IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2020, accepted March 22, 2020, date of publication April 2, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985095

Noise-Robust Pupil Center Detection Through CNN-Based Segmentation With Shape-Prior Loss

SANG YOON HAN, HYUK JIN KWON, YOONSIK KIM, (Student Member, IEEE), AND NAM IK CHO, (Senior Member, IEEE)

Department of Electrical and Computer Engineering, Institute of New Media and Communication (INMC), Seoul National University, Seoul 08826, South Korea

Corresponding author: Nam Ik Cho (nicho@snu.ac.kr)

This work was supported by the Samsung Electronics Company, Ltd.

ABSTRACT Detecting the pupil center plays a key role in human-computer interaction, especially for gaze tracking. The conventional deep learning-based method for this problem is to train a convolutional neural network (CNN), which takes the eye image as the input and gives the pupil center as a regression result. In this paper, we propose an indirect use of the CNN for the task, which first segments the pupil region by a CNN as a classification problem, and then finds the center of the segmented region. This is based on the observation that CNN works more robustly for the pupil segmentation than for the pupil center-point regression when the inputs are noisy IR images. Specifically, we use the UNet model for the segmentation of pupil regions in IR images and then find the pupil center as the center of mass of the segment. In designing the loss function for the segmentation, we propose a new loss term that encodes the convex shape-prior for enhancing the robustness to noise. Precisely, we penalize not only the deviation of each predicted pixel from the ground truth label but also the non-convex shape of pupils caused by the noise and reflection. For the training, we make a new dataset of 111,581 images with hand-labeled pupil regions from 29 IR eye video sequences. We also label commonly used datasets (*ExCuSe* and *ElSe* dataset) that are considered real-world noisy ones to validate our method. Experiments show that the proposed method performs better than the conventional methods that directly find the pupil center as a regression result.

INDEX TERMS Convex shape prior, deep learning, pupil segmentation, U-Net.

I. INTRODUCTION

Eye tracking or gaze tracking is one of the most important techniques for Human-Computer Interaction (HCI) and its applications. For example, it is essential for pointing (virtual) objects in AR/VR environment [1]–[3], detecting drowsiness that improves driver safety [4], [5], analyzing the human behavior such as eye-tracking heat map [6], [7], etc. These methods usually need real-time eye-tracking, for interacting with virtual objects or for the foveated image rendering in VR environments [8]. In most VR/AR devices, IR cameras are mounted inside the device, and the gaze-point is estimated by using features such as the shapes and positions of pupil, eyelash, eye corners, etc. Of course, finding the pupil's center point is the most important for the accurate estimation of gaze-point in many applications.

In indoor situations where external lights are blocked, acquired eye images usually have less noise. A relatively simple algorithm [9] can find necessary features such as pupil or eye corners when the eye areas are taken clearly with less noise. However, naive algorithms fail when pupils are not well seen due to occlusion by eyelid and eyelash, or when there are reflections around the pupil area. The reflections are caused by lights from displays in the case of VR devices, or by external ambient light in the case of AR devices. For example, Fig. 1 lists the challenging images such as strong reflections, occluded pupil region, and spurious pupil center. Non-learning-based methods attempted to address these problems by hiring adaptive edge detection or iterative methods such as StarBurst [10], ExCuSe [11], and ElSe [12]. But it seems that the performance of these methods is still insufficient for finding accurate gaze point in a noisy environment, as will be addressed in our experiments (see Fig. 9 and Table 1).

With the remarkable development of deep learning methods, there have also been several attempts to apply CNNs for pupil detection. For some examples, Fuhl *et al.* [13], Chinsatit and Saitoh [14], and Kondo *et al.* [15] proposed CNNs that take the eye image as the input and directly generate the

The associate editor coordinating the review of this manuscript and approving it for publication was Manik Sharma.

FIGURE 1. Highly challenging images in real-world scenarios that are affected by various factors: strong reflections, bad illumination, mascara, occluded pupil, recording errors, contact lenses, and additional black dot on iris.

gaze point. They showed improved results than the traditional approaches to a certain extent. However, the performance gain is not so significant, considering that the CNNs for other problems generally produce significant gains over the conventional feature engineering methods. Learning results for some sequences are even worse than traditional ones, as will be shown in the experiments (see Table 1).

To be more precise with the above mentioned CNN-based methods, they generally use the ConvNet [16] to produce the center or bounding box of pupils, following previous object detectors. Specifically, there have been many works that successfully use CNNs for the regression problem, especially for finding the bounding box of objects in video sequences [17]–[19]. However, it seems that the coordinates of box corners are usually fluctuating very much, which may not be a big problem in many object tracking applications [20], [21]. That is, just a few-pixel error in the bounding box corners is not much of concern in general object tracking problems. But, the estimation of the pupil center should be very accurate and robust to noise because a pixel difference results in quite a significant angle error in gaze direction as experimented in [22]–[24]. For example, a pixel error in estimating pupil center location may cause a gaze estimation error of 5 degrees or more when a user is gazing at the corner of the screen [25].

In the estimation problems, it is generally considered that the integration or ensemble approach makes the result more robust to the noise. Hence, we do not directly obtain the pupil center from the features of a CNN, but estimate all the pixels that may belong to the pupil and integrate the pixel positions

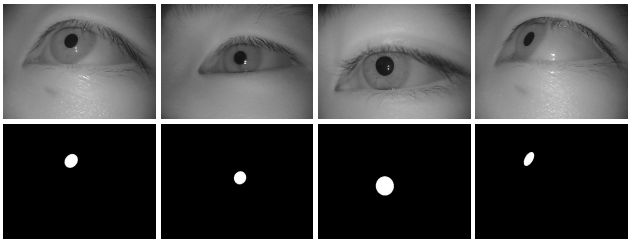


FIGURE 2. Our dataset and ground truth binary image for the pupil area segmentation.

to find its center. In other words, we use the CNN not for the pupil center regression but for the pupil segmentation, and then find the center of mass of the segment as the pupil center. For the CNN model, we use the UNet [26] instead of the plain VGG-style network, as it is shown to provide excellent performance in segmentation problem.

Meanwhile, the loss term commonly used in segmentation problems with fully convolutional networks (FCNs) [27]–[29] is the unary term that consists of pixel-wise cross-entropy. As the FCNs do not consider high-level label dependencies, the segmentation result usually suffers from noises due to occlusion, strong reflections, etc. Hence, DeepLab V2 [30] and Noh *et al.*'s methods [31] employed Fully Connected CRF [32] as a postprocessing tool to consider the prior knowledge of the shapes of target objects. Specifically, these methods combine information about the classifier output and local pixel priors such as shape, color, and position, which enforces the results to have similar properties to the ones with corresponding labels. Furthermore, instead of postprocessing, more recent researches encoded the shape prior into a loss function for directly manipulating the shape of output. Specifically, Lin *et al.* [33] and Tang *et al.* [34] chose to put a pairwise term in the loss term to consider the structure prior of the target object.

As will be shown in Section VI-B, our segmentation result using only pixel-wise cross-entropy is good enough compared to other methods. However, it cannot prevent failures for some challenging images, like others. Hence, we encode the prior knowledge on the shape of pupils, which should be elliptical [35], into the loss term with some relaxation. Specifically, we relax the elliptic-shape condition to the convex prior, and encode this term in addition to the cross-entropy. In the experiments, we will show the effectiveness of our new loss term with shape prior (see Fig. 11).

Also, since existing datasets provide only the pupil center as the ground truth label, we make a dataset with the binary segmentation map of the pupil area, as shown in Fig. 2. In addition, we also make pupil-region maps for commonly used datasets (ExCuSe dataset, ElSe dataset) that are regarded as real-world ones containing many challenging images, as shown in Fig. 3.

In summary, we believe that our contributions are as follows.

1) Unlike the existing CNN-based pupil detection methods based on the regression, we use a CNN for the pupil segmentation and obtain its center as the center of mass.

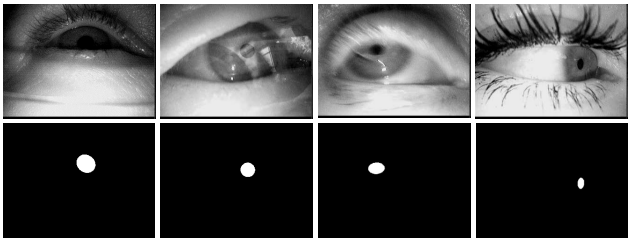


FIGURE 3. *ExCuSe* & *ElSe* datasets for which we annotate binary segmentation map.

This method is shown to be robust to reflection, noise, and occlusion.

2) We propose a new loss term that encodes convex shape prior for the training of segmentation networks, which further increases the accuracy of pupil center estimates.
3) We make a new dataset for the pupil segmentation and also add the annotations to the existing IR eye image datasets.

We published a preliminary work of pupil center estimation in a conference [36], where we also segmented the pupil region by UNet. The difference of this work from our conference version is that we propose a new loss term that brings better performance. Also, we annotate more segmentation map for the existing datasets and perform more experiments to validate the performance of our indirect segmentation-based method.

II. RELATED WORK

A. TRADITIONAL NON-LEARNING-BASED PUPIL DETECTION ALGORITHMS

Many pupil detection algorithms have been proposed so far, which are used mainly for the gaze estimation or for biometrics applications to locate the iris or face. Some algorithms for gaze estimation and iris recognition use active IR lightings for capturing eye images, which incurs reflections on the image. Hence, one of the well-known non-learning-based algorithms, StarBurst [10] removes the reflection region using the adaptive threshold algorithm and employs an iterative method to estimate the edge of the pupil. Then the pupil center is found by the ellipse fitting using RANSAC [37]. A more recently developed ExCuSe method [11] employs edge detection and morphology operations for detecting and polishing the pupil area. The algorithm first calculates the brightness histogram of the image, and if there is a peak in the histogram, it finds the pupil by edge analysis. If there is no peak in the histogram, the Angular Integral Projection Function (AIPF) [38] is applied to extract the pupil contour. In the case of another widely used method ElSe [12], it also uses the Canny-edge filtered image similar to ExCuSe [11]. The algorithm finds features such as the straightness and the inner intensity value from the detected connected edge components. If the algorithm finds a valid ellipse that describes the pupil through these features, it derives this ellipse as a result. Otherwise, the algorithm derives the pupil center through a proper convolution and thresholding. This algorithm shows state-of-the-art performance among the non-learning based algorithms. However, its pupil detection rate (with under 5-pixel error tolerance) is only under 70% in the presence of severe noise, especially on the ExCuSe and ElSe dataset. We use datasets from these non-learning methods as they are still widely used in recent learning methods. However, since they provide only pupil center position as a ground truth label, we annotate segmentation maps on these data for using them in our work.

B. LEARNING-BASED PUPIL DETECTION ALGORITHMS

It is well expected that the learning-based methods would bring better results than the non-learning ones. Considering the performance gains brought by CNNs in many object tracking and detection problems, there have also been several methods to apply the CNN to the pupil detection. For example, like recent object detectors that find bounding box over an object of interest, Chinsatit and Saitoh [14] used the CNN-based regression method. Specifically, they take the eye image as the input to the CNN, which directly produces the pupil center as a regression result. However, as mentioned above, this method is not robust to the noise, and hence the center point fluctuates quite widely. This is because the regression network based on the VGG-style CNN iteratively applies the pooling layer to grow the receptive field too much compared to the pupil size. Precisely, typical regression networks such as ResNet or ConvNet apply five max-pooling layers of size 2×2, and the receptive field of feature pixel on the last fully convolutional layer is 32. This seems to be too larger than the pupil diameter, considering that the size of input IR eye images is usually very small (VGA size or less). Also, since the pixel intensities inside and around pupil are generally near zero, there are too few features to be used. Due to the too-large receptive field and too-few features, the network gives any point near the center as the pupil center. In the case of the PupilNet [13], they tried to detect the pupil using a CNN classification model. By applying the sliding window technique, the patches corresponding to the pupil region are classified as 1 and the others as 0. This method is based on the multiscale approach, i.e., it first downscales the original image and finds a position of maximum network output as a coarse location of the pupil. In the same way, they find a fine position in the cropped region of original images. The reason for resizing the input is to reduce the noise and also to see more contexts around the pupil.

Unlike these CNN-based methods, we perform pixel-wise segmentation based on the belief that integrating many classification results will bring the point estimation more robust to noise. That is, instead of regressing the center point or patch-wise segmentation, we perform the pupil region segmentation and then find the pupil center as the center of mass of the region.

C. SHAPE-PRIOR TERM FOR THE LOSS FUNCTION

As mentioned in Section I, there are several ways to impose some constraints to the loss function for the segmentation

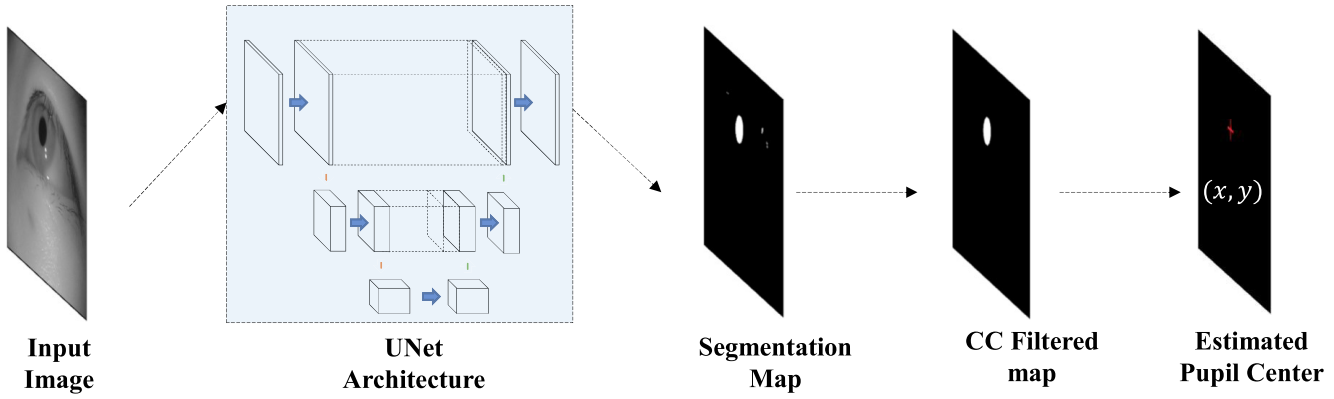


FIGURE 4. The overall workflow of the proposed method.

network. We first review how to directly affect the result of object segmentation by applying a loss term that embeds prior knowledge on the targeting object's properties. For this purpose, the typical loss function is composed of two terms: a unary term that confines the pixel-wise output to the annotated label, and a pairwise term that constrains the segment shape. Let us denote the pixel space as Ω, one of the pixels of Ω as p or q, the sigmoid output of p of the i-th image in a training image set I as S_{ip}, and the ground truth label of the pixels as y_{ip}. Then, an example of the loss function can be written as

$$\sum_{i\in I,\ p\in\Omega} l\left(S_{ip}, y_{ip}\right) + \sum_{i\in I,\ p,q\in\Omega} W^i_{pq}\|S_{ip} - S_{iq}\|_n \qquad (1)$$

where l is any pixel-wise loss function on S_{ip} and y_{ip}. The second term is the pairwise term that penalizes disagreements between the pair of p and q, with predefined affinity $W = W^i_{pq}$ between the points. The term $\|S_{ip} - S_{iq}\|_n$ measures the label compatibility between the points for encoding the object's constraints, which can be defined in several ways. For example, the affinity $W = \sum_{m=1}^{M} w^{(m)} k_G^{(m)}(f_{ip}, f_{iq})$ of the pairwise term that encodes CRF-RNN [39] uses Gaussian kernels $k_G^{(m)}(f_{ip}, f_{iq}) = \exp(\frac{1}{2}(f_{ip} - f_{iq})^T \Lambda^{(m)}(f_{ip} - f_{iq}))$ that depends on relative positions and relative intensities between the pair of f_{ip} and f_{iq}. In this case, the label compatibility function $\|S_{ip} - S_{iq}\|_n$ is replaced by the Potts model [40], $\mu(S_{ip}, S_{iq}) = [S_{ip} \neq S_{iq}]$. Norm cut [34] also uses the affinity W that encodes the sum of ratios between the cuts and the volumes based on RGBXY Gaussian kernel, but the weights are divided by the volume of each segment area. Hence, a quadratic relaxation of the Potts model $S_{ip} * (1 - S_{iq})$ is used as a label compatibility function. In [41], they defined "Star Shape Prior," by setting $W = [y_{ip} = y_{iq}]$ as a conditional term between two points p and q. The two points are on the line extended from the center of the segmented object so that the loss penalizes a non-star shaped object. Also, $|S_{ip} - S_{iq}|$ is chosen as a label compatibility function in star shape prior.

D. PRIOR KNOWLEDGE ON CONVEX POLYGON

As stated previously, the ideal shape of a pupil is circular, which appears elliptic in the images acquired by the perspective camera. Hence, previous non-learning methods usually fit the edges of the pupil to an ellipse. However, the appearance of the pupil in the image is not perfectly modeled as an ellipse; rather, it has a near-elliptic shape. Hence, in our method, we relax the shape to a convex polygon to allow for some deviation from the ellipse and use the shape-prior term that enforces the segmentation result to be a convex polygon.

The definition of a convex polygon is that all internal angles $\angle\gamma$ of the polygon $\mathbb{O}$ are less than π. However, this definition requires complex computation to determine polygon convexity, and cannot directly penalize pixels inside the polygon. Hence, we use another equivalent definition of convexity as follows:

- For every $p, q \in \Omega_{\mathbb{O}}$, every r on the line segment $\in l_{pq}$ is in the polygon region $\Omega_{\mathbb{O}}$.

This property can be easily encoded as an equation that directly penalizes the pixels inside the segmented area, as will be discussed in Section III-B.

III. PROPOSED METHOD

As stated in the introduction, we attempt to derive a robust pupil center estimation method by integrating the information from the deep network. For this, we do not directly use the deep network to regress the pupil center point but to classify the pupil region, and then calculate the center of mass of the region. The overall workflow on the inference phase is shown in Fig. 4. In the first stage, we downscale the resolution of the image obtained from the IR camera by half (320×240) and give it as an input to the network. In the second stage, we perform the segmentation using the sigmoid output obtained from the network, i.e., we threshold the magnitude of the final feature map. At the third stage, we perform the connected component (CC) analysis, which is to connect the same-labeled neighboring pixels into a blob, and find the largest blob $\mathbb{O}$, which is considered the pupil area. At the last stage, we find the center of the largest blob as a pupil center. When there exists no segmented area, we may select the pixel location that has the highest sigmoid output as a pupil center, or just keep the previous frame's center point.

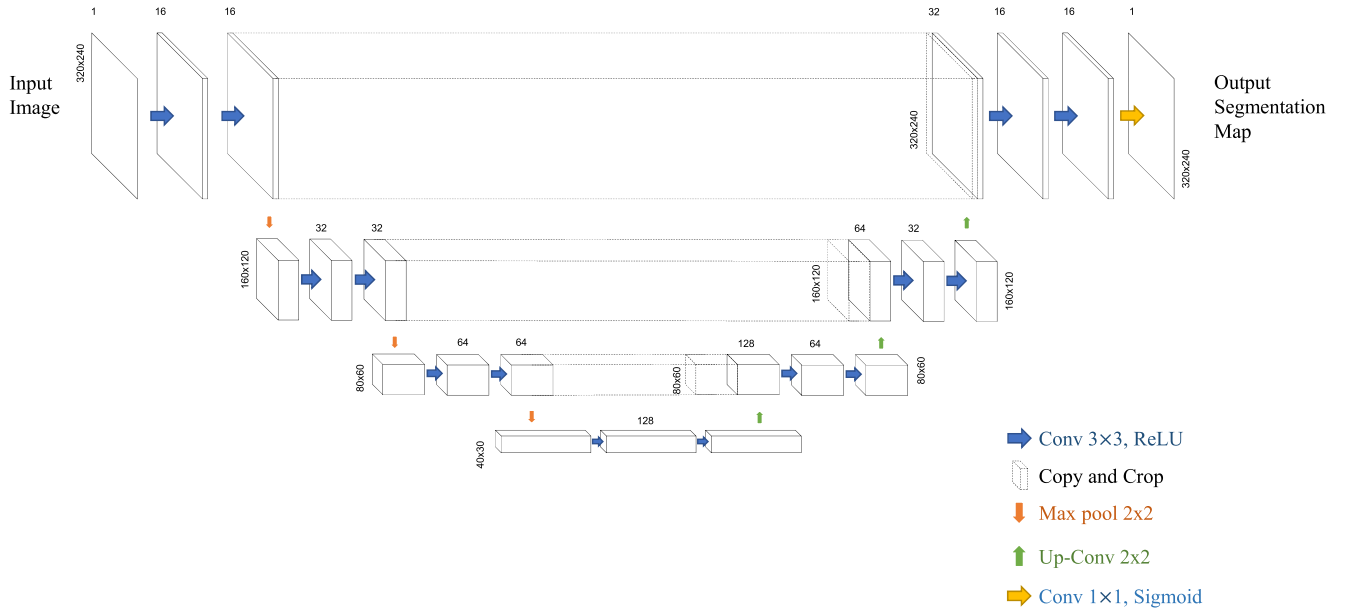


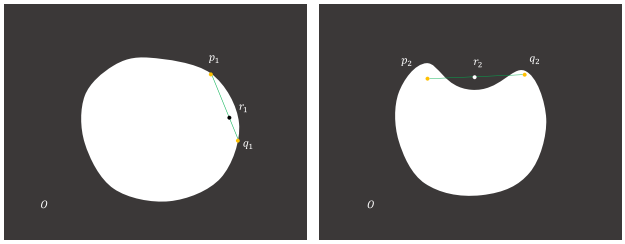
FIGURE 5. A light UNet architecture for pupil segmentation. Each square box represents a feature map, and the number on the box represents the number of channels in the feature. The x-y size of the feature map is shown on the left side of the box, and the meaning of each arrow is on the bottom right.

In the experiment, it is shown that the latter approach brings more stable results.

A. NETWORK ARCHITECTURE

We adopt the UNet for the pupil segmentation, as shown in Fig. 5. It is a kind of FCN, which is suitable for our work because it can perform the classification and localization at once [26]. There are also some other methods that perform both tasks at the same time. For example, Ciresan *et al.* [42] made a class-labeled patch around each pixel in the original image and used it for training the classifier. By applying the classifier to all the pixels in the input image with the sliding window method, they obtained the segmentation results. The PupilNet [13] also classifies all the test patches, although its purpose is different. In the case of patch-based methods, we may generate many training samples from a single training image, which can be an advantage in some cases. However, these patch-based binary labeling methods have some problems. First, there is a lot of redundancies in the overlapped patches, which influence the performance both in training and inferencing. Second, there is a trade-off between the localization accuracy and the abundance of context depending on patch size. Third, to classify using CNN, the fully connected layer is required at the end of CNN, which increases the size of the network. Hence, the reason for using the UNet is that it does not require fully connected layers (it consists of only convolutional layers, max pooling layers, and concatenating paths).

The left side of the UNet structure in Fig. 5 is called contracting path, which consists of 3×3 convolutional layers and max pooling layers, and the right side is called expansive path which consists of convolutional layers and concatenating paths from the layers from the contracting path.



(a) (b)

FIGURE 6. Example object $\mathbb{O}$ that (a) satisfies convex shape condition and (b) does not satisfy convex shape condition.

Maxpooling layer in UNet also grows receptive field, which causes fluctuating error like VGG-style network. However, in case of UNet, all points around the pupil are classified as pupil area, and the results are integrated so that the error can be sufficiently diminished. The left side enables the multiscale analysis by stacking the image pyramids, and the right side enables to acquire the pixel-wise classification results. The UNet also concatenates the reduced feature map in the contracting path and upsampled output that enables the output to assemble the information of various scales. This resolves the trade-off between localization accuracy and the abundance of context mentioned above. Moreover, since there is no FC layer, the network requires a less number of parameters compared to the regression network at the same depth.

B. CONVEX SHAPE PRIOR LOSS

Typical FCN frameworks employ the pixel-wise binary cross-entropy loss as a unary loss term, which can be written as

$$L_{ce} = -\sum_{i \in I} \sum_{p \in \Omega} y_{ip} \log S_{ip} + (1 - y_{ip}) \log(1 - S_{ip}). \quad (2)$$

In this case, only the ground truth of the pixel corresponding to the pupil area is 1, and the rest is 0. Hence, this equation can be rewritten as

$$L_{ce} = -\sum_{i\in I}\Big(\sum_{p\in\Omega_{\mathbb{O}}}\log S_{ip} + \sum_{p\in\Omega_{\mathbb{O}^c}}\log(1-S_{ip})\Big) \quad (3)$$

where $\Omega_{\mathbb{O}}$ is pixel space on the pupil area. As mentioned in Section II-D, we add a convex-shape-prior term that any line connecting two arbitrary points of the segmentation region is inside the region. The entire loss term incorporating the convex shape prior is

$$L_{total} = L_{ce} + \sum_{i\in I}\sum_{p,q\in\Omega}\sum_{r\in l_{pq}} B^i_{pqr} \times \left|y_{ip} - S_{ip}\right| \times \left|y_{iq} - S_{iq}\right| \times \left|S_{ip} + S_{iq} - 2S_{ir}\right| \quad (4)$$

where

$$B^i_{pqr} = \begin{cases} 1, & \text{if } y_{ip} = y_{iq} = y_{ir} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where the second term in the eq. (4) is the shape-prior term. To explain the second term in detail, let us consider a point pair (p,q) inside the segmented region, and a point r lying on the line that is bounded by p and q. Following the convex shape property, the point r must have the same ground truth label. Hence, the shape-prior term of eq. (4) is designed to be activated when p, q and r have the same label ($B^i_{pqr} = 1$). The next terms ($|y_{ip} - S_{ip}|$ and $|y_{iq} - S_{iq}|$) ensures that the predicted results on p and q do not differ from the ground truth label, and the last term means the convex shape prior that penalizes r to have the same label as p and q, as illustrated in Fig. 6.

In the training phase, encoding the prior knowledge of convex shape into the network is a time-consuming task, due to the times taken for finding the pupil center point in the ground truth segment and for randomly selecting points that are used to calculate the loss. But in the inference phase, the parameters of the network already contains prior knowledge, so the computation time is the same as the baseline that does not have the shape-prior term.

IV. IMPLEMENTATION

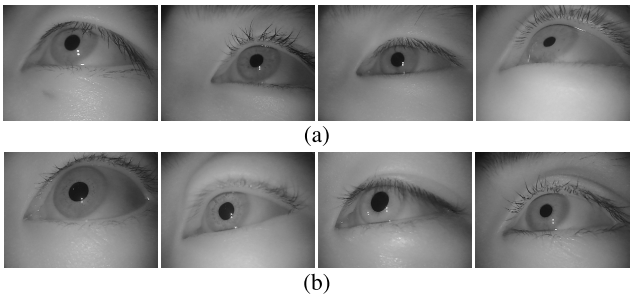
The UNet for the pupil area segmentation in our work is shown in Fig. 5, the complexity of which is reduced from the original network for several reasons. First of all, we deal with a rather simple shape compared to the diverse cell images for which the original UNet [26] was designed. Specifically, the UNet consists of several floors, where the first floor receives the input at the left side and generates output at the right. The number of channels is increased and their size is reduced, as the floor is increased downwards (see Fig. 5). While the original UNet consists of five floors with 64 channels on the first floor and 1024 channels on the fifth, we reduce the number of channels on the first by half (64 to 32) and the number of floors from five to four.

For training the network, the batch size is set to 20, and the training epoch 10. All weights of our network are initialized using the Glorot and Bengio [43],' and its learning rate is set to $(1e-4) * \frac{1}{10}^{(iter/maxiter)}$, and batch normalization is not used. In the test stage, a binary image is acquired by thresholding the UNet response by 0.5. When performing the CC analysis, we adopt 4-point neighborhood system to partition the binary map into pupil region candidates. For the binary image, the largest CC is regarded as the pupil region $\mathbb{O}$, and the center of mass is obtained as

$$(m, n) = \sum_{p\in\Omega_{\mathbb{O}}} (X_p, Y_p)/Area(\Omega_{\mathbb{O}}) \quad (6)$$

where (X_p, Y_p) is the location of the pixel p. In the implementation, we use the Pupil-Labs equipment [44] for acquiring the IR eye image, which has the VGA resolution. We reduce it by half (320 × 240) to save the computations and also to reduce the inherent noise from the camera.

In the implementation of eq. (4), the loss is not computed for every possible point because of the implementation efficiency. Instead of calculating the loss for every point on the line l_{pq}, we randomly select m pixels on the line. Also, p and q are randomly selected from the intersection points between the line of d directions extending from the center of the segmented polygon and the contour of the polygon, not all possible combination inside the polygon. Thus, all possible combinations of p and q are ${}_dC_2$ per image. Also, by setting m random points r for each combination, the number of points r participating in the loss is ${}_mC_2$. Since we use $d = 16$, $m = 8$ in the experiment setting, the total number of r is $8 \times {}_{16}C_2 = 960$.


(a)

(b)

FIGURE 7. Our dataset images used for (a) training and (b) test.

V. DATASET

We make a dataset from 29 IR eye video sequences (111,581 frames with VGA resolution) acquired from the Pupil Mobile Eye Tracking Headset for 12 participants, under the laboratory environment. The participants are requested to gaze at several target points, and some of the images are shown in Fig. 7. Also, we make binary ground-truth labels on the pupil region for all the frames. The dataset used for training is composed of former 10 sequences (000 ∼ 009) among 29, which are the sequences taken from five subjects. All the gaze tracking methods mentioned in this paper, including ours, show satisfying results with our dataset because the sequences are obtained under the laboratory condition that has little noise. We also perform

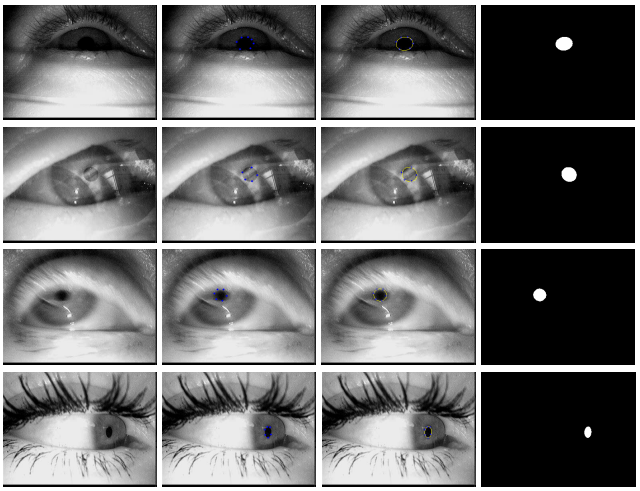


FIGURE 8. The process of annotating binary ground-truth map: (a) Examples of input images, (b) dotting the pupil boundary (blue dots), (c) ellipse fitting, and (d) labeling the area inside the ellipse.

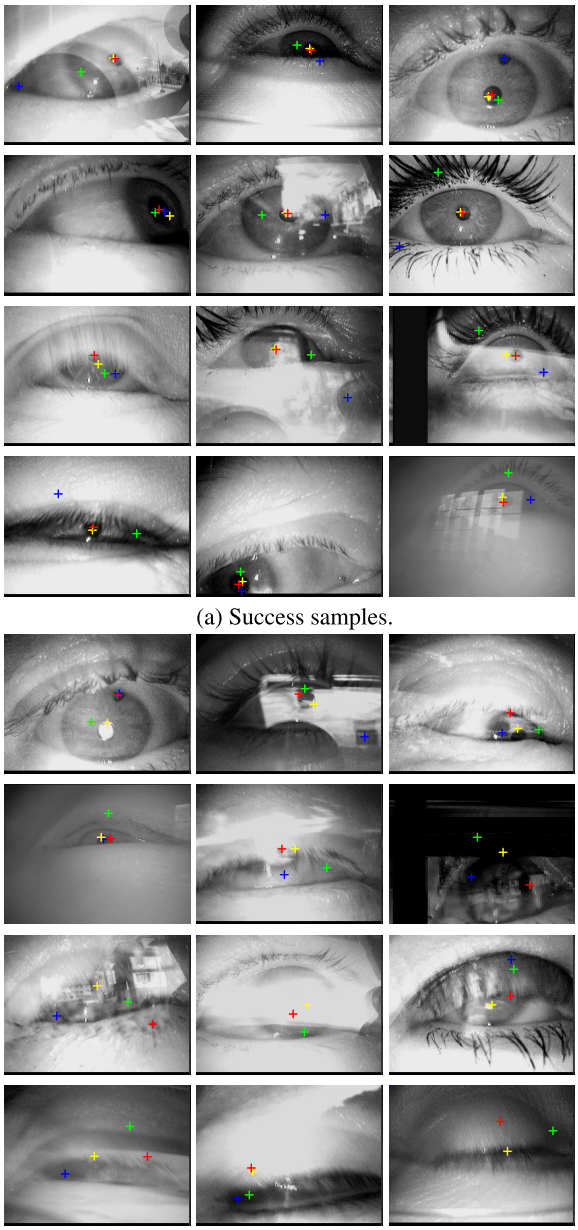
experiments on *ExCuSe* and *ElSe* dataset that are considered the real-world ones, containing many challenging images due to reflection, occlusion, blur, and other kinds of noises. These datasets are composed of 24 IR sequences (94,113 frames with 384 × 288 resolution), and are available at `ftp://emmapupildata@messor.informatik.uni-tuebingen.de`. However, since they provide only center points of pupils as annotation data while our main objective is the segmentation, we also annotate the segmentation map for these datasets.

Ten researchers participated in labeling the segmentation map, who have experiences in annotating hand regions in video sequences, and also in building semantic segmentation maps. More precisely, to create a binary ground-truth label, we first created a label for pupil boundary. Since it is impossible to annotate ground truth for all boundary points, we selected several points of the pupil boundary manually and applied the OpenCV ellipse fitting algorithm, as shown in Fig. 8. The inner region of the completed boundary was annotated as the pupil region. The results are double-checked by several industry researchers who are developing AR glasses. Hence, we believe that our ground-truth segmentation maps are quite reliable, though not perfect.

VI. EVALUATION

We perform the training and evaluation on a PC with GTX980 GPU, i5-4570 CPU @ 3.20 GHz, and 16GB RAM, and implement the CNN using the Pytorch library [45]. In the training phase, our approach requires 14 GPU-hours, while the traditional UNet method requires only 2 GPU-hours, because of our loss term that encodes the convex shape prior. However, in the inference phase, our approach shows almost the same inference time (about 7ms), as the traditional UNet method.

All the parameters in the compared methods, including non-learning-based and learning-based ones, follow their



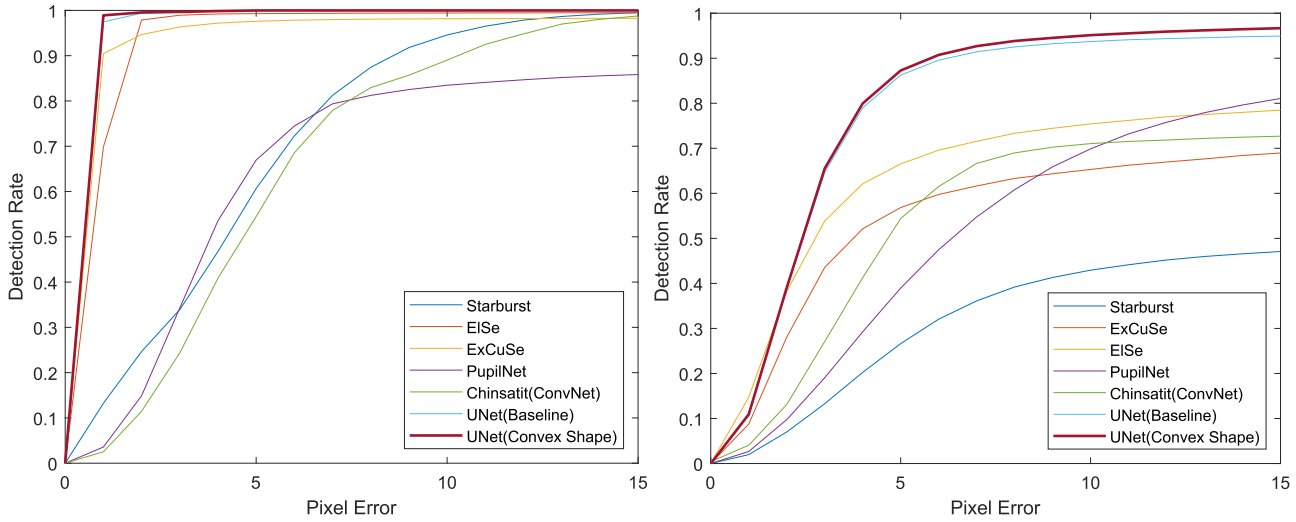
(a) Success samples.

(b) Failure samples.

FIGURE 9. Examples of detection results. Plus marker of red, green, blue, yellow indicate the results of the proposed method, ***StarBurst***, ***ElSe*** and ***Chinsatit***, respectively.

default settings. Since the trained weights of learning methods are not available or not trained for given datasets, we trained each method to get the detection rate curve in Fig. 10. The dataset of the eye image is much more redundant than other image datasets for other tasks because most of the eye images from different people look similar. Hence, we focus on the rate of pupil detection, instead of the accuracy of pupil center position, when the validation is performed for a person who did not participate in the creation of dataset.

We compare the pupil detection rates by calculating the Euclidean distance between the predicted pupil center and the ground truth label. Precisely, we count the number of pupil centers that are within N-pixel error, and denote it as the



(a) Train : Seq. 000 ∼ 009, Test : Seq. 010 ∼ 028

(b) Test : Each data sequence, Train : Remainder in Each data sequence

FIGURE 10. All datasets are split into training sequences and validation sequences. Performance comparison on (a) our dataset, and (b) *ExCuSe* & *ElSe* datasets.

TABLE 1. Performance comparison on each data sequence from *ExCuSe* & *ElSe*, in terms of Detection rate (%) allowing 5-pixel error tolerance. In the case of PupilNet*, we report the performance shown in the original publication.

	Detection Rate(%)						
Dataset	StarBurst	ExCuSe	ElSe	Chinsatit	PupilNet*	UNet (baseline)	UNet (Convex Shape)
I	7	71	86	51	82	89	**90**
II	25	34	65	49	79	79	**80**
III	11	39	64	56	66	**95**	**95**
IV	31	82	83	79	**92**	**92**	**92**
V	21	77	85	43	92	**98**	**98**
VI	9	53	78	71	79	94	**94**
VII	5	47	60	64	73	81	**82**
VIII	21	57	68	41	81	87	**91**
IX	11	75	87	56	86	93	**93**
X	59	80	79	50	81	**94**	**94**
XI	22	56	75	31	91	96	**98**
XII	51	79	79	48	85	85	**87**
XIII	33	70	74	28	83	80	**85**
XIV	54	58	84	69	95	97	**98**
XV	36	52	57	51	**81**	59	70
XVI	45	49	60	23	80	65	**66**
XVII	83	78	90	61	**97**	88	91
XVIII	3	22	51	42	62	93	**95**
XIX	6	26	33	50	37	**97**	**97**
XX	5	52	68	56	79	81	**82**
XXI	6	44	41	54	**83**	77	78
XXII	5	28	49	62	58	76	**80**
XXIII	64	94	94	20	90	98	**98**
XXIV	19	45	53	26	55	55	**60**
AVERAGE	26.3	57.1	69.3	49.1	78.6	85.3	**87.3**

detection rates at pixel error N. Note that some existing gaze estimation methods find the pupil center in sub-pixel resolutions. In this case, the maximum tolerance for the algorithm on locating the pupil center point is about one pixel, i.e., we need to evaluate the detection rate at $N = 1$. However, this seems rather impractical because creating the ground-truth of pupil center location is a hard manual task, and hence a certain amount of error in the ground-truth is inevitable. Therefore, the results are usually discussed in the tolerance of five-pixel errors [46], i.e., detection rates at $N = 5$. We will also show how pupil center detection using the segmentation works well under challenging situations. In order to reduce the computational cost, we will also show how adjusting the numbers of channels and floors affects accuracy. Also, we will show how incorporating the prior knowledge on the loss term affects the shape of the segmented result and the improvement of the overall detection accuracy.

A. ANALYSIS ON OUR DATASET

For the comparison with our dataset, we train the network using 10 data sequences from 5 subjects, among the 29 sequences. The results of 5-pixel error for the compared methods are shown in Fig. 10a, where we can see that our approach shows slightly better results than the traditional non-learning-based ones, whereas the other learning-based methods perform worse than the non-learning ones.

B. ANALYSIS ON VARIOUS CHALLENGING SITUATIONS

The datasets *ExCuSe* and *ElSe* have challenging images such as the ones with severe reflection, poor illumination, long mascara, which works as severe noise, etc. First, we show some qualitative comparisons in Fig. 9, where the red dot is the result obtained using the proposed method, and the green, blue, and yellow are the results using the *StarBurst*, *ElSe* algorithm, and *Chinsatit et al.*, respectively. We can see that the proposed method shows robust results for the images with a certain degree of reflection, shaky images, dark images with mascara, and dark images, as shown in Fig. 9a. The proposed algorithm does not work well when the pupil is not visible at all because of the reflected light when the image is very shaky or when the image is very dark, as shown in Fig. 9b. However, other algorithms also fail to detect the pupil center in these cases.

TABLE 2. Size of the network according to the number of floors and the number of channels.

UNet	Numb. floors	Channels on the 1st floor.	Network Size (Mb)
F3CH16	3	16	1.33
F4CH16	4	16	5.33
F3CH32	3	32	5.51
F4CH32	4	32	22.0

For the quantitative analysis, we also show the detection rates on $N = 5$ with cross-validation on each data sequence. Our UNet with unary loss term with cross-entropy is denoted as UNet (Baseline). Also, we compare our method with three traditional non-learning algorithms and two learning-based algorithms. As shown in Fig. 10b, *ElSe* shows the highest detection rate (average about 69.3% at $N = 5$) among the non-learning based algorithms. However, the results are not satisfactory to be used for gaze tracking. Existing CNN-based methods were expected to overcome these limitations, but they did not show better performance. Specifically, *Chinsatit et al.* shows the detection rate of 49.1% at $N = 5$. It even shows lower performance on average by 20.2%p than *ElSe*, probably due to the limitation of the regression approach on pupil center detection. On the other hand, our method shows 85.3%, which is the best among all the compared non-learning and learning-based methods.

C. ANALYSIS ON CONVEX SHAPE PRIOR

Although our baseline UNet (without shape-prior term) shows satisfying results, there are some failure cases. These are the cases when the segmentation is not performed properly, as shown in Fig. 11. This figure also shows that the proposed shape-prior term brings better results than the baseline, which results in the improved N-pixel error, as shown in Table 1 and Fig. 10b.

D. ANALYSIS ON NETWORK SIZE

In this subsection, we analyze the performance vs. network complexity. As stated previously, our UNet has a simpler architecture than the original UNet. However, the reduced network needs 22 Mb, which is still larger than the *PupilNet* with 1.33 Mb. Hence, we attempt to reduce the size of UNet down to 1.33 Mb and compare the performance in Fig. 12. Specifically, we reduce the UNet as shown in Table 2, i.e., we reduce the number of floors and/or the number of channels in several ways, and plot their detection rate vs. N in Fig. 12. We can see that the reduction does not much decrease the detection rates, and we can achieve a better detection rate than the PupilNet with almost the same complexity.

E. COMPARISON OF UNet WITH OTHER NETWORKS FOR PUPIL SEGMENTATION

As stated previously, we adopt UNet in this paper because it is known to provide state-of-the-art performance in many applications, including the segmentation. In this section, we validate this by comparing the overall performance when the segmentation is performed by other well-known image segmentation networks such as FCN [27] and Deeplab [30]. For the fair comparison, the numbers of weights of compared networks are set to be almost the same. The results are

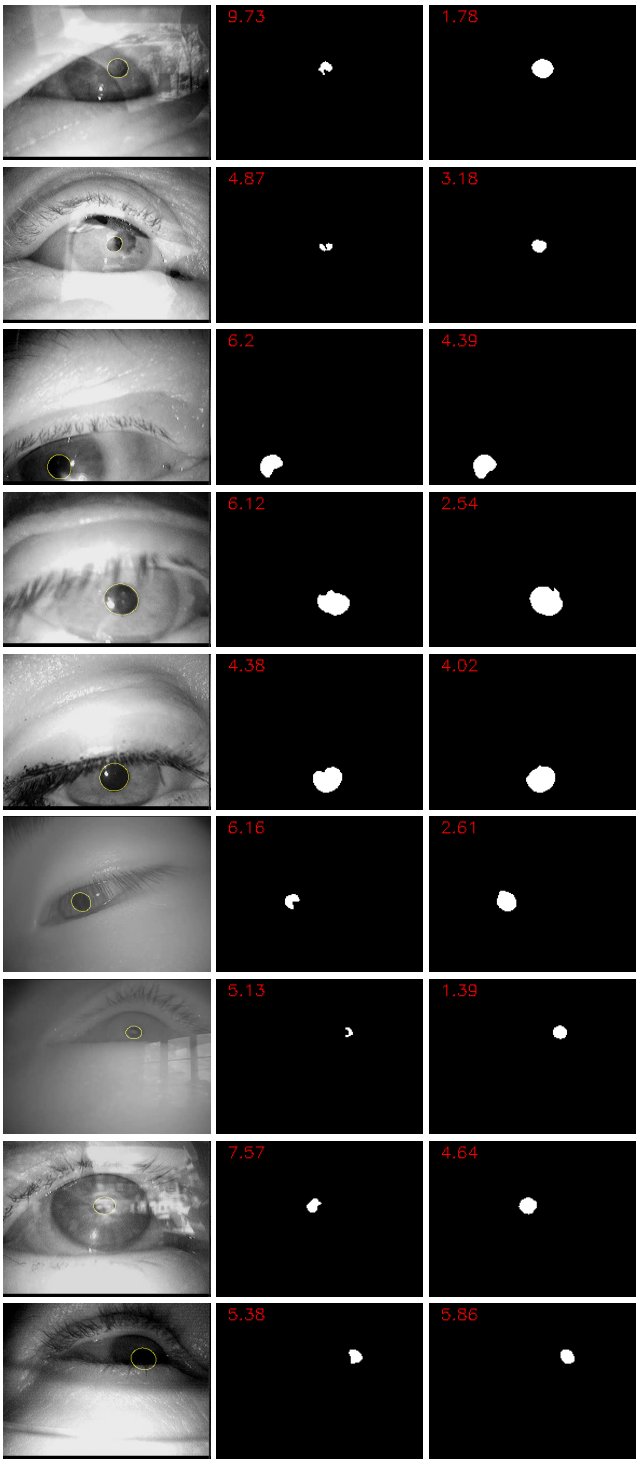


FIGURE 11. Qualitative comparison of segmentation results with/without shape-prior term: (a) input image with ground truth (b) result with only unary term (c) result with additional shape-prior term that encodes the convex shape prior. Red numbers on the figures denote the errors of pupil center estimation.

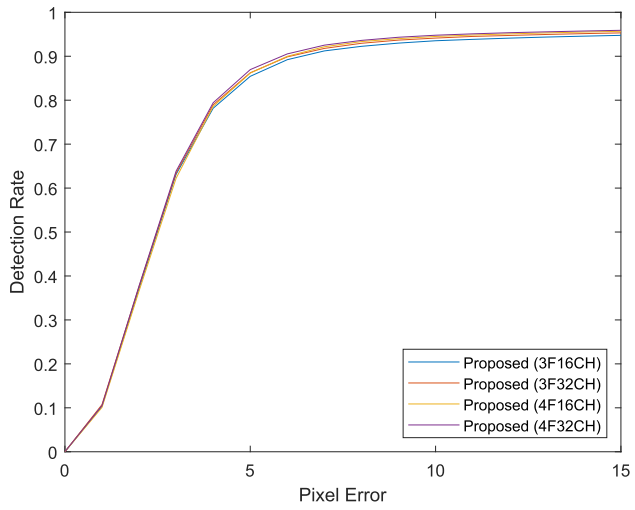


FIGURE 12. Performance comparison on *ExCuSe* & *ElSe* dataset for each network size.

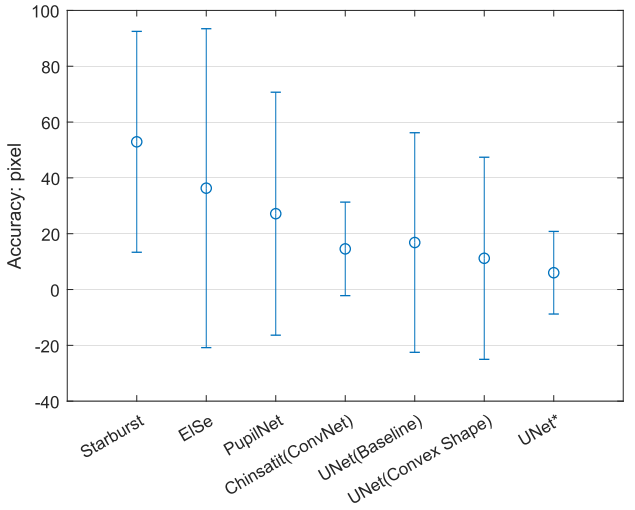


FIGURE 13. Comparison of the accuracy of pupil center coordinates. UNet* is the result after the simple postporcessing that keeps the center position when the pupil area is not found in the current frame.

summarized in Table 3, which shows that the UNet performs better than or comparable to the FCN and Deeplab for the pupil segmentation. The result also indicates that adding our shape-prior term enhances the performances regardless of the type of networks used for the segmentation.

F. ANALYSIS OF CENTER POSITION ERROR

Finally, we compare the accuracy of the pupil center positions in Fig. 13. Specifically, the figure shows the average and variance of the estimation error of each algorithm. We can see that the proposed method shows the least mean value but a relatively higher variance than *Chinsatit*(ConvNet). This is because the network output of the proposed method has a large error when there exists no segmented area. For example, failure cases such as the bottom row of Fig. 9b has no detected segments since the eyelid covers the entire pupil area. We can add a simple postprocessing step to alleviate this problem, specifically we keep the same center position

TABLE 3. Performance comparisons when different networks (FCN [27] and DeepLab [30]) are used for the segmentation, for each data sequence from *ExCuSe* & *ElSe*, in terms of detection rate (%) allowing 5-pixel error tolerance.

Dataset	Detection Rate(%)					
	Deeplab (baseline)	Deeplab (Convex Shape)	FCN (baseline)	FCN (Convex Shape)	UNet (baseline)	UNet (Convex Shape)
I	89	89	81	84	89	90
II	70	77	73	71	79	80
III	95	96	78	88	95	95
IV	92	92	88	89	92	92
V	98	98	94	94	98	98
VI	94	94	90	91	94	94
VII	82	82	70	74	81	82
VIII	89	89	81	84	87	91
IX	94	94	89	90	93	93
X	96	95	85	87	94	94
XI	99	99	86	95	96	98
XII	86	86	84	85	85	87
XIII	80	79	75	80	80	85
XIV	97	96	96	95	97	98
XV	75	72	37	51	59	70
XVI	65	64	35	47	65	66
XVII	90	91	90	91	88	91
XVIII	93	94	91	86	93	95
XIX	98	98	96	95	97	97
XX	82	82	46	59	81	82
XXI	78	77	68	70	77	78
XXII	81	81	49	64	76	80
XXIII	97	99	92	77	98	98
XXIV	55	56	48	51	55	60
AVERAGE	86.5	86.6	75.9	79.1	85.3	87.3

when the segmentation area is lost at the current frame. The UNet* in Fig. 13 is the estimation error after this postprocessing, which shows the least mean and variance, i.e., the most accurate estimation of pupil center coordinates. The increase of estimation accuracy by this simple postprocessing may be another advantage of the segmentation-based method, which is not easy in the case of regression methods. Precisely, we can easily see whether there is a pupil segment or not in our case, and we just do not use the result when the segment is absent. On the other hand, it is difficult to tell whether the center point brought by the regression is out of tolerance or not.

VII. CONCLUSION

We have proposed a method for the detection of the pupil center in the IR eye images, which can be used for many human-machine interfaces. Unlike the existing CNN-based regression methods that directly obtain the pupil center coordinate as the network output, our method segments the pupil region and calculates the pupil center as the center of mass of the segmented region. Also, unlike the conventional multi-scale sliding-window method that uses the VGG-style CNN, our system exploits the UNet that can naturally exploit the multiscale features. Moreover, we designed a shape-prior term for the loss function, which increases the robustness to the noise. We have also created datasets for training and testing, on our videos and also on widely used datasets. Experiments show that our method detects the pupil center robustly and yields accurate pupil center positions. Our

dataset and code are available at `https://github.com/jaegal88/pupil-shape-prior`.

REFERENCES

[1] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," in *Adv. Physiol. Comput.* London, U.K.: Springer, 2014, pp. 39–65.

[2] J. Cannan and H. Hu, "Human-machine interaction (HMI): A survey," Univ. Essex, Colchester, U.K., Tech. Rep. CES-508, 2011.

[3] J.-Y. Lee, H.-M. Park, S.-H. Lee, T.-E. Kim, and J.-S. Choi, "Design and implementation of an augmented reality system using gaze interaction," in *Proc. Int. Conf. Inf. Sci. Appl.*, Apr. 2011, pp. 1–8.

[4] B. Mandal, L. Li, G. S. Wang, and J. Lin, "Towards detection of bus driver fatigue based on robust visual analysis of eye state," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 545–557, Mar. 2017.

[5] W. G. Aguilar, J. I. Estrella, W. López, and V. Abad, "Driver fatigue detection based on real-time eye gaze pattern analysis," in *Proc. Int. Conf. Intell. Robot. Appl.* Cham, Switzerland: Springer, 2017, pp. 683–694.

[6] X. Drèze and F.-X. Hussherr, "Internet advertising: Is anybody watching?" *J. Interact. Marketing*, vol. 17, no. 4, pp. 8–23, Jan. 2003.

[7] M. Resnick and W. Albert, "The impact of advertising location and user task on the emergence of banner ad blindness: An eye-tracking study," *Int. J. Hum.-Comput. Interact.*, vol. 30, no. 3, pp. 206–219, Mar. 2014.

[8] T. Ohshima, H. Yamamoto, and H. Tamura, "Gaze-directed adaptive rendering for interacting with virtual space," in *Proc. IEEE Virtual Reality Annu. Int. Symp.*, Mar. 1996, pp. 103–110.

[9] J. Orlosky, Y. Itoh, M. Ranchet, K. Kiyokawa, J. Morgan, and H. Devos, "Emulation of physician tasks in eye-tracked virtual reality for remote diagnosis of neurodegenerative disease," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 4, pp. 1302–1311, Apr. 2017.

[10] D. Li, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Sep. 2005, p. 79.

[11] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci, "Excuse: Robust pupil detection in real-world scenarios," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Cham, Switzerland: Springer, 2015, pp. 39–51.

[12] W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci, "ElSe: Ellipse selection for robust pupil detection in real-world environments," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl. (ETRA)*, 2016, pp. 123–130.

[13] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci, "PupilNet: Convolutional neural networks for robust pupil detection," 2016, *arXiv:1601.04902*. [Online]. Available: http://arxiv.org/abs/1601.04902

[14] W. Chinsatit and T. Saitoh, "CNN-based pupil center detection for wearable gaze estimation system," *Appl. Comput. Intell. Soft Comput.*, vol. 2017, pp. 1–10, 2017.

[15] N. Kondo, W. Chinsatit, and T. Saitoh, "Pupil center detection for infrared irradiation eye image using CNN," in *Proc. 56th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*, Sep. 2017, pp. 100–105.

[16] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 538–552.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[21] Y. Luo, D. Yin, A. Wang, and W. Wu, "Pedestrian tracking in surveillance video based on modified CNN," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 24041–24058, Sep. 2018.

[22] S. Y. Han, I. Hwang, S. H. Lee, and N. I. Cho, "Gaze estimation using 3-D eyeball model and eyelid shapes," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2016, pp. 1–4.

[23] J. Wang, G. Zhang, and J. Shi, "2D gaze estimation based on pupil-glint vector using an artificial neural network," *Appl. Sci.*, vol. 6, no. 6, p. 174, 2016.

[24] S. Y. Han, S. H. Lee, and N. I. Cho, "Gaze estimation using 3-D eyeball model under HMD circumstance," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–4.

[25] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[28] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[31] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[32] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

[33] G. Lin, C. Shen, A. V. D. Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3194–3203.

[34] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised CNN segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1818–1827.

[35] H. J. Wyatt, "The form of the human pupil," *Vis. Res.*, vol. 35, no. 14, pp. 2021–2036, Jul. 1995.

[36] S. Y. Han, Y. Kim, S. H. Lee, and N. I. Cho, "Pupil center detection based on the UNet for the user interaction in VR and AR environments," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, Mar. 2019, pp. 958–959.

[37] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[38] G. J. Mohammed, B. R. Hong, and A. A. Jarjes, "Accurate pupil features extraction based on new projection function," *Comput. Informat.*, vol. 29, no. 4, pp. 663–680, 2012.

[39] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[40] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Jul. 2001, pp. 105–112.

[41] Z. Mirikharaji and G. Hamarneh, "Star shape prior in fully convolutional networks for skin lesion segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 737–745.

[42] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2843–2851.

[43] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[44] M. Kassner, W. Patera, and A. Bulling, "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Computing: Adjunct Publication*, 2014, pp. 1151–1160.

[45] N. Ketkar, "Introduction to PyTorch," in *Deep Learning With Python*. Berkeley, CA, USA: Apress, 2017.

[46] W. Fuhl, M. Tonsen, A. Bulling, and E. Kasneci, "Pupil detection in the wild: An evaluation of the state of the art in mobile headmounted eye tracking," *Mach. Vis. Appl.*, vol. 27, no. 8, pp. 1275–1288, Nov. 2016.