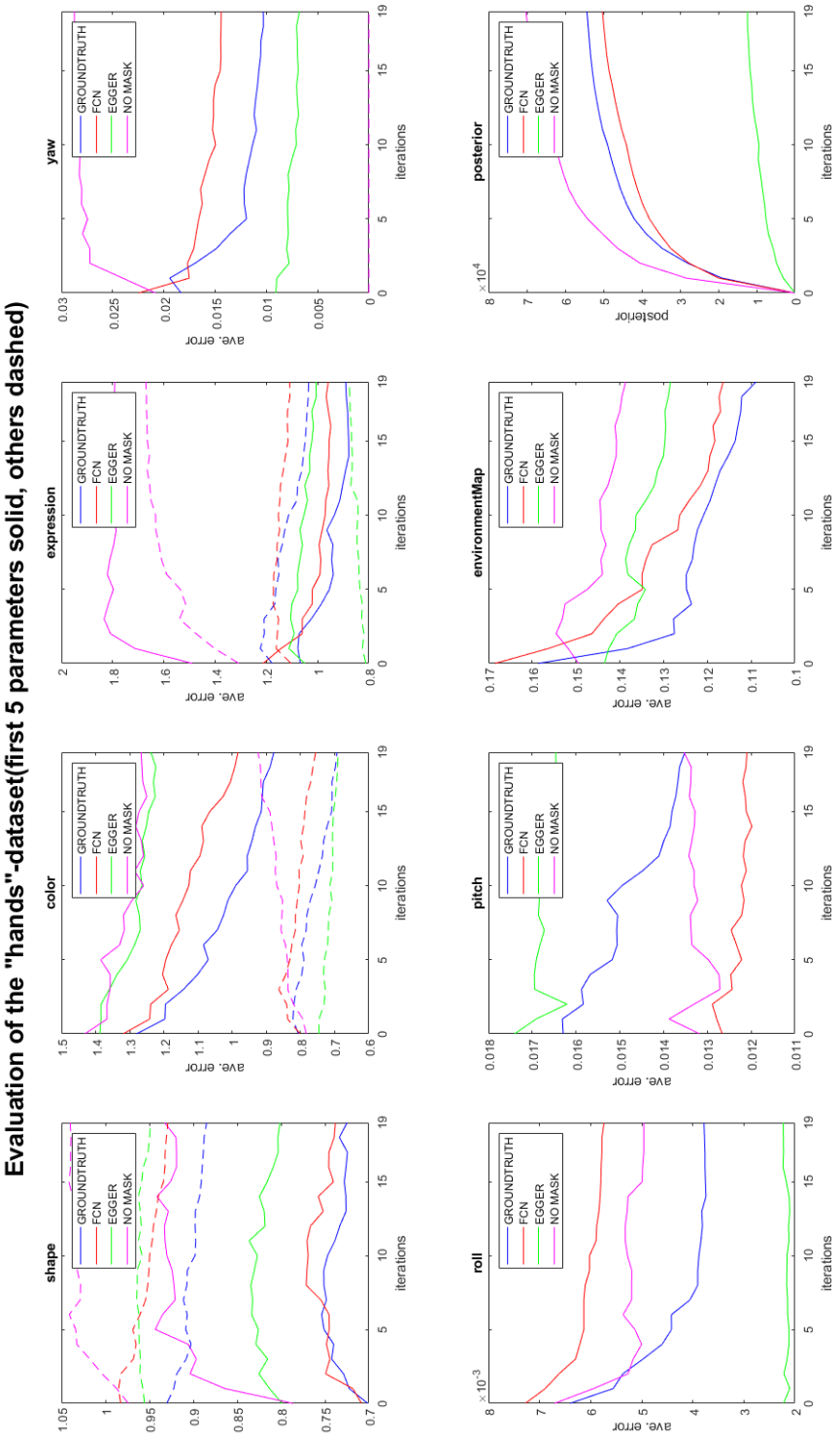


Figure 3.3: The first three plots show the errors of the first 50 shape, color and expression parameters of the Basel Face Model (in standard deviations). The next three show the errors of the Euler angles in radian. The 'EnvironmentMap'-plot shows the illumination chosen by all methods. In the last plot, we see the unnormalized posterior-probability, with which an Metropolis-Hastings sample was accepted.

3.3 Evaluation on the Original Face

In most cases, the fit with the FCN mask is not as different from the fit with the Egger mask itself as shown in [Figure 3.2]. However, we find that Egger's method tends to segment all skin pixels, whether they belong to the face or not. That's why in a next step we create faces with the 'bfm' version of the Basel Face Model. These faces are not tailored and include ears, hairline, and neck. We expect the segmentation of Egger et al to segment these areas too.

(a) original facial image (b) facial image with occlusion

Figure 3.4: These pictures are different from the ones in Figure 3.1 because we now use the 'bfm' version of the Basel Face Model which contains more skin pixels than the 'face12' version. The fitting algorithm takes 3.4(b) as input and tries to approximate 3.4(a).

	ground truth mask	Egger	FCN	no mask
z-labels				
fits				

Figure 3.5: In the top row, the different mask can be seen. The segmentation of Egger is updated in every iteration of the fitting process and only the mask in the last step is given. It is obvious that Egger oversegments and labels too much as face. The bottom row shows the fits with the masks above.

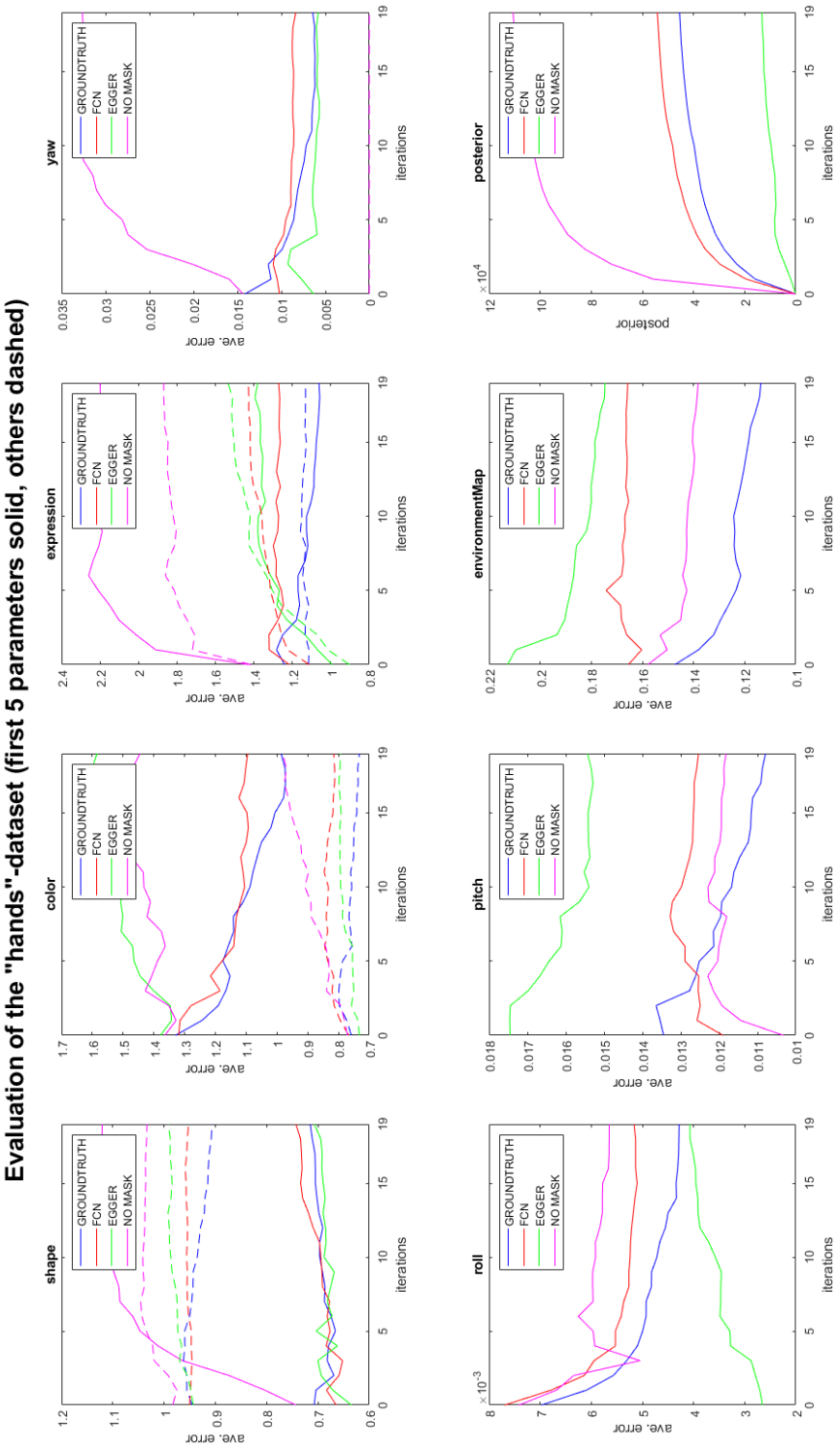


Figure 3.6: Due to the oversegmentation by Egger, a clear difference in the error of the color parameters can be seen. The mask of the FCN models the color significantly better.

From these plots, we see that the method of Egger et al oversegments the synthetic face and segments regions which are not visible on the final fits (the 'face12' version of the Basel Face Model is used for fitting). Since these regions are also evaluated and have a slightly different colour from the rest of the face, the final colour is wrong. This is easy to see in the error-plots for the colour parameter (second plot) where the spread between the fit with the FCN mask and the one with the Egger mask is approximately 0.5 standard deviations.

3.4 Runtime

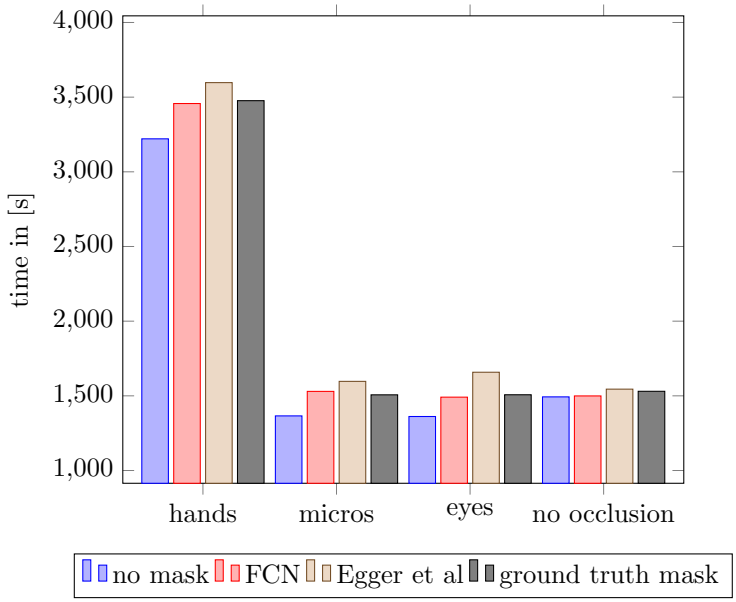


Figure 3.7: Comparison of the Wall-Clock Time for the fitting process. The times were measured with the tailored 'face12' mask and the 'bfm' rendering.

Since we evaluated our experiments on a compute-server where we had no control of the priority of the process, we can't make a general statement about the absolute duration. Within a dataset, all fits for the different segmentations were computed simultaneously. From this, we can conclude that the fit with the FCN mask tends to require less time especially if the face is occluded by objects which can be recognised by the FCN.

However, it is very difficult to make a general statement about the runtime behavior, because we cannot control whether the algorithm writes results in the cache and just loads them when desired, or computes them. As already mentioned, the Metropolis-Hastings algorithm proposes a randomly chosen sample to be the next one. If the evaluator rejects it, the algorithm uses the old sample as the next one and just has to load it from the cache.

4

Integration of the FCN into the original work of Egger et al

The aim of this chapter is to combine the advantages of the iterative segmentation method of Egger et al and of the segmentation by the FCN. In the previous chapters, we saw that both approaches have their strengths. Let us summarise them:

- Strengths of the segmentation of the FCN:
  - Although the borders of the segmentation are sometimes a bit fuzzy, almost only skin pixels are segmented (very few false positives).
  - As long as the face is not rotated (roll rotation), the FCN is very sturdy against variations in pitch and yaw.
  - Even if the occlusion has approximately the same colour as the face, the FCN may recognises the occlusion [Figure 3.2].
- Strengths of the segmentation of Egger et al
  - It is an iterative algorithm, the segmentation is always updated, based on the ever-changing 3DMM fit.
  - The algorithm can find occlusions that are thin and small, and the borders of the segmentation are very sharp [Figure 4.2].
  - Egger et al use a beard prior to exclude beards from the segmentation [Figure 4.1].

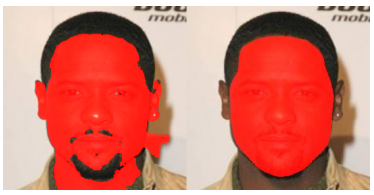


Figure 4.1: On the left-hand side, the segmentation of Egger which excludes the beard from the segmentation (but oversegments the face). The right-hand side shows the same target image overlaid with the segmentation of the FCN.

We combined both approaches so that the fitting algorithm of Egger et al takes the segmentation of the FCN as the initial mask and refines it during the following iterations. The biggest weakness of the FCN is its inability to detect small or thin occlusions [Figure 4.2].

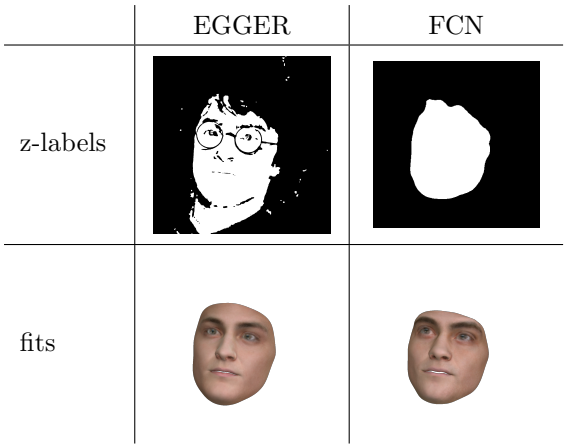


Figure 4.2: This example illustrates a major advantage and a disadvantage of Egger's approach. The FCN can't find the thin glasses but segments the eyes as face while Egger does not.

It is precisely this weakness of the FCN that is a strength of the approach of Egger et al. It is therefore very suitable to take the FCN's mask as a first segmentation, which contains almost only facial pixels and then refine this segmentation and search for thin occlusions by applying the occlusion-aware segmentation method of Egger to it.

However, our assumption is that the final segmentation does not differ much from the mask according to Egger et al alone because of the Metropolis-Hastings sampling in every iteration. The random walk always ends in the same region, regardless of the starting point.

We first show an example with a tailored (face12) target face [Figure 4.3 and Figure 4.4], then an example where the target face is a 'bfm'-Version face [Figure 4.6 and Figure 4.7].

Both experiments in this chapter show that the error of a fit with a combined mask is very close to the error of a fit with the iterative segmentation of Egger et al. It is interesting that the model parameters get a little better with the combined segmentation but the lighting gets worse in both cases.

4.1 Evaluation on the Tailored Face

In the following experiment, we use a 'face12'-version rendering of a face as input for the fit. The facial image which should be modelled is usually called the *target image*.

(a) initial mask of the FCN (b) segmentation of Egger et al (c) combined segmentation (d) fit with the combined mask (Subfigure (c))

Figure 4.3: An example of the combination of the two masks. [Figure 4.3(a)] shows the segmentation of the FCN while [Figure 4.3(b)] shows the segmentation by Egger et al. The combination is depicted in [Figure 4.3(c)]. The target image for this example is the same as in [Figure 3.1]. We see our suspicion that the initial segmentation only makes a tiny difference confirmed. [Figure 4.3(d)] shows the resulting fit with the combined mask.

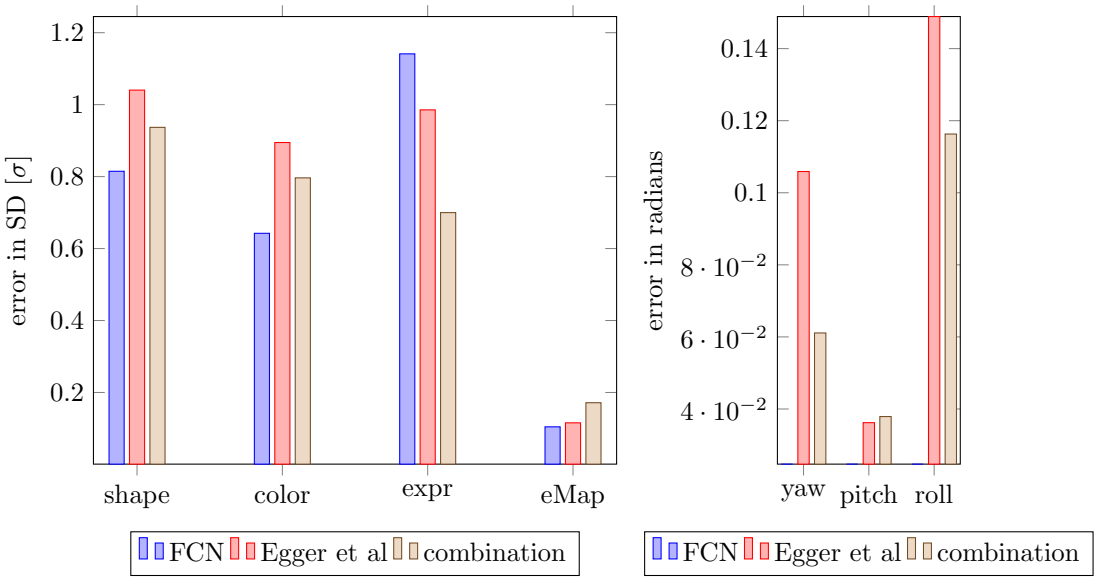
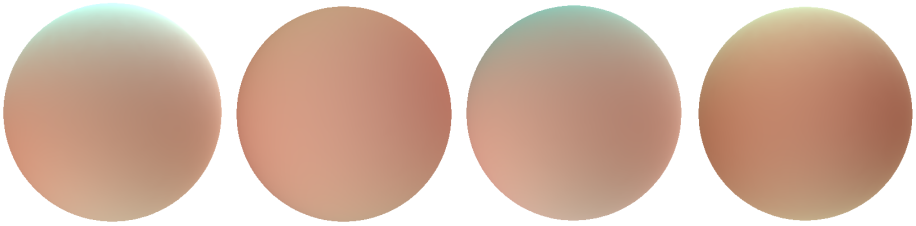


Figure 4.4: A comparison of the Basel Face Model parameters. In this plot, the fits with the masks of Egger, of the FCN [Figure 3.2], and with the combined mask [Figure 4.3(d)] are compared. Per parameter, only the first 5 dimensions are considered. In all considered Basel Face Model parameters (shape, color, and expression), the version with the combined mask has a lower error than the slower fit with the mask of Egger et al alone.

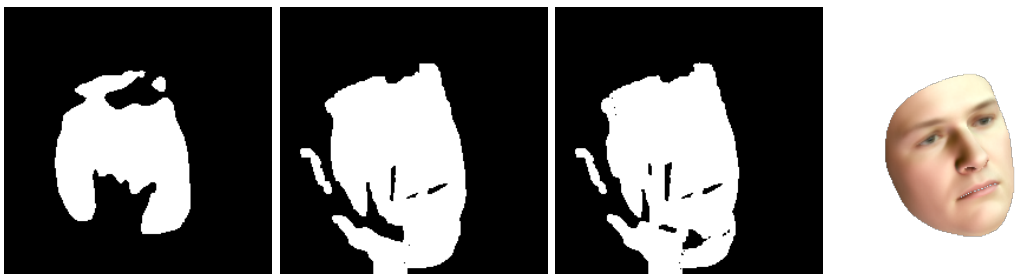


(a) Egger et al (b) FCN (c) combination (d) groundtruth

Figure 4.5: The illuminations of the last example rendered on a sphere. It seems like the illumination with the FCN mask is dull and has no specular term (only ambient). However, the illumination with the combined mask is strongly based on illumination (a).

4.2 Evaluation on the Original Face

Because we showed in Section 3.3 that the iterative algorithm of Egger et al tends to oversegment and labels skin-parts other than the face, we repeat the experiment with a rendering that shows ears, hairline, and neck too - the 'bfm' version of the Basel Face Model.



(a) initial mask of the FCN (b) segmentation of Egger et al (c) combined segmentation (d) fit with the combined mask (Subfigure (c))

Figure 4.6: An example of the combination of the two masks. [Figure 4.6(a)] shows the segmentation of the FCN while [Figure 4.6(b)] shows the segmentation by Egger et al. The combination is depicted in [Figure 4.6(c)]. The target image for this example is the same as in [Figure 3.4(b)]. [Figure 4.6(d)] shows the resulting fit with the combined mask.

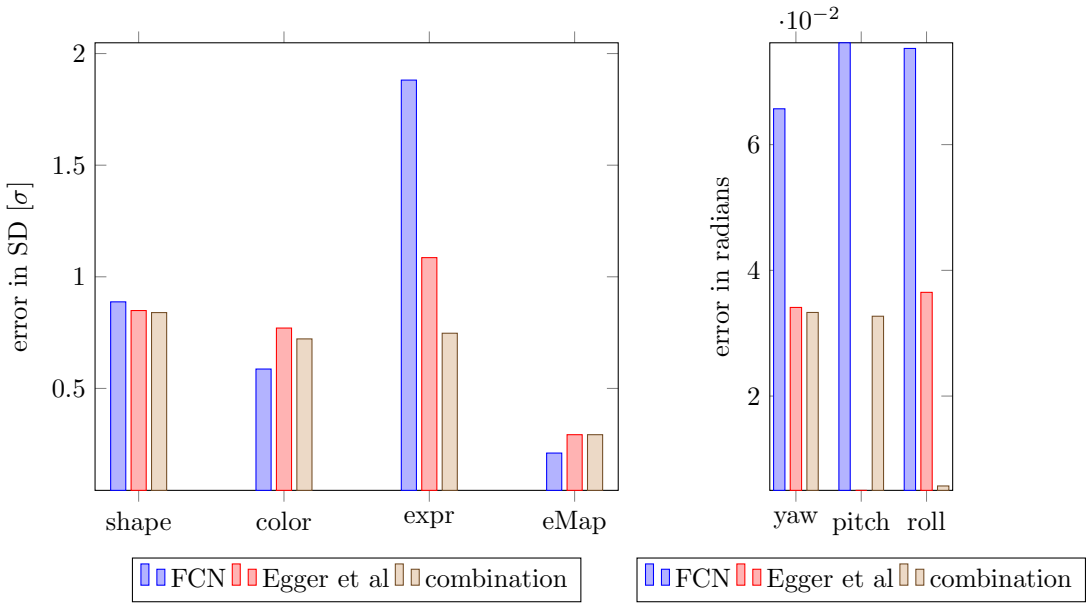
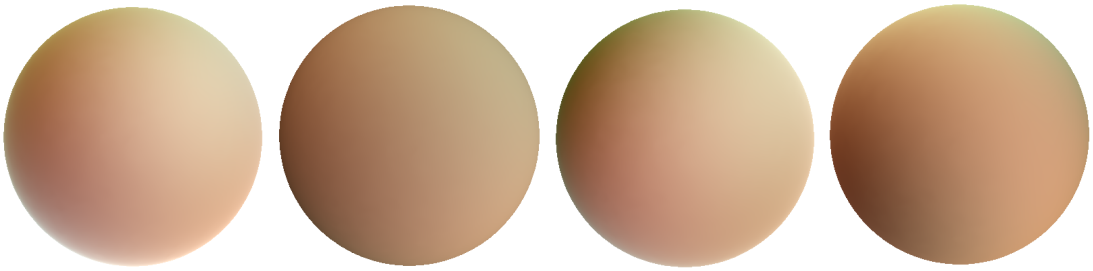


Figure 4.7: A comparison of the Basel Face Model parameters. In this plot, the fits with the masks of Egger, of the FCN [Figure 3.5], and the fit with the combined mask [Figure 4.3(d)] are compared. For each parameter, all dimensions are considered. With the combined segmentation, the fit is a tiny bit better than with the segmentation of Egger et al alone.



(a) Egger et al (b) FCN (c) combination (d) groundtruth

Figure 4.8: The illuminations of the last example rendered on a sphere. It seems like (as in the previous example) the illumination with the FCN mask is dull and has no specular term (only ambient). Again, the illumination with the combined mask is strongly based on illumination (a).

5 Conclusion

This thesis tested and analysed a fully convolutional network (FCN) for segmenting facial images under occlusion. The used network was trained by Nirkin et al [1] on a rich and diverse dataset. They claim that the speed and the accuracy of this segmentation method outperforms other approaches that were made especially for this task.

We evaluated the segmentation accuracy of the network on two real-life datasets. In one of them, every image was supplied with the ground truth labels, which determine whether a pixel belongs to the face or not. A comparison of both segmentations showed that the segmentation of the FCN contained very few false positives. However, the FCN only recognises about 4/5 of the actual facial region (false negatives). In a further step, the network was evaluated on synthetic images in order to be in full control of all parameters and to be able to simulate every possible face. The results show that there is a hierarchy among the rotations that determine the orientation of the face. Regardless of whether the face is hidden or not, the FCN is the most vulnerable to large roll rotations, then pitch rotations, and least important are the yaw rotations. That is a strange result because no matter how the face is rolled, the information does not change. We expect that the reason for this is the training of the FCN.

Figure 5.1: Three examples of training images used by Nirkin et al. These are pictures of the recent Janus CS2 dataset by Klare et al [6]. It looks as if the face had not been turned on any of the images in the whole dataset. The first two images are overlaid with synthetic occlusions. The third one depicts the interface used for semi-supervised labeling.

Egger et al [2] propose an EM-like method to simultaneously segment a face out of a given image and reconstruct it. We compare the segmentations of Egger et al with the segmentations of the FCN and find that: [1] The approach of Egger et al oversegments the image. [2] The method of Egger et al tends to exclude important details like the eyes due to their strong variability in color and shape. [3] The FCN often fails in recognising and segmenting thin occlusions which is possible with the approach of Egger et al.

The runtime of the FCN is much faster compared with the iterative segmentation of Egger et al. On our GPU, the segmentation with the FCN takes approximately 3 seconds, while the algorithm of Egger et al takes 2 minutes (according to the paper [2] of Egger et al).

(a) original (b) segmentation of Egger et al (c) segmentation of FCN

Figure 5.2: Comparison of the two segmentations ((b) and (c)) of the same facial image (a).

It is difficult to say which mask is better. In order to compare these masks in numbers, both segmentations are used as a mask for fitting synthetic face images, of which we know the correct 3DMM parameters. It turns out that due to the very few false positives of the FCN segmentation, the fit in many cases gets better (in 3DMM parameters) than that with the mask of Egger et al. On the other hand, real facial images are obscured by a variety of objects (not just microphones, hands, and glasses). The method of Egger can recognise these and therefore their mask often leads to better fits on real-world data than with the mask of the FCN [Appendix A.2].

5.1 Future Work

The Integration in Chapter 4 was not optimal, because the resulting mask came very close to the one estimated by Egger et al itself. Our setting executed the FCN and then let the combined segmentation and fitting algorithm update the mask in all 20 iterations. It would be interesting to see what happens if the algorithm of Egger et al could only improve the initial FCN-Segmentation in a few last iterations. It is strange that the illumination estimation does not improve although an initial segmentation is used for this task (the original approach of egger et al does not use a mask/segmentation to determine the illumination parameters). This is an interesting starting point for further research on how to optimally combine both segmentations.

Bibliography

[1] Nirkin, Y., Masi, I., Tran, A. T., Hassner, T., and Medioni, G. On Face Segmentation, Face Swapping, and Face Perception. In *IEEE Conference on Automatic Face and Gesture Recognition* (2018).

[2] Egger, B., Schönborn, S., Schneider, A., Kortylewski, A., Morel-Forster, A., Blumer, C., and Vetter, T. Occlusion-aware 3D Morphable Models and an Illumination Prior for Face Image Analysis (2018).

[3] McCulloch, W. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133 (1943).

[4] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, page 3431–3440 (2018).

[5] K. Simonyan, A. Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *arXiv:1409.1556* (2015).

[6] Klare, Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Burge, M., and Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 1931–1939. IEEE Computer Society (2015).

[7] Grundmann, M., Kwatra, V., Han, M., and Essa, I. Efficient hierarchical graph-based video segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition* (2010).

[8] Uricar, M., Franc, V., Thomas, D., Sugimoto, A., and Hlavac, V. Real-time multi-view facial landmark detector learned by the structured output SVM. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 02, pages 1–8 (2015).

[9] Saito, S., Li, T., and Li, H. Real-Time Facial Segmentation and Performance Capture from RGB Input. *CoRR*, abs/1604.02647 (2016). URL http://arxiv.org/abs/1604.02647. Last visited 2018-07-22.

[10] Morel-Forster, A. *Generative shape and image analysis by combining Gaussian processes and MCMC sampling*. Ph.D. thesis, University of Basel, Faculty of Science (2016).

[11] X. P. Burgos-Artizzu, P. P. and Dollár, P. Robust face landmark estimation under occlusion. CCV 2013, Sydney, Australia (2013).

[12] Kae, A., Sohn, K., Lee, H., and Learned-Miller, E. Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling. In *CVPR* (2013).

[13] Egger, B., Kortylewski, A., Morel-Forster, A., and Schneider, A. parametric-face-image-generator (2017).

[14] Gerig, T., Forster, A., Blumer, C., Egger, B., Lüthi, M., Schönborn, S., and Vetter, T. Morphable Face Models - An Open Framework. *CoRR*, abs/1709.08398 (2017). URL http://arxiv.org/abs/1709.08398. Last visited 2018-07-22.

[15] Blanz, V. and Vetter, T. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1999). URL http://dx.doi.org/10.1145/311535.311556. Last visited 2018-07-22.

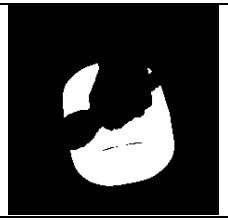
A
Appendix

A.1 COFW-Images and Fits

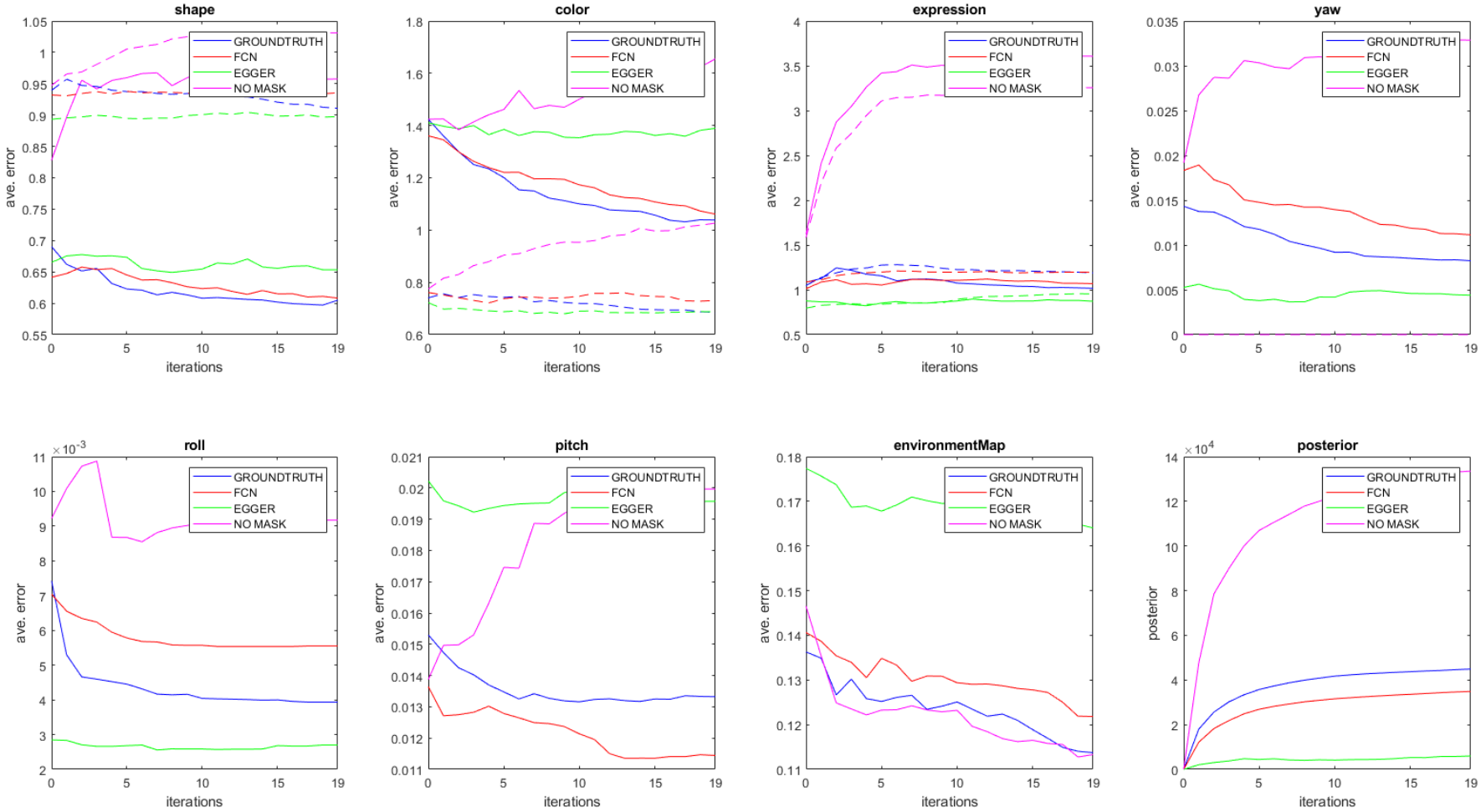
Figure A.1: The additional images of Figure 2.3. In each tuple the first plot shows the fit with the mask of egger et al. and the second plot is made with the segmentation of the FCN.

A.2 Datasets other than Hands(which are shown in the thesis)

micros (mask: face12, rendering: face12):

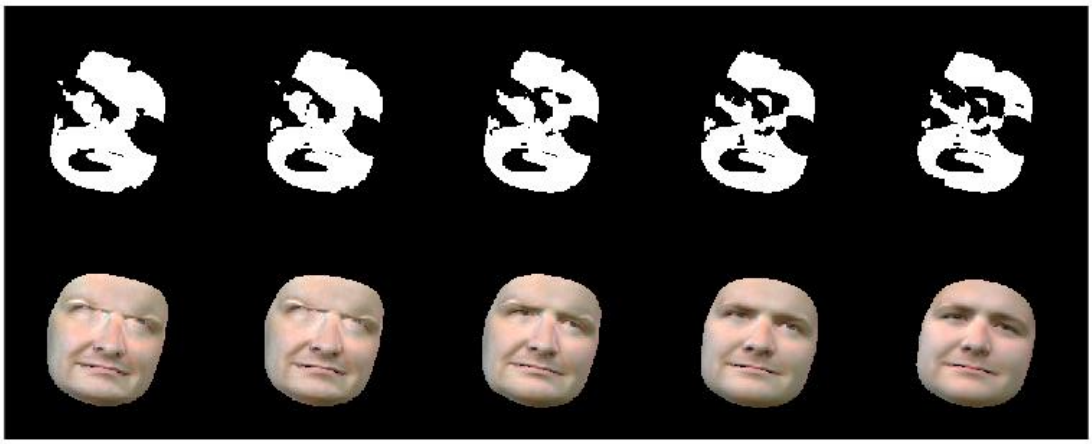
	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

Evaluation of the "micros"-dataset(first 5 parameters solid, others dashed)



segmentation and mask of test3 in every 5th iteration with mask: GROTRU(from right to left)

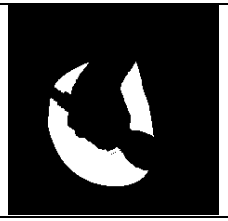
segmentation and mask of test3 in every 5th iteration with mask: EGGER(from right to left)



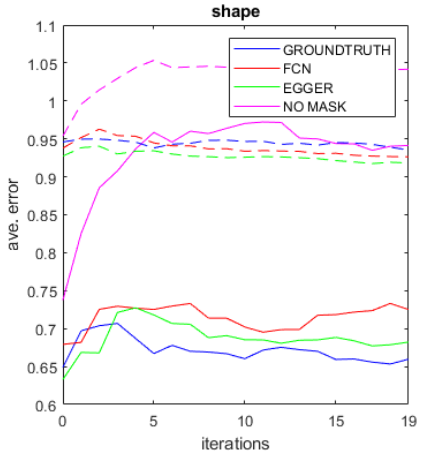
segmentation and mask of test3 in every 5th iteration with mask: FCN(from right to left)

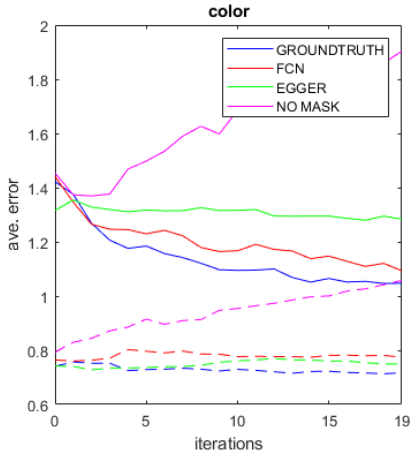
segmentation and mask of test3 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

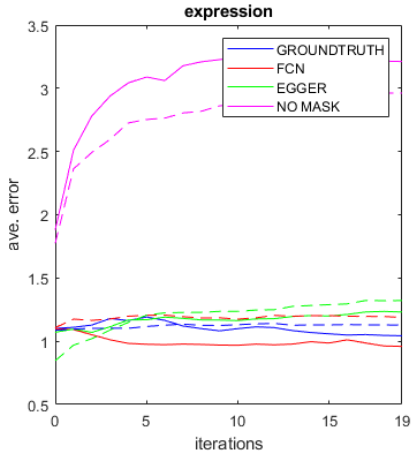
micros (mask: face12, rendering: bfm):

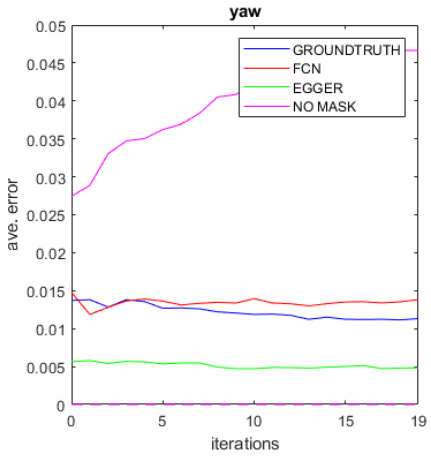
	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

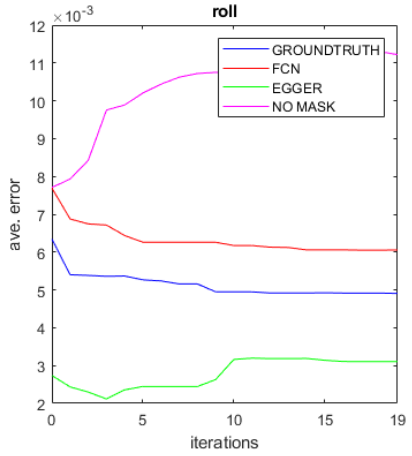
Evaluation of the "micros"-dataset(first 5 parameters solid, others dashed)

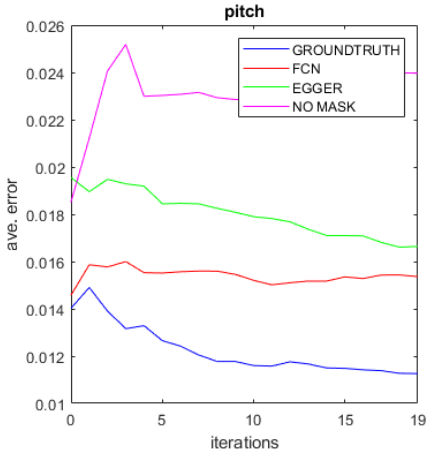


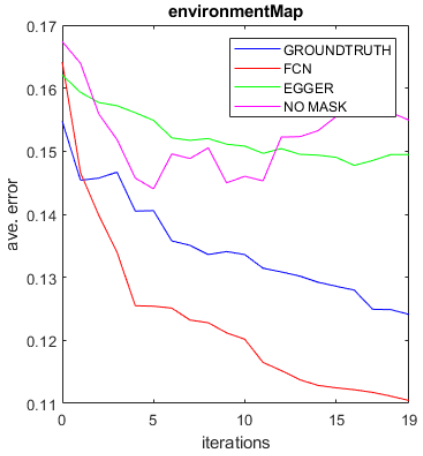


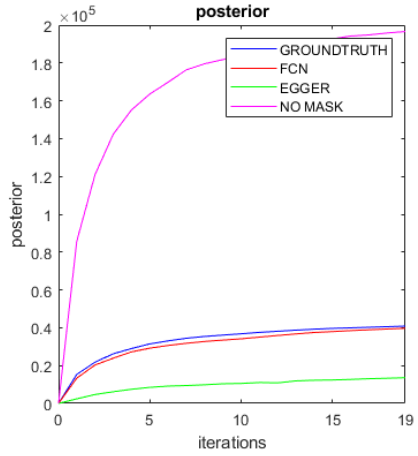








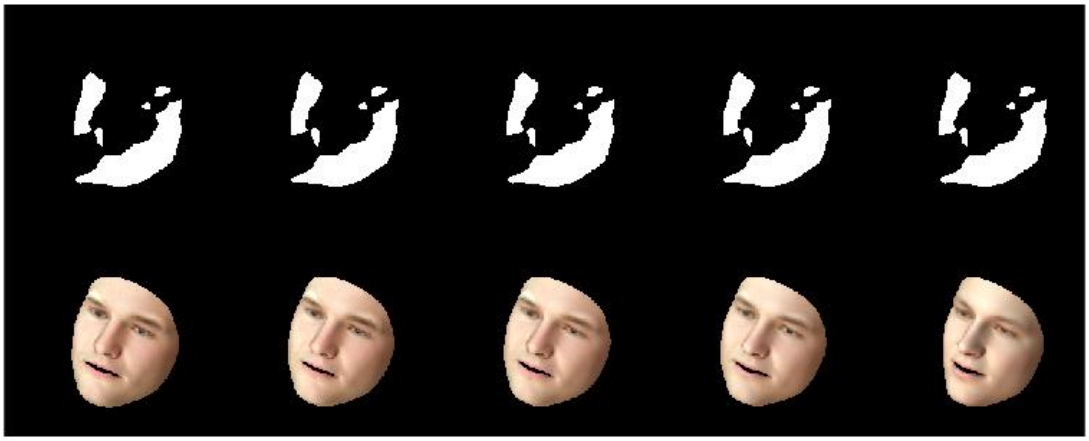




segmentation and mask of test0 in every 5th iteration with mask: GROTRU(from right to left)

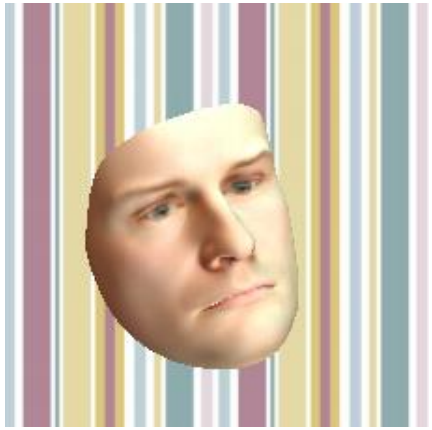
segmentation and mask of test0 in every 5th iteration with mask: EGGER(from right to left)

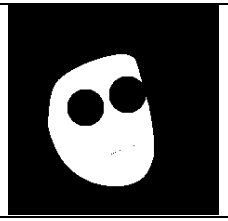
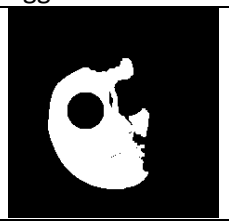
segmentation and mask of test0 in every 5th iteration with mask: FCN(from right to left)



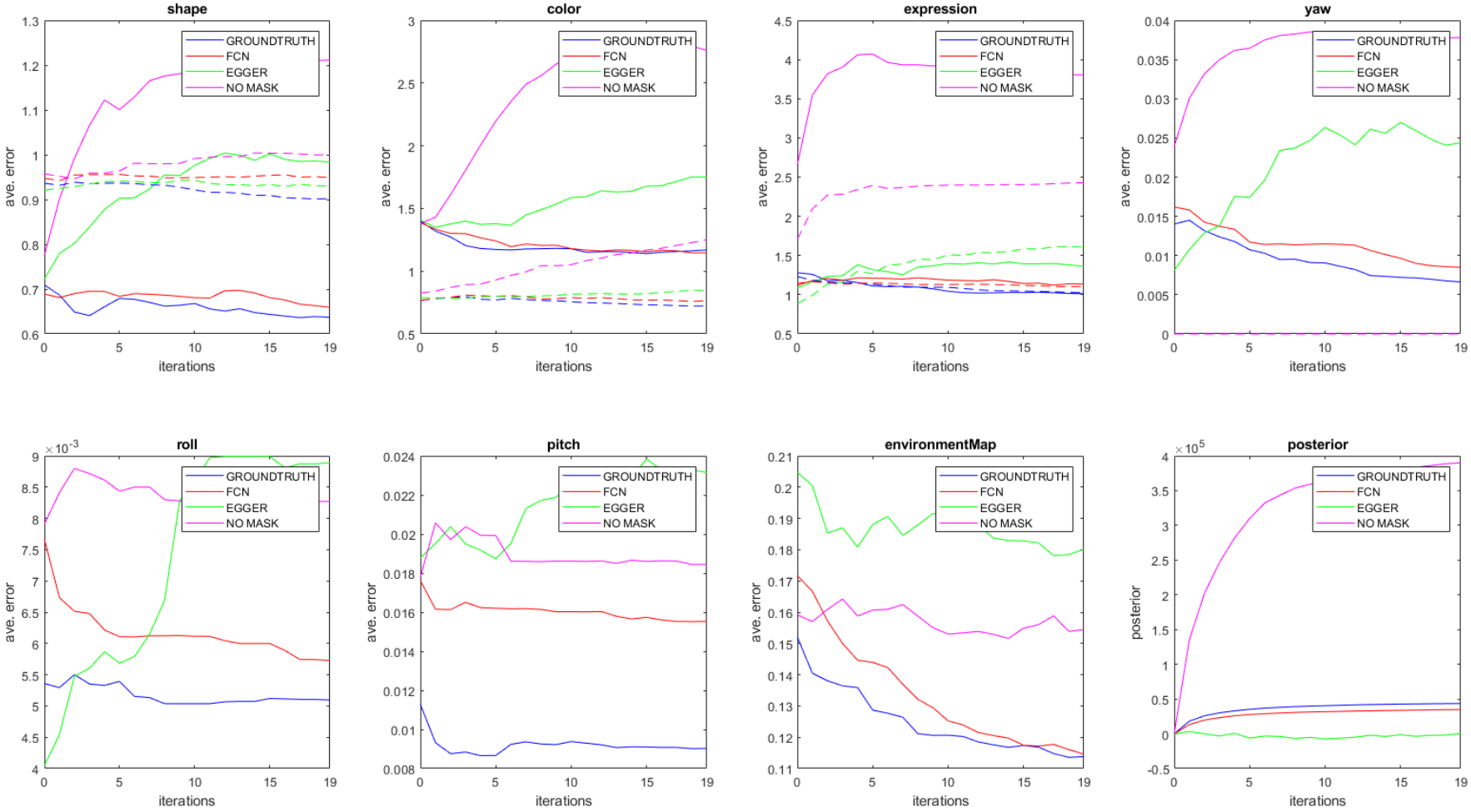
segmentation and mask of test0 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

glasses (mask: face12, rendering: face12):

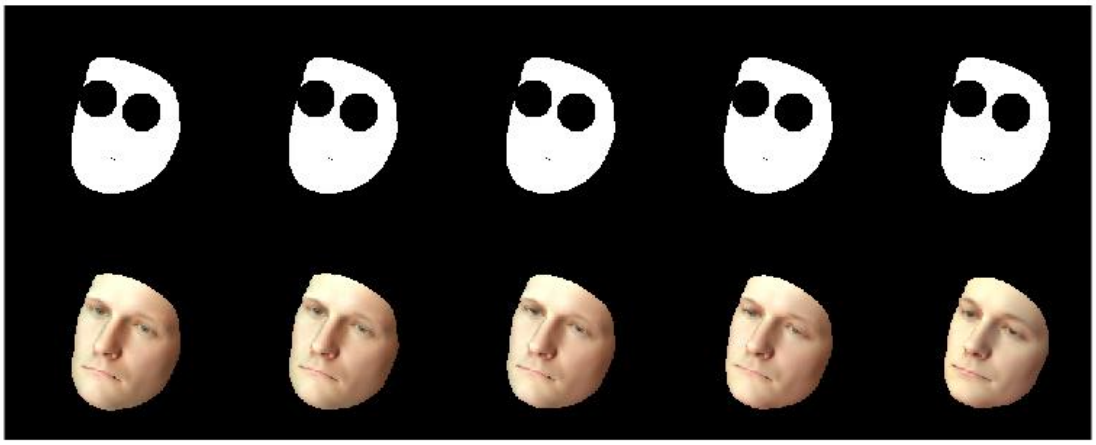


	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

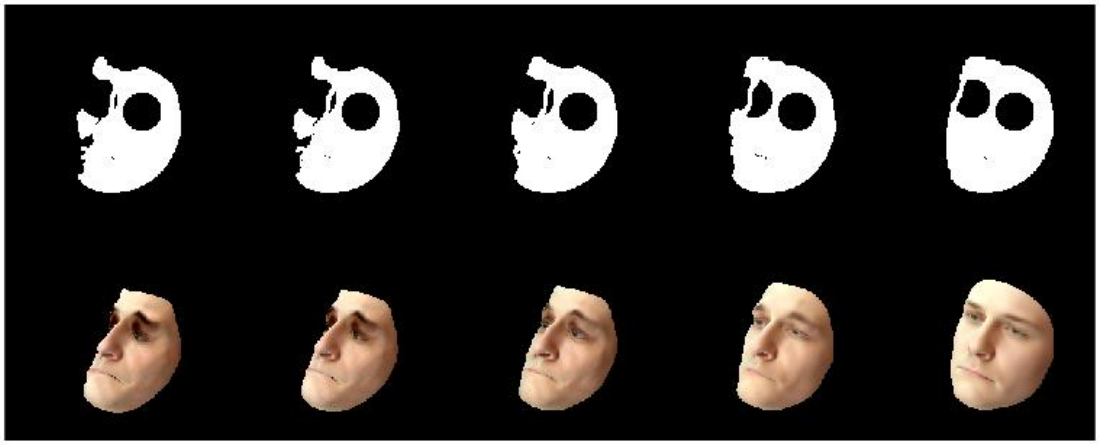
Evaluation of the "glasses"-dataset(first 5 parameters solid, others dashed)



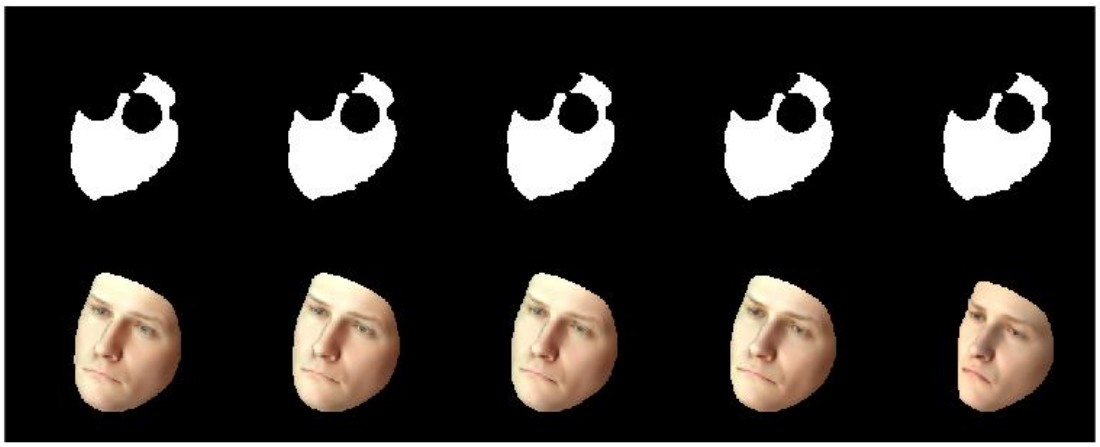
segmentation and mask of test0 in every 5th iteration with mask: GROTRU(from right to left)



segmentation and mask of test0 in every 5th iteration with mask: EGGER(from right to left)

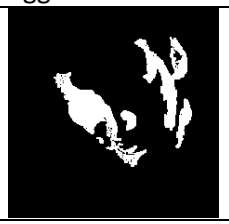


segmentation and mask of test0 in every 5th iteration with mask: FCN(from right to left)

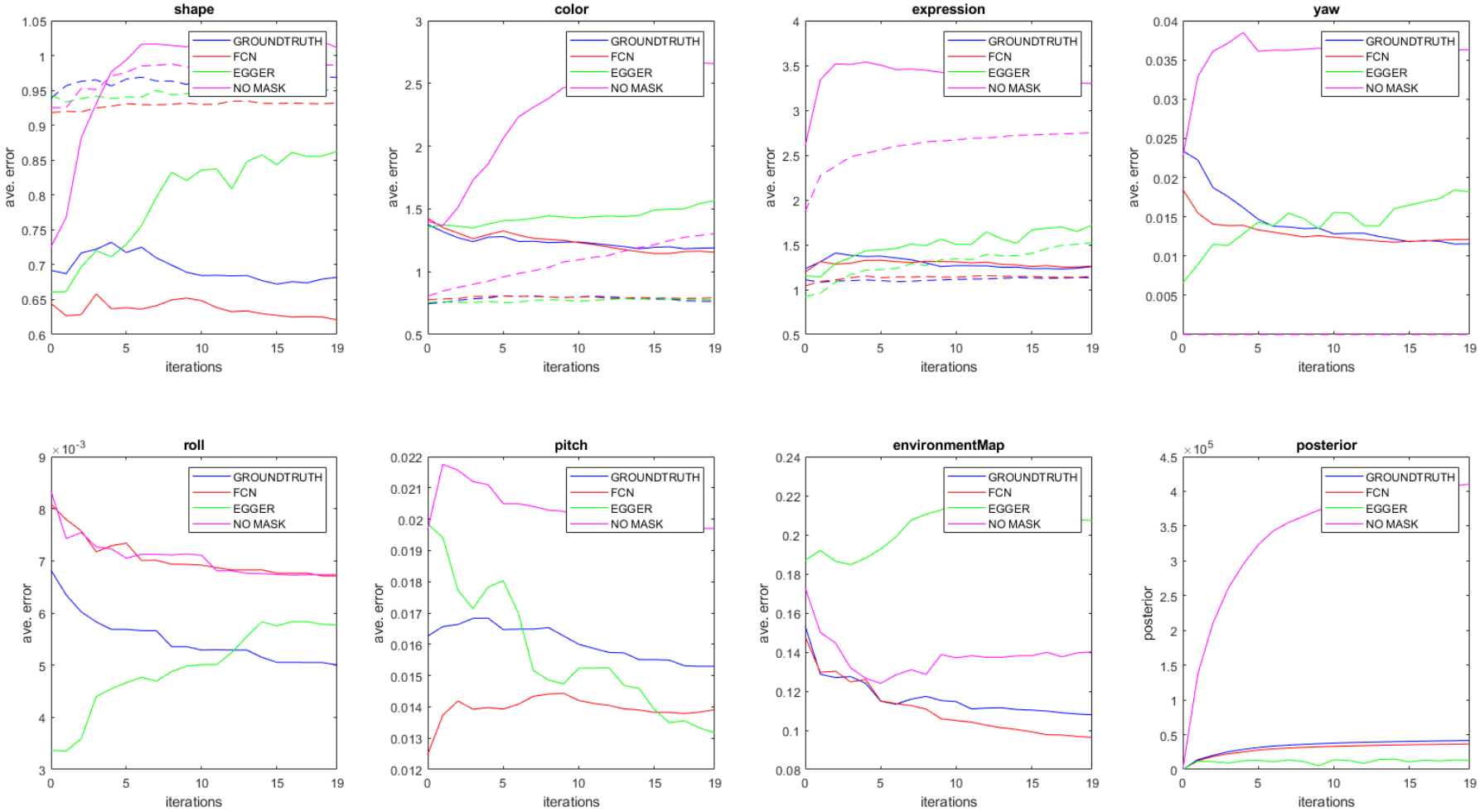


segmentation and mask of test0 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

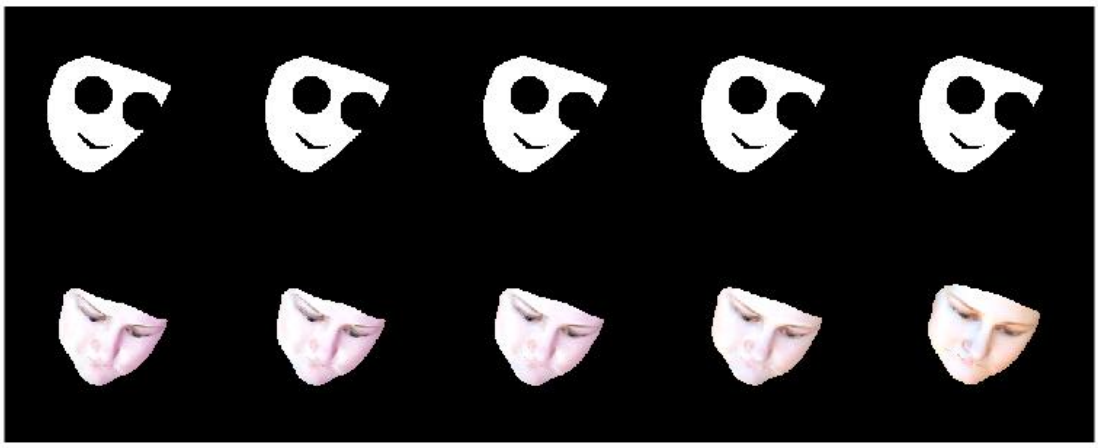
glasses (mask: face12, rendering: bfm):

	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

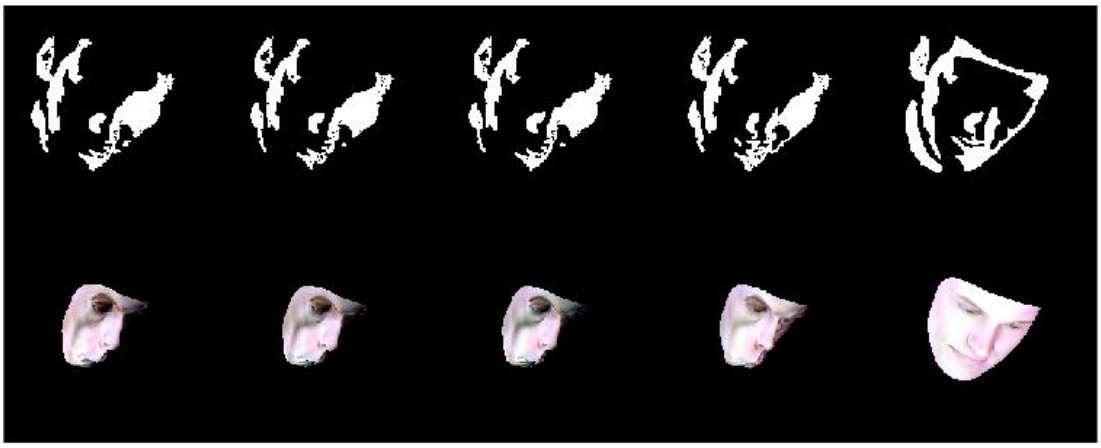
Evaluation of the "glasses"-dataset (first 5 parameters solid, others dashed)



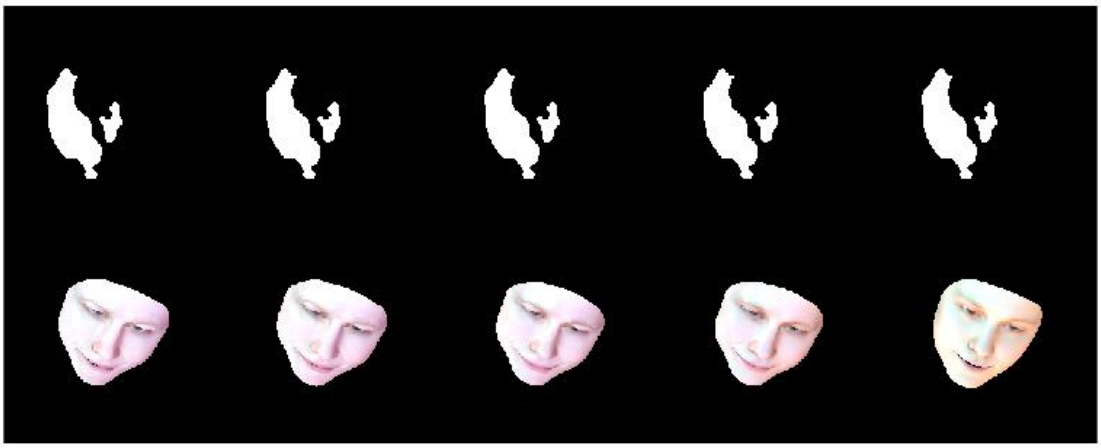
segmentation and mask of test2 in every 5th iteration with mask: GROTRU(from right to left)



segmentation and mask of test2 in every 5th iteration with mask: EGGER(from right to left)

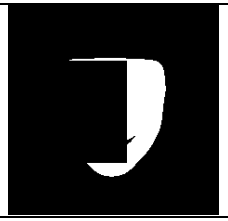
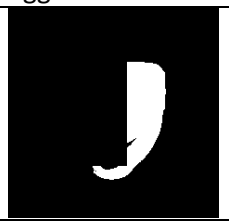
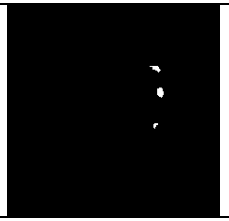


segmentation and mask of test2 in every 5th iteration with mask: FCN(from right to left)

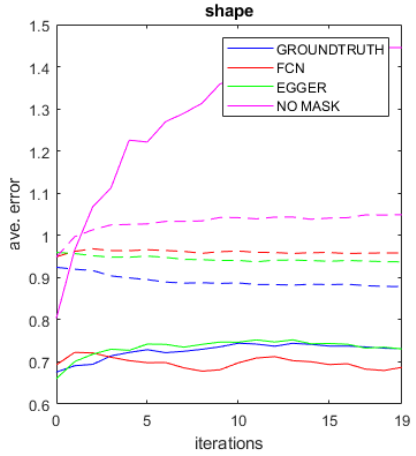


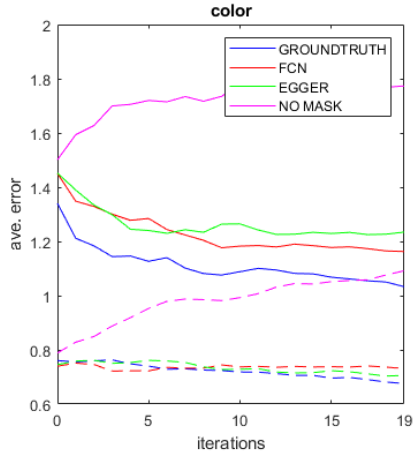
segmentation and mask of test2 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

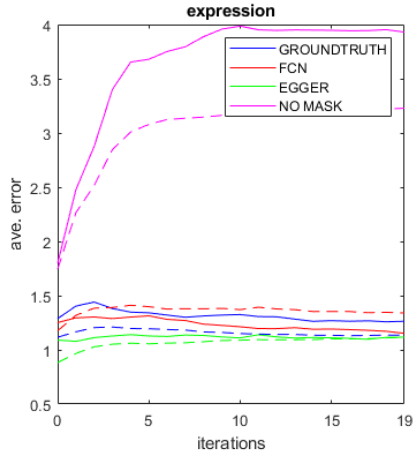
random boxes (mask: face12, rendering: face12):

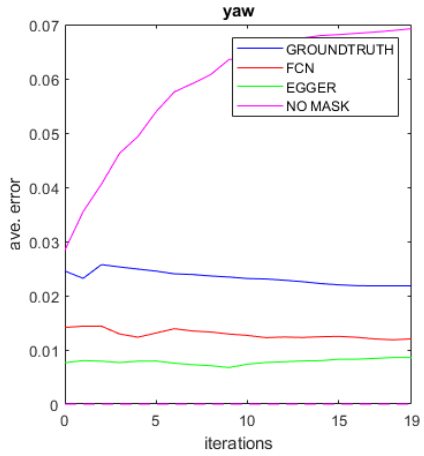
	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

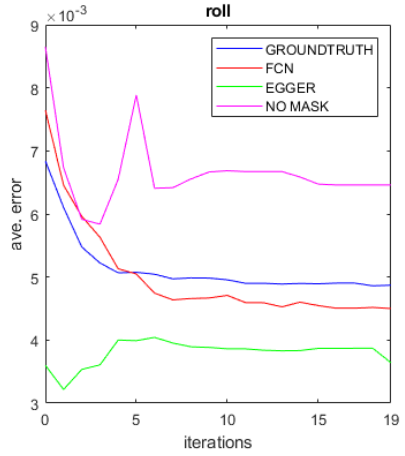
Evaluation of the "random boxes"-dataset(first 5 parameters solid, others dashed)

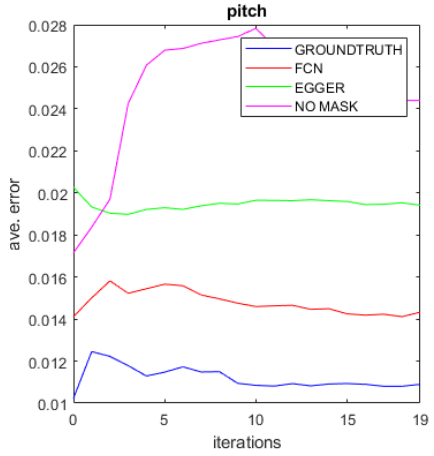


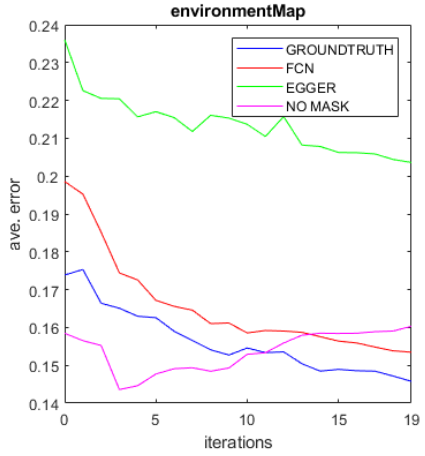


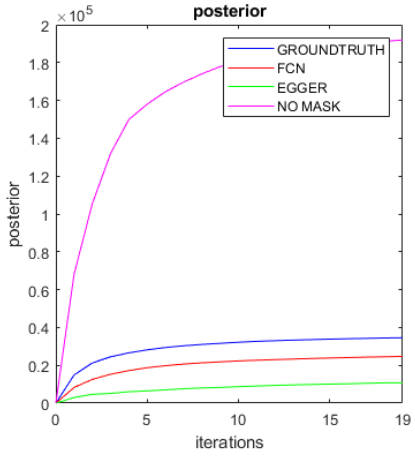




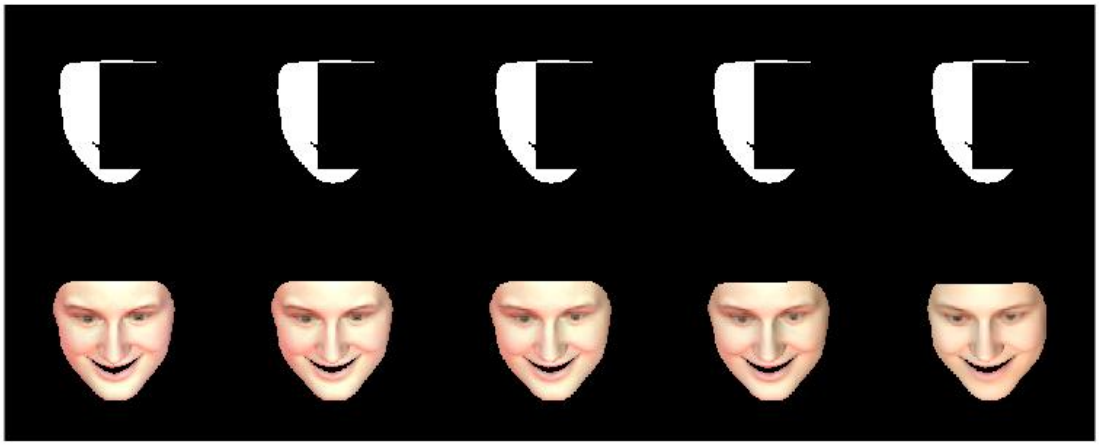








segmentation and mask of test1 in every 5th iteration with mask: GROTRU(from right to left)

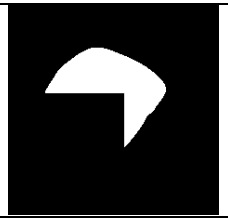
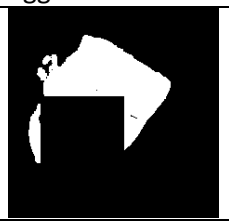
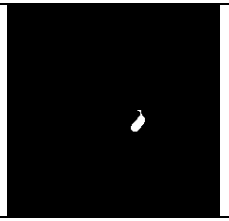


segmentation and mask of test1 in every 5th iteration with mask: EGGER(from right to left)

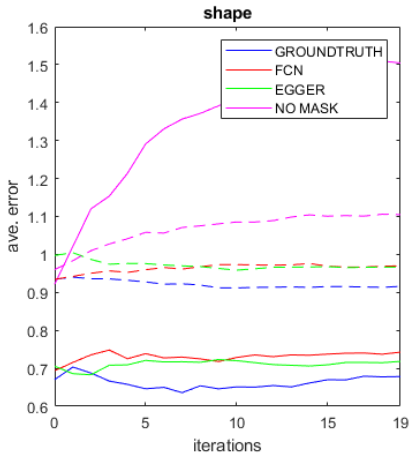
segmentation and mask of test1 in every 5th iteration with mask: FCN(from right to left)

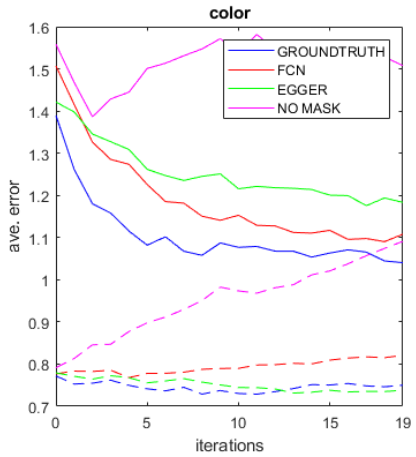
segmentation and mask of test1 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

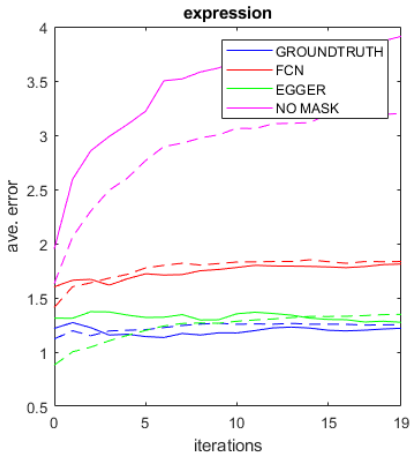
random boxes (mask: face12, rendering: bfm):

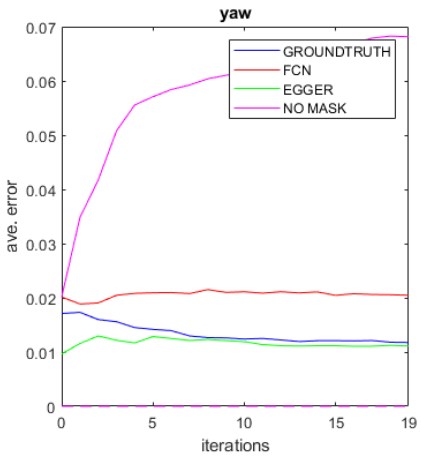
	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

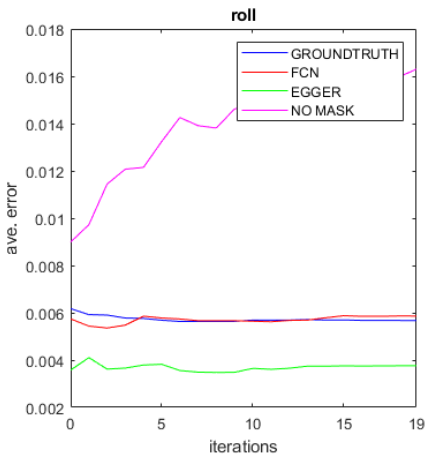
Evaluation of the "random boxes"-dataset(first 5 parameters solid, others dashed)

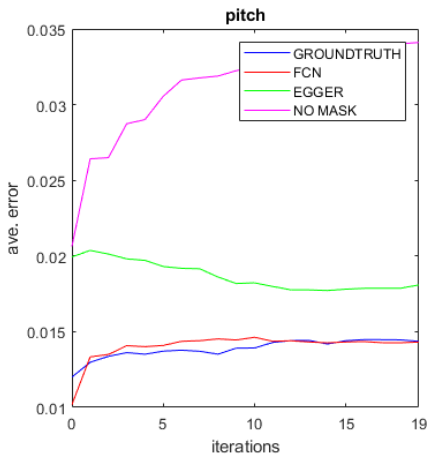


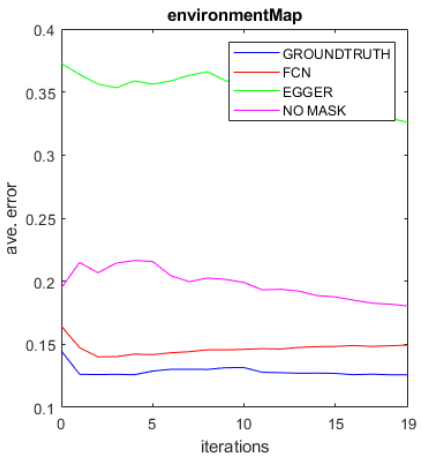


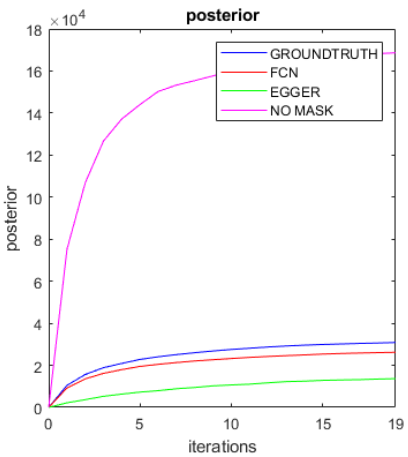




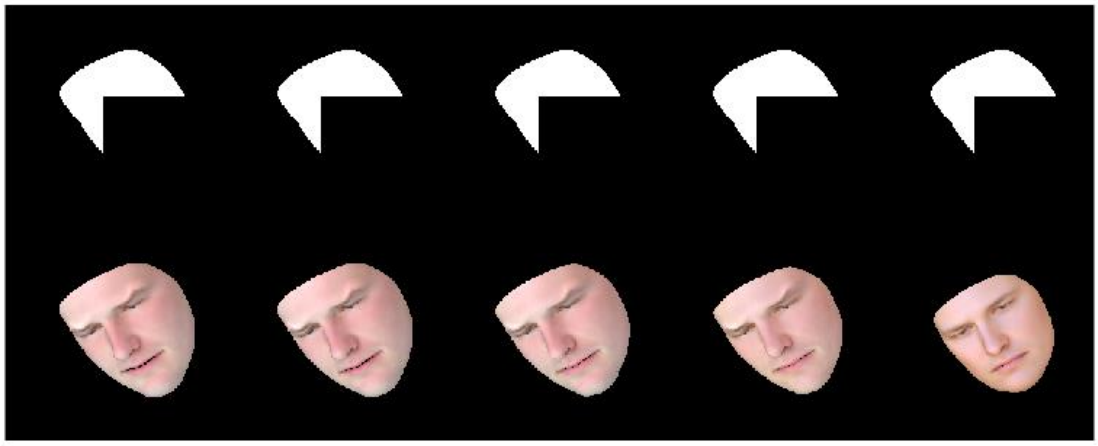






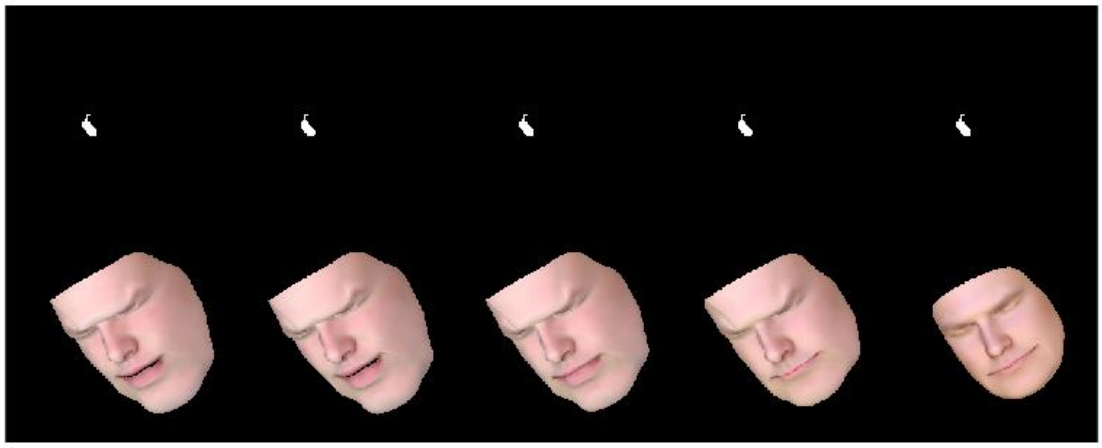


segmentation and mask of test4 in every 5th iteration with mask: GROTRU(from right to left)



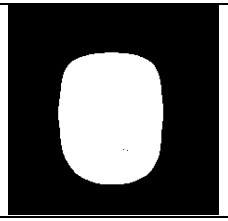
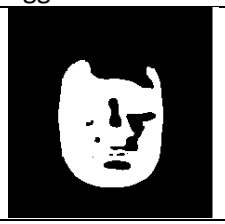
segmentation and mask of test4 in every 5th iteration with mask: EGGER(from right to left)

segmentation and mask of test4 in every 5th iteration with mask: FCN(from right to left)

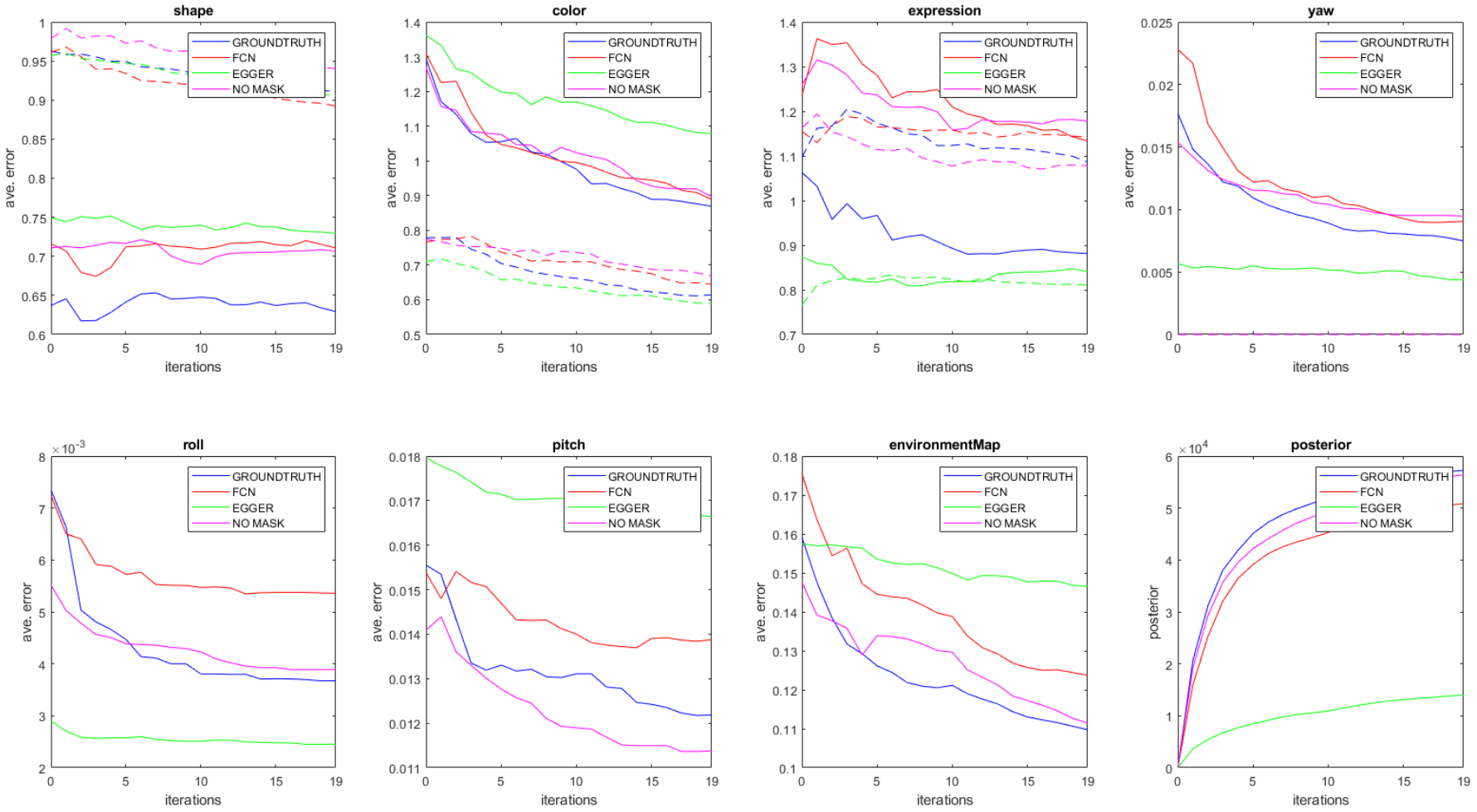


segmentation and mask of test4 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

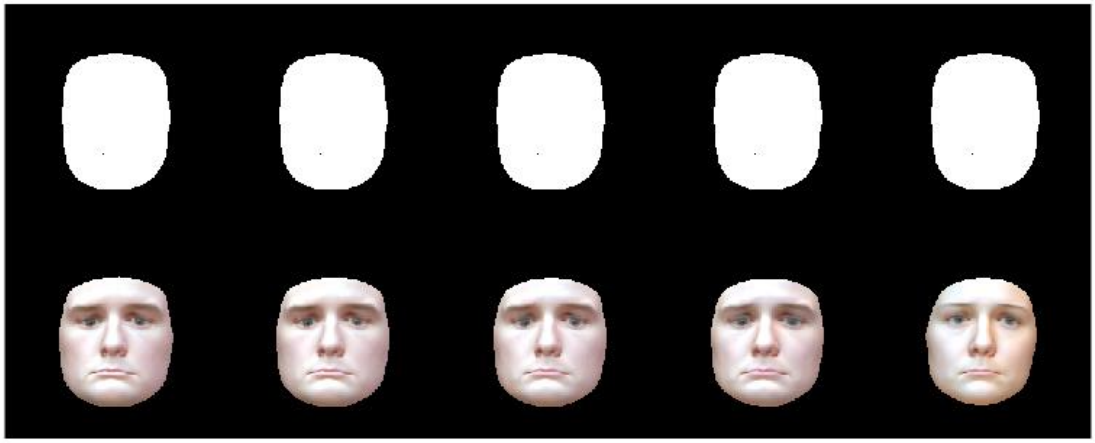
no occlusion (mask: face12, rendering: face12):

	Groundtruth	Egger	FCN	No mask
Mask				
Fit				

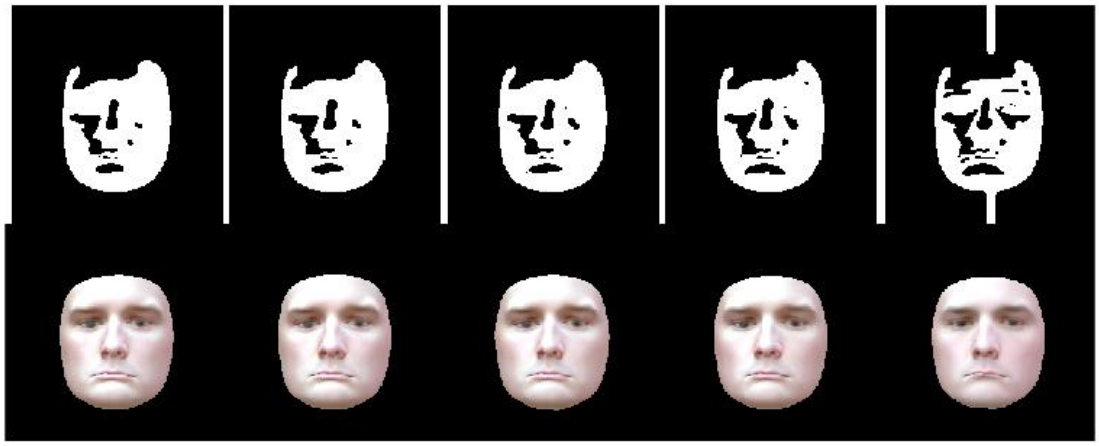
Evaluation of the "no occlusions"-dataset(first 5 parameters solid, others dashed)



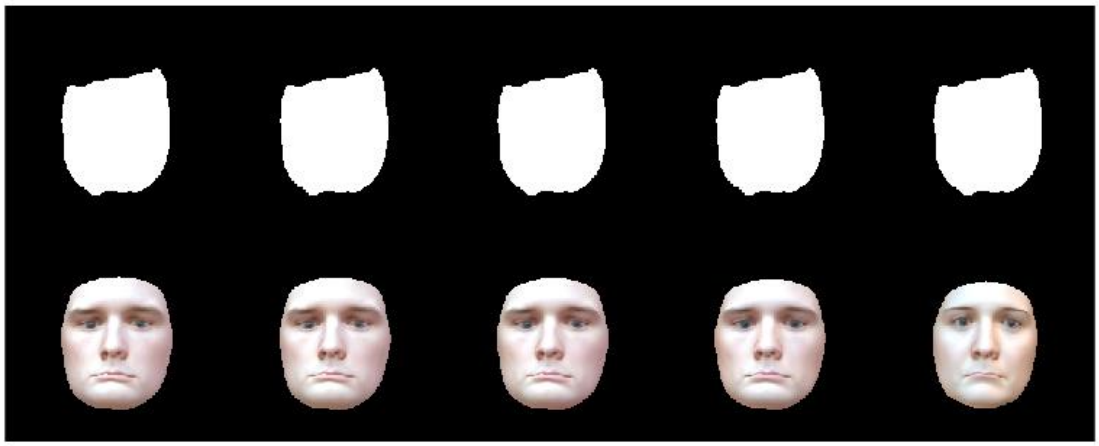
segmentation and mask of test7 in every 5th iteration with mask: GROTRU(from right to left)



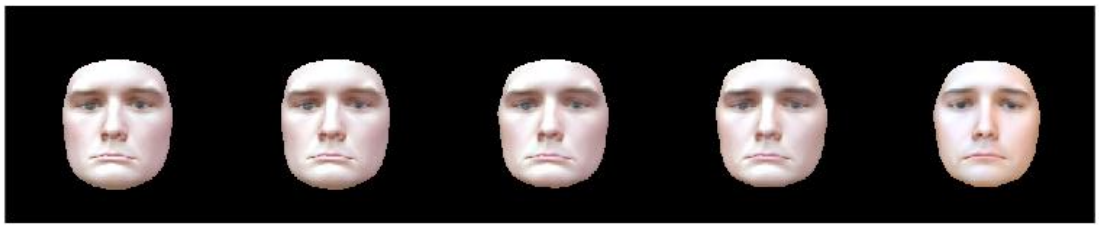
segmentation and mask of test7 in every 5th iteration with mask: EGGER(from right to left)



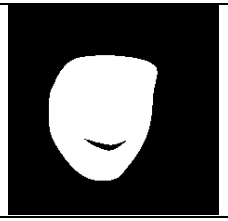
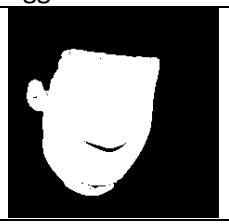
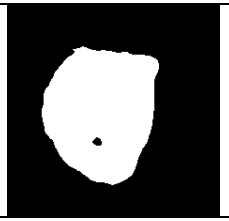
segmentation and mask of test7 in every 5th iteration with mask: FCN(from right to left)

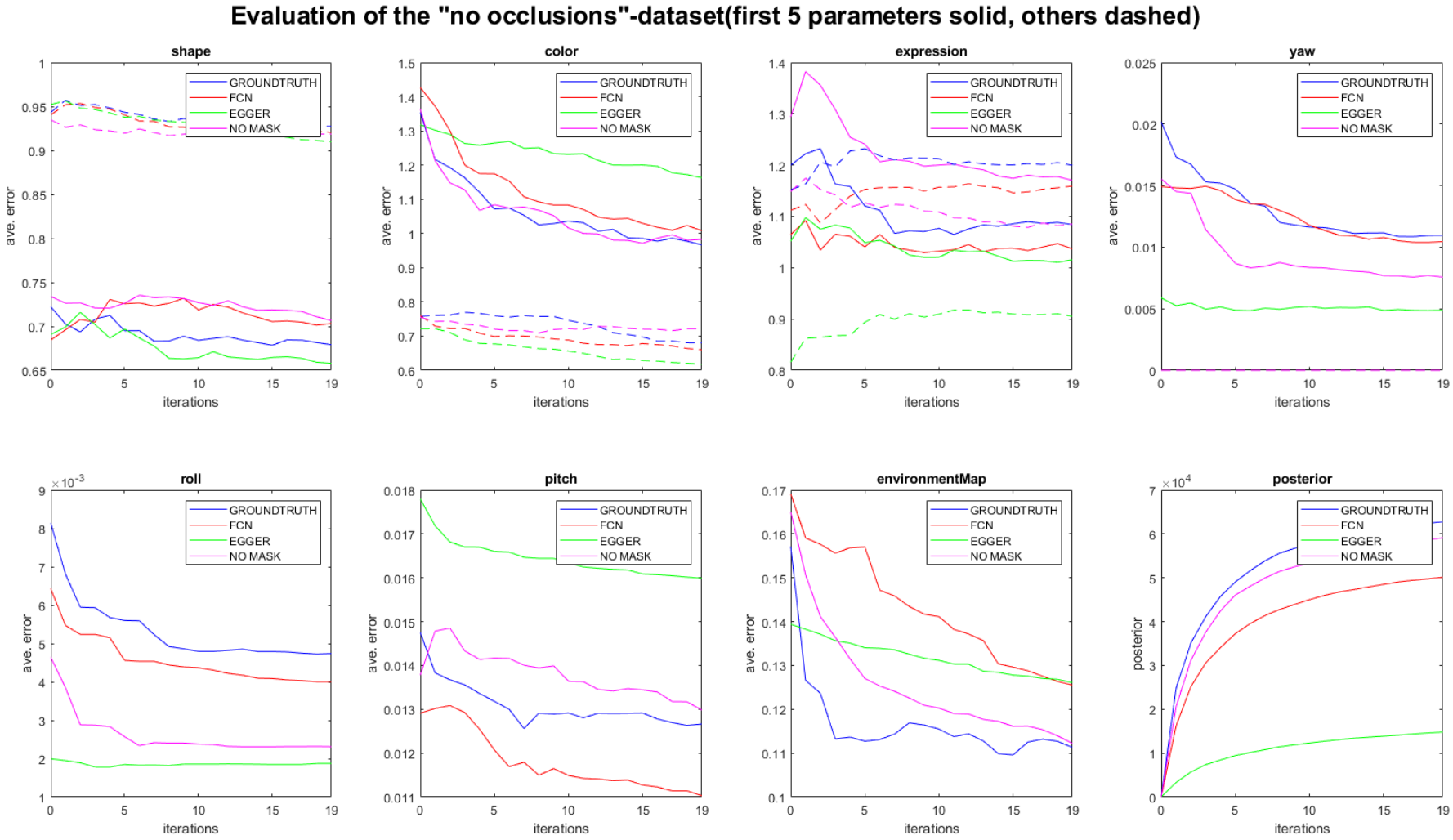


segmentation and mask of test7 in every 5th iteration with mask: NO_OCCLUSION(from right to left)

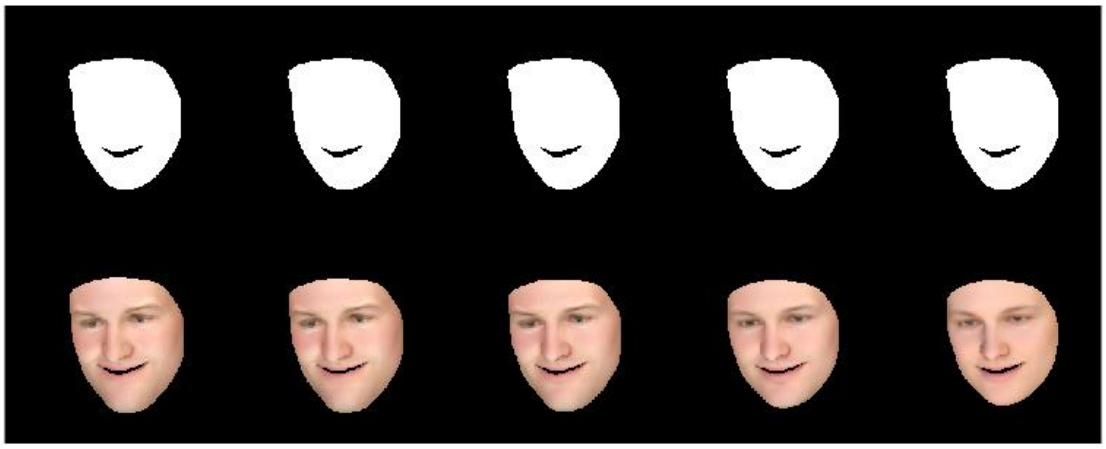


no occlusion (mask: face12, rendering: bfm):

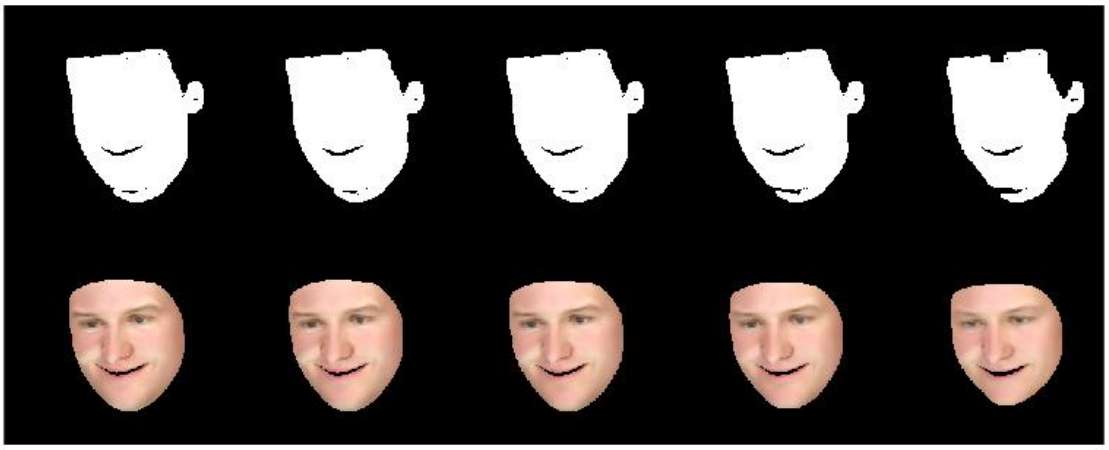
	Groundtruth	Egger	FCN	No mask
Mask				
Fit				



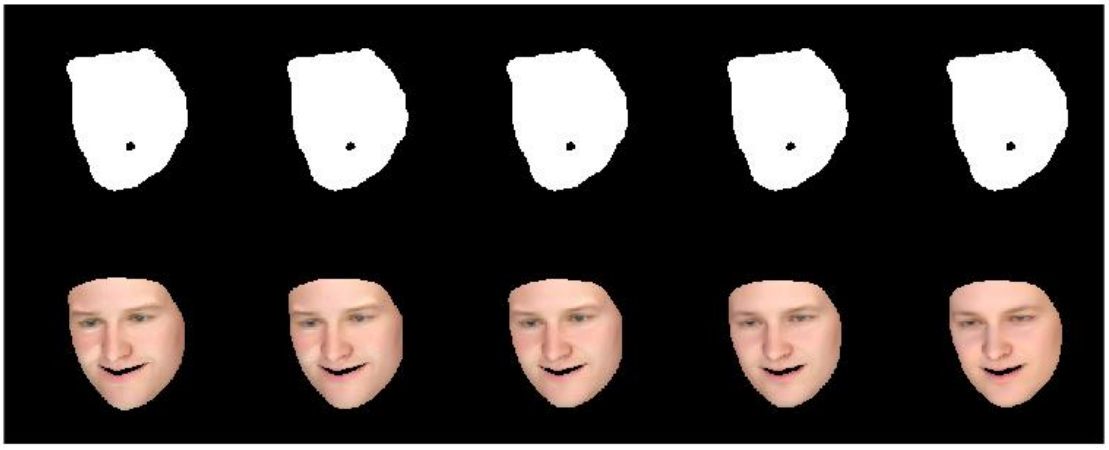
segmentation and mask of test6 in every 5th iteration with mask: GROTRU(from right to left)



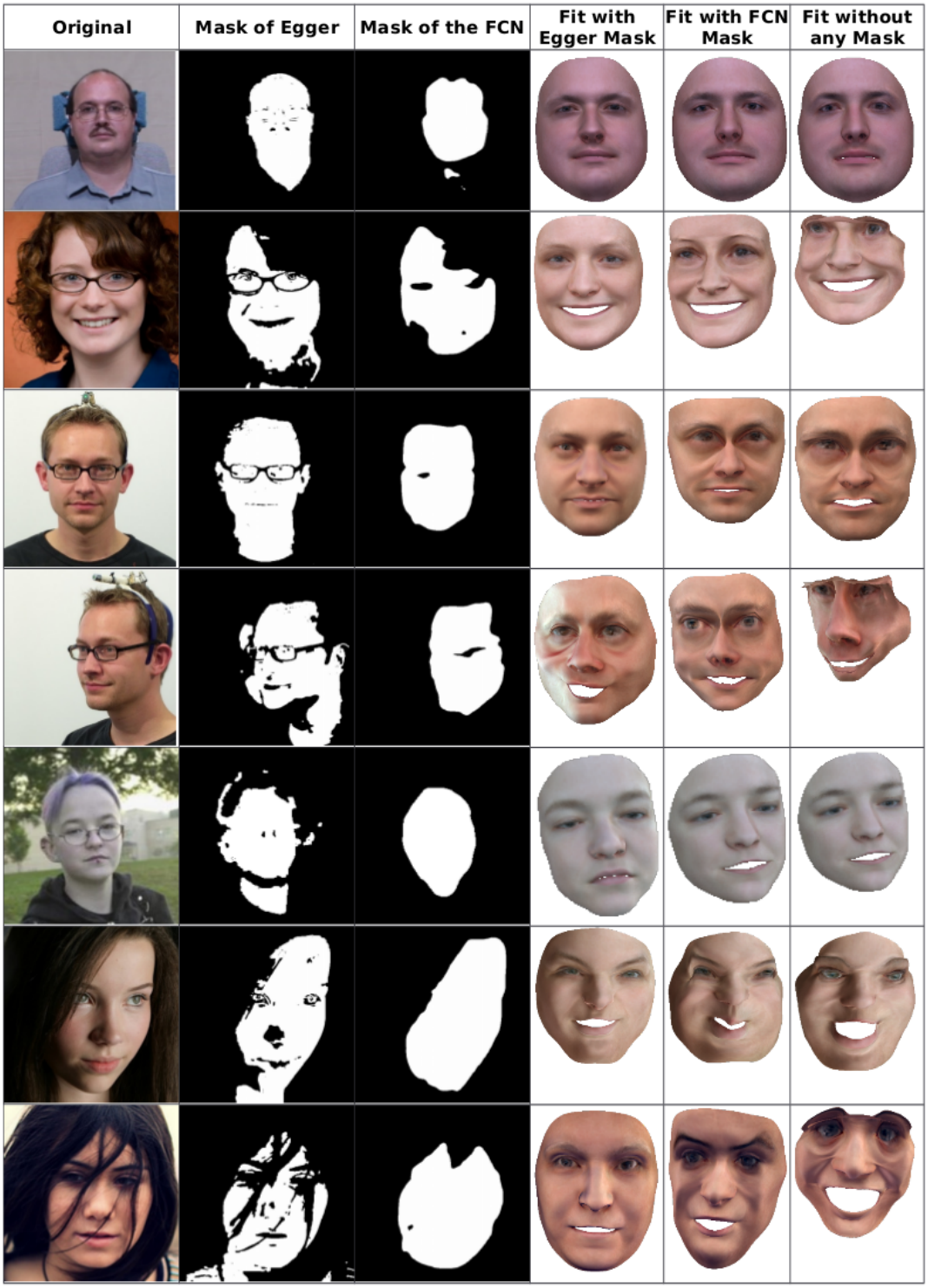
segmentation and mask of test6 in every 5th iteration with mask: EGGER(from right to left)

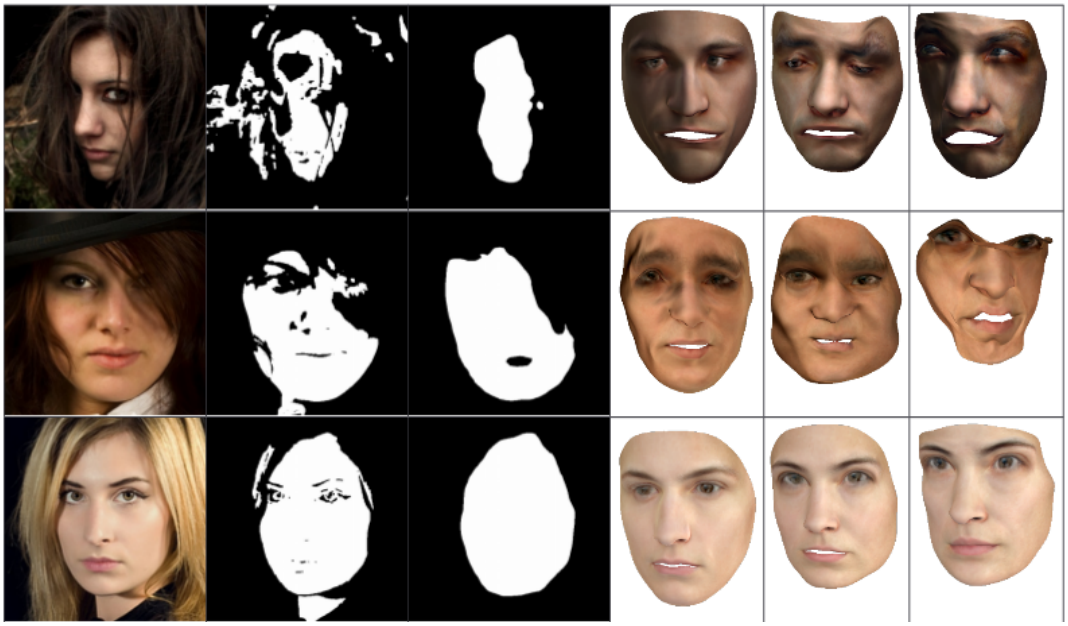


segmentation and mask of test6 in every 5th iteration with mask: FCN(from right to left)



segmentation and mask of test6 in every 5th iteration with mask: NO_OCCLUSION(from right to left)


Original
Mask of Egger
Mask of the FCN
Fit with Egger Mask
Fit with FCN Mask
Fit without any Mask



A.3 Discussion of the Results

For the synthetic data, a distinction must be made between the occlusions on which the FCN was trained on (hands, micros, glasses) and those unknown to the FCN (random boxes). For known occlusions, the FCN's mask is often more accurate than the one of Egger et al. If these masks are used for fitting, the errors of the parameters ('shape', 'color', 'expression', 'environmentMap') are either very close to those of the fit with the Egger mask, or even better.

The random boxes are not recognised by the FCN. The mask is very bad compared to Egger. However, just a few segmented points are enough to get a good fit. Because of the very low false positive rate of the FCN, the fits are comparable to those with the mask of Egger et al. Without occlusion, the masks of the FCN are clearly better than those of Egger. The resulting fits are almost impossible to compare with the naked eye. Due to the Basel Face Model parameters, the fit with the FCN mask is sometimes better than the one with the segmentation of Egger et al.

An eye-catching detail is that with the segmentation of Egger et al on the setting with the 'bfm' rendering (on the no occlusion dataset) a very small error of the shape parameter can be achieved. This could be because much more of the face shape is known due to the over-segmentation.
The real life data shows the enormous advantage of Egger. Although the segmentation of the FCN is not bad. The occlusion-aware method of Egger et al can recognise very thin partial coverings such as glasses and separate them from the facial region.

A.4 Parametric-Face-Image-Generator

We extended the Parametric-Face-Image-Generator of Kortylewski et al [13]. In our version the option "occlusionMode" in the configuration files can now be set to:

- **"eyes"**: A circle occludes the picture centered on the pupil centers of the face.
- **"random-1"**: A hand hides the picture in a random place in a random orientation.
- **"random-2"**: A microphone hides the picture in a random place in a random orientation.
- **"random"**: A randomly chosen occlusion image hides the picture in a random place.
- **"box"**: A box filled with an arbitrary color hides the image.
- **"box-whiteNoise"**: A box filled with Gaussian white noise hides the image.
- **"box-skinColor"**: Boxes filled with the colour at the tip of the chin hide the image on a random place.
- **"box-[0-100]"**: Boxes filled with a random colour sized that they occlude the specified amount of face pixels.
- **"loop"**: Produces 20 copies of the same image, each with a box occluding 2,4,6,...,40 % of the face.
- **"texture"**: Fills a randomly placed box with a specified texture image.

(a) A facial image of the parametric face image generator with all landmarks activated.

(b) The same image as in (a) but with an occlusion and two underlying landmarks which are disabled because they're not visible anymore.

Figure A.2: Because of the occlusion, certain landmarks had to be disabled. The image shows original landmarks (green dots) and the landmark which had to be deactivated (red dots).

The software provides csv-files, rps-files, tlms-files, ground truth masks, images with occlusions and images without occlusions for both the 'bfm' version and for the tailored 'face12' version of the Basel Face Model. If an occlusion gets rendered over a landmark, it gets disabled.

Declaration on Scientific Integrity
Erklärung zur wissenschaftlichen Redlichkeit

includes Declaration on Plagiarism and Fraud
beinhaltet Erklärung zu Plagiat und Betrug

Author — Autor
Elias Arnold

Matriculation number — Matrikelnummer
14-930-770

Title of work — Titel der Arbeit
An Empirical Comparison of Deep Learning and 3DMM-Based Approaches for Facial Occlusion Segmentation

Type of work — Typ der Arbeit
Bachelor-Thesis

Declaration — Erklärung
I hereby declare that this submission is my own work and that I have fully acknowledged the assistance received in completing this work and that it contains no material that has not been formally acknowledged. I have mentioned all source materials used and have cited these in accordance with recognised scientific rules.

Hiermit erkläre ich, dass mir bei der Abfassung dieser Arbeit nur die darin angegebene Hilfe zuteil wurde und dass ich sie nur mit den in der Arbeit angegebenen Hilfsmitteln verfasst habe. Ich habe sämtliche verwendeten Quellen erwähnt und gemäss anerkannten wissenschaftlichen Regeln zitiert.

Basel, 10.08.2018

Signature — Unterschrift