IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 3, 2020, accepted July 14, 2020, date of publication July 17, 2020, date of current version July 29, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3010011*

# Semantic Segmentation of the Eye With a Lightweight Deep Network and Shape Correction

**VAN THONG HUYNH, (Graduate Student Member, IEEE), HYUNG-JEONG YANG, (Member, IEEE), GUEE-SANG LEE, (Member, IEEE), AND SOO-HYUNG KIM, (Member, IEEE)**

School of Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, South Korea

Corresponding author: Soo-Hyung Kim (shkim@jnu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A3A03000947 and Grant NRF-2020R1A4A1019191.

**ABSTRACT** This paper presents a method to address the multi-class eye segmentation problem which is an essential step for gaze tracking or applying a biometric system in the virtual reality environment. Our system can run on the resource-constrained environments, such as mobile, embedded devices for real-time inference, while still ensuring the accuracy. To achieve those ends, we deployed the system with three major stages: obtain a grayscale image from the input, divide the image into three distinct eye regions with a deep network, and refine the results with image processing techniques. The deep network is built upon an encoder-decoder scheme with depthwise separation convolution for the low-resource systems. Image processing is accomplished based on the geometric properties of the eye to remove incorrect regions as well as to correct the shape of the eye. The experiments were conducted using OpenEDS, a large dataset of eye images captured with a head-mounted display with two synchronized eye-facing cameras. We achieved a mean intersection over union (mIoU) of 94.91% with a model of size 0.4 megabytes and 16.56 seconds to iterate over the test set of 1,440 images.

**INDEX TERMS** Eye segmentation, image processing, virtual and augmented reality, human computer interaction.

## I. INTRODUCTION

Understanding the motion and appearance of the human eye is an active area of research with applications in many fields, such as psychology, biometrics, and human-computer interactions. In the digital world, virtual reality (VR) involves placing a screen in front of the eyes to create a virtual environment which simulates the user's physical presence. Accurate and precise eye tracking can assist in rendering only on those parts of a virtual scene that the user is focusing at full resolution, which significantly reduces the computational burden of VR [1]. Eye detection, which localizes eye regions in the image, is an essential component of any eye tracking system [2], [3].

Iris segmentation has been drawing significant attention from the research community due to the popularity of iris recognition technology. In [4], the authors presented an algorithm that segmented the iris in RGB eye images taken under visible and near-infrared light. Their approach involves five sequential steps: reflection localization, filling in of reflections, localization of iris boundaries, and determination of the lower and upper eyelid boundaries. In each step, the system does image processing and analyzes RGB values to achieve its goals. Tan and Kumar [5] extracted the features of Zernike moments [6] at different radii to classify each pixel into iris or non-iris with support vector machines. The authors in [7] used a random walker algorithm [8] in which images are modeled with graph theory, such that each pixel corresponds to the vertex (node) and the linkage between any two pixels corresponds to the edge of a graph. The weight in the graph is calculated by exploiting gradient information. Liu *et al.* [9] deployed multi-scale fully convolutional networks (MFCNs) allowing for arbitrary input image size. The approach in [10] exploited feature learning with ATTention U-Net (ATT-UNet) which used the attention mechanism to guide the model to learn more discriminative features for separating iris from

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

non-iris pixels. ATT-UNet deployed bounding box regression to generate an attention mask for the iris, which was used as a weighted function to make the model pay more attention to the iris region in the enhanced eye image. These methods based on deep representation significantly outperform those methods with handcrafted features such as [5], [7].

Sclera segmentation is typically considered as a sub-problem of a broader task such as iris recognition or gaze estimation [11]. Most recent approaches focused on features from deep architecture. The authors in [12] proposed a method with two steps: periocular region localization and sclera segmentation in the detected region based on a fully convolutional network. In [13], the author presented Sclera-Net, a residual encoder-decoder network based on SegNet, to segment the sclera into various sensor images. Skip connection was employed to reduce the loss of information during down-sampling as well as up-sampling. Wang *et al.* [14] inserted an attention module into the contracting path and expansive path of U-Net [15]. They achieved the best performance with channel-wise attention, which is similar to a squeeze-and-excitation block [16].

In multi-class eye segmentation, [17] trained a convolutional encoder-decoder network based on SegNet [18] with 4-fold cross-validation on a small dataset of 120 images from 30 participants. The system classifies pixels in an image into six classes based on region: pupil, iris, sclera, eyelashes, medial canthus, and periocular. A study based on atrous convolution with a conditional random field for post-processing is presented in [19]. Luo *et al.* [20] proposed a shape constrained network which employed VAE-GAN [21] to learn the shape first, and SegNet [18] was used to incorporate this information into model. The authors in [22] leveraged separable convolution [23] to reduce the computational cost of SegNet when applied to the multi-class eye segmentation problem. They also utilized boundary refinement [24] to improve performance. In [25], the authors designed a multi-scale segmentation solution (Eye-MS) which consists of inter-connected refinement modules. They created the miniature multi-scale segmentation network (Eye-NMS), a light version of Eye-MS, by reducing the feature map size with a concomitant drop in performance of 2.5%. Perry and Fernandez [26] leveraged dilated and asymmetric convolution, while Kansal and Devanathan [27] utilized squeeze-and-excitation [16] block as well as spatial attention on channel attetion [28]. Chaudhary *et al.* [29] presented an architecture based on DenseNet [30] and UNet [15]. They performed a lot of augmentation operations during training, such as Gaussian blur, image translation, and corruption.

Although many studies have segmented the components of the eye in an image, it remains a challenge because of the difficulties such as blurred or defocused images, eye makeup, or eyeglasses. In order to apply these methods in real-world situations, a system must be able to run on embedded devices, mobile devices, or other resource-constrained environments. In this work, we propose an integrated system composed of traditional image processing and lightweight deep network architecture which has high performance where the influences of above mentioned effects are minimized, and that can be used in real-world applications. This study extends the work in [31] with additional analysis and a new method to make further improvement of the system. We investigate on the incorrect shape produced by the deep network because of heavy makeup, the loss of focus, or eyeglasses. These procedures are detected and calibrated with our proposed image processing.

## II. PROPOSED APPROACH

Our work utilizes convolutional encoder-decoder architecture to segment a 2D grayscale eye image into three distinct classes: the sclera, pupil, and iris. We build a simple decoder and leverage the effect of depthwise separable convolution in an encoder module to achieve a small but accurate model. Firstly, we resize the input by a factor of 0.5 with bilinear sampling to reduce the computation cost of the system. Encoder module performs the down-sampling process on resized input with stride parameters in convolution operations. Depthwise separable convolutions are deployed to form bottleneck blocks in the middle of encoder module. Post-processing step including our image processing operation, is applied to correct the failure part of the deep network model in all input image. A detail of our approach is illustrated in Figure 1.

### A. DEPTHWISE SEPARABLE CONVOLUTIONS

Depthwise separable convolutions are a core component of most efficient deep neural networks that work in mobile and other resource-constrained environments [23], [32]–[35]. In standard convolution, a new representation is obtained in a single step with a huge number of parameters to allow for feature filtering and computation. Assuming that the input tensor $F$ has a size $h \times w \times d_F$ and the output tensor $G$ has a size $h \times w \times d_G$, where $h$ and $w$ are spatial height and width, $d_F$ and $d_G$ are the number of input and output channels, respectively. With the base convolution kernels of size $d_K \times d_K$, standard convolution uses kernel $K$ of size $d_K \times d_K \times d_F \times d_G$ to obtain $G$ from $F$ as

$$G_{k,\ell,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i,\ell+j,m} \tag{1}$$

where $m = \overline{1, d_F}$ and $n = \overline{1, d_G}$. Consequently, standard convolution has a computational cost of

$$C_{trad\_conv} = h \cdot w \cdot d_F \cdot d_G \cdot d_K^2 \tag{2}$$

Depthwise separable convolution breaks standard convolution down into two separate layers: depthwise convolution and pointwise convolution. The first layer performs filtering with kernel $\hat{K}$ which included $d_F$ kernel of size $d_K \times d_K$, where the $m^{th}$ filter is applied to the $m^{th}$ channel of $F$ to obtain the $m^{th}$ channel in the output tensor $\hat{G}$. This operation can be written as

$$\hat{G}_{k,\ell,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i,\ell+j,m} \tag{3}$$

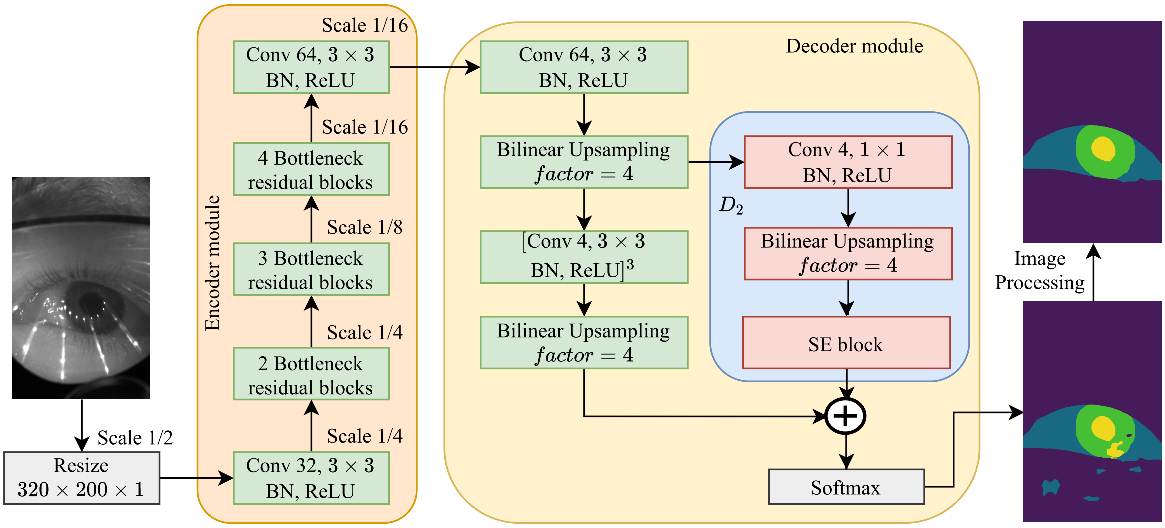


**FIGURE 1.** Overall system architecture.

which leads to a computational cost of

$$C_{dwise\_conv} = h \cdot w \cdot d_F \cdot d_K^2 \tag{4}$$

For the purpose of generating new features by computing linear combinations of input channels, pointwise convolution performs traditional convolution based on $d_G$ kernels of size $1 \times 1$. As in Equation 2 with $d_K = 1$, the pointwise convolution has a computational cost of

$$C_{pwise\_conv} = h \cdot w \cdot d_F \cdot d_G. \tag{5}$$

Thus, the computational cost of depthwise separable convolution can be expressed as the summation of Equation 4 and Equation 5

$$\begin{aligned} C_{dsepa\_conv} &= C_{dwise\_conv} + C_{pwise\_conv} \\ &= h \cdot w \cdot d_F \cdot d_K^2 + h \cdot w \cdot d_F \cdot d_G \\ &= h \cdot w \cdot d_F \cdot \left(d_K^2 + d_G\right). \end{aligned} \tag{6}$$

Empirically, by factorizing convolution into two steps, depthwise separable convolution works almost the same as regular convolution [23], while the computational cost is reduced by

$$\begin{aligned} C_{reduced} &= \frac{C_{dsepa\_conv}}{C_{trad\_conv}} \\ &= \frac{h \cdot w \cdot d_F \cdot (d_K^2 + d_G)}{h \cdot w \cdot d_F \cdot d_G \cdot d_K^2} \\ &= \frac{1}{d_G} + \frac{1}{d_K^2} \end{aligned} \tag{7}$$

which is almost a factor of $d_K^2$.

### B. BOTTLENECK BLOCKS

The idea of the bottleneck residual block was originally introduced in [33]. It is like a residual block [36], but it is more memory efficient as well as slightly better. It takes an input with $d$ channels of spatial height $h$ and width $w$. The ratio between the depth of the inner feature map and the input is referred to as the expansion ratio $t$. When $t \neq 1$, a linear transform with $1 \times 1$ convolution is used to project the $d$ channels input onto a new $t \cdot d$ channel space. When $t < 1$, it is a classical residual convolution block [36]. Down-sampling is handled by stride $s$ in the first step of depthwise separable convolution. The detailed structure of this block is shown in Figure 2.

A squeeze-and-excitation (SE) block [16] produces significant improvements in performance with slightly additional computational costs for state-of-the-art deep architectures by performing feature re-calibration through two major operations: *squeeze* to produce a channel descriptor, and *excitation* to obtain per-channel modulation weights, which are used to scale the input. In the squeeze operation, an SE block exploits channel dependencies with channel-wise statistics by using global average pooling which aims to allow information from the global receptive field to be used by all layers in the network. Then, a simple gating mechanism with sigmoid activation is applied to aggregate information from the *squeeze* operation and produce a non-mutually exclusive relationship between channels. The structure of an SE block is shown in Figure 3.

### C. ENCODER-DECODER ARCHITECTURE

The architecture of our encoder includes a full convolution with 32 and 64 filters for the initial and the last layer, respectively; 9 residual bottleneck layers, as shown in Figure 2, are inserted between the two convolution layers. These bottleneck residual blocks are organized into 3 groups with 16, 24, and 32 output channels, respectively. The expansion ratio is 1 for the first group and 6 for the next two groups. Down-sampling is done by setting the stride to 2 in the initial convolution layer of the network and the first block of the

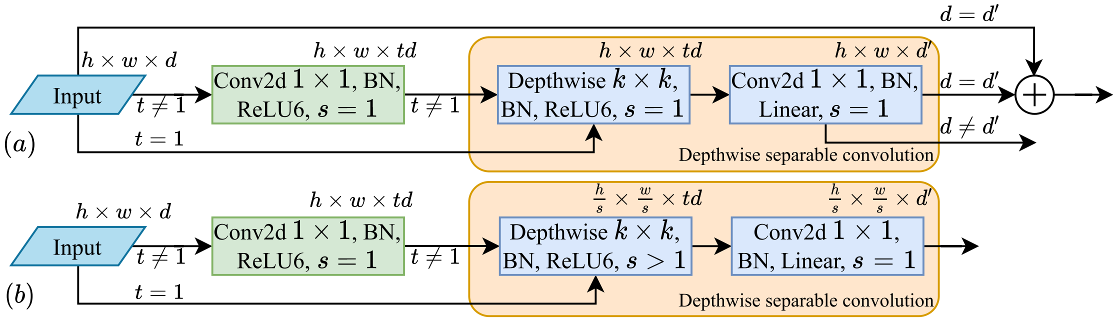

$d = d'$
$h \times w \times d$
Input
$t \neq 1$
$t = 1$
(a)
$h \times w \times td$
Conv2d $1 \times 1$, BN, ReLU6, $s = 1$
$t \neq 1$
$h \times w \times td$
Depthwise $k \times k$, BN, ReLU6, $s = 1$
$h \times w \times d'$
Conv2d $1 \times 1$, BN, Linear, $s = 1$
$d = d'$
$d \neq d'$
Depthwise separable convolution
$h \times w \times d$
Input
$t \neq 1$
$t = 1$
(b)
$h \times w \times td$
Conv2d $1 \times 1$, BN, ReLU6, $s = 1$
$t \neq 1$
$\frac{h}{s} \times \frac{w}{s} \times td$
Depthwise $k \times k$, BN, ReLU6, $s > 1$
$\frac{h}{s} \times \frac{w}{s} \times d'$
Conv2d $1 \times 1$, BN, Linear, $s = 1$
Depthwise separable convolution

**FIGURE 2.** A visualization of bottleneck residual block [33] transformation from *d* to *d′* channel with expansion ratio *t* and stride *s*. (a) Depthwise convolution with *s* = 1. (b) Depthwise convolution with *s* = 2.

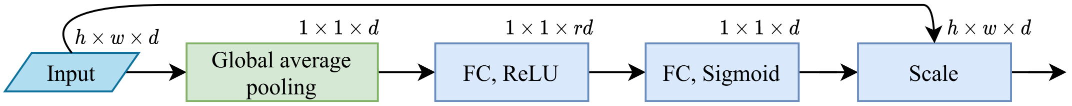


**FIGURE 3.** The schema of the SE block [16]. FC indicates a fully connected layer. 0 < *r* ≤ 1 denotes the reduction factor of depth size in the *excitation* operation.

last two groups. We always use the kernel size of 3 × 3, as is standard for modern networks. The details of our encoder architecture are described in Table 1.

**TABLE 1.** The encoder architecture of our method. Each line describes a group which included *n* blocks of the operator. The first block of each group has stride *s* and all others use a stride of 1.

| Input | Operator | Expansion ratio $t$ | Output channels $c$ | Number of operators $n$ | Stride $s$ |
|---|---|---|---|---|---|
| $320 \times 200$ | conv2d | - | 32 | 1 | 2 |
| $320 \times 200$ | bottleneck | 1 | 16 | 2 | 1 |
| $160 \times 100$ | bottleneck | 6 | 24 | 3 | 2 |
| $80 \times 50$ | bottleneck | 6 | 32 | 4 | 2 |
| $40 \times 25$ | conv2d | - | 64 | 1 | 1 |

In the decoder module, we build an architecture with a structure similar to squeeze and excitation (SE) block in [16], as described briefly above. The goal of an SE block is to acquire the global information necessary to selectively emphasize informative features and suppress less useful ones by explicitly modeling the interdependencies between channels [16]. Our decoder begins with a regular component in the segmentation network: a convolution with 64 filters of kernel size 3 × 3 followed by bilinear up-sampling, which increases the input size four times. At this point, we create two different streams in order to learn from and make an ensemble of them at the end of the decoder module. In the first stream, we use three convolutions with kernel size 3 × 3 followed by bilinear up-sampling to get the same size as the original input. In the other stream, we use only a 1 × 1 convolution and up-sampling of the output. In order to learn vital information and eliminate trivial information, we set the reduction ratio of $r$ to 4 in the SE block. After each convolution operation, we use batch normalization and ReLU as the non-linearity function. Finally, we applied softmax activation on the summation of two streams in order to obtain the probabilities that each pixel belong to 4 classes of background, sclera, iris, and pupil.

### *D. IMAGE PROCESSING*

To reduce incorrect region classification in the predictions of the deep network, we analyze the properties of the connected components with 8-connectivity. Each mask contains at most four values 0, 1, 2, and 3, corresponding to the background, sclera, iris, and pupil. In each class except the background, we keep only the biggest region, which is considered the correct region. The sclera covers the iris, and the iris wraps the pupil. Consequently, we filled black holes and removed the small connected components for the sclera, iris, and pupil sequentially, as in algorithm 1. Figure 4 shows a visualization of the steps in *forloop* in algorithm 1.

**Algorithm 1** Filtering With Connected Components

```
Input:   Predicted mask Mp of size $640 \times 400$
Output:  Filtered mask Mf of size $640 \times 400$
Initialize Mf with zeros
for $i \leftarrow 0, 2$ do
    Bw $\leftarrow$ Binary image from Mp with threshold $i$
    Bw $\leftarrow$ Fill all black holes in Bw
    Lc $\leftarrow$ Largest connected component with 8-connectivity in Bw
    Lr $\leftarrow$ Bw\Lc
    Fill the region in Mf corresponding with Lc with the value of $i + 1$
    Fill the region in Mp corresponding with Lr with 0 values
end for
```

We apply a horizontal projection on a binary image, in which pixels belonging to the eye region are marked as foreground. We observe that there is only one peak in the image projection with the correct eye shape; incorrect shape

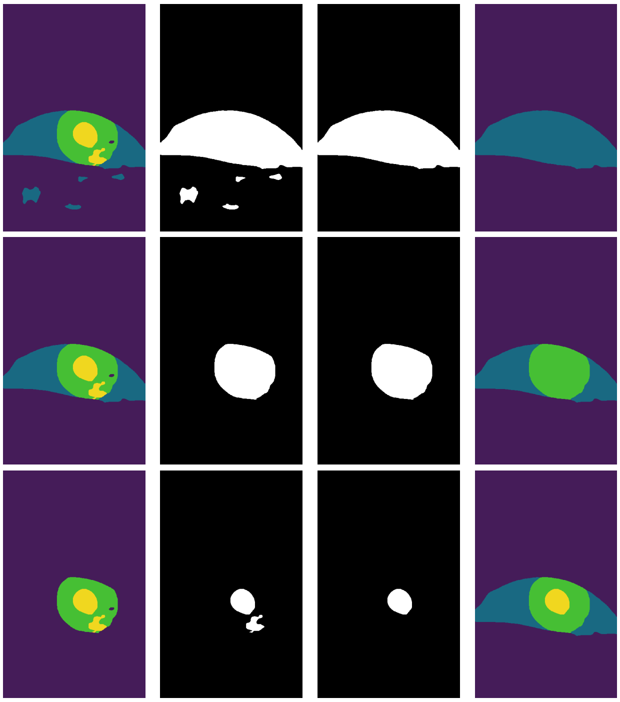


**FIGURE 4.** Example of filtering with connected components. From top to bottom, $i = 0, 1, 2$. From left to right: input, binary image Bw (step 2), Lc (step 5), and Mf (step 7).

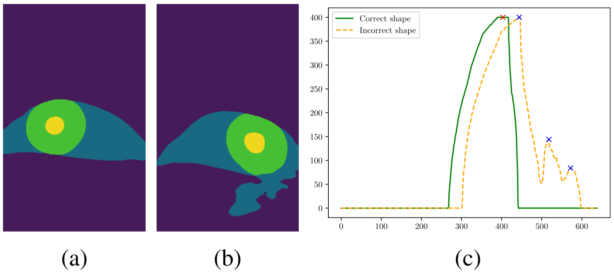


**FIGURE 5.** Horizontal projection profiles of a binary image with the eye region as the foreground. (a) Correct shape. (b) Incorrect shape. (c) Horizontal projection profiles of (a) and (b). Marks in (c) indicate peaks.

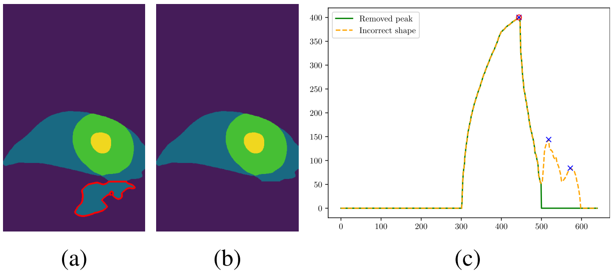


**FIGURE 6.** An example of peaks removal from the horizontal projection. (a) Region to remove. (b) After removal of peaks. (c) Horizontal projection profiles of (a) and (b). The region inside the red line of (a) corresponds to removed peaks.

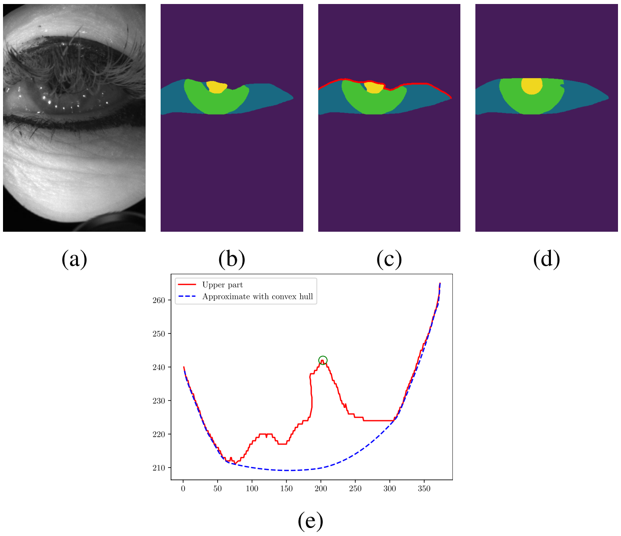


**FIGURE 7.** Approximation of the upper part of the eye with a convex hull. (a) Input image. (b) Output of deep network. (c) Visualization of upper part in red line. (d) After correction. (e) Illustration of region to extend where the peak is marked with green circle.

contains more than two peaks, as in Figure 5. These differences can be explained by the shape of the human eye, which approximates an ellipse with the longest diameter on the major axis and the shortest diameter on the minor axis. Apparently, in the set of lines which connect two points on the boundary of the ellipse and parallel with the major (minor) axis, the length of a line will increase if it come closer to the major (minor) axis. This property is also true when we use horizontal (vertical) axis instead of major (minor) axis, meaning that for each axis, we have only one line with maximum length and the length decreases gradually when we move further away from that line. Consequently, projection of a binary image of the eye region will increase or decrease gradually. Thus, we keep the largest peak and remove the others by setting zero values to pixels of those peaks. An example of this operation is shown in Figure 6.

In addition, deep networks miss pixels in the upper part of the eye due to loss of focus or heavy makeup (Figure 7b). We observe that they create a defect region in the results with different sizes depending on the influence. To deal with this problem, we first collect pixels in the upper boundary, as shown by the red line in Figure 7c and Figure 7e. Then, we determine whether a peak on that line exists or not, and if it does, the correction process continues. At this point, we find their convex hull, which is the smallest convex line that contains all of them and approximate a curve for that convex (blue line in Figure 7e). The reason for this approach is that, in real-world, the shape of the upper part of the eye is always a convex polygon. We extend the eye region with pixels belonging to the region between the red and blue lines. We also extend the visible area of the pupil (iris) by approximating a circle.

## III. EXPERIMENTAL EVALUATION

The experiment and evaluation of our approach have been done with Open Eye Dataset (OpenEDS) [22]. The data were collected from 152 individual participants using a head-mounted display (HMD) with two synchronized cameras. The semantic segmentation data contained 12,759 images annotated at a resolution of 640 × 400 with three

**TABLE 2.** Results of applying our model to the test set with model size is computed by Equation 12.

| Model | mIOU | Number of parameters ($T$) | Model size ($S$) | Overall ($M$) |
|---|---|---|---|---|
| mSegnet [22] | 0.907 | $350,000$ | 13.3 | 0.4911 |
| mSegnet w/ BR [22] | 0.914 | $350,000$ | 13.3 | 0.4946 |
| mSegnet w/ SC [22] | 0.895 | $40,000$ | 1.5259 | 0.7751 |
| MinENet [26] | 0.9230 | $222,440$ | 0.8485 | 0.9615 |
| Eye-MMS80 [25] | 0.9068 | $80,000$ | 0.3052 | 0.9534 |
| Eye-MS [25] | 0.9275 | $6,574,000$ | 25.08 | 0.4837 |
| Eyenet [27] | 0.949 | $258,021$ | 0.9843 | 0.9745 |
| RITnet [29] | 0.9528 | $248,900$ | 0.9495 | 0.9764 |
| $A_1$ [31] | 0.9482 | $104,456$ | 0.3985 | 0.97408 |
| $A_2$ [31] | 0.9483 | $104,720$ | 0.3995 | 0.97417 |
| $A_3$ [31] | 0.9485 | $104,728$ | 0.3995 | 0.97425 |
| **Our method** | **0.9491** | $\mathbf{104,728}$ | **0.3995** | **0.9746** |

components of an eye: the sclera, iris, and pupil. The data was divided into three sections for training, validation, and testing, which had 8,916, 2,403, and 1,440 samples, respectively. Our deep architecture is optimized with the training set. The validation data was only used to evaluate and select the model for testing on the unknown test set.

### A. IMPLEMENTATION

We implemented our networks in PyTorch [37] and trained for 200 epochs with a combination of the Adam optimizer and Stochastic Weight Averaging (SWA) technique [38], which has been proposed to substantially improve generalization in computer vision tasks by performing an equal average of the weights traversed by an optimizer. We applied SWA from the $51^{st}$ epoch. Our network is trained from scratch with weights from He initialization [39]. We used a batch size of 32 and weight decay of 1e−4. The network started with a learning rate of $1e-3$. After the $27^{th}$ epoch, we decreased it to $5e-4$ in 28 epochs, and retained that value in the rest of the training.

We observed that the eye region accounts for from 10 to 30 percent of an input image in most cases. Furthermore, each eye contains 3 parts, of which the sclera and pupil are the largest and the smallest region, with a difference of about 15 times or more in area. This led to a large imbalance between the 4 classes (background, sclera, iris, and pupil). To handle this complication, we applied generalized dice loss (GDL) [40] as an objective function for training our networks. For each image, it takes the form

$$GDL = 1 - 2\frac{\sum_c w_c \sum_{i,j}\left(p_{i,j,c}\cdot \hat{p}_{i,j,c}\right)}{\sum_c w_c \sum_{i,j}\left(p_{i,j,c} + \hat{p}_{i,j,c}\right)} \tag{8}$$

where $p_{i,j,c}$ and $\hat{p}_{i,j,c}$ are probabilities that the pixel located at $(i, j)$ in the image belongs to class $c$ in ground truth and output of the network, respectively. The weight attribute $w_c$ for class $c$ considers the contribution of each label and is used to formulate the following equation

$$w_c = \frac{\sum_c N_c}{C \cdot N_c} \tag{9}$$

where $C$ and $N_c$ are the number of classes and pixels belonging to class $c$. We changed the brightness of each image used for the training by a factor which was chosen uniformly from [0.5, 2.0] to tackle the difference in brightness due to light reflection. During the training, we kept only the model which had the least validation loss.

### B. EVALUATION

In our evaluation, the efficiency of a method is evaluated by balancing two aspects: performance and model complexity. To accommodate these requirements, the following equation [41] is used as the metric to measure efficiency

$$M = 50\left[mIOU + \min\left\{1, \frac{1}{S}\right\}\right] \tag{10}$$

where mIOU takes account of model accuracy and is defined by

$$\text{mIOU} = \frac{1}{C}\sum_{c=1}^{C}\frac{|P_c \cap G_c|}{|P_c \cup G_c|} \tag{11}$$

where $P_c$ and $G_c$ are the region of class $c$ from the ground truth and predicted mask, respectively. Model complexity is quantified by the size of the model with the value of $S$ in Equation 10 which takes the form

$$S = \frac{T \times 4}{1024 \times 1024} \tag{12}$$

where $T$ is the number of trainable parameters in the deep network. Stated differently, the metric $M$ is the combination of the mean intersection over union (mIOU) and model size in megabytes $S$.

Generally, our approach achieves a competitive result, with less than half of the number of trainable parameters compared to the best result on the OpenEDS dataset as shown in Table 2. In terms of speed, our system took only 16.56 seconds while RITnet [29] took 22.75 seconds to iterate over a set of $1,440$ test images on an NVIDIA 1080Ti GPU. A comparison between our predictions and those of RITnet [29] is shown in Figure 8. Our network is sensitive to reflection from eyeglasses resulted in an additional prediction below the eye, but it is eliminated by the post-processing step with horizontal prediction. In images with blur on the eye region, most networks produce the missing region. The missing region can be fixed in our system by using the convex hull of the eye as well as extending a region based on its properties.

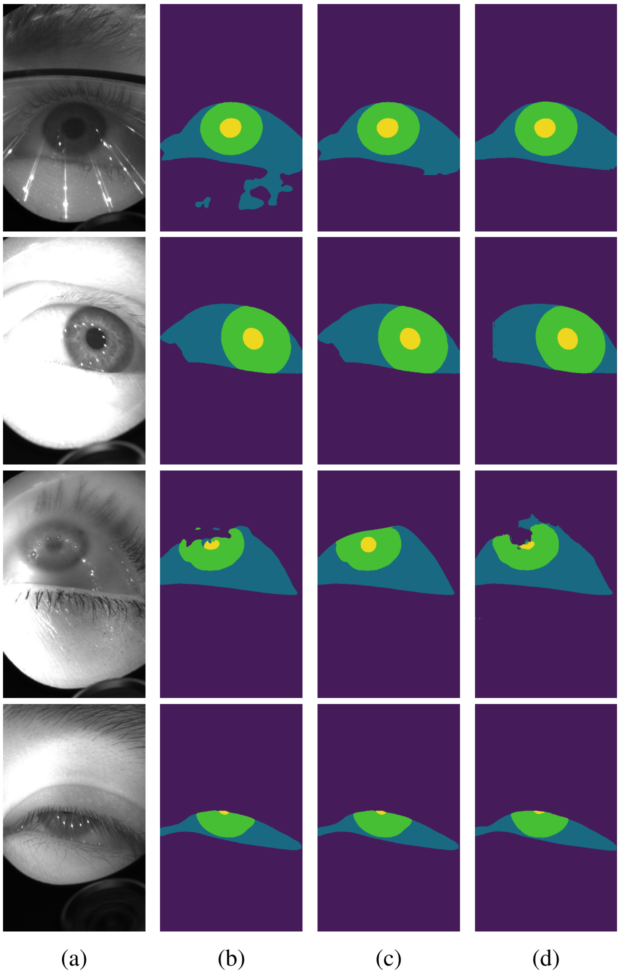


**FIGURE 8.** Example results on OpenEDS dataset. (a) Input image. (b) Result from our deep network. (c) Results of our system after the image processing on (b). (d) Results from RITnet [29].

We also evaluate our method with some reduction [31]. In $A_1, A_2$, and $A_3$, we exclude the shape correction with horizontal projection and convex hull. $A_3$ is created from the whole architecture, as  in Figure 1, whereas $A_1$ exclude $D_2$ and $A_2$ exclude the SE block. In terms of architecture, the network $A_3$ with $D_2$ as well as the *SE* block produces slightly better results, compared to $A_1$ or $A_2$. With an additional step to correct the shape with geometrical properties, our model proposed in this paper produces the best result among the variations of our system with less computational cost.

## IV. CONCLUSION

We present a lightweight deep architecture to localize 3 separate regions of the eye (sclera, iris, and pupil), along with a background. Using the geometrical properties of the eye, we employed a convex hull and horizontal projection to obtain the best result, with 0.4% lower accuracy but 1.4 times faster than RITnet [29]. More research is needed to apply the results from eye localization to the next step in gaze tracking or biometric systems, and to produce a fast and accurate system in real-world applications.

## REFERENCES

[1] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke, "Perceptually-based foveated virtual reality," in *Proc. ACM SIGGRAPH Emerg. Technol. (SIGGRAPH)*. New York, NY, USA: ACM Press, 2016, pp. 1–2.

[2] E. Wood and A. Bulling, "EyeTab: Model-based gaze estimation on unmodified tablet computers," in *Proc. Symp. Eye Tracking Res. Appl. (ETRA)*, New York, NY, USA: ACM Press, 2014, pp. 207–210.

[3] Z. Wang, J. Chai, and S. Xia, "Realtime and accurate 3D eye gaze capture with DCNN-based Iris and pupil segmentation," *IEEE Trans. Vis. Comput. Graphics*, early access, Aug. 28, 2019, doi: 10.1109/TVCG.2019.2938165.

[4] W. Sankowski, K. Grabowski, M. Napieralska, M. Zubert, and A. Napieralski, "Reliable algorithm for iris segmentation in eye image," *Image Vis. Comput.*, vol. 28, no. 2, pp. 231–237, Feb. 2010.

[5] C.-W. Tan and A. Kumar, "Unified framework for automated iris segmentation using distantly acquired face images," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4068–4079, Sep. 2012.

[6] A. Khotanzad and Y. H. Hong, "Invariant image recognition by zernike moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 489–497, May 1990.

[7] C.-W. Tan and A. Kumar, "Towards online iris and periocular recognition under relaxed imaging constraints," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3751–3765, Oct. 2013.

[8] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[9] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan, "Accurate iris segmentation in non-cooperative environments using fully convolutional networks," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–8.

[10] S. Lian, Z. Luo, Z. Zhong, X. Lin, S. Su, and S. Li, "Attention guided U-Net for accurate iris segmentation," *J. Vis. Commun. Image Represent.*, vol. 56, pp. 296–304, Oct. 2018.

[11] P. Radu, J. Ferryman, and P. Wild, "A robust sclera segmentation algorithm," in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2015, pp. 1–6.

[12] D. R. Lucio, R. Laroca, E. Severo, A. S. Britto, and D. Menotti, "Fully convolutional networks and generative adversarial networks applied to sclera segmentation," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–7.

[13] R. A. Naqvi and W.-K. Loh, "Sclera-net: Accurate sclera segmentation in various sensor images based on residual encoder and decoder network," *IEEE Access*, vol. 7, pp. 98208–98227, 2019.

[14] C. Wang, Y. He, Y. Liu, Z. He, R. He, and Z. Sun, "ScleraSegNet: An improved U-net model with attention for accurate sclera segmentation," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2019, pp. 1–8.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2015, pp. 234–241.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[17] P. Rot, Z. Emersic, V. Struc, and P. Peer, "Deep multi-class eye segmentation for ocular biometrics," in *Proc. IEEE Int. Work Conf. Bioinspired Intell. (IWOBI)*, Jul. 2018, pp. 1–8.

[18] 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.

[19] B. Luo, J. Shen, Y. Wang, and M. Pantic, "The iBUG eye segmentation dataset," in *Proc. Imperial College Comput. Student Workshop (ICCSW)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019, pp. 7:1–7:9.

[20] B. Luo, J. Shen, S. Cheng, Y. Wang, and M. Pantic, "Shape constrained network for eye segmentation in the wild," 2019, *arXiv:1910.05283*. [Online]. Available: https://arxiv.org/abs/1910.05283

[21] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: http://arxiv.org/abs/1512.09300

[22] S. J. Garbin, Y. Shen, I. Schuetz, R. Cavin, G. Hughes, and S. S. Talathi, "OpenEDS: Open eye dataset," 2019, *arXiv:1905.03702*. [Online]. Available: http://arxiv.org/abs/1905.03702

[23] 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

[24] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.

[25] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Eye-MMS: Miniature multi-scale segmentation network of key eye-regions in embedded applications," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–6.

[26] J. Perry and A. Fernandez, "MinENet: A dilated CNN for semantic segmentation of eye features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–6.

[27] P. Kansal and S. Devanathan, "EyeNet: Attention based convolutional encoder-decoder network for eye region segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3688–3693.

[28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2018, pp. 3–19.

[29] A. K. Chaudhary, R. Kothari, M. Acharya, S. Dangi, N. Nair, R. Bailey, C. Kanan, G. Diaz, and J. B. Pelz, "RITnet: Real-time semantic segmentation of the eye for gaze tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3698–3702.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[31] V. T. Huynh, S.-H. Kim, G.-S. Lee, and H.-J. Yang, "Eye semantic segmentation with a lightweight model," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3694–3697.

[32] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[34] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, in Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[35] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 1314–1324.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Las Vegas, NV, USA: IEEE Computer Society, Jun. 2016, pp. 770–778.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS Autodiff Workshop*, 2017, pp. 1–4.

[38] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. 34rd Conf. Uncertainty Artif. Intell. (UAI)*, 2018, pp. 1–12.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

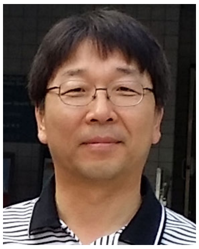
[40] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2017, pp. 240–248.

[41] R. Cavin, Y. Chen, I. Demir, S. J. Garbin, O. Komogortsev, I. Schuetz, A. Sharma, and S. S. Talathi. (2019). *Eye Tracking for VR and AR*. [Online]. Available: https://research.fb.com/wp-content/uploads/2019/05/Eye_Tracking_VR_AR_Proposal.pdf

**VAN THONG HUYNH** (Graduate Student Member, IEEE) received the B.S. degree from the Department of Computer Science and Engineering, Ho Chi Minh City University of Technology (VNUHCM), Vietnam, in 2018, and the M.S. degree from the School of Electronics and Computer Engineering, Chonnam National University, South Korea, in 2020, where he is currently pursuing the Ph.D. degree with the School of Electronics and Computer Engineering. His research interests include pattern recognition, human–computer interaction, and facial behavior analysis.

**HYUNG-JEONG YANG** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Chonbuk National University, South Korea. She is currently a Professor with the Department of Electronics and Computer Engineering, Chonnam National University, Gwangju, South Korea. Her main research interests include multimedia data mining, medical data analysis, social network service data mining, and video data understanding.

**GUEE-SANG LEE** (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from Seoul National University, South Korea, in 1980 and 1982, respectively, and the Ph.D. degree in computer science from Pennsylvania State University, in 1991. He is currently a Professor with the Department of Electronics and Computer Engineering, Chonnam National University, South Korea. His primary research interests include image processing, computer vision, and video technology.



**SOO-HYUNG KIM** (Member, IEEE) received the B.S. degree in computer engineering from Seoul National University, in 1986, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, in 1988 and 1993, respectively. Since 1997, he has been a Professor with the School of Electronics and Computer Engineering, Chonnam National University, South Korea. His research interests include pattern recognition, document image processing, medical image processing, and deep learning.