Real Time Eye Gaze Tracking System using CNN-based Facial Features for Human Attention Measurement

Oliver Lorenz and Ulrike Thomas
Professorship of Robotics and Human-Machine-Interaction,
Chemnitz University of Technology, Reichenhainer Str. 70, Chemnitz, Germany

Keywords: Eye Gaze Tracking, Human-robot Interaction, Facial Features, Head Pose, Face Detection, Human Attention.

Abstract: Understanding human attentions in various interactive scenarios is an important task for human-robot collaboration. Human communication with robots includes intuitive nonverbal behaviour body postures and gestures. Multiple communication channels can be used to obtain a understandable interaction between humans and robots. Usually, humans communicate in the direction of eye gaze and head orientation. In this paper, a new tracking system based on two cascaded CNNs is presented for eye gaze and head orientation tracking and enables robots to measure the willingness of humans to interact via eye contacts and eye gaze orientations. Based on the two consecutively cascaded CNNs, facial features are recognised, at first in the face and then in the regions of eyes. These features are detected by a geometrical method and deliver the orientation of the head to determine eye gaze direction. Our method allows to distinguish between front faces and side faces. With a consecutive approach for each condition, the eye gaze is also detected under extreme situations. The applied CNNs have been trained by many different datasets and annotations, thereby the reliability and accuracy of the here introduced tracking system is improved and outperforms previous detection algorithm. Our system is applied on commonly used RGB-D images and implemented on a GPU to achieve real time performance. The evaluation shows that our approach operates accurately in challenging dynamic environments.

1 INTRODUCTION

Analyses of human eye gazes are important for recognizing attentions in various applications. Eye motions and eye directions indicate human attentions to communicate to each other and can thus be used to study their willingness to interact and their intentions. Based on these knowledge, accurate and robust detection of eye gazes are essential components in active research topics, while used among many other indicators for action recognition (Liu et al., 2017) and saliency detection (Parks et al., 2015). Besides hand gestures and body postures, facial features are considered as the most important interactive features to understand a human desire, needs and cognitive processes (Palermo and Rhodes, 2007). In order to enable interactive communication between humans and robots, eye gazes are used, which clearly describe human behaviour to communicate. Looking on eye gazes, it offers other humans a stimulus to direct their attentions immediately to their counterparts. Human actions are always performed in the direction of eyes in order to maintain control and the ability to react. Face detection tasks are affected strongly by various human characteristics and images based on strong assumptions. Rowley et al. (Rowley et al., 1998) describe a neural network trained by arbitration on multiple networks to improve performance over a single network. In (Yang et al., 2015) a deep CNN-based (DCNN) face detector detects faces from a new perspective view through scoring facial parts responses by spatial structures and arrangements. Fischer et. al. (Fischer et al., 2018) generate own datasets with annotations based on an external unit of measurement in order to obtain eye gazes and head poses. A cascading CNN (Li and Fu, 2018), which operates on multiple resolutions, quickly rejects the background regions in the fast low-resolution step. Cascaded shape regression models (Dollár et al., 2010) apply regression to establish the relation between appearance to shape by directly learning the mapping function. Deep learning combined with shape regression can improve the accuracy of localisation of facial features. DCNNs can be classified within the large framework of cascaded shape regression models. Multi task-oriented CNNs establish the inherent correlation between these two tasks. Eye gaze tracking is defined as tracking the eye movements, e.g. the directions of

Lorenz, O. and Thomas, U.
Real Time Eye Gaze Tracking System using CNN-based Facial Features for Human Attention Measurement.
DOI: 10.5220/0007565305980606
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 598-606
ISBN: 978-989-758-354-4

eye gazes and their gradient over time. Special geometric relationships of the features (Zhao et al., 2018) reduce the complexity for calibration and image processing time. Chinsatit and Saitoh (Chinsatit and Saitoh, 2017) suggest two CNNs as feature points to recognise the pupil centres. The first CNN is used to classify the eye state and the second CNN estimates the position of the pupil centres. The authors (Mukherjee and Robertson, 2015) use low-resolution multi modal RGB-D images and regression based on learned depth classifiers. They combine the two models to approximate regression confidences. In (George and Routray, 2016), a CNN is used to classify the eye gaze and to estimate eye-accessing cues in real time. Lemley et. al. (Lemley et al., 2018) evaluate different datasets for designing a CNN with minimal computational requirements for gaze estimation. Recently, Deng and Zhu (Deng and Zhu, 2017) suggest a new two-step training policy, where a CNN for head recognition and a CNN for eye recognition are trained separately and then jointly fine-tuned with a geometrically constrained gaze transform layer. In general, the robustness and reliability of current approaches for eye gaze tracking have been improved. However, the accuracy and repeatability of current detection methods are not sufficient. In order to predict human intentions in different situations and in realistic work environments, a new tracking system based on cascaded CNNs for extreme conditions and face alignments is proposed. It is able to recognise the willingness of interaction and measures the attention of humans because of eye gazes and head orientations. It applies in the first recognised 2D facial feature and maps it into 3D, then a further CNN is applied to track the head orientation. These models allow real time tracking of eye gaze independently from orientation of the head. Due to the facial features and applied facial symmetries, the eye gaze recognition is robust towards face and image-based occlusions. As a result, the eye gaze direction should be considered as a key indicator for the ability to interact.

2 PROPOSED METHOD

Figure 1 shows the proposed tracking pipeline for eye gazes. A RGB-D image is obtained from a low-cost RGB-D sensor and transferred to the Multi-Task CNN (MTCNN). The MTCNN detects face regions and locates the facial landmarks of an individual face. The eye regions are separated and the head pose is computed. If the yaw angle of the head is beyond a certain threshold τ, the eye region is replaced with the tracking result from the MTCNN. Alternatively, the eye region is taken as the initialisation input for the Kernelized Correlation Filter (KCF). Furthermore, the two eye regions are taken as input for another CNN. The model of the second CNN explicitly encodes coordinates in the CNN-layers and calculates the coordinates of eye corners. Finally, the eye gaze of each individual eye is computed by the usage of different facial features.

2.1 Face Detection

By various human characteristics and different training datasets, we recognise all human faces in the field of view (FOV) and then determine the face alignment for each face. We introduce a cascaded structure to establish our detection from coarse to fine, see Figure 1. The cascaded network contains three sub networks: a Proposal Network (P-Net), a Refinement Network (R-Net) and an Output Network (O-Net). The P-Net is used to generate bounding boxes around detected faces. The trained P-Net only outputs N bounding boxes with four coordinates and their quality scores. The R-Net is used to remove a large number of regions where no faces are detected. The input of the R-Net is the resulted bounding box of the former P-Net, with the size of 24×24 pixels. The O-Net is similar to the R-Net by one exception that this the O-Net with the size of 48×48 pixels includes the task for landmark regression. The final output includes four (x,y)-coordinates for the bounding box, designed by scores, that are in particular the (x,y)-coordinate of left top point, height and width of the bounding box, and can be described by a vector $y_i^{box} \in \mathbb{R}^4$. The values $\hat{y}_i^{lm}$ are the facial coordinates and the values are y_i^{lm} the Ground Truth coordinates. Altogether, there are five facial landmarks as a vector $y_i^{lm} \in \mathbb{R}^5$, including rough centre of left and right eye, nose tip, left mouth corner and right mouth corner. The loss function of the face classification branch is given by L_i^{det} (1), which adopts the cross entropy

$$L_i^{\text{det}} = -(y_i^{\text{det}} \log(p_i) + (1 - y_i^{\text{det}})(1 - \log(p_i))), \quad (1)$$

where p_i is the probability produced by the network that indicates a sample being a face. The $y_i^{\text{det}} \in \{0,1\}$ denotes the Ground Truth label. The second loss function is for bounding box regression $L_i^{\text{box}} = || \hat{y}_i^{\text{box}} - y_i^{\text{box}} ||^2$ and the third is for facial landmark regression. The difference $L_i^{\text{lm}} = || \hat{y}_i^{\text{lm}} - y_i^{\text{lm}} ||^2$ is the euclidean distance loss. The $\hat{y}_i^{box}$ describes the regression target and y_i^{box} represents the Ground Truth coordinates. We trained the MTCNN with several different datasets (Sun et al., 2013; Yang et al., 2016), which contain front view images and side view images. Moreover, we annotated these datasets to im-

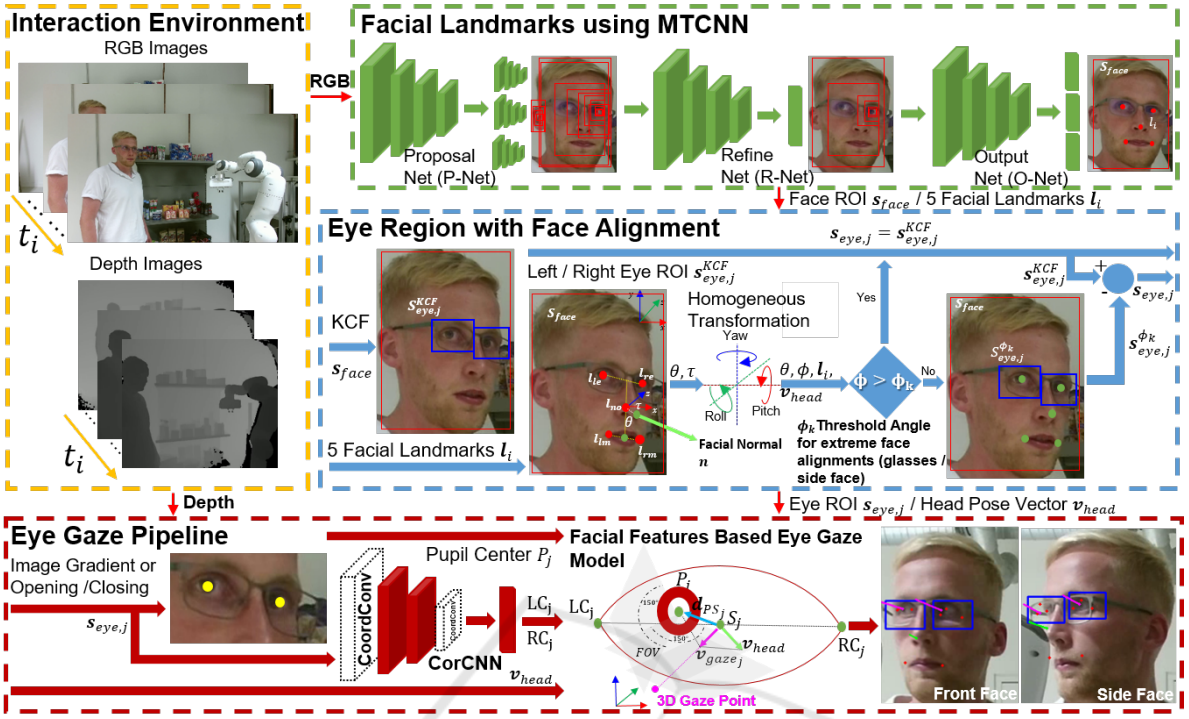

Figure 1: Proposed pipeline of CNN-based facial features and tracking of eye gazes.

prove the accuracy and robustness against the approach from (Zhang et al., 2016). Zhang et. al. only use datasets with front view images. For eliminating the background we compute the loss function L_i^{det}. The Equation (2) is defined to control the calculation of different losses for various inputs.

$$\min \sum_{i=1}^{N} \sum_{j \in \{\text{det,box,lm}\}} \alpha_j \beta_i^j L_i^j \qquad (2)$$

where N is the number of training samples. $j \in \{\text{det}, \text{box}, \text{lm}\}$ and α_j denotes the task importance. In the case of P-Net and R-Net, the weights for the key points regression are less than those for the O-Net part due to the focus of the first two stages on filtering out the bounding boxes of which are no faces. $\beta_i^j \in \{0,1\}$ is a sample type indicator and L_i^j denotes the respective loss function for three different tasks. As the information provided in the various datasets can be assumed as known, the number of images contained in the original training set is very small, which is not enough to train MTCNN. Therefore, we increase the number of input images by augmenting training dataset with annotated information. Our approach includes a) *random rotations to deal with face rotations*, b) *random increase and decrease of intensity values to deal with different lighting conditions* and c) *addition of Gaussian noise to enhance generalisation performances*. We select the 70 % largest loss samples in each mini batch as hard samples and we only use this part to calculate the gradient in the back propagation. Some simple samples, which have little effect on the enhanced classifications, are discarded. Finally, it turned out that our training with stochastic gradient descent, an initial learning rate of 0.01 with decay is assumed.

2.2 Head Orientation

Based on the five facial landmarks $\boldsymbol{l}_i$ the key features are determined and with the symmetry assumption of the face, we compute the orientation of the head. In (Gee and Cipolla, 1994) are taken four key features to calculate the head orientation and determine the centre of the mouth by linearly extending the tip of the nose downwards. We use an improved and more accurate approach with more CNN-based facial features as well as the calculation of the pupil centre. In addition, we estimate the symmetry centre of the mouth and the length of the mouth L_m based on the left and right corner of the mouth. The facial symmetry axis is found by connecting a line between the eye midpoint and the mouth midpoint. Assuming a fixed ratio between these facial features and fixed distance ratios between eye to mouth, nose to mouth, nose base to nose tip as well as the length of the mouth. The facial direction can be determined under weak perspective geometry

from the 3D angle of the nose. The $\boldsymbol{l}_i$ are used to determine the head pose from the normal by the vector $\boldsymbol{d}$ to the image plane, which can be found from planar skew symmetry and a coarse estimate of the nose tip. The angle σ is formed by the face normal $\hat{\boldsymbol{n}}$ and the normal vector $\boldsymbol{d}$ of the image plane. Using the length between eyes to mouth centre L_e and nose tip to mouth centre L_b, the position of the nose base l_{no} can be found along the symmetry axis following the image of the facial normal and producing an immediate estimate of the tilt τ. Our Algorithm 1 calculates

Algorithm 1: Tracking for face alignment.

Data: Face ROI $\boldsymbol{s}_{\text{face}}$, 5 facial landmarks $\boldsymbol{l}_i = (x_i, y_i)$ where $i \in \{\text{le, re, lm, rm, no}\}$
Result: Aligned eye regions $\boldsymbol{s}'_{\text{eye},j}$,where $j \in \{\text{l,r}\}$

Initialize KCF using dot product kernel;
Compute head orientation (θ, ϕ) based on $\hat{\boldsymbol{n}} = [\sin\sigma\cos\tau, \sin\sigma\sin\tau, -\cos\sigma]$ with $\sigma = \cos^{-1} \mid \hat{\boldsymbol{d}}_z \mid$ and angle τ;
if $\phi >$ *yaw threshold* ϕ_k **then**
    Result as $\boldsymbol{s}'_{eye,j}$;
else
    if $\tau > 5$ **then**
        Rotate transformation with rotation angle -τ and rotation centre $\boldsymbol{l}_{lm}$ on $\boldsymbol{s}_{face}$ and landmarks $\boldsymbol{l}_i$;
        Aligned face ROI $\boldsymbol{s}'_{face}$ and transformed landmarks $\boldsymbol{l}'_i$;
        Crop 2 eye ROIs $\boldsymbol{s}'_{eye,j}$ in $\boldsymbol{s}_{face}$;
        Location of $\boldsymbol{s}_{eye,j}$ to initialize the KCF using $\hat{f}(\boldsymbol{z}) = \hat{\boldsymbol{k}}^{xz} \odot \frac{\hat{\boldsymbol{y}}^*}{\hat{\boldsymbol{k}}^{xx}+\lambda}$;
        Result as $\boldsymbol{s}'_{eye,j}$;
    end
end

the head orientation resulting in a pitch angle ψ and yaw angle ϕ obtained from the angles θ and τ. Since the roll angle θ describes the alignment of a human faces, the process of feeding the aligned human eye regions can improve the robustness. For computing a roll angle θ, we use the two mouth corners and the mouth length to determine the angle θ relative to the x-axis. Then we select the left mouth corner as the centre of rotation to carry out the rotation transformation of $-\theta$. This step can eliminate the influence of the roll angle. Through the yaw angle ϕ, we determine lateral head rotation which has an influence on the detection of the face. If the yaw angle ϕ is too large due to the head orientation, the key features can no longer be detected exactly, because the MTCNN was only trained with front view images and a few multi view images. It is well known that when the face is tilted at a larger angle, the other eye is not visible in the camera-centred image, and the eye gaze tracking depends on the eye region that is visible. Afterwards, the eye corners and pupil centres can be extracted in the aligned face images and then they are reversed to obtain their true coordinates in the origin image. For tracking the eye region under large yaw angle ϕ, we set a threshold angle ϕ_k for the yaw angles. The threshold angle ϕ_k is statistically determined by the usage of training data. If the yaw angle ϕ is greater than the threshold angle ϕ_k, we will start to use the tracking results of KCF instead of these cropped eye regions. The KCF constructs the training sample of classifier through loop shifting, which turns the data matrix into a circular matrix. Based on the properties of circular matrix, the solution of the problem is transformed to the frequency domain, which avoids the process of matrix inversion and greatly reduces the complexity.

2.3 Eye Features and Eye Gaze

CNN models have some excellent properties, such as weight sharing and local connections. However, in the task involving coordinate modelling, its advantages become eliminated and potentially affect the model performance. The nature of the problem lies on the invariant translation of convolution. Accordingly, we proposed the corresponding "CoordConv-layer". "CoordConv-layer" solves the problem of coordinate transformation and it has an improved generalisation ability. The "CoordConv-layer" allows the convolution filter to observe the coordinates that will break invariant transformations and learn a function for the translation of invariants. If weights converge toward zero, the layer behaves exactly like the standard convolution. If weights are nonzero, the function will contain some degree of translation of dependence. The precise form of which will ideally depend on the task being solved. The "CoordConv-layer" accomplishes a mapping by first concatenating extra channels to the incoming representation. It allows the Corner CNN (CorCNN) as a single stage CNN to learn to keep or to discard the third translation of invariance as is needed for the task being learned, see in Figure 2. For the training datasets, we selected

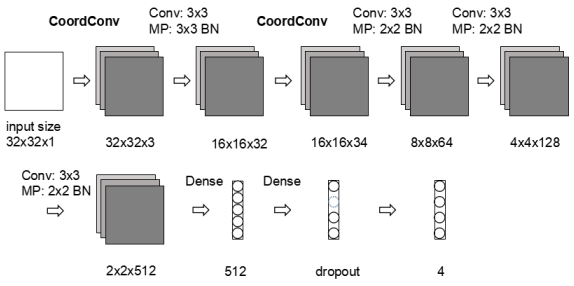

Figure 2: Model of our CorCNN.

the Kaggle Facial Landmarks Dataset [1] and also extended the training data. We use the loss function of landmark regression to control the training process of our CorCNN. The loss function, based on Mean Squared Error (MSE), is a *L2* regularisation with factor 0.0001 and a simple *L2* distance between the respective predicted output and (x,y)-coordinates of Ground Truth landmarks. Our CorCNN computes the loss between actual targets and predicted targets and learns the weights by reducing this loss. The regression loss does not change after 10000 iterations and the minimum of the regression loss is 0.0136 by a learning rate of 0.01. From there, the regression loss converge towards zero. We apply a feature-based approach by using combined image gradients and morphological operations for the localisation of pupil centre P. In order to directly track the eye movements and to calculate the eye gaze vector, we propose a facial features method based on the pupil centre P, eye corners LC_j and RC_j and the particular head orientations $\boldsymbol{v}_{\text{head}}$, which allows image-based estimation of eye gaze directions. Humans have the intuition to use the pupil centre positions in relation to the corners of eyes to determine the direction of eye gazes. Eye regions can be considered as two small rectangles located in the face plane. When the position of the corners of both eyes, LC_j and RC_j, are determined, we can define the centre points of the eyes S_j. The vertical field of view of the eyes is about 150°. The distance between the centre of the pupils P_j and the centre of the eyes S_j is determined by adding the calculated vector of the head pose $\boldsymbol{v}_{\text{head}}$ on the shift vector $\boldsymbol{d}_{\text{PS}}$ with two scaling factors k_{gaze} and k_{head}. The result is the respective gaze direction of both eyes. The algorithm corrects this result based on the head pose. One eye far away from the camera may not appear in the image when the head is moved relatively to the camera, thus we add a weight w_{pose} to determine the importance of head poses in the direction of gaze, which points away from the camera as described in Algorithm 2.

3 EXPERIMENTS AND RESULTS

All components are considered separately or independently of each other and then statistically evaluated on the basis of various image datasets and compared to the current state of art. Experiments are conducted in dynamic environments in real time and for side faces. It allows different interaction scenarios to be simulated and adjusted. The goal is to evaluate the reliability and accuracy of the whole tracking system and all associated components. Therefore, we neglect extremely side faces with an occlusion of more than a half of the face. The error function is given by the absolute mean error (AME):

$$\text{AME} = \frac{1}{N}\sum_{i=0}^{N} \mid g - \hat{g} \mid \tag{3}$$

where g denotes the facial feature values and $\hat{g}$ the Ground Truth values.

Algorithm 2: Eye gaze vector estimation.

Data: Eye corners LC_j and RC_j, pupil centre P_j for $j \in \{\text{l, r}\}$, where denotes left l and right left r eye, head pose vector $\boldsymbol{v}_{\text{head}}$, scaling factors for local eye gaze and head pose direction k_{gaze}, k_{head}

Result: Eye gaze vector $\boldsymbol{v}_{\text{gaze},j}$

w_{pose} according to $\boldsymbol{v}_{\text{head}}[0]$;
for $j \in \{l, r\}$ **do**
  Results as S_j of LC_j and RC_j; $\boldsymbol{d}_{\text{PS},j}$;
  $\boldsymbol{v}_{\text{gaze},j} = k_{\text{gaze}}\boldsymbol{d}_{\text{PS},j} + k_{\text{head}}\boldsymbol{v}_{\text{head}}$;
end
if $\boldsymbol{v}_{head}[0] > 0$ **then**
  $\boldsymbol{v}_{\text{gaze,right}} \leftarrow (1 - w_{\text{pose}})\boldsymbol{v}_{\text{gaze,right}} + w_{\text{pose}}\boldsymbol{v}_{\text{gaze,right}}$;
else
  $\boldsymbol{v}_{\text{gaze,left}} \leftarrow (1 - w_{\text{pose}})\boldsymbol{v}_{\text{gaze,left}} + w_{\text{pose}}\boldsymbol{v}_{\text{gaze,left}}$;
end

3.1 Facial Features Detection

The evaluation of facial features detection is based on the HELEN dataset (Le et al., 2012) and is compared to the Dlib-ml method (E. King, 2009). The HELEN dataset provides a large collection of annotated facial images, exhibiting a large variety in appearance as well as general images and environmental conditions. The facial landmark detector proposed inside Dlib-ml produces 68 coordinates of facial landmarks that map to specific facial structures. Through the annotations and Ground Truth data, both algorithms can be compared directly. We calculate the AME for the Ground Truth data of all previously detected facial features. Our first CNN (Multi-Data MTCNN) does not achieve the detection accuracy of the Dlib-ml. For all extracted facial key features the AME is smaller as shown in Figure 3. The largest deviation can be seen at the corners of the mouth. Due to the inaccurate recognition of our Multi-Data MTCNN, we refined and improved the inaccurate output by a second CorCNN in which is especially designed for the detection of eye corners in the eye regions. It turns out that the "CoordConv-layer" significantly improves the accuracy for the regression results of eye corners. We do not need an exact output of the corners of both eyes,

[1] https://www.kaggle.com/c/facial-keypoints-detection

because cascading the two CNNs improves the recognition accuracy for the eye regions sufficiently. Our coarse-to-fine approach realised by the two CNNs optimises the accuracy and reliability. We defined eye regions of the Multi-Data MTCNN as a new input for the CorCNN to improve the recognition of both eye corners. As underlined by our experiments, it is shown that the CorCNN detects the corners of the eye better than before, no matter in which direction the eyes or in which orientation the head is. Compared to the detection accuracy for the eye corners in Figure 3 e) and Figure 4 b), an improved detection is achieved as well with the cascaded two CNNs. Further examples are illustrated in Figure 3 f) Figure 4 a) and Figure 3 h) and Figure 4 c). Our approach is not as accurate in mouth detection as the Dlib-ml method, but we recognise facial features in side faces in which the Dlib-ml method does not recognise the landmarks. The multi view training of our Multi-Data MTCNN without the addition of the KCF can identify some side faces. If the head orientation converges to the limited angle for sides face during tracking, we switch to our KCF based tracker. Moreover, it improves the results of real time detection of the eye regions in side faces. For the permanent calculation of the head orientation, we achieve an average tracking rate of 20 ms. The Dlib-ml method is not robust against lateral faces because it is not designed for lateral faces, but the achieved tracking rate can be higher 18 ms. In addition, it is worth mentioning that our detection rate is 100 %, while Dlib-ml only scores to 96.8 %.

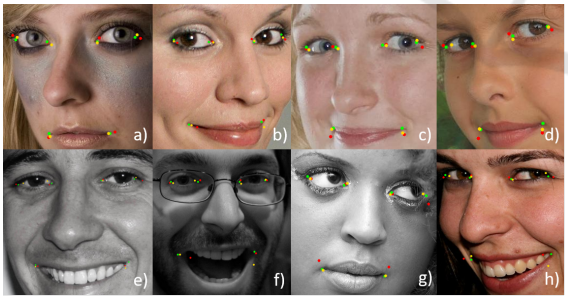

Figure 3: Detection results of facial features for our method (red), Dlib-ml (green) and Ground Truth Data (yellow).

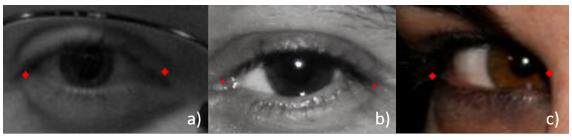

Figure 4: Coarse-to-fine results of our CorCNN.

3.2 Head Pose Estimation

For head pose estimations we tested with the Head Pose Images (Gourier et al., 2004) using the DeepGaze method (Patacchiola and Cangelosi, 2017). The Head Pose Images consists of 15 images of 15 different people, wearing glasses or not and of various skin colors. Each set contains 2 series of 93 images of the same person in different poses. The head orientation is determined by 2 angles, which varies from $-90°$ to $90°$. The head orientations are not continuously recorded, but distributed at intervals of $15°$. The average detection time of our method is about 44 ms and the from DeepGaze 5 ms. The difference can be explained by the need to first extract facial features with our Multi-Data MTCNN and then refine our coarse-to-fine CorCNN to improve the accuracy. As a comparison DeepGaze is a rapid end-to-end structure that directly outputs the head pose orientation. We only need the head orientation to distinguish between front faces and side faces and then we switch to our MTCNN and our KCF tracker. Figure 6 shows the AME and standard deviation (STD) for the two approaches. The angular deviations of a vertical (yaw) or horizontal (pitch) head rotation is significantly smaller by our method. The estimated average angular deviation of our method is $10°$ in comparison to $15°$ with a STD of $0.1°$ to $0.2°$. DeepGaze has a deviation greater than or equal to $25°$ for both angles with a STD of $0.2°$ to $0.4°$. Due to our high accuracy, the calculation time increases in comparison to DeepGaze.

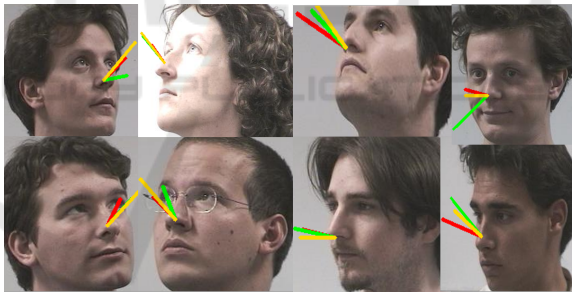

Figure 5: Head orientation results of our method (red), DeepGaze (green) and Ground Truth data (yellow).

3.3 Eye Gaze Direction

For the evaluation of eye gaze tracking, all previously determined facial features are transferred into the Facial Features Based Eye Gaze Model, see Figure 1. Our whole CNN-based tracking pipeline is then compared in terms of accuracy and robustness with the OpenFace approach (Amos et al., 2016) evaluated with the Columbia Gaze dataset (A. Smith et al., 2013). The Columbia Gaze dataset is a large publicly available gaze dataset with 5,880 images of 56 people (21 of them wearing glasses) over varying gaze directions and head poses. There are 5 head poses and 21 gaze directions per head pose. Figure 7

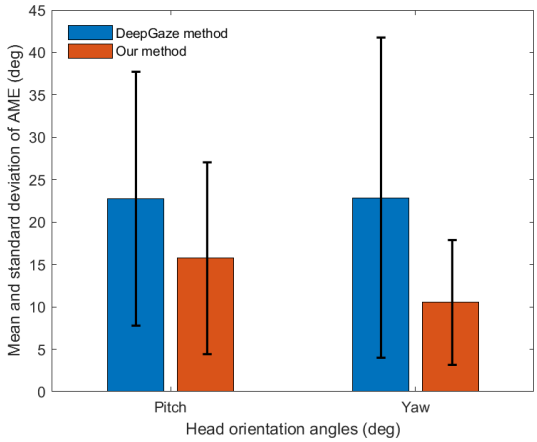

Figure 6: Mean and standard deviation of the head pose.

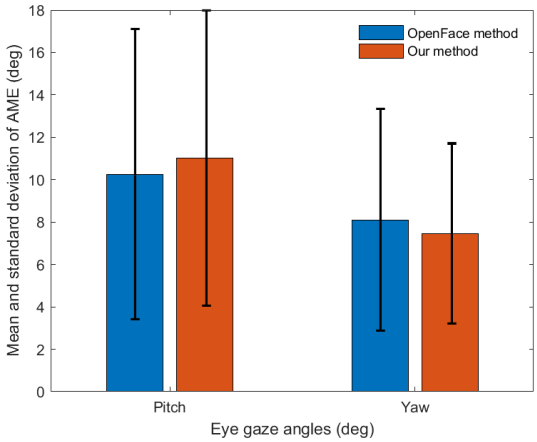

Figure 9: Absolute mean and standard deviation of eye gaze direction.

and Figure 8 show that there are no significant differences in the accuracy of the gaze direction between our approach and the OpenFace. However, our method is robust against head orientations, which shape lateral facial views in the image plane. The detection accuracy of OpenFace decreases in side faces. In contrast to OpenGaze, we achieve a stable detection rate for front faces as well as side faces. As illustrated in Figure 7 and 8, we estimate the gaze direction in most cases, but as shown in Figure 8 the black background seriously affected the pupil centre calculation, because they are computed based on the image gradient, and the black regions gain a large weight according to intensity values. Each method has its strengths and weaknesses in precision, but both are better in the pitch angle than in the yaw angle.

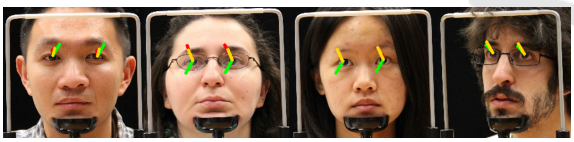

Figure 7: Eye Gaze results of our method (red), OpenFace (green) and Ground Truth data (yellow).

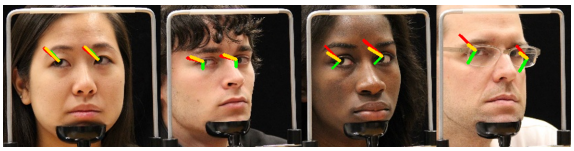

Figure 8: Eye Gaze results of our method (red), OpenFace (green) and Ground Truth data (yellow).

3.4 Dynamic Environments

For some applications, we focus on industrial work cells and service-oriented supermarket environments. In both environments various humans appear where some of them wear glasses and keep their heads in different orientations. Many people wear glasses that partially cover the human eyes or create reflections through reflective lenses. Our method robustly detects the eye gaze direction in many different scenarios where some of the humans wear glasses, refer to Figure 11. Many methods from the state of the art show inaccuracies and errors. By using the KCF, we can eliminate some of these errors, e.g. the limitation due to reflection of lenses have been reduced. This elimination is important, because landmark detection for the faces and eye regions have been previously inaccurate and produced a large error rate, which can be reduced by our new approach. Limitations of current methods concern tracking of eye gazes in lateral faces. All CNNs are based on training datasets, which consist of series of frontal images. Therefore, the tracking of faces seen in side views is difficult, because of missing training datasets. With previous solutions, facial occlusions often result in failures for eyes gaze estimation. Our method overcomes this problem by exploiting symmetry using the KCF instead of the facial features to calculate the eye gaze direction after a certain vertical head rotations. When faces are seen from the side view, half of the faces are occluded. Our algorithm assumes symmetry and calculates the eye gaze direction robustly. This solution works because the eyes perform a synchronous movement in the gaze direction normally. Seen in Figure 10, the eye gaze is always only for the non-occluded eye. In the middle lower picture in Figure 10 it can be seen, that the symmetry of the gaze can be projected onto the other eye.

4 CONCLUSIONS

We propose a CNN-based facial features gaze tracking system to track eye gaze based on facial features and head orientations in dynamic environments. It is able to estimate human eye gazes as a significant measurement for the ability of humans to interact with their counterparts. Our approach recognises eye gazes in extreme conditions robustly and with a good relationship between accuracy and real time performance. Our angle-based approach of the head orientation differentiates between front faces and side faces and allows us to switch specifically between the different CNNs and the detection of image filters. With the aid of the suggested cascaded CNNs, the eye regions can be further refined and the detection of facial features are improved in order to counteract reflections and by masking in a robust manner. Furthermore, distinctions can be made in the future between open and closed eyes in order to improve safety in human-robot interaction.

Figure 10: Eye gaze tracking for side faces.

Figure 11: Eye gaze tracking when wearing eyeglasses.

REFERENCES

A. Smith, B., Yin, Q., K. Feiner, S., and K. Nayar, S. (2013). Gaze locking: Passive eye contact detection for human-object interaction.

Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. Technical report.

Chinsatit, W. and Saitoh, T. (2017). Cnn-based pupil center detection for wearable gaze estimation system. 2017:1–10.

Deng, H. and Zhu, W. (2017). Monocular free-head 3d gaze tracking with deep learning and geometry constraints. *2017 IEEE International Conference on Computer Vision*, pages 3162–3171.

Dollár, P., Welinder, P., and Perona, P. (2010). Cascaded pose regression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085.

E. King, D. (2009). Dlib-ml: A machine learning toolkit. 10:1755–1758.

Fischer, T., Jin Chang, H., and Demiris, Y. (2018). Rt-gene: Real-time eye gaze estimation in natural environments. In *The European Conference on Computer Vision*.

Gee, A. and Cipolla, R. (1994). Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647.

George, A. and Routray, A. (2016). Real-time eye gaze direction classification using convolutional neural network. pages 1–5.

Gourier, N., Hall, D., and Crowley, J. L. (2004). Estimating face orientation from robust detection of salient facial structures. In *FG Net Workshop on Visual Observation of Deictic Gestures*.

Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). Interactive facial feature localization. In *Computer Vision*, pages 679–692.

Lemley, J., Kar, A., Drimbarean, A., and Corcoran, P. (2018). Efficient cnn implementation for eye-gaze estimation on low-power/low-quality consumer imaging systems.

Li, B. and Fu, H. (2018). Real time eye detector with cascaded convolutional neural networks. 2018:1–8.

Liu, Y., Wu, Q., Tang, L., and Shi, H. (2017). Gaze-assisted multi-stream deep neural network for action recognition. *IEEE Access*, 5:19432–19441.

Mukherjee, S. S. and Robertson, N. M. (2015). Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11):2094–2107.

Palermo, R. and Rhodes, G. (2007). Are you always on my mind? a review of how face perception and attention interact. *Neuropsychologia*, 45(1):75–92. The Perception of Emotion and Social Cues in Faces.

Parks, D., Borji, A., and Itti, L. (2015). Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision Research*, 116:113–126. Computational Models of Visual Attention.

Patacchiola, M. and Cangelosi, A. (2017). Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. 71.

Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 20(1):23–38.

Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483.

Yang, S., Luo, P., Loy, C. C., and Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. *CoRR*, abs/1509.06451.

Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). Wider face: A face detection benchmark. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878.

Zhao, X., Meng, C., Feng, M., Chang, S., and Zeng, Q. (2018). Eye feature point detection based on single convolutional neural network. *IET Computer Vision*, 12(4):453–457.