# Accurate and Robust Eye Center Localization via Fully Convolutional Networks

Yifan Xia, Hui Yu, Fei-Yue Wang*

***Abstract*— Eye center localization is one of the most crucial and basic requirements for some human-computer interaction applications such as eye gaze estimation and eye tracking. There is a large body of works on this topic in recent years, but the accuracy still needs to be improved due to challenges in appearance such as the high variability of shapes, lighting conditions, viewing angles and possible occlusions. To address these problems and limitations, we propose a novel approach in this paper for the eye center localization with a fully convolutional network (FCN), which is an end-to-end and pixels-to-pixels network and can locate the eye center accurately. The key idea is to apply the FCN from the object semantic segmentation task to the eye center localization task since the problem of eye center localization can be regarded as a special semantic segmentation problem. We adapt contemporary FCN into a shallow structure with a large kernel convolutional block and transfer their performance from semantic segmentation to the eye center localization task by fine-tuning. Extensive experiments show that the proposed method outperforms the state-of-the-art methods in both accuracy and reliability of eye center localization. The proposed method has achieved a large performance improvement on the most challenging database and it thus provides a promising solution to some challenging applications.**

***Index Terms*—Human-computer interaction, eye tracking, eye gaze estimation, eye center localization, deep learning, FCN.**

## I. Introduction

EYE center localization refers to localizing the centers of human's pupil on given face images. Locating these centers means that we could establish correspondence between two eyes of the person and the focused targets, which has been proven to be useful for computer vision and human computer interaction tasks such as eye gaze estimation and eye tracking. Eye center localization is the first step towards eye gaze tracking and estimation in images and video [1]. During the process of eye gaze estimation and tracking, we need to determine the precise pixel location of important key points of the eye center for a single given RGB image. Moreover, achieving accurate eye center localization is useful for higher level tasks [2-7] such as human attention control, driver monitoring system and sentiment analysis, and also serves as a fundamental tool in fields such as human computer interaction and animation.

Eye center localization has been an interesting topic in the field of computer vision in recent years. There are many factors that can affect performance of the eye center localization such as the significant variability situation of eye appearance from different illumination, shape, color and viewing angles. A good eye center localization system must be accurate and robust to these factors. Early works tackle such difficulties using specialized devices like infrared cameras or head-mounted devices. This kind of devices is very popular in commercial areas since they could apply infrared illumination to localize the eye centers through corneal reflections. In that case, these devices could obtain a high accurate eye center location. However, it has some limitations in applications such as the high cost devices and the uncomfortable wearing experience. Compared with these specialized devices, the approaches which directly localizing key point positions of eye center through computer vision and image processing techniques are more efficient since they only need a low-cost webcam instead of specific hardware devices and can be easily implemented. This method is often used as an alternative approach of infrared illumination in terms of the high accuracy and robustness.

The success of deep learning methods for various computer vision tasks in recent years motivates us to investigate it in the task of eye center localization [65-68]. Traditional methods have recently been reshaped by emerging deep learning techniques, which are the main driver behind an explosive rise in performance across many computer vision tasks [69-74]. Fully convolutional network (FCN) has been proved to be successful not only in object semantic segmentation tasks, but also in other applications such as image classification or object detection. However, deep learning has rarely been mentioned and used for eye center localization. Therefore, in this paper, we introduce a novel end-to-end and pixels-to-pixels method for the eye center localization via FCN.

The designed FCN takes an entire image of face as input and the predicted heatmaps as output. And then we transform the predicted heatmaps to landmark coordinates to get the eye center location. The designed network follows two design principles: 1) we design a shallow structure rather than a deep one,

This work was supported by National Natural Science Foundation of China (61533019, U1811463), the Open Fund of the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences (Y6S9011F51), and the EPSRC project (EP/N025849/1).

Y. Xia, and H. Yu are with the School of Creative Technologies, University of Portsmouth, Portsmouth, PO1 2DJ, UK (e-mail: Yifan.Xia@myport.ac.uk; hui.yu@port.ac.uk).

F.-Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feiyue@ieee.org).

*Corresponding Author: F.-Y. Wang, E-mail: feiyue@ieee.org

which makes a good balance between performance and computational resources due to limited publicly available databases with accurate eye center annotations. 2) inspired by [53] and [54], we use a large kernel convolutional block instead of stacking small size ($1 \times 1$ or $3 \times 3$). The key idea is based on the assumption that the eye center localization can be considered as a special semantic segmentation problem. For the eye center localization and semantic segmentation task, images are taken as the input, but the output of the former task is coordinates of landmarks and the latter one is the object's class at each pixel. Thus, the key to implementing this assumption is that we need to establish correspondence between coordinates of landmarks and object's class at every pixel. To this end, we preprocess the images in which the coordinates of the eye centers are first transformed to a heatmap using Gaussian kernels. Then the problem becomes estimating the value of the heatmap at each pixel, which is equivalent to the semantic segmentation problem, where the goal is to estimate the object's class at each pixel. Thanks to the strong performance of FCN for semantic segmentation, we design a shallow FCN network, which is similar to the one in [8] with a large kernel convolutional block and finetune it to transfer their performance from semantic segmentation to the eye center localization task. The detailed experimental results show that the proposed approach outperforms state-of-the-art methods for eye center localization in terms of accuracies and reliability.

The major contributions of this work are as follows:

- We design a fully convolutional network (FCN) with a shallow structure and a large kernel convolutional block to accurately locate the eye center, which well balances the performance and the computational costs.
- We regard the problem of eye center localization as a special semantic segmentation problem, which is a novel and important solution regarding the key and future directions for this area of research.

Here is a brief introduction of the structure of this paper. In Section II, we describe the related work on eye center localization and fully convolutional network. In Section III, we describe the methodology about our proposed network. We show results of experiment on the public dataset to evaluate the performance of our proposed method and other existing methods in Section IV. Finally, Sections V and VI are the general discussion and conclusion.

## II. Related Work

This section reviews related works on eye center localization and fully convolutional networks.

**Eye center localization** Localizing the eye center is a critical requirement for eye gaze estimation and eye tracking and has attracted a huge interest in recent years. Existing works for eye center localization can be roughly divided into three categories: 1) appearance-based methods, (2) model-based methods, and (3) hybrid methods. Early works tackle this problem mainly using appearance-based methods, which use priori eye knowledge about appearance information such as the color, circle structure and other geometric characteristics of the eye to localize the eye center [9, 10, 11]. Valenti and Gevers [12] proposed a method using the isophote curvature method according to circle shape of eye to localize the eye center. Moreover, based on the circle property of the eye, the means of gradient method proposed by Timm and Barth [13] is a milestone in the development of eye center localization tasks. It can localize the eye center by calculating the dot product of gradient vector and displacement vector. Based on means of gradient method, there are many improved or similar methods over recent years like [14,15]. Asadifard *et al.* [16] proposed a method based on the cumulative density function (CDF), which mainly filters the image to determine which pixel is the eye center. A method proposed by Leo *et al.* [17] used the local variability of the appearance and image intensities to determine the eye center. Araujo *et al.* [19] described an Inner Product Detector for eye localization based on correlation filters. The appearance-based methods have achieved good performance, but under some challenging scenarios like poor illumination they are not robust and accurate enough. Zhang *et al.* [47] introduced a modular approach making use of isophote and gradient features simultaneously to estimate the eye center locations. Villanueva *et al.* [48] proposed a method to detect the eye center using a multiresolution and topographic method. George *et al.* [51] used geometrical characteristics for eye center localization. Choi *et al.* [52] reviewed the local structure patterns (LSPs) and extended them by using several hybrid local structure patterns (LSPs) for accurate eye detection.

Model-based and hybrid methods are alternative solutions for eye center localization. Model-based methods mainly use machine learning algorithms. It first extract key features of images to train a model regarding appearance or structures of eye and then fit the learned model to determine eye centers. Many machine learning algorithms have been used for eye center localization such as Bayesian models [20], hidden Markov models (HMMs) [21], support vector machines (SVM) [22, 23, 24] and AdaBoost [25]. Kim *et al.* [26] localized eye centers using a multi-scale approach, which was based on Gabor vectors. A multi-layer perceptron was used by Jesorsky *et al.* [27] to determine the position of eye center. Kroon *et al.* [28] employed a Fisher Linear Discriminant to filter the face image and selected the highest responses as the eye center. Chen *et al.* [29] used a hierarchical FloatBoost and MLP classifier simultaneously to localize the eye center. Cristinacce *et al.* [30] used Active Appearance Model (AAM) to find the eye center positions. Behnke [31] proposed a hierarchical network with local recurrent connectivity for this task. A cascade regression model was trained by Gou *et al.* [32, 49, 50] using synthetic photorealistic data, which was used to determine the eye center. Markus *et al.* [33] localized the eye pupil by using an ensemble of randomized regression trees. Chen *et al.* [34] used clustering-based discriminant analysis (CDA) models to localize the eye center. Ren *et al.* [35] proposed a codebook of invariant local features

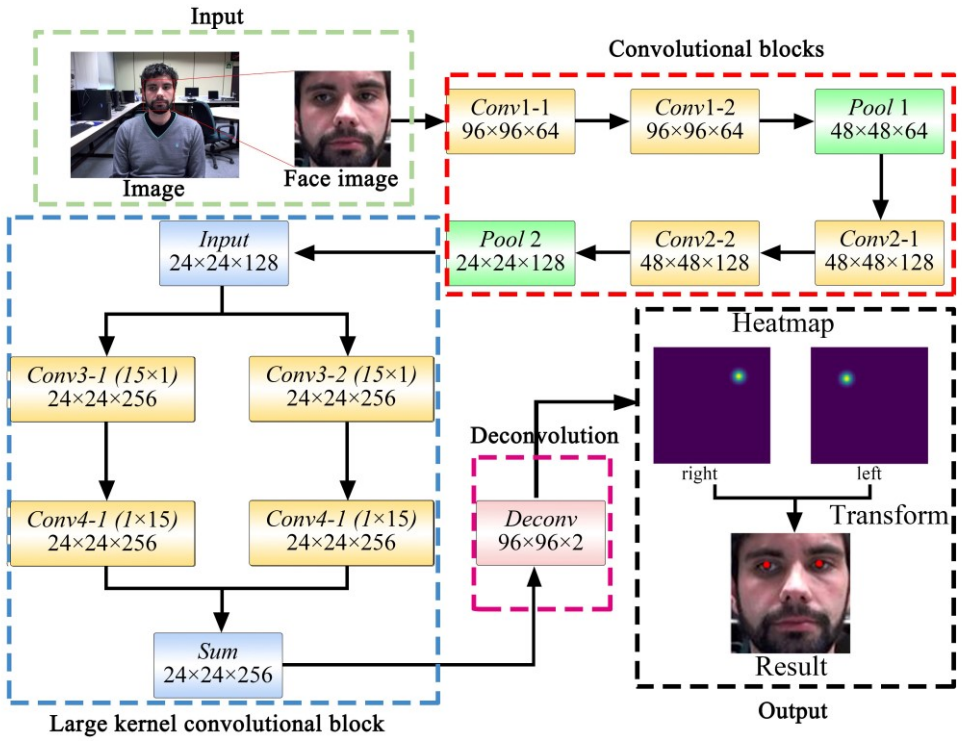


Fig. 1. Overview of our method for eye center localization using shallow fully convolutional network. First, given an input image, the image is cropped to the size of face bounding box provided by face detection algorithm. Then, the face image is fed into shallow FCN with a large kernel convolutional block. Then the feature map is mapped to each pixel by deconvolution operation to predict the per-pixel eye region. And the network outputs the heatmap. Finally, the heatmap generated by the network is transformed back to the normal landmark coordinates. In this way, we can get the final results of eye center position.

and a pyramid-like sparse representation classifier to locate the eyes. Hamouz *et al*. [36] used a GMM-based feature detector and an enhanced appearance mode to localize the eye center. Compared with appearance-based methods, the model-based method is more robust. However, this kind of methods relies on lots of annotated training data, which is difficult to obtain in many cases. Hybrid methods integrate the advantages of appearance-based and model-based method simultaneously in one method like [37,38]. In order to deal with occlusions of the eyelids under certain lighting conditions, Valenti *et al.* [39] proposed a hybrid method using mean shift and machine learning algorithm to improve their previous isophote method [12].

**Fully Convolutional Network** In the domain of deep learning, fully convolutional network (FCN) is widely used for semantic segmentation to predict object's class at each pixel in an image according to its semantic meaning. Semantic segmentation is one of the most active research areas over recent years in computer vision. Early works [40, 41] mainly relied on low level or hand-crafted features to generate the label map to solve this problem. Fully Convolutional Networks (FCN) proposed by Long *et al*. [8] is a special variant of Convolutional Neural Networks. This method is an encoder-decoder architecture taking the existing CNNs model like VGG-16 as powerful tools to learn hierarchical features, which transform these models into a fully convolutional form by replacing the original fully connected layers with convolutional layers. Then upsampling or deconvolution is used to output the class of prediction for each pixel. FCN is the first end-to-end and pixel-wise predicting model, which provides a novel and milestone solution and opens a new research area for semantic segmentation. It is also the foundation for other contemporary semantic segmentation algorithms. Based on the principle of FCN, many variations have been proposed for semantic segmentation over recent years [42, 55, 56, 57]. Note that all the aforementioned methods are used for semantic segmentation. Recently, however, the FCN-like network structure has been also applied successfully to other keypoint detection problems such as human pose estimation [58], facial landmark detection [59] and eye gaze estimation [60, 61]. They all have an encoder-decoder architecture and used a FCN-like network structure called hourglass network which borrows the idea from FCN.

The method proposed in this paper is inspired by both the semantic segmentation task and FCN, which regard eye center localization as a special semantic segmentation task. Therefore, we design a shallow FCN network with a large kernel convolutional block to overcome the limitations of previous works for eye center localization. It is a feasible and high-efficiency solution for eye center localization, which leads to high performance outperforming many existing state-of-the-art methods.

## III. Methodology

In this paper, we mainly focus on designing a network to achieve the task of localization of left eye center and right eye center. In this section, we give a detailed description of the proposed deep learning approach for eye center localization. Fig. 1 shows the brief flowchart of the proposed method. We design a

shallow FCN network inspired by [8, 53, 54] with a large kernel convolutional block. The major advantage of our approach is that we regard the eye center localization as a special semantic segmentation problem. And the transformation of the image to a heatmap allows the network of semantic segmentation to focus on the landmark detection of the eye center.

### A. Preprocessing

The key to transforming eye center localization to the sematic segmentation problem is the preprocessing stage. The images of the training set are first cropped based on the face bounding box provided by the database. Knoche *et al*. [43] researched the effect of the image resolution on performance of facial landmark prediction and found that there was a decline of performance when the image resolution is smaller than 50×50 px. We thus, resize all the cropped face images to be an equal size of 96×96px. And then we transform all the processed images to a gray level for a stable performance. This can also improve the efficiency in processing and training.

Finally, according to the landmarks of eye centers, we transform these images to a heatmap as inputs of the network. Note that the successful use of the network of sematic segmentation on eye center localization heavily depends on the generation of heatmap. We transform each landmark to a single heatmap using Gaussian kernel. For the eye center localization problem, there are two landmarks (left and right eye center). This means that we need to generate 2 heatmaps for each eye image, which can be interpreted as a grayscale image in the range [0,1]. The ground-truth landmark coordinates are set to white and the other position as black. In other words, a black heatmaps indicates that some landmarks are not recorded, so all pixels on this heatmap are set to 0. We use two formulas based on the Gaussian kernel to generate heatmaps of eye center landmarks:

$$H_l = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x-x_l)^2+(y-y_l)^2}{2\sigma^2}\right) \quad (1)$$

$$H_r = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x-x_r)^2+(y-y_r)^2}{2\sigma^2}\right) \quad (2)$$

where $(x_l, y_l)$ and $(x_r, y_r)$ are the ground truth landmarks of left and right eye center, $H_l$ and $H_r$ are corresponding values of the heatmap at position $(x, y)$ of the image. And σ is the standard deviation. The value of σ is an important parameter, which needs to be appropriately adjusted. The choice of σ is important to get sensible results. If the value of σ is too small, the heatmap becomes too sparse (mostly zero). If the value of σ is too big, the trained model focuses too much on estimating coordinates of other positions instead of eye center positions. For generating heatmaps, we set $\sigma = 3$, which achieves the best results in our experiment. The further discussion can be found in Section IV. Fig. 2 shows the examples of the generated heatmaps of left and right eye center.

### B. Network Architecture

In this section, we introduce the proposed network architecture. We use the VGG16-FCN [8] architecture as a basis for developing our eye center localization network. Classical CNN uses the convolutional layers to extract local features in an image. On the top of convolutional layers, the fully connected layers use the inner product operation to integrate high-level local feature maps into a single feature vector to predict the label of each image. Therefore, it is not able to predict the label for each pixel. Recently, the trend has shifted towards using FCN to solve the dense prediction of each pixel. FCN is a special type of CNN, which replaces all fully connected layers with the convolutional layers and adds additional upsampling or deconvolution layers.

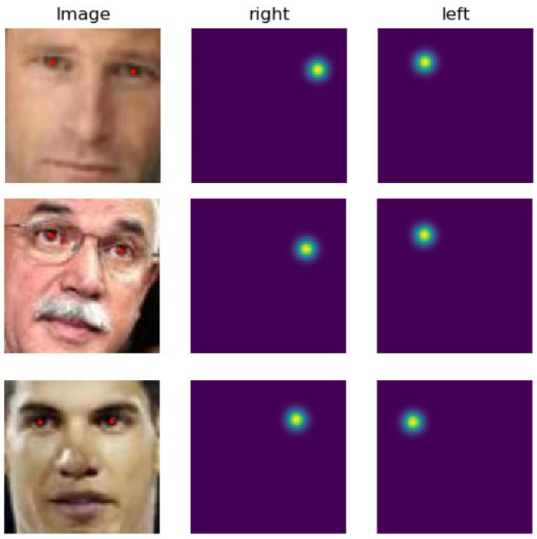


Fig. 2. The sample heatmaps generated using Gaussian kernel.

After upsampling or deconvolution layers, the output feature maps of the network can be transformed to probability maps with sigmoid outputs $f_{y_i}$ by passing through a perceptron layer. $f_{y_i}$ represents the probability of predicting class $y_i$ at pixel $i$. The final results of class prediction $\hat{y}_i$ can be represented as the formula:

$$\hat{y}_i = \underset{y_i \in Y}{\operatorname{argmax}} f_{y_i} \quad (3)$$

where $Y$ is a set of possible categories. Unlike a typical CNN, FCN could perform end-to-end and pixel-to-pixel classification and output a tensor of pixel-wise class predictions without additional post-processing. The spatial size of the tensor is equal to the input image, which is implemented by using several upsampling or deconvolution layers. Since the output of the deep layer lacks location and edge clues, the FCN combines feature maps of deep and shallow layers to obtain finer results called "FCN-xs" (like FCN-8s). For more details on FCN, see [8].

Our network architecture is a shallow and simplified version of FCN with a large kernel convolutional block, which is also an encoder-decoder structure shown in Fig. 1. In this work, we use the first two convolutional blocks from VGG16-FCN [8] for encoders. Each convolutional block includes two convolutional layers and one Maxpooling layer. The parameters are set as the same as those in [8]. And the remaining layers of VGG16-FCN are discarded. The numbers of channels of two

convolutional blocks at different resolutions are 64 and 128 respectively.

For traditional network architectures, stacking convolutional blocks with small size kernels ($1 \times 1$ or $3 \times 3$) in the entire network is more efficient than using large kernels. However, in the experiment, Zhou et al. [53] proposed the concept of valid receptive field (VRF) and claimed that the sizes of the actual receptive were always smaller than the theoretical receptive fields for traditional network architectures. Based on this work, Peng et al. [54] concluded that the large kernel size which could lead to more effective receptive field played an important role in the field of semantic segmentation and could improve the performance.

Inspired by [54], we propose to use large kernel convolutional blocks in our network after the outputs from previous encoder convolutional blocks. However, the direct use of a large kernel size will increase the computational burden due to the large number of parameters. In our method, we employ a simulation of a $K \times K$ convolutional kernel including a combination of one $K \times 1$ convolutional kernel and one $1 \times K$ convolutional kernel to replace the direct use of a large kernel size. For the large kernel convolutional block, we set $K = 15$ and use two path convolutional operations shown in Fig. 1. Each path contains two convolutional layers with kernel the size of $15 \times 1$ and $1 \times 15$ respectively. Therefore, the output feature maps of the convolutional block in the encoder pass through a large kernel convolutional block with a kernel size of 15 and a filter number of 256 for a large receptive field. After two path convolutional operations, we aggregate the feature maps of two paths. Finally, the output of the large kernel convolutional block is upsampled with a deconvolution layer, which is used to output the prediction results. The input of the network is the cropped face image and the heatmap. The cropped face image input to the network is a grey level image with a resolution of $96 \times 96$ px. The ground truth label generated by using the Gaussian kernel is to generate heatmaps of two eye center landmarks. The output of the network is the heatmap with the size of $96 \times 96$ px.

The training procedure for eye center localization is similar to the one training FCN for sematic segmentation, which uses the images and labels of each pixels as input. We use heatmaps as the labels, which are generated by using landmarks of eye centers. Landmarks of eye centers are encoded using the Gaussian kernel to generate heatmaps at the provided location of the eye center landmarks. Each eye center landmark has its own heatmap and allows the network to distinguish between two points more easily. The eye center localization network training is formulated as a per-pixel regression problem based on the ground-truth segmentation masks. Formally, the objective function can be represented as the following formula:

$$\varepsilon(\theta) = \sum_p e\big(X_\theta(p), l(p)\big) \qquad (4)$$

where $p$ is the index of the pixel, $l(p)$ is the ground truth heatmap which represents ground truth label of the pixel, and $X_\theta(p)$ is the predicted heatmap which indicates estimated label

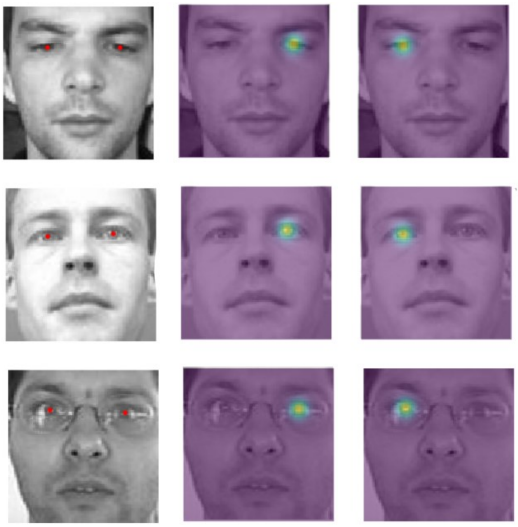

Fig. 3. Example output produced by our network. On the left we see the final eye center positions provided by weighted average method across each heatmap. On the right we show sample heatmaps (From left to right: right eye center and left eye center).

predicted by the fully convolutional network with parameters $\theta$. The network parameters $\theta$ are updated by using RMSprop optimizer. And $e\big(X_\theta(p), l(p)\big)$ is the loss function.

During the training stage, all parameters $\theta$ are learned and updated via minimizing a loss function, which are computed as errors between the predicted heatmap and the ground truth data. Usually, the Mean Squared Error (MSE) loss function is used for this kind of problems. However, research shows that the use of an asymmetric weighted loss can improve the performance when the data is unbalanced [18] [63]. Since the number of eye centers and non-center pixels are imbalanced, we compute a weighted MSE in our experiment. Given an image $I$ with a size of $h \times w$, we can get ground truth heatmap $G \in [0,1]^{h \times w}$ using the gaussian kernel, and the network predicts a heatmap $P \in [0,1]^{h \times w}$. During the training procedure, the variant of MSE loss function of the proposed network is thus given by

$$L_{(P,G)} = \frac{1}{h \times w} \sum_{i=1}^{h \times w} ((1-\alpha)(p_i - g_i)^2 + \alpha((1-p_i) - (1-g_i))^2) \qquad (5)$$

where $g_i \in G$ and $p_i \in P$ represent the ground truth and prediction of each pixel location, respectively. And $\alpha$ refers to the weight. We set $\alpha = 0.15$ empirically, which achieves the best results in our experiment.

### C. *From Heatmaps to Coordinates*

In order to evaluate the training model performance, we need to transform heatmaps generated by the network to the normal landmark coordinates as shown in Fig. 3. To this end, a straightforward method is to use the landmark coordinates of the pixel

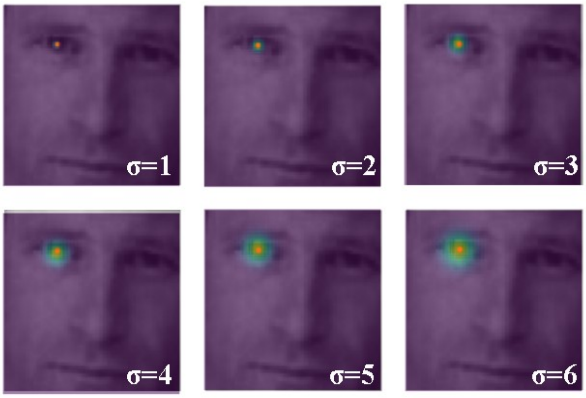


Fig. 4. The generated heatmap using different values of standard deviation σ.

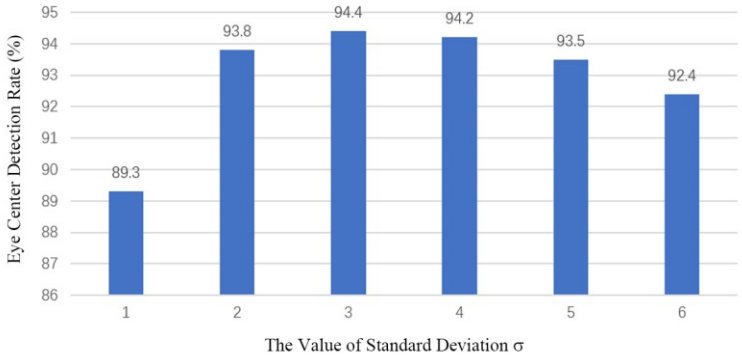


Fig. 5. Experimental results on different values of standard deviation σ on BioID database. The eye center detection rate is evaluated by maximum normalized error. And Y coordinate denotes the rate with the normalized error less than 0.05.

with the largest estimated density in the heatmap as the estimated landmark coordinates. We find that this method usually works well, but sometimes it is not accurate enough and thus results in outliers. To solve this issue, we use the weighted average of these coordinates corresponding to the pixels instead. To improve the accuracy and reduce the impact of outliers, the result of weighted average is further refined by considering only those pixels with the top *N* largest estimated density. Therefore, the problem becomes how to determine the value of *N*, which is used to calculate the weighted average of the coordinates. In our experiment, we set *N*=36 which achieves the best performance for eye center localization.

## IV. Experimental Results

In this section, we first introduce the database including the training and test set, experimental settings and the evaluation metric. We have also compared with other existing state-of-the-art methods on the public database including BioID [46] and GI4E [48].

### A. Database

In the experiment, we use the database from [44] as the training set. This dataset consists of 13,466 face images from real-world conditions, among which 5,590 images are selected from LFW database [45] while the remaining 7,876 images are downloaded from the web. These facial images have a clear difference in shape, expression and occlusions. Each face in this database is manually labeled with 5 landmarks including left and right eye center. We only use landmarks of left and right eye center in our experiment.

Moreover, for a fair comparison with other existing state-of-art methods, we choose two public databases as the test set and evaluate the proposed method on this database. The test set BioID [46] is composed of 1,521 grey level images taken from 23 different subjects under various illumination, poses and locations. This database is regarded as the most challenging and realistic databases and then widely used for eye center localization. Images of this database have a low-resolution size of 286 × 384px. The left and right eye center of each image are labeled in this database.

Another test set GI4E [48] contains 1236 high quality RGB images from 103 subjects with 12 different gaze directions. These images have a resolution of 800 × 600px, which are similar to images acquired by a normal camera. The eye centers are also labeled in this database. Both of the training and test sets are challenging and realistic in terms of appearance, pose, expression and occlusion.

### B. Evaluation criteria

In the testing stage, performance is measured with the maximum normalized error [27], which is the standard evaluation metric for eye center localization. It indicates the accuracy and reliability of each algorithm by calculating the maximum error from the worst estimations of both eyes. The detection error is measured as

$$err = \frac{max\left(\sqrt{(x_l'-x_l)^2+(y_l'-y_l)^2}, \sqrt{(x_r'-x_r)^2+(y_r'-y_r)^2}\right)}{\sqrt{(x_l-x_r)^2+(y_l-y_r)^2}} \quad (6)$$

where $(x_l', y_l')$ and $(x_l, y_l)$ are the estimated position and the ground truth of left eye center, and $(x_r', y_r')$ and $(x_r, y_r)$ refer to that of the right eye center. During evaluation, if the maximum normalized error is larger than 0.25, it is regarded as failure. There are some special thresholds which are meaningful and usually used to evaluate algorithms for eye center localization: $err = 0.05 \approx$ the diameter of pupil; $err = 0.10 \approx$the diameter of iris; $err = 0.25 \approx$ the distance between the eye center and the eye corners. Therefore, in order to estimate the eye center point located in the eye region, the error should be less than or equal to 0.25.

### C. Experimental Settings

All images used in our experiment including the training and test set are cropped using a bounding box to obtain a clear face area. And then the cropped face images are further processed to gray level images with a size of 96×96px. And the ground truth heatmap of two eye center landmarks is generated by using the

TABLE 1. THE PERFORMANCE OF OUR PROPOSED APPROACH ON BIOID AND GI4E DATABASE

| | $err \leq 0.05$ | $err \leq 0.10$ | $err \leq 0.25$ |
|---|---|---|---|
| ***BioID-Max*** | *94.4%* | *99.9%* | *100%* |
| ***BioID-Min*** | *98.9%* | *100%* | *100%* |
| ***BioID-Avg*** | *96.9%* | *100%* | *100%* |
| ***GI4E-Max*** | *99.1%* | *100%* | *100%* |
| ***GI4E-Min*** | *100%* | *100%* | *100%* |
| ***GI4E-Avg*** | *99.8%* | *100%* | *100%* |

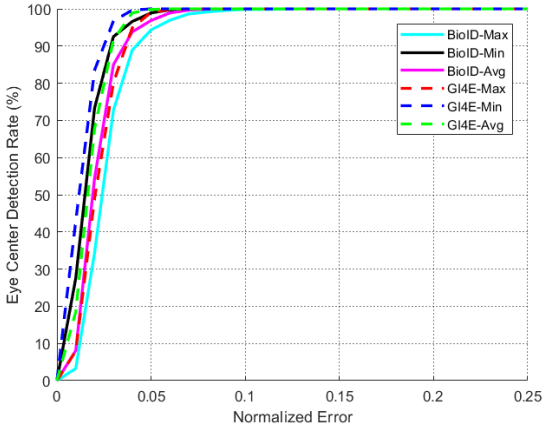


Fig. 6. Normalized error curves of the proposed approach on the BioID and GI4E database.


Fig. 7. Qualitative results of our approach on the BioID and GI4E database. The images are sorted according to the maximum normalized error. The red points represent the estimated eye center positions by our proposed approach. And the green points represent the ground truth. (a) For the BioID database, the first two rows are the best which $err \leqslant 0.05$ and $0.05 \leqslant err \leqslant 0.10$, and the bottom row is the worst which $0.10 \leqslant err \leqslant 0.25$. Note that all the results within the eye region which meet minimum standards of the eye center localization ($err \leqslant 0.25$). (b) For the GI4E database, the first two rows are the best which $err \leqslant 0.05$, and the bottom row is the worst which $0.05 \leqslant err \leqslant 0.10$. Note that all the results of this database meet $err \leqslant 0.10$.

Gaussian kernel with a size of 96×96px.

During the training process, we use 10,000 images from database [44] for training and 3466 images for validation. But for deep learning methods, the amount of data has a significant impact on the performance. But to the best of our knowledge, only a few databases provide annotations of eye centers, which are not enough to support the FCN network. Therefore, we use a data augmentation method to augment the available training images to improve the model performance on validation data. After splitting, the training set is augmented via affine transformation including rotation (+/- 30 degrees) and scaling (.75-1.25) and horizontal flipping to increase size of the training set.

In our experiment, the network is trained using TensorFlow on a desktop PC with the specification of an Intel Core i7 at 4.20GHz processor, 16 GB of RAM memory and 8 GB NVIDIA GeForce GTX 1080 GPU. And we use RMSprop with a leaning rate of 5e-4 for optimization and set the batch size to 32. A weighted Mean Squared Error (MSE) loss is computed comparing with the predicted heatmap to the ground truth heatmap generating from a 2D gaussian kernel (with standard deviation $\sigma = 3$) on the eye centers. To improve the performance of transforming heatmaps to coordinates, we use a weighted average of the top $N$=36 largest estimated density instead of the largest one.

### *D. Quantitative Results*

We first explore the impact of standard deviation σ on generating the heatmaps for our method. As mentioned in Section III, we use the Gaussian kernel to generate heatmaps of eye center landmarks according to Eq. (1) and Eq. (2) and set $\sigma = 3$ in our experiment. The impact on the generated heatmap using different parameters σ shown in Fig. 4. From the Fig. 4, we can find that the size of heatmaps of the eye center consistently increases with standard deviation σ. It demonstrates the changes of performance of eye center localization with different values of σ. We have tested different values of standard deviation σ ranging from 1 to 6 and obtained the performance on the BioID database shown in Fig. 5. From the results we can see

that for eye center localization, when $\sigma \leq 3$, a larger value of standard deviation σ will result in better performance yet for $\sigma \geq 4$ the performance drops. The possible reason for this is that a very high value of standard deviation σ could lead to too many values of non-eye center positions in the generated heatmap which reduce the effectiveness of heatmaps.

Based on the previous choice of standard deviation σ, we have obtained the overall performance of the proposed approach on BioID [46] and GI4E [48] database using three metrics shown in Table. 1 including the maximum normalized error, the minimum normalized error and the average normalized error. And we also show their corresponding normalized error curves in Fig. 6. For quantitative results, we mainly focus on metric of the maximum normalized error in this work.

For the BioID database, the maximum normalized error in Table. 1 shows that our approach can reach an accuracy of 94.4%($err \leqslant 0.05$) which means that the estimated eye centers are located within the pupil with a high probability. Moreover, Table. 1 also shows the accuracy of 99.9%($err \leqslant 0.10$) indicating that eye centers estimated by our approach well lie within the iris. Finally, our approach yields an accuracy of 100%($err \leqslant 0.25$) for localizing eye center, which means that the method could meet the minimum standards of the eye center localization and all the estimated eye center points locate exactly within the eye region. Table. 1 shows that our method has good performance with the accuracy of 99.1%($err \leqslant 0.05$), 100%($err \leqslant 0.10$), 100%($err \leqslant 0.25$) on the GI4E database.

To further demonstrate the overall performance of our method, we also use the minimum normalized error and the average normalized error to evaluate the performance to give an upper bound and an average error. The minimum normalized error and the average normalized error replace the maximum function in Eq. (6) with the minimum and average function respectively. In Table. 1, we can find that accuracy of almost all errors are 100%, indicating a reliable accuracy for eye center localization.

### *E. Qualitative Results*

The qualitative results of the proposed approach on BioID and GI4E database are shown in Fig. 7. The red points are used to represent the estimated eye center positions by the proposed approach. And the green points represent the ground truth positions of eye center provided by the database. The first two rows show a selection of images of different subjects with various

TABLE 2. COMPARISON OF OUR METHOD WITH OTHER METHODS ON BIOID DATABASE (BOLD VALUE INDICATES BEST ACCURACY)

| Method | $err \leq 0.05$ | $err \leq 0.10$ | $err \leq 0.15$ | $err \leq 0.20$ | $err \leq 0.25$ |
|---|---|---|---|---|---|
| Asteriadis *et al.* [9] | 44.0% | 81.7% | 92.6% | 96.0% | 97.4% |
| Zhou *et al.* [10] | - | - | - | - | 94.8% |
| Bai *et al.* [11] | 37.0% | 64.0% | - | - | 96.0% |
| Timm and Barth [13] | 82.5% | 93.4% | 95.2% | 96.4% | 98.0% |
| Cai *et al.* [14] | 84.1% | 95.6% | - | - | 99.8% |
| Xia *et al.* [62] | 87.1% | 98.7% | - | - | 99.9% |
| Valenti *et al.* [12] | 84.1% | 90.9% | 93.8% | 97.0% | 98.5% |
| Soelistio *et al.* [15] | 80.8% | 95.2% | 97.8% | 98.9% | 99.4% |
| Leo *et al.* [17] | 80.7% | 87.3% | 88.8% | 90.9% | - |
| Leo *et al.* [64] | 78.0% | 86.0% | - | - | 90.0% |
| Asadifard *et al.* [16] | 47.0% | 86.0% | 89.0% | 93.0% | 96.0% |
| Araujo *et al.* [19] | 88.3% | 92.7% | 94.5% | 96.3% | 98.9% |
| Niu *et al.* [25] | 75.0% | 93.0% | 95.8% | 96.4% | 97.0% |
| Chen *et al.* [29] | - | 89.7% | - | - | 95.7% |
| Jesorsky *et al.* [27] | 38.0% | 78.8% | 84.7% | 87.2% | 91.8% |
| Gou *et al.* [32] | 89.2% | 98.0% | - | - | 99.8% |
| Behnke [31] | 37.0% | 86.0% | 95.0% | 97.5% | 98.0% |
| Markus *et al.* [33] | 89.9% | 97.1% | - | - | 99.7% |
| Kim *et al*. [26] | - | 96.4% | - | - | 98.8% |
| Everingham *et al*. [20] | 45.87% | 81.35% | - | - | 91.21% |
| Ren *et al*. [35] | 77.08% | 92.25% | - | - | 98.99% |
| Campadelli *et al*. [23] | 80.7% | 93.2% | - | - | 99.3% |
| Chen *et al*. [24] | 88.79% | 95.2% | - | - | 98.98% |
| Chen *et al*. [34] | 87.3% | 94.9% | - | - | 99.2% |
| Hamouz *et al.* [22] | 58.6% | 75.0% | 80.8% | 87.6% | 91.0% |
| Kroon *et al.* [28] | 65.0% | 87.0% | - | - | 98.8% |
| Cristinacce *et al.* [30] | 57.0% | 96.0% | 96.5% | 97.0% | 97.1% |
| Hamouz *et al.* [36] | 50.0% | 66.0% | - | - | 70.0% |
| Turkan *et al.* [37] | 18.6% | 73.7% | 94.2% | 98.7% | 99.6% |
| Campadelli *et al.* [38] | 62.0% | 85.2% | 87.6% | 91.6% | 96.1% |
| Valenti *et al.* [39] | 86.1% | 91.7% | - | - | 97.9% |
| Zhang *et al.* [47] | 85.7% | 93.7% | - | - | 99.2% |
| Gou *et al.* [49] | 91.2% | 99.4% | 99.6% | - | 99.8% |
| Gou *et al.* [50] | 92.3% | 99.1% | 99.7% | - | - |
| George *et al.* [51] | 85.1% | 94.3% | 96.7% | 98.1% | - |
| Choi *et al.* [52] | 91.1% | 98.4% | - | - | 99.7% |
| **Our Method** | **94.4%** | **99.9%** | **100%** | **100%** | **100%** |

TABLE 3. COMPARISON OF OUR METHOD WITH OTHER METHODS ON GI4E DATABASE (BOLD VALUE INDICATES BEST ACCURACY)

| Method | $err \leq 0.05$ | $err \leq 0.10$ | $err \leq 0.15$ | $err \leq 0.20$ | $err \leq 0.25$ |
|---|---|---|---|---|---|
| Timm and Barth [13] | 92.4% | 96% | 96.9% | - | 97.5% |
| Villanueva *et al.* [48] | 93.9% | 97.3% | 98% | - | 98.5% |
| Zhang *et al.* [47] | 97.9% | 99.6% | - | - | 99.9% |
| Gou *et al.* [32] | 98.2% | 99.8% | - | - | 99.8% |
| Gou *et al.* [49] | 94.2% | 99.1% | 99.6% | - | 99.8% |
| George *et al.* [51] | 89.3% | 92.3% | 93.6% | 94.2% | - |
| Gou *et al.* [50] | 98.3% | 99.8% | 99.8% | - | - |
| **Our Method** | **99.1%** | **100%** | **100%** | **100%** | **100%** |

TABLE 4. COMPARISON OF OUR METHOD WITH OTHER METHODS IN AVERAGE PROCESSING TIME.

| Method | Araujo *et al.* [19] | Leo *et al.* [17,64] | Gou *et al.* [32,49] | Gou *et al.* [50] | Our Method |
|---|---|---|---|---|---|
| **Time(ms)** | 83 | 333 | 67 | 63 | **5** |

poses, facial expressions, occlusions and lighting conditions. Row three shows the worst results estimated by the proposed approach due to occlusion from glasses, strong reflection and shadows making pupils invisible. Nevertheless, our method could obtain accurate eye center points locating exactly within the eye region and meet the minimum standards of the eye center localization ($err \leqslant 0.25$).

For BioID database, the results of the first two rows demonstrate that the proposed method is very accurate and robust under different challenging situations such as closed eyes, occlusion from glasses or hair, affection from shadows and far away from the camera. All estimated eye centers using the proposed method fall within the corresponding eye region, which meets the minimum standards ($err \leqslant 0.25$) of the eye center localization. It is worth noting that even the four worst examples demonstrated are not failure cases. For GI4E database, it has similar performance on qualitative results to BioID database, reaching an accuracy of 100% when $err \leqslant 0.10$ which is more accurate than BioID database.

### *F. Comparison with Existing Approaches*

We have extensively compared the proposed approach with the state-of-art methods on BioID and GI4E database using the maximum normalized error as the metric. The comparison results are shown in Table. 2 and Table. 3.

BioID is one of the most widely used databases with low quality images for eye center localization. Many previous research results are available and easy to compare with using the same experiment protocol. In order to further investigate the overall performance of the proposed method, we first show results of 36 state-of-art methods for eye center localization including appearance-based, model-based and hybrid method which include almost all-important eye center localization methods published in recent years. Furthermore, we also increase the number of thresholds *err* of the evolution metric of previous research. We employ five types of *err* thresholds {0.05,0.10,0.15,0.20,0.25} instead of {0.05,0.10,0.25}.

The comparison results between our approach and state-of-art methods on BioID database are shown in Table. 2. From the Table. 2, we can make the following observations. Firstly, it is obvious that the proposed approach achieves the best performance for all kinds of thresholds *err* on BioID database compared with existing methods. Secondly, it is worth noting that the proposed approach obtains an accuracy of 94.4% at $err \leqslant 0.05$. This is a milestone achievement, since the most majority of existing methods maintains an accuracy of around 80% or lower. Finally, it shows that with the increasing maximum normalized error metric, the performance of the proposed method gets better. Compared with other methods, the accuracy of our method is the first one close to 100% as the maximum normalized error increase. Except for at $err \leqslant 0.05$, the accuracy is almost 100% when it comes to $err \leqslant 0.10$, $err \leqslant 0.15$, $err \leqslant 0.20$ and $err \leqslant 0.25$.

GI4E is another evaluation database for eye center localization, which contains images with high quality taken by normal cameras. The results on GI4E are listed in Table. 3. We have compared the proposed method with 7 state-of-art methods. As shown in Table. 3, the performance on GI4E database is better than that on BioID in general. The proposed approach still achieves the best performance on GI4E database with the accuracy of 94.4%($err \leqslant 0.05$), 99.9%($err \leqslant 0.10$), 100%($err \leqslant 0.15$), 100%($err \leqslant 0.20$) and 100%($err \leqslant 0.25$) respectively. What's more, the accuracy of our method is the only one that can achieve 100% accuracy at $err \leqslant 0.10$.

Another important consideration in evaluating the algorithm for eye center localization is its computational complexity. The computational complexity is measured by average processing time for each input image. We have conducted a comparison in the processing time of locating the eye centers on BioID database. We train a network model of the proposed method first through a desktop PC. And then deploy it on a standard laptop with an Intel Core i5 at 2.50GHz processor and 16GB of RAM memory for eye center localization. The comparison of our method and other methods in average processing time is shown in Table 4. In our experiment, the proposed method is more efficient and faster than all other state-of-the-art methods, taking 5ms per image on average. This shows that our proposed

method is suitable for real time applications and embedded systems.

## V. Discussion

In this paper, the proposed FCN approach shows a significant improvement for eye center localization, which is more robust and accurate on the BioID and GI4E database. It can maintain enough accuracy especially for images with visible pupils, open eyes, no strong reflection and no occlusion.

We notice that several issues still need to be discussed. First, there is still a space to improve performance. The proposed FCN is a shallow and simplified version in terms of the architecture. Due to limited training databases which provide annotations of eye centers and limited computational resources, it is only possible to design a shallow network rather than a deep one. Therefore, adding more training data or synthetic data and using deeper networks such as hourglass networks could potentially improve the performance for eye center localization. But at the same time the training time and complexity will also increase. We thus need to find a balance between performance and efficiency. Second, despite the MSE loss could have a minimum value, it cannot guarantee that the improvement of MSE loss could lead to the improvement of results for localizing eye centers. This is because the MSE loss function is used directly for optimizing the metric for the whole heatmaps instead of the eye centers. And we need to further transform the predicted heatmap to coordinates. Third, the generation and transformation of heatmaps is crucial to the proposed approach. We use the Gaussian kernel to generate heatmaps during the process of preprocessing and employ the weighted average method to transform generated heatmaps to coordinates in the testing stage. Though both Gaussian kernel and weighted average method work well in this case, more effective methods or strategies could lead to a better performance. The proposed method cannot handle perfectly for cases such as closed eyes, occlusion from glasses or hair and affection from shadows. As shown in Fig. 7, the position of eye centers can still be improved.

## VI. Conclusion

In this paper, we propose an accurate and robust network architecture for eye center localization via a shallow FCN with a large kernel convolutional block. The key idea is regarding the eye center localization as a special semantic segmentation problem, which leads to the transformation of heatmaps of eye center positions. In the preprocessing stage, we first use the Gaussian kernel to generate heatmaps of eye center landmarks, which are then used to train the network. In the testing stage, we transform the heatmaps generated by the network to coordinates to evaluate the performance. Our experimental results on testing database show that the proposed approach outperforms the state-of-the-art methods. We understand that more training dataset will potentially improve the performance.

In the future, we will explore the use of synthetic data as an alternative solution to this problem [50, 75]. And a deeper and complex network architecture and a more efficient strategy for the transformation of heatmaps will be explored to improve the performance though at the cost of more computation.

## Acknowledgment

This work was supported by National Natural Science Foundation of China (61533019, U1811463). This work was supported by the Open Fund of the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences (Y6S9011F51). This work was supported by the EPSRC project (EP/N025849/1).

## References

[1] Cai, Haibin, Hui Yu, Xiaolong Zhou, and Honghai Liu. "Robust gaze estimation via normalized iris center-eye corner vector." In International conference on intelligent robotics and applications, pp. 300-309. Springer, Cham, 2016.

[2] Lin, Min, Bin Li, and Qiao-Hong Liu. "Identification of eye movements from non-frontal face images for eye-controlled systems." International Journal of Automation and Computing 11, no. 5 (2014): 543-554.

[3] Xing, Yang, Chen Lv, Zhaozhong Zhang, Huaji Wang, Xiaoxiang Na, Dongpu Cao, Efstathios Velenis, and Fei-Yue Wang. "Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition." IEEE Transactions on Computational Social Systems 5, no. 1 (2018): 95-108.

[4] Yu, Hui, and Honghai Liu. "Regression-based facial expression optimization." IEEE Transactions on Human-Machine Systems 44, no. 3 (2014): 386-394.

[5] Liu, Zhentao , et al. "A facial expression emotion recognition based human-robot interaction system." IEEE/CAA Journal of Automatica Sinica 4.4(2017):668-676..

[6] Cyganek, Bogusław, and Sławomir Gruszczyński. "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring." Neurocomputing 126 (2014): 78-94.

[7] Jang, Young-Min, Rammohan Mallipeddi, Sangil Lee, Ho-Wan Kwak, and Minho Lee. "Human intention recognition based on eyeball movement pattern and pupil size variation." Neurocomputing 128 (2014): 421-432.

[8] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.

[9] Asteriadis, Stylianos, Nikos Nikolaidis, Andras Hajdu, and Ioannis Pitas. "An eye detection algorithm using pixel to edge information." In Int. Symp. on Control, Commun. and Sign. Proc. 2006.

[10] Zhou, Zhi-Hua, and Xin Geng. "Projection functions for eye detection." Pattern recognition 37, no. 5 (2004): 1049-1056.

[11] Bai, Li, Linlin Shen, and Yan Wang. "A novel eye location algorithm based on radial symmetry transform." In 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, pp. 511-514. IEEE, 2006.

[12] Valenti, Roberto, and Theo Gevers. "Accurate eye center location and tracking using isophote curvature." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. IEEE, 2008.

[13] Timm, Fabian, and Erhardt Barth. "Accurate eye centre localisation by means of gradients." Visapp 11 (2011): 125-130.

[14] Cai, Hai-Bin, Hui Yu, Chun-Yan Yao, Shen-Yong Chen, and Hong-Hai Liu. "Convolution-based means of gradient for fast eye center localization." In Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 759-764. IEEE, 2015.

[15] Soelistio, Yustinus Eko, Eric Postma, and Alfons Maes. "Circle-based eye center localization (CECL)." In 2015 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 349-352. IEEE, 2015.

[16] Asadifard, Mansour, and Jamshid Shanbezadeh. "Automatic adaptive center of pupil detection using face detection and cdf analysis." In Proceedings of the international multiconference of engineers and computer scientists, vol. 1, p. 3. 2010.

[17] Leo, Marco, Dario Cazzato, Tommaso De Marco, and Cosimo Distante. "Unsupervised eye pupil localization through differential geometry and local self-similarity matching." PloS one 9, no. 8 (2014): e102829.

[18] Mostajabi, Mohammadreza, Payman Yadollahpour, and Gregory Shakhnarovich. "Feedforward semantic segmentation with zoom-out features." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3376-3385. 2015.

[19] Araujo, Gabriel M., Felipe ML Ribeiro, Eduardo AB Silva, and Siome K. Goldenstein. "Fast eye localization without a face model using inner product detectors." In 2014 IEEE International Conference on Image Processing (ICIP), pp. 1366-1370. IEEE, 2014.

[20] Everingham, Mark, and Andrew Zisserman. "Regression and classification approaches to eye localization in face images." In 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 441-446. IEEE, 2006.

[21] Samaria, Ferdinando, and Steve Young. "HMM-based architecture for face identification." Image and vision computing 12, no. 8 (1994): 537-543.

[22] Hamouz, Miroslav, Josef Kittler, Joni-Kristian Kamarainen, Pekka Paalanen, Heikki Kalviainen, and Jiri Matas. "Feature-based affine-invariant localization of faces." IEEE transactions on pattern analysis and machine intelligence 27, no. 9 (2005): 1490-1495.

[23] Campadelli, Paola, Raffaella Lanzarotti, and Giuseppe Lipori. "Precise eye and mouth localization." International Journal of Pattern Recognition and Artificial Intelligence 23, no. 03 (2009): 359-377.

[24] Chen, Shuo, and Chengjun Liu. "Eye detection using discriminatory Haar features and a new efficient SVM." Image and Vision Computing 33 (2015): 68-77.

[25] Niu, Zhiheng, Shiguang Shan, Shengye Yan, Xilin Chen, and Wen Gao. "2d cascaded adaboost for eye localization." In 18th International Conference on Pattern Recognition (ICPR'06), vol. 2, pp. 1216-1219. IEEE, 2006.

[26] Kim, Sanghoon, Sun-Tae Chung, Souhwan Jung, Dusik Oh, Jaemin Kim, and Seongwon Cho. "Multi-scale gabor feature based eye localization." World Academy of Science, Engineering and Technology 21 (2007): 483-487.

[27] Jesorsky, Oliver, Klaus J. Kirchberg, and Robert W. Frischholz. "Robust face detection using the hausdorff distance." In International conference on audio-and video-based biometric person authentication, pp. 90-95. Springer, Berlin, Heidelberg, 2001.

[28] Kroon, Bart, Alan Hanjalic, and Sander MP Maas. "Eye localization for face matching: is it always useful and under what conditions?." In Proceedings of the 2008 international conference on Content-based image and video retrieval, pp. 379-388. ACM, 2008.

[29] Chen, Dan, Xusheng Tang, Zongying Ou, and Ning Xi. "A hierarchical floatboost and mlp classifier for mobile phone embedded eye location system." In International Symposium on Neural Networks, pp. 20-25. Springer, Berlin, Heidelberg, 2006.

[30] Cristinacce, David, Timothy F. Cootes, and Ian M. Scott. "A multi-stage approach to facial feature detection." In BMVC, vol. 1, pp. 277-286. 2004.

[31] Behnke, Sven. "Learning face localization using hierarchical recurrent networks." In International Conference on Artificial Neural Networks, pp. 1319-1324. Springer, Berlin, Heidelberg, 2002.

[32] Gou, Chao, Yue Wu, Kang Wang, Fei-Yue Wang, and Qiang Ji. "Learning-by-synthesis for accurate eye detection." In 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3362-3367. IEEE, 2016.

[33] Markuš, Nenad, Miroslav Frljak, Igor S. Pandžić, Jörgen Ahlberg, and Robert Forchheimer. "Eye pupil localization with an ensemble of randomized trees." Pattern recognition 47, no. 2 (2014): 578-587.

[34] Chen, Shuo, and Chengjun Liu. "Clustering-based discriminant analysis for eye detection." IEEE Transactions on Image Processing 23, no. 4 (2014): 1629-1638.

[35] Ren, Yan, Shuang Wang, Biao Hou, and Jingjing Ma. "A novel eye localization method with rotation invariance." IEEE Transactions on Image Processing 23, no. 1 (2014): 226-239.

[36] Hamouz, Miroslav, Josef Kittler, J-K. Kamarainen, Pekka Paalanen, and H. Kalviainen. "Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model." In Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings., pp. 67-72. IEEE, 2004.

[37] Türkan, Mehmet, M. M. Pardàs, and A. Enis Cetin. "Human eye localization using edge projections." In VISAPP 2007-2nd International Conference on Computer Vision Theory and Applications, Proceedings, no. IA/-, pp. 410-415. 2007.

[38] Campadelli, Paola, Raffaella Lanzarotti, and Giuseppe Lipori. "Precise Eye Localization through a General-to-specific Model Definition." In BMVC, vol. 1, pp. 187-196. 2006.

[39] Valenti, Roberto, and Theo Gevers. "Accurate eye center location through invariant isocentric patterns." IEEE transactions on pattern analysis and machine intelligence 34, no. 9 (2012): 1785-1798.

[40] Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. "Indoor segmentation and support inference from rgbd images." In European Conference on Computer Vision, pp. 746-760. Springer, Berlin, Heidelberg, 2012.

[41] Deng, Zhuo, Sinisa Todorovic, and Longin Jan Latecki. "Semantic segmentation of rgbd images with mutex constraints." In Proceedings of the IEEE international conference on computer vision, pp. 1733-1741. 2015.

[42] Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40, no. 4 (2018): 834-848.

[43] Knoche, Martin, Daniel Merget, and Gerhard Rigoll. "Improving facial landmark detection via a super-resolution inception network." In German Conference on Pattern Recognition, pp. 239-251. Springer, Cham, 2017.

[44] Sun, Yi, Xiaogang Wang, and Xiaoou Tang. "Deep convolutional network cascade for facial point detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3476-3483. 2013.

[45] Huang, Gary B., Marwan Mattar, Tamara Berg, and Eric Learned-Miller. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments." In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition. 2008.

[46] Bioid dataset, https://www.bioid.com/About/BioID-Face-Database.

[47] Zhang, Wenhao, Melvyn L. Smith, Lyndon N. Smith, and Abdul Farooq. "Eye center localization and gaze gesture recognition for human–computer interaction." JOSA A 33, no. 3 (2016): 314-325.

[48] Villanueva, Arantxa, Victoria Ponz, Laura Sesma-Sanchez, Mikel Ariz, Sonia Porta, and Rafael Cabeza. "Hybrid method based on topography for robust detection of iris center and eye corners." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 9, no. 4 (2013): 25.

[49] Gou, Chao, Yue Wu, Kang Wang, Kunfeng Wang, Fei-Yue Wang, and Qiang Ji. "A joint cascaded framework for simultaneous eye detection and eye state estimation." Pattern Recognition 67 (2017): 23-31.

[50] Gou, Chao, Hui Zhang, Kunfeng Wang, Fei-Yue Wang, and Qiang Ji. "Cascade learning from adversarial synthetic images for accurate pupil detection." Pattern Recognition 88 (2019): 584-594.

[51] George, Anjith, and Aurobinda Routray. "Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images." IET Computer Vision 10, no. 7 (2016): 660-669.

[52] Choi, Inho, and Daijin Kim. "A variety of local structure patterns and their hybridization for accurate eye detection." Pattern Recognition 61 (2017): 417-432.

[53] Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Object detectors emerge in deep scene cnns." arXiv preprint arXiv:1412.6856 (2014).

[54] Peng, Chao, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. "Large Kernel Matters--Improve Semantic Segmentation by Global Convolutional Network." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4353-4361. 2017.

[55] Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs." arXiv preprint arXiv:1412.7062 (2014).

[56] Zheng, Shuai, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. "Conditional random fields as recurrent neural networks." In Proceedings of the IEEE international conference on computer vision, pp. 1529-1537. 2015.

[57] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." In Proceedings of the IEEE international conference on computer vision, pp. 1520-1528. 2015.

[58] Newell, Alejandro, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation." In European Conference on Computer Vision, pp. 483-499. Springer, Cham, 2016.

[59] Yang, Jing, Qingshan Liu, and Kaihua Zhang. "Stacked hourglass network for robust facial landmark localisation." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 79-87. 2017.

[60] Park, Seonwook, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. "Learning to find eye region landmarks for remote gaze estimation in unconstrained settings." In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, p. 21. ACM, 2018.

[61] Park, Seonwook, Adrian Spurr, and Otmar Hilliges. "Deep pictorial gaze estimation." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 721-738. 2018.

[62] Xia, Yifan, Jianwen Lou, Junyu Dong, Gongfa Li, and Hui Yu. "SDM-based means of gradient for eye center localization." In 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp. 862-867. IEEE, 2018.

[63] Wang, Wenguan, Jianbing Shen, and Ling Shao. "Video salient object detection via fully convolutional networks." IEEE Transactions on Image Processing 27, no. 1 (2017): 38-49.

[64] Leo, Marco, Dario Cazzato, Tommaso De Marco, and Cosimo Distante. "Unsupervised approach for the accurate localization of the pupils in near-frontal facial images." Journal of Electronic Imaging 22, no. 3 (2013): 033033.

[65] Zheng, Naning. "The new era of artificial intelligence." CHINESE JOURNAL OF INTELLIGENT SCIENCE AND TECHNOLOGIES, 2019, 1(1): 1-3.

[66] Zhang, Bo. "Artificial intelligence is entering the post deep-learning era." CHINESE JOURNAL OF INTELLIGENT SCIENCE AND TECHNOLOGIES, 2019, 1(1): 4-6.

[67] Zhang, Jun Jason, Fei-Yue Wang, Xiao Wang, Gang Xiong, Fenghua Zhu, Yisheng Lv, Jiachen Hou et al. "Cyber-physical-social systems: The state of the art and perspectives." IEEE Transactions on Computational Social Systems 5, no. 3 (2018): 829-840.

[68] Wang, Fei-Yue, Yong Yuan, Juanjuan Li, Dongpu Cao, Lingxi Li, Petros A. Ioannou, and Miguel Ángel Sotelo. "From intelligent vehicles to smart societies: A parallel driving approach." IEEE Transactions on Computational Social Systems 5, no. 3 (2018): 594-604.

[69] Li, Li, Yisheng Lv, and Fei-Yue Wang. "Traffic signal timing via deep reinforcement learning." IEEE/CAA Journal of Automatica Sinica 3, no. 3 (2016): 247-254.

[70] Tian, Yonglin, Xuan Li, Kunfeng Wang, and Fei-Yue Wang. "Training and testing object detectors with virtual images." IEEE/CAA Journal of Automatica Sinica 5, no. 2 (2018): 539-546.

[71] Chen, Long, Xuemin Hu, Wei Tian, Hong Wang, Dongpu Cao, and Fei-Yue Wang. "Parallel planning: a new motion planning framework for autonomous driving." IEEE/CAA Journal of Automatica Sinica 6, no. 1 (2018): 236-246.

[72] Wang, Qiang, Xiaojing Yang, Zhigang Huang, Shiqian Ma, Qiao Li, David Wenzhong Gao, and Fei-Yue Wang. "A novel design framework for smart operating robot in power system." IEEE/CAA Journal of Automatica Sinica 5, no. 2 (2018): 531-538.

[73] Wang, Shiping, Jinyu Cai, Qihao Lin, and Wenzhong Guo. "An Overview of Unsupervised Deep Feature Representation for Text Categorization." IEEE Transactions on Computational Social Systems (2019).

[74] Chen, Rung-Ching. "User Rating Classification via Deep Belief Network Learning and Sentiment Analysis." IEEE Transactions on Computational Social Systems (2019).

[75] Yu, Hui, Oliver GB Garrod, and Philippe G. Schyns. "Perception-driven facial expression synthesis." Computers & Graphics 36, no. 3 (2012): 152-162.