2D RGB Head Pose Estimation For Face Occlusion Scenarios

José Carlos Faria Celestino

***Abstract*—Head pose estimation, the task that deals with the prediction of the orientation of human heads, is a challenging Computer Vision problem that has been extensively researched and has a wide variety of applications. Despite the many studies carried out to achieve a more accurate pose prediction, current state of the art systems still under perform in the presence of occlusions. This 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. The implemented methodologies are based on the development of personalized occluded training and testing sets and the adaptation of deep learning network frameworks and strategies.

We show that our models improve occluded head pose estimation and equal or surpass 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.

***Index Terms*—Head Pose Estimation; Euler Angles; Occlusion; Neural Networks**

I. Introduction

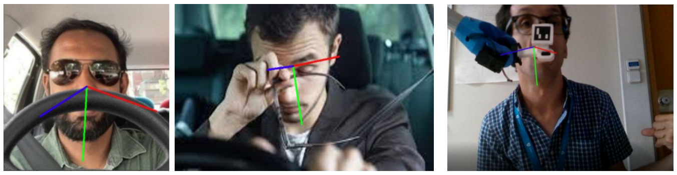
EXTENSIVELY researched over the last 25 years [1], 2D head pose estimation (HPE) is a challenging but compelling and relevant computer vision problem, essentially due to the wide variety of applications for which it can be used, such as driving aid systems [2], motion capture [3] and gaze estimation. Succinctly, this problem consists in approximately determining the orientation of a head in a 2D image.

Despite recent advances aided by deep learning, current state of the art systems scarcely approach one of the most challenging and common problems in HPE, the occurrence of facial occlusions, and underperform in such scenarios (figure 1). To address the issue, this thesis aims to study different methods to approach the occluded head pose estimation challenge, all based on the use of deep learning solutions and with the aid of synthetic occluded datasets. Our purpose is 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. 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.

II. State Of The Art

A. Model-Based Strategies

Yinguobing [4] provides a simple way which of performing head pose estimation by using 68 detected 3D facial landmarks to calculate the rotation R and translation T that define the head pose. This corresponds to solving the PnP problem [5], which is modeled by the pinhole camera model. The authors from [6] propose a method that similarly uses 68 landmark prediction but also computes the pose using predicted 2D keypoints of the head without fixed locations. They use the Features from Accelerated Segment Test (FAST) [7] and a pyramidal Lucas-Kanade feature tracker [8] to detect and track the points. They use a Kalman filter to blend both keypoint (prediction step of filter) and landmark detection (correction step). This fusion method shows better results than either using only keypoints or landmarks.



(a) Driver attention systems **(b)** Autonomous assisted feeding

Fig. 1. Head pose problem for occlusion scenarios (Blue axis points towards estimated face direction).

The work developed in [11] approaches the 3D face alignment and pose estimation problems as a 3D Morphable Model (3DMM) [12] parameters regression problem. They regress the rotation matrix and translation vector to estimate the pose in order to avoid the ambiguity cause by the gimbal lock [13] problem that occurs when faces get close to profile view. They use a fast lightweight backbone convolutional neural network and apply both a cost function with two terms, the weighted parameter distance cost (WPDC) and the vertex distance cost (VDC), and a landmark regressor. The cost function minimizes the vertex distances between a fitted 3D face and the ground truth. They also establish a 3D aided short-video-synthesis method which helps to achieve smoother estimation results in videos.

B. Occlusion Related Works

The authors in [14] estimate the head pose of partially occluded faces by tracking the displacement of a face feature with respect to the center of the head. They use CamShift [15] to track the center of the head and a iterative Lucas-Kanade optical flow tracker [8] to track the feature face point. This method requires the mouth not to be occluded and it is based on outdated software and hardware. The authors of [16] focus on achieving robust facial landmark detection for severe occlusions and images with large head poses.

They use landmark visibility probabilities to measure if a landmark is visible, and perform occlusion prediction. They add a prior occlusion pattern loss to aid the performance of the prediction. This work, however, does not have real-time tracking capabilities and does not specifically focus on estimating poses.

The method of [17] estimates facial landmark locations, head pose and facial deformation under facial occlusions. This procedure updates each estimation parameter based on the previously estimated values of the others. According to the authors, the combined framework 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. However, this work only evaluates yaw angles and has low accuracy for larger yaws.

C. Learning-Based Strategies

The authors from [18] claim that model-based methods rely on the chosen head model and are senstive to errors in landmark/keypoint detection. To avoid these drawbacks, they propose using a deep learning framework to estimate the pose directly from 2D RGB images. They input the images into a backbone neural network and augment it with three fully-connected layers, each one used to predict a different Euler angle. They introduce a multi-loss approach that combines a classification loss with a weighted regression loss for each angle. They use a cross-entropy loss for the classification component and a mean squared error loss for the regression component. 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 [19], applies the soft-stage wise regression problem defined in [20] to solve the HPE challenge. Feature maps from input images are extracted and fused together across several stages. Stage outputs are probabilities distributions for the angle interval classes. Each successive stage refines the decision within an angle interval assigned by the previous stage. The estimated pose is given by the soft-stage regression function, which corresponds to the sum of the product between probability distribution and the values of pose groups at each stage.

The method *img2pose* presented in [21] propose a novel real-time capable solution to simultaneously perform face detection and head pose estimation with 6 degrees of freedom (Euler rotation and 3D translation vectors) in an image without requiring a prior face detection step. This estimation is computationally much easier than the one of model-based approaches which regress 68x2D=136 elements, instead of only 6. Moreover, this pose allows to align the 3D face with its location in an image, which eliminates the need for face detectors.

The authors of [1] extend the multi-loss approach of *Hopenet* [18] for full 360° yaw estimation. They generate a new dataset with full range of yaws by combining 300W-LP with computed Euler angle data from the CMU Panoptic Dataset [22]. They use binary-cross entropy as for the classification loss and introduce a new wrapped loss for the regression component. They also utilize a lighter backbone network to facilitate real-time applications. The modifications made to *Hopenet* achieved state of the art of performance for full-range head pose estimation.

D. Summary

We saw that the literature on the challenge of occlusion in head pose estimation is scarce. The system in [14] requires the mouth not to be occluded and uses outdated software and hardware. The procedure in [16] addresses occlusions but focuses on landmark detection and is not extended for real-time tracking, and while the method in [17] includes pose estimation, it only evaluates yaw angles and displays low accuracy for large yaw values.

Model-based methods rely on the chosen head/face model and are very sensitive to landmark detection and tracking errors. They are also more susceptible both to self-occlusions (extreme poses for example) and object occlusions.

Learning-based methods do not require the detection of landmarks and therefore avoid the occlusion problem mentioned above, while outperforming model-based methods. For these reasons, our work will follow a learning-based approach, develop strategies based on some of the studied end-to-end deep learning frameworks and adapt them to the challenge of facial occlusions in head pose estimation.

III. Generating a Synthetic Occluded Dataset

A. Synthetic Occlusion Generation Procedure

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 cameras which combine RGB and depth sensors and are capable of simultaneously recording the necessary data.

The first step in our procedure is to record RGB and depth data in a video where an object occludes a person's face. The face is not occluded in the first frame since we need to first detect the face in the image in order to find the depth points that correspond to the image pixels within the face detection box. Afterwards, we determine the face point at minimum distance from the camera. This distance serves 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 2.

We implement DBSCAN [23], a density-based clustering algorithm, to remove outliers from the selected depth points and therefore correctly determine the threshold distance. When this threshold is determined we can start to reproduce the synthetic occlusions through the procedure illustrated in figure 3.

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

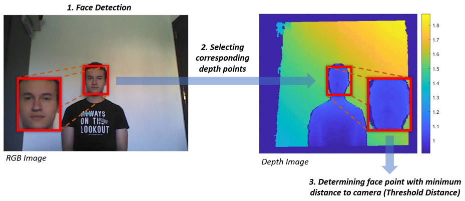

1. Face Detection
2. Selecting corresponding depth points
RGB Image
Depth Image
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
3. Determining face point with minimum distance to camera (Threshold Distance)

Fig. 2. Determining threshold depth for occlusion segmentation.

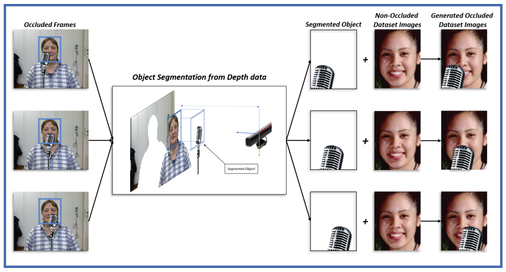

Occluded Frames
Object Segmentation from Depth data
Segmented Object
Segmented Object
Non-Occluded Dataset Images
Generated Occluded Dataset Images
+
+
+

Fig. 3. Synthetic occlusion generation in non-occluded images.

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.

The next step is to insert the object into a face image and generate the proposed synthetic facial occlusion. It is necessary to first re-scale the object image to the dimensions of the non-occluded face image, and only then superimpose the object in the original image.

Having determined 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 for each one of the images and for each occluded frame, and therefore generate a new occluded dataset.

B. Head Pose Datasets

We use the 300W-LP [24], BIWI [25] and AFLW2000 [26] datasets to implement and test our methodologies.

The 300W-LP dataset consists of 61225 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 provides facial landmark annotations 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 [18] [1].

The BIWI dataset contains over 15000 images of 20 people and covers about ±75 degrees yaw and ±60 degrees pitch. It is one of the most commonly benchmarked datasets. For each frame, it provides a depth image, the corresponding RGB image (both 640x480 pixels), and the ground truth pose annotation.

AFLW2000 is a dataset that contains 2000 images of diverse head poses under challenging conditions. It contains annotations for 3D facial landmarks from which the pose can be extracted.

We use occluded versions of the 300W-LP dataset in training and test our methodologies in occluded and non-occluded versions of the BIWI and AFLW datasets.

IV. Methodologies For Head Pose Estimation With Occlusions

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

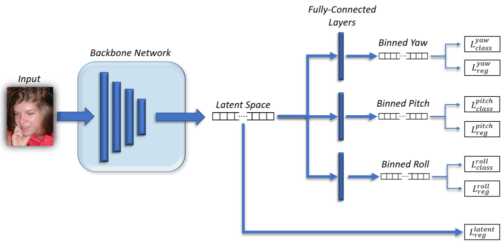


Fig. 4. Multi-Loss head pose estimation framework with latent space regression.

The framework is as exemplified in figure 4: A 2D RGB image is input to any backbone network of choice. This backbone network is expanded with three extra fully-connected layers which will be used to output the predictions for each Euler angle. The output of the final layer in the backbone network is flattened into a vector which becomes the input for each fully-connected layer. The output for these layers will be a vector of logits, which are raw prediction scores (real numbers in range $[-\infty, +\infty]$) for the predicted angle belonging to a certain angle bin of w degrees. 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.

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

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

Afterwards the cross-entropy loss result is computed by equation 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.

$$L_{class} = CE = -\sum_{i}^{C} t_i log(S(y_i)) \tag{2}$$

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:

$$\theta_{pred} = w \sum_{i=1}^{N} p_i (i - \frac{1+N}{2}) \tag{3}$$

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, 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 [1]. The loss used for the regression component is the mean squared error(MSE) between the predicted angle θ_{pred} and the ground truth angle θ_{gt}, for N predictions.

$$L_{reg} = MSE = \frac{1}{N} \sum_{i=1}^{N} (\theta_{pred} - \theta_{gt})^2 \tag{4}$$

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. 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. 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.

Our procedure is the following: Firstly, we either train or use a pre-trained model for head pose estimation in non-occluded images, with the same framework as figure 4 apart from the latent space loss. Then we perform inference with this model for each non-occluded image and store the flattened output of the final layer in the backbone network, which corresponds to the latent space representation for that given image. Finally, we use the occluded dataset and train the full framework of figure 4, 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, and another parameter β regulates the weight of the latent space regression loss. Overall, four losses are used to train the model:

$$\begin{aligned} L_{yaw} &= L^{yaw}_{class}(y, \hat{y}) + \alpha \, L^{yaw}_{reg}(y, \hat{y}) \\ L_{pitch} &= L^{pitch}_{class}(y, \hat{y}) + \alpha \, L^{pitch}_{reg}(y, \hat{y}) \\ L_{roll} &= L^{roll}_{class}(y, \hat{y}) + \alpha \, L^{roll}_{reg}(y, \hat{y}) \\ L_{latent} &= \beta \, L^{latent}_{reg}(y, \hat{y}) \end{aligned} \tag{5}$$

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 stored inference latent space output for the non-occluded images is used as ground truth in the latent loss. With a parameter β to weight the influence of each loss, the total combined loss is:

$$L_{total} = (1 - \beta)(L_{yaw} + L_{pitch} + L_{pitch}) + \beta L_{latent} \tag{6}$$

B. Occluded Head Pose Estimation Through Face Reconstruction

In this method, instead of directly adapting the head pose estimation model to deal with occlusions as we did previously, we train an autoencoder to output reconstructed non-occluded faces from facial occluded inputs. Autoencoders are a larger kind of unsupervised neural network composed of a encoder that maps the input into the code and a decoder that maps the code into a reconstruction of the input. The idea behind this approach is to use the occlusion-free outputs of the autoencoder as input to a trained head pose estimation network. We use the HPE multi-loss pipeline defined in [18]. The framework is as exemplified in figure 5.

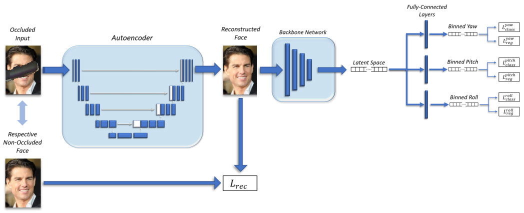


Fig. 5. Framework for occluded head pose estimation through face reconstruction.

Since standard autoencoders suffer from a loss of feature information when the input dimensions are reduced in the encoder, we base ourselves in the approach carried out by the authors of *mask2face* [27] and employ the U-Net [28] architecture as the chosen autoencoder structure. 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 accelerates the learning process and reduces the information loss. A simplified representation of the U-Net architecture is presented in figure 6.

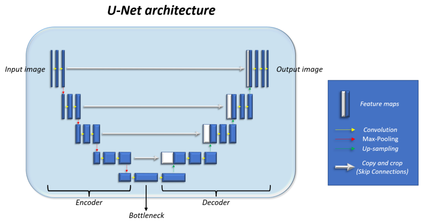


Fig. 6. Simplified standard U-Net architecture.

1) Reconstruction Loss: The autoencoder 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 images as ground truth in the reconstruction loss function, L_{rec}. We define this function as the combination of two losses: The l_1

the SSIM losses. The l_1 loss or Mean Absolute Error (MAE) corresponds to:

$$L_{l_1}(I_{rec}, I_{gt}) = \frac{1}{N} \|I_{rec} - I_{gt}\|_1 = \frac{1}{N} \sum_{p \in I_{rec}, I_{gt}} |I_{rec}(p) - I_{gt}(p)| \tag{7}$$

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.

The structural similarity index measure equation (SSIM) [29] is a metric used for the measurement of the similarity between two images. It extracts and compares three different measures between images: the luminance (l) , the contrast (c) and the structure (s). The equation for this metric is obtained by combining the three measures:

$$SSIM(x,y) = [l(x,y)]^{\mu} \cdot [c(x,y)]^{\phi} \cdot [s(x,y)]^{\psi} = \frac{(2u_x u_y + c_1)(2\sigma_{xy} + c_2)}{(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8}$$

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 ${}^2_x + {}^2_y$ and $\sigma_x^2 + \sigma_y^2$ are close to zero. $\mu > 0$, $\phi > 0$ and $\psi > 0$ are 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:

$$L_{SSIM}(I_{rec}, I_{gt}) = \frac{1}{N} \sum_{x,y \in I_{rec}, I_{gt}} 1 - SSIM(I_{rec}(x), I_{gt}(y)) \tag{9}$$

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, 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:

$$L_{rec}(I_{rec}, I_{gt}) = L_{l_1}(I_{rec}, I_{gt}) + \gamma L_{SSIM}(I_{rec}, I_{gt}) \tag{10}$$

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

C. Multi-Loss Autoencoder For Occluded Head Pose Estimation

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 7.

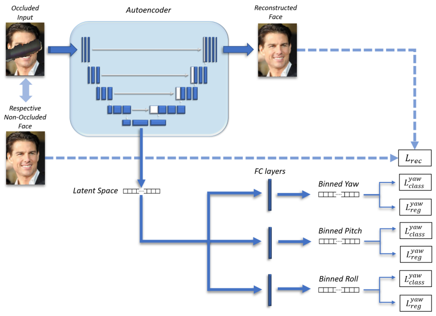


Fig. 7. Multi-loss autoencoder high-level pipeline.

We add a decoder to the head pose estimation encoder 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, in particular, the ResNet-Unet architecure described by the authors of [30] with ResNet-50 [31] as the encoder for pose estimation.

The training plan 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. 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. 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. The third stage stems from both and combines them so that the pose estimation is adapted to the reconstruction of occluded faces.

The 1st stage involves losses L_{yaw}, L_{pitch}, L_{roll} defined in section IV-A (head pose estimation training), the 2nd stage involves the loss L_{rec} defined in section IV-B1 (face reconstruction), and the 3rd stage (combined training) requires the combined minimization of all losses (equation 11) with ρ parameter to define the weight of angle components:

$$L_{total} = L_{rec} + \rho(L_{yaw} + L_{pitch} + L_{roll}) \tag{11}$$

V. Results and Discussion

A. 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. We use ResNet-50 as the backbone network for this method. 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 [18]. To optimize the parameters we use the Adam [32] optimization algorithm with a learning rate of 10^{-5}.

We train the framework defined in section IV-A using synthetically occluded versions of 300W-LP. The face images are cropped to the pre-defined input dimension of the ResNet-50 network, 224x224 pixels, and the mean and standard deviation of ImageNet [9] 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.

1) Angle Regression Weight Study: We train the framework with 5 different α parameter values to determine the best one. The head pose estimation MAE results in synthetically occluded and non-occluded datasets are listed in tables I and II. 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. We can also observe that the largest errors tend to occur for $\alpha = 1$. 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					
Reg. Weight (α)	yaw	pitch	roll	Avg. $MAE°$	Combined Avg.	
0.5	5.250	6.529	4.255	5.345	alpha	$MAE°$
1	5.333	6.916	4.036	5.495	0.5	4.763
2	5.110	6.832	3.629	**5.190**		
5	5.477	6.542	4.304	5.441	1	4.950
10	5.218	7.565	4.344	5.709		
	Non-Occluded Images				2	**4.617**
0.5	4.259	4.704	3.580	5.441		
1	4.765	4.493	3.956	5.768	5	4.723
2	4.242	4.041	3.845	4.043		
5	4.474	4.043	3.494	**4.004**	10	4.909
10	4.196	4.503	3.628	4.109		

TABLE I
Head pose estimation MAE ° tests with BIWI for different angle regression weights (α).

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

TABLE II
Head pose estimation MAE ° tests with AFLW2000 for different angle regression weights (α).

2) Latent Space Regression Weight Study: Having determined the best α, we trained 4 different head pose estimation networks, one for each different β, the parameter that defines the weight of the latent and angle losses, and compared the results with *Hopenet* [18]. 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 III and IV display the results for each dataset. We observe that when the latent space regression loss is not used ($\beta = 0$), despite substantially decreasing the error for occluded images, the results are worse 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 β parameter increases, the MAE becomes lower for both occluded images and non-occluded images. In AFLW the non-occluded estimation results improve on 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. While for $\beta = 1$ the average MAE errors are the best in BIWI, we verify that this parameter leads to a worse estimation of yaw values, which is the most varied and relevant Euler angle in head pose estimation. $\beta = 0.999$ provides the most balanced results in average MAE and yaw estimation.

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

TABLE III
Head pose estimation MAE ° tests with BIWI for different latent space regression weights (β). LSR stands for latent space regression.

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

TABLE IV
Head pose estimation MAE ° tests with AFLW2000 for different latent space regression weights (β). LSR stands for latent space regression.

B. Occluded Head Pose Estimation Through Face Reconstruction

We train the autoencoder for 25 epochs and use a batch size of 116. The training and testing datasets are the same we used in the previous approach. 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. We randomly adjust the brightness, contrast, saturation and hue during training to enhance the robustness of the reconstruction.

We train 3 models with distinct weights for the SSIM loss in the reconstruction loss function defined in equation 7 (section IV-B1). 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 $\gamma = 2$.

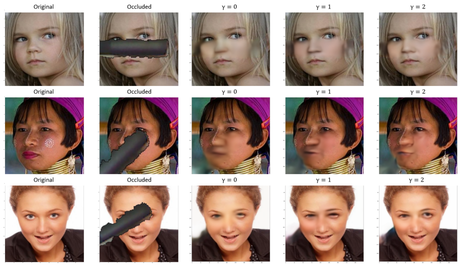


Fig. 8. Reconstruction comparison for different loss weights.

Figure 8 displays examples of the different reconstructions generated by each model. 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 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 significant improvement over the first model. Lips become more defined, and complicated details such as the 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 9 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 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 evaluate and compare the pose estimation performance for the reconstructed images. The network and model of *Hopenet* is used as the head pose estimator. The results for each dataset are listed in tables V and VI. We can observe that the estimation error decreases for all reconstruction models, in both occluded and non-occluded images. The improvements for the average MAE in occluded image range from around 1.5° in BIWI to nearly 5° in AFLW2000. The lower resolution of the images in the BIWI dataset leads to the less improved results. 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 lead to reduce the error in head pose estimation.

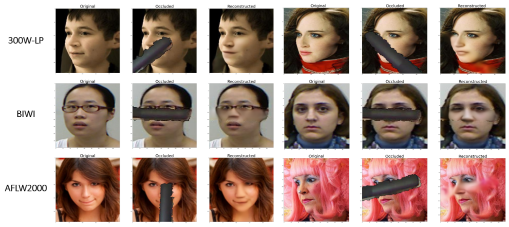


Fig. 9. Examples of face reconstruction results for each dataset.

BIWI	Occluded Images				Combined Avg.	
Methods	yaw	pitch	roll	Avg. MAE		
Hopenet	6.725	8.616	7.338	7.560	γ	MAE
Rec. ($\gamma = 0$)	5.979	6.526	5.755	6.087	HPN	5.661
Rec. ($\gamma = 1$)	5.755	6.306	5.852	5.971		
Rec. ($\gamma = 2$)	5.847	6.323	5.727	**5.966**	0	4.853
	Non-Occluded Images					
Hopenet	4.375	3.559	3.348	3.761	1	**4.789**
Rec. ($\gamma = 0$)	4.137	3.392	3.324	3.618		
Rec. ($\gamma = 1$)	4.117	3.389	3.606	**3.606**	2	4.792
Rec. ($\gamma = 2$)	4.133	3.391	3.617	3.617		

TABLE V
HEAD POSE ESTIMATION MAE° RESULTS FOR RECONSTRUCTED IMAGES IN BIWI. THE RESULTS OF HOPENET CORRESPOND TO THE ORIGINAL INPUTS TO THE NETWORK.

AFLW2000	Occluded Images				Combined Avg.	
Methods	yaw	pitch	roll	Avg. MAE		
Hopenet	12.438	10.277	8.586	10.434	γ	MAE
Rec. ($\gamma = 0$)	5.738	6.392	4.962	5.697	HPN	7.589
Rec. ($\gamma = 1$)	5.674	6.196	4.812	5.560		
Rec. ($\gamma = 2$)	5.546	6.138	4.690	**5.458**	0	5.335
	Non-Occluded Images					
Hopenet	4.965	5.250	3.956	4.724	1	5.113
Rec. ($\gamma = 0$)	4.978	5.250	3.959	4.729		
Rec. ($\gamma = 1$)	4.676	5.353	3.965	4.665	2	**5.058**
Rec. ($\gamma = 2$)	4.675	5.345	3.956	**4.658**		

TABLE VI
HEAD POSE ESTIMATION MAE° RESULTS FOR RECONSTRUCTED IMAGES IN AFLW2000. THE RESULTS OF HOPENET CORRESPOND TO THE ORIGINAL INPUTS TO THE NETWORK.

C. Multi-Loss Autoencoder For Occluded Head Pose Estimation

We trained 3 different models for this methodology. In the first stage we use the pre-trained encoder provided by the authors of [18] 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 $\gamma = 0$, 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 $\rho = 0.1$. 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 $\alpha = 2$ in all models. The networks were trained for 25 epochs in each stage with 116 image batches. 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. Tables VII and VIII list the results for all models in each dataset.

BIWI	Occluded Images					
Methods	yaw	pitch	roll	Avg. MAE	Combined Avg.	
Hopenet	6.725	8.616	7.338	7.560	Method	MAE
ResUnet 1	6.874	6.462	4.994	6.110	HPN	5.661
ResUnet 2	6.192	7.218	4.335	5.922		
ResUnet 3	5.377	6.318	4.564	**5.420**	1	5.127
	Non-Occluded Images					
Hopenet	4.375	3.559	3.348	**3.761**	2	5.384
ResUnet 1	4.586	4.193	3.654	4.144		
ResUnet 2	4.798	4.944	3.712	4.845	3	**4.747**
ResUnet 3	4.380	4.175	3.669	4.075		

TABLE VII
HEAD POSE ESTIMATION MAE° RESULTS FOR RESUNET IN BIWI.

AFLW2000	Occluded Images					
Methods	yaw	pitch	roll	Avg. MAE	Combined Avg.	
Hopenet	12.438	10.277	8.586	10.434	Method	MAE
ResUnet 1	5.557	6.859	5.372	5.929	HPN	7.579
ResUnet 2	5.329	6.550	4.978	5.619		
ResUnet 3	5.123	6.354	4.633	**5.370**	1	5.304
	Non-Occluded Images					
Hopenet	4.965	5.250	3.956	4.724	2	5.189
ResUnet 1	4.237	5.623	4.173	4.678		
ResUnet 2	4.380	5.669	4.196	4.758	3	**5.001**
ResUnet 3	4.235	5.569	4.126	**4.643**		

TABLE VIII
HEAD POSE ESTIMATION MAE° RESULTS FOR RESUNET IN AFLW.

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 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.

D. Method Results Comparison and Discussion

BIWI	Occluded Images					
Methods	yaw	pitch	roll	Avg. MAE	Combined Avg.	
Hopenet	6.725	8.616	7.338	7.560	Method	MAE
LSR 3	5.174	6.622	4.117	5.304	HPN	5.661
LSR 4	5.429	4.823	3.467	**4.573**		
Rec. 3	5.847	6.323	5.727	5.966	LSR 3	4.669
ResUnet 3	5.377	6.317	4.564	5.420		
	Non-Occluded Images				LSR 4	**4.046**
Hopenet	4.375	3.559	3.348	3.761		
LSR 3	4.297	4.186	3.617	4.033	Rec. 3	4.792
LSR 4	4.291	3.086	3.179	**3.519**		
Rec. 3	4.133	3.391	3.326	3.617	ResUnet 3	4.747
ResUnet 3	4.380	4.175	3.669	4.075		

TABLE IX
METHOD COMPARISON IN BIWI.

AFLW2000	Occluded Images					
Methods	yaw	pitch	roll	Avg. MAE	Combined Avg.	
Hopenet	12.438	10.277	8.586	10.434	beta	MAE
LSR 3	4.741	6.254	4.765	**5.253**	HPN	7.579
LSR 4	5.117	6.075	4.590	5.261		
Rec. 3	5.674	6.138	4.690	5.458	LSR 3	**4.833**
ResUnet 3	5.123	6.354	4.633	5.370		
	Non-Occluded Images				LSR 4	4.867
Hopenet	4.965	5.250	3.956	4.724		
LSR 3	3.813	5.420	4.003	**4.412**	Rec. 3	5.058
LSR 4	4.258	5.272	3.888	4.473		
Rec. 3	4.675	5.345	3.956	4.658	ResUnet 3	5.001
ResUnet 3	4.235	5.569	4.126	4.643		

TABLE X
METHOD COMPARISON IN AFLW2000.

Tables IX and X display the head pose estimation results in the respective dataset, for the best models of each method. LSR 3 and 4 stand for the latent space regression methods with a weight of $\beta = 0.999$ and $\beta = 1$, respectively. Rec. 3 stands for the reconstruction method with a weight of $\gamma = 2$ for the SSIM loss. 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 verify that reconstructing the images to input in an head pose estimator produces the worst results in occluded images, specially in the BIWI dataset. This is mainly due to its sensitivity to the lower resolution of the images in this dataset, which is the main disadvantage of this method. However, it improves the head pose estimation for non-occluded images in both datasets when compared to inputting the original image directly in *Hopenet*. ResUnet 3 produces the third best results in BIWI and AFLW occluded results, 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 model 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. 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 LSR 3 to be the best for such scenarios.

E. Testing Pose Estimation In The Feedbot Scenario

We tested our best method in the feeding context of Feedbot [33], an autonomous feeding robot, to find out how our model performs and compare it to *Hopenet*. We recorded a video where Feedbot's robotic arm executes the feeding task and occludes the face of the user. We performed inference for the entire video using both our and *Hopenet's* models. Since we do not have the ground truth pose in this testing conditions, we carry out a qualitative analysis and evaluation in this section.

Figure 10 displays a set of occluded example frames which show clear improvements in the head pose estimation when compared to *Hopenet*. In these examples, the *Hopenet* 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 its towards the right. Our model, on the contrary, indicates head poses much closer to reality despite the partial occlusions, particularly improving the yaw rotation around the green axis when compared to *Hopenet*.

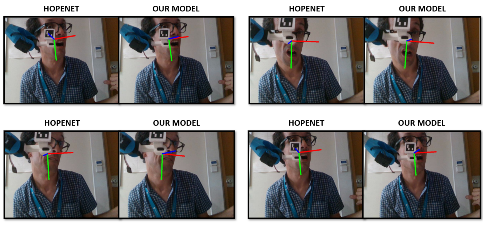


Fig. 10. Feedbot: Comparison between *Hopenet* and our model - Occluded frames.

There are, however, frames of the video for which our model does not exhibit the desired behaviour. Figure 11 displays some examples of large occlusions where the head pose estimation of our model is not as accurate. Despite the user looking straight at the camera, the estimated yaw rotation indicates the head is slightly tilted to the left of the image. Nonetheless, both pitch and roll estimation seem to be valid and our model still performs better than *Hopenet* in these cases.

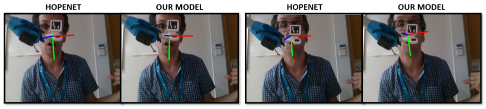


Fig. 11. Feedbot: Comparison between *Hopenet* and our model - Occluded frames (continuation).

For non-occluded frames, our model's pose estimations are good and identical to those of state of the art *Hopenet*, as seen in figure 12. This is the desired behaviour since we intend to improve on occluded head pose estimation, while preserving the accuracy for non-occluded images that state of the art works exhibit.

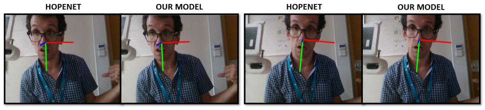


Fig. 12. Feedbot: Comparison between *Hopenet* and our model - Non-occluded frames.

VI. Conclusion

A. 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 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 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 also studied 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 an l_1 and SSIM losses leads to more fine-grained face reconstructions, which 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 applying the SSIM loss and increasing the weight of angle losses in the overall framework leads to better pose estimation results, which surpassed usign two separate networks.

By performing ablation studies and measuring the influence of losses we determined the best training configurations and models. We verified that all methodologies improved occluded head pose estimation and equaled or surpassed the estimation state of the art 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. 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.

B. 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. As a consequence he segmented occlusions occupy a small region in the image and have low resolution. An RGB-D sensor of higher resolution would allow to generate more natural synthetic occlusions.

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 lighter, accurate network such as EfficientNet [10], with 11 million parameters, could be used to improve this aspects.

The performance of our reconstruction models depends on the resolution of the detected face. The face generation capabilities of a Generative Adversarial Network could be explored 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.

References

[1] 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

[2] A. Fern andez Vill an, R. Usamentiaga, J. Car us Cand as, and R. Casado, “Driver distraction usingvisual-based sensors and algorithms,”Sensors, vol. 16, p. 1805, 10 2016.

[3] M. C. d. F. Macedo, A. L. Apolin ario, and A. C. d. S. Souza, “A robust real-time face trackingusing head pose estimation for a markerless ar system,” in2013 XV Symposium on Virtual andAugmented Reality, 2013, pp. 224–227.

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

[5] OpenCV, “Opencv: Solvepnp.” [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group calib3d.htmlga549c2075fac14829ff4a58bc931c033d

[6] 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.

[7] 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.

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

[9] 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.

[10] 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

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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

[17] 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

[18] 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.

[19] 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.

[20] 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 TwentySeventh International Joint Conference on Artificial Intelligence, IJCAI-18.

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

[22] 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.

[23] 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.

[24] 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

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

[26] 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.

[27] 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

[28] 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

[29] 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.

[30] 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.

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

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

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