# EyeSeg: Fast and Efficient Few-Shot Semantic Segmentation

Jonathan Perry[0000−0002−1981−9740] and Amanda S. Fernandez[0000−0003−2397−0838]

University of Texas at San Antonio
{jonathan.perry,amanda.fernandez}@utsa.edu
http://www.cs.utsa.edu/∼fernandez/vail

**Abstract.** Semantic segmentation is a key component in eye- and gaze-tracking for virtual reality (VR) and augmented reality (AR) applications. While it is a well-studied computer vision problem, most state-of-the-art models require large amounts of labeled data, which is limited in this specific domain. An additional consideration in eye tracking is the capacity for real-time predictions, necessary for responsive AR/VR interfaces. In this work, we propose EyeSeg, an encoder-decoder architecture designed for accurate pixel-wise few-shot semantic segmentation with limited annotated data. We report results from the OpenEDS2020 Challenge, yielding a 94.5% mean Intersection Over Union (mIOU) score, which is a 10.5% score increase over the baseline approach. The experimental results demonstrate state-of-the-art performance while preserving a low latency framework. Source code is available: http://www.cs.utsa.edu/ fernandez/segmentation.html

**Keywords:** semantic segmentation, eye tracking, computer vision, OpenEDS2020

## 1 Introduction

The concept of foveated rendering has significant potential to improve upon the visual and computational performance of VR/AR applications. A critical underlying component of this technique is eye-tracking, which often relies on semantic segmentation - accurately and efficiently identifying regions of the eye.

Supervised training of neural networks for these segmentation tasks often requires extremely labor-intensive annotations as well as a relatively high volume of samples. In addition, these vision models are intended for embedded systems, such as head-mounted displays (HMD), and therefore we must consider an additional constraint of computational complexity in terms of the number of trainable or learned parameters.

Several efficient approaches to semantic segmentation for eye tracking on HMDs have shown the ability to reduce model complexity and demonstrate accurate performance in terms of mean Intersection Over Union (mIOU) [2, 3, 8, 11]. However, in this work, we additionally focus on the limited availability of large, fully-labeled datasets for this task. While VR/AR technologies continue

to increase in popularity, diversity in implementations reduces the consistency of such available and labeled data for evaluating deep learning models. We therefore explore the efficacy of existing models on datasets with limited amount of labeled data, as defined in the OpenEDS 2020 Challenge for Semantic Segmentation [9], and find a reduction in this measure of performance.

In response, we propose a new encoder-decoder framework, EyeSeg, which is designed for training where there is scarcity of annotated data as well as optimized for embedded systems. Our architecture improves on related state-of-the-art approaches [10, 11] in four main ways.

First, it improves upon the constraint of computational complexity by reducing the number of trainable parameters in our framework.

Second, it leverages a customized combined loss function of the standard categorical cross entropy (CCE) and generalised dice loss (GDL)[13].

Third, it applies well-established targeted data manipulation and augmentation techniques which have been demonstrated for their performance optimization [3].

Finally, it will utilize two different training methods to leverage capabilities of semi-supervised learning and identify the performance gain from a standard supervised learning approach.

In evaluation of our proposed approach, we measure the performance of EyeSeg against the Open Eye Dataset [9] for the 2020 Semantic Segmentation Challenge. The performance metric chosen for this challenge is mIOU. Additionally, we compare model complexity, as defined in the previous OpenEDS 2019 challenge [6], in order to thoroughly evaluate and compare with existing approaches. Our method demonstrates a significant improvement over the baseline model, and we additionally compare our proposed method with current state-of-the-art models for eye segmentation.

## 2 Related Works

As the availability of high-resolution digital media datasets continues to increase, research in segmentation algorithms has kept pace through strategic optimization and deep neural networks. Building on convolutional neural networks (CNNs) and fully convolutional networks (FCNs), segmentation architectures have benefited from techniques in pooling, filtering, and dilation [16]. In this section, related approaches to semantic segmentation are respectively described for accurate pixel-wise classification, complexity reduction methods, and imbalanced class representations.

**Encoder-Decoder Frameworks.** Convolutional encoder-decoder frameworks have been widely used for robust feature extraction in a range of computer vision applications.

SegNet [1] utilized this framework to improve upon scene understanding with a non-linear upsampling augmentation for FCNs. Chen et al[5] employed the

DeepLab[4] atrous convolution module, a dilation to further increase the performance of an encoder-decoder framework through exponentially larger receptive fields without an increased computational cost. UNet[12] introduced a patterned encoder-decoder design, containing residual connections in order to maintain spatial information from earlier layers within the encoder. A demonstrable trend in segmentation is to leverage a general encoder-decoder design for increasing performance and decreasing parameterization.

**Lightweight Frameworks.** The emerging technologies such as AR/VR or autonomous vehicles have shown that model complexity is a key factor in the application of segmentation models in real-world environments. ENet [10] improved upon model complexity towards real-time segmentation for autonomous vehicles. More recently, frameworks have been presented from OpenEDS challenge that improved in both computational complexity and performance capabilities towards AR/VR applications[2, 3, 8, 11]. These frameworks optimize for the number of trainable parameters within a deep neural network.

**Semi-Supervised and Unsupervised Training.** Domain adaptation and self-training have been widely adopted as techniques for a structured method of training with data that has a low amount of labeled samples, and this has been successful in many different domains, including synthetic to real domain adaptation for vehicle video sequences [14]. Recent work on self-training [17] that utilized a student-teacher format demonstrated state-of-the-art performance with a fast training schedule. Both of these works [14, 17] used a type of entropy based approximation for determining quality or confidence of inference.

## 3 EyeSeg Architecture

Our primary aim is to improve the performance and efficiency of semantic segmentation, especially for situations where there is limited availability of labeled data. In this section, we outline our proposed neural network architecture and describe its total loss function.

### 3.1 Network Architecture

Figure 1 provides a high-level view of the composition of EyeSeg, an encoder-decoder architecture. EyeSeg consists of 4 encoder blocks which store feature maps learned at each step prior to the down sampling portion connecting to the subsequent encoder blocks. The decoder portion of EyeSeg upsamples in a mirrored or patterned fashion with respect to the encoder and utilizes the store feature maps from the encoding as an alternative path to sustain simple high level features.

Recent approaches[3, 11] specifically for eye-tracking have shown an increase in accuracy from applying different mechanisms to an encoder-decoder.

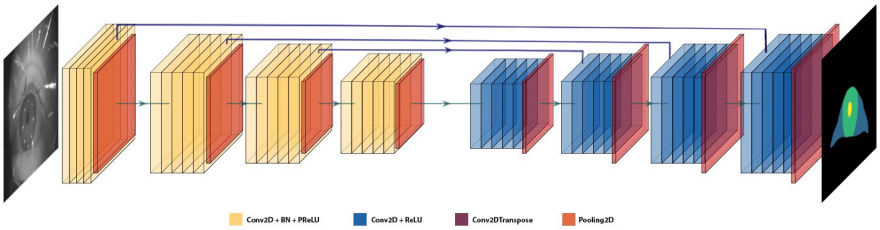


**Fig. 1.** Visualization of the proposed framework (high-level view). From left: Input image $640 \times 640$ in gray scale format through $4\times$Encoder blocks & $4\times$Decoder blocks to a predicted mask of background, sclera, iris, and pupil.

Similarly, EyeSeg employs two of these components, residual connections and dilated convolutional layers, where these different components combined substantially increase the performance without impacting computational complexity drastically. However, both the encoder and decoder proposed are distinct variants from existing architectures [3, 11] in both size and structure. Table 1 outlines the framework of EyeSeg in more detail, breaking down the internal blocks, output sizes, and types within layers.

### 3.2 Encoder

Each encoder block consists of 4 convolutional layers, that are paired with Parametric rectified Linear Unit (PReLU) [7] and Batch Normalization (BN) layers per convolution. There are 2 variants to the basic structure of an encoder block, which modify one of the convolutional layers. The variant will be a dilated convolutional layer or a pooling layer. Finally, our encoder block utilizes average pooling layers for a more accurate localization than what is provided in max pooling layers. A single encoder block is visualized in Figure 2.

### 3.3 Decoder

As illustrated in Figure 2, the decoder blocks each have 4 primary components that consists of convolutional, activation, upsampling, and normalization layers. Each convolutional layer is paired with a Rectified Linear Unit (ReLU). A convolutional transpose layer is leveraged for the task of upsampling.

With an emphasis in reduction of computational complexity, we forgo the additional BN layers commonly incorporated into the decoder blocks at this stage [11].

In order to sustain spatial information from earlier layers, we implemented residual connections pairing the appropriate encoder blocks to the respective decoder blocks.

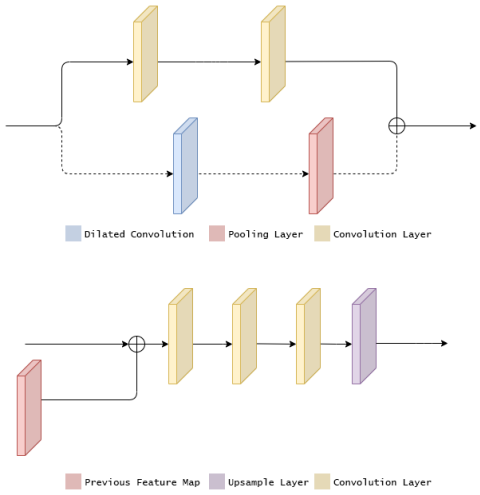


**Fig. 2.** From top: A flowchart of a single encoder block where the dotted line represents conditionally applicable layers, a single decoder block where the previous feature map shown in red is a residual connection.

### 3.4 Loss Function

With motivation from RITNet [3], our implementation utilizes a customized total loss function which can be represented in 2 parts: cross entropy and generalised dice loss. The total loss function for our proposed method is as follows:

$$\mathcal{L}_{loss} = \mathcal{L}_{cce} + \mathcal{L}_{gdl} \tag{1}$$

$$\mathcal{L}_{cce} = -\frac{1}{N}\Sigma_{l=1}\Sigma_{n=1}^{N} r_{ln}\log(p_{ln}) \tag{2}$$

$$\mathcal{L}_{gdl} = 1 - 2\frac{\Sigma_{l=1} w_l \Sigma_{n=1}^{N} r_{ln} p_{ln}}{\Sigma_{l=1} w_l \Sigma_{n=1}^{N} r_{ln} + p_{ln}} \tag{3}$$

where $\mathcal{L}_{cce}$ is a standard implementation of categorical cross entropy and $\mathcal{L}_{gdl}$ is an implementation of a generalized dice loss function [13] for imbalanced class features. The aim of this combined loss function is to mitigate the over-representation of one or many class features $l$ within each sample $n$, comparing the ground truth (target) $r$ with the predicted values $p$.

## 4 Experiments and Results

### 4.1 openEDS 2020 Challenge Data

In this work, the eye segmentation subset of the Open Eye Dataset 2020 [9] is used for evaluation. This dataset consists of 29,476 images, from 74 different

**Fig. 3.** Sampled images from the OpenEDS2020[9] dataset. From left column: Image with reflection and participants eye partially open, participants eye occluded by the eye lid, participants eye fully open.

participants, ranging in ethnicity, gender, eye color, age, and accessories (such as make-up and glasses). Shown in Figure 3, the dataset has variations of images that allow for a range of different real world challenges for applications of eye tracking including participants with accessories, nearly visible eyes, partially occluded eyes, or fully observable eyes. Further, the images were from 200 different sequences of 30 second video recordings from 74 participants. The sequences of recorded sessions contain only a few annotations, where approximately 5% of the entire dataset contained annotations. Labels provided in the form of pixel-level masks denoting eye region, iris, and pupil were manually annotated by two or more individuals. Overall, the labeled portion of this dataset consists of only 2,605 annotated images. For the purposes of challenge, a hidden test set is made unavailable, comprised of five of the annotated images per sequence. This leaves only 1605 total annotated images for our training and validation purposes.

### 4.2 Data Augmentation

In order to account for the variety of challenging categories within the dataset viewed in Figure 3, we propose two approaches to reduce the most noticeable undesirable properties of the original images. First, we utilize a technique to amplify the contrast of the image to improve upon low light areas of the original image. Second, we apply image denoising techniques to smooth the prominent reflections caused by participants wearing accessories. Additionally, we performed data manipulation techniques such as horizontal flipped or mirrored samples in order to combat the detriment to training on sparse amount of data. The process of both denoising techniques and contrast amplification are shown in Figure 4.

**Adaptive Histogram Equalization.** Contrast Limited Adaptive Histogram Equalization (CLAHE) [15] is an enhancement method for improving the quality of images and video where visibility is less than satisfactory. A key component to CLAHE is the clipped or limited range of its visibility enhancement, whereas the standard AHE algorithm will allow for overly amplified images to occur and

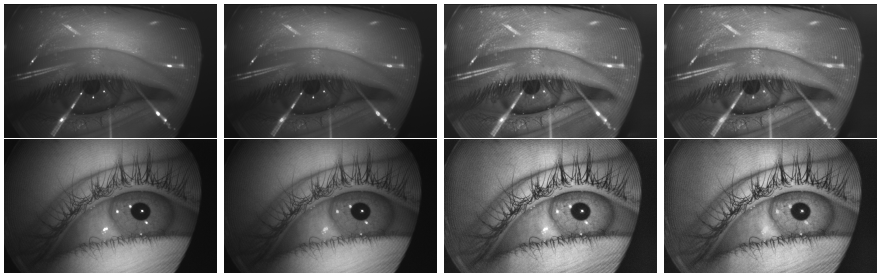

**Fig. 4.** This figure shows the application of both CLAHE and noise reduction from two different participants where the top row is a participant with reflection from glasses as well as low light and the bottom row is a participant with partial reflection. From Left: Original image, noise reduction applied, CLAHE applied, and fully pre-processed with both CLAHE and noise reduction.

presents the problem of trading off low light samples with over exposed samples. Our proposed EyeSeg utilizes this CLAHE enhancement to employ a contrast amplification to the images in order to achieve more visibility of class features.

**Image Noise Reduction.** The front facing sensors of AR/VR devices such as HMDs usually are accompanied with both visible and infrared light to illuminate the participants eye for more accurate eye-tracking. The trade-off with deploying these types of emitters will result in glare and reflections from the participants eye itself and additional noise will be caused if the participant is wearing glasses. We use Gaussian filtering as a noise reduction method to provide more clear representations of the images.

### 4.3 Training

The lack of manually annotated images causes a significant detriment to the capabilities of the traditional methods of training. In this section, we discuss the two experimental training methods applied to EyeSeg. Initially, the 1,605 annotated images are utilized in a supervised learning environment. Second, we describe the semi-supervised training method applied.

**Supervised Training.** We trained EyeSeg with an ADAM optimizer at a initial learning rate of 1e-3, and is lowered to 1e-4 once there is a plateau. The training process is terminated within 200 epochs. The training and inference of this model were performed on the padded image size of $640 \times 640$. Our network was tested on the Open Eye Dataset [9] hidden test samples achieving a mIOU score of 0.945 shown in Table 2. Additionally, we trained segmentation models[3, 11] from OpenEDS 2019 Challenge [6] using the same training method to encompass a more robust comparison of EyeSeg.

**Semi-Supervised Training.** Utilizing the entire dataset, We trained EyeSeg with pseudo labels generated for the portion of the dataset without annotations. Our proposed semi-supervised method aims to minimize entropy to ensure quality pseudo labels similar to recent works [17, 14]. Additionally, we incorporated the 1,605 annotated images from the Open Eye Dataset [9]. The pseudo labels generated were utilized only whenever the entropy of each pixel-wise classification demonstrated high confidence or low entropy. Entropy for a single sample is written as follows:

$$\mathcal{E}_{entropy} = -\Sigma_{i,j=0}^{N} \Sigma_{l=1} p_{ijl} \log(p_{ijl}), \tag{4}$$

where $\mathcal{E}_{entropy}$ will demonstrate the confidence of the pseudo label determined by evaluating each pixel at $ij$ until $N^{th}$ pixel for each label or class $l$. Entropy will be low if only one class per $ij$ is classified with high confidence from inference. Our network achieved a marginal score increase of 0.06 over the supervised learning method to a total score of 0.951. However, due to a lack of accessibility to the Open Eye Dataset [9] hidden test samples we could not accurately compare this iteration to the previous supervised method.

### 4.4 Results

The 2020 OpenEDS Semantic Segmentation Challenge includes a leaderboard, ranking submissions by mIOU score. A baseline encoder-decoder network [9] was provided, which was loosely based upon SegNet[1], an encoder-decoder architecture with relatively few parameters and a base score of 0.84. Additional results are provided in Table 2, including mIOU, but also breaking down the performance across the 4 semantic categories and including the number of parameters in the models. Related works in this table include top-performing models from the 2019 OpenEDS challenge, RITNet[3] and MinENet[11]. EyeSeg demonstrates a higher mIOU score, consistently improving across the background, sclera, iris, and pupil semantic categories. While the number of parameters in EyeSeg are streamlined in comparison with related works, the baseline model was significantly smaller than our proposed architecture. This trade-off provides our model with improved performance, but we will discuss further plans to reduce size in the following section.

A visual evaluation of the effectiveness of EyeSeg is shown in Figures 5 and 6. In Figure 5, images from the dataset are shown in the first column, followed by the ground truth annotation, and our predicted segmentation. Despite reflections within the eye, partial occlusion by eyelid, and differences in lighting, the EyeSeg predictions are fairly close to the ground truth. In comparison, Figure 6 looks at challenging edge cases in the dataset - reflections from glasses, severe occlusion by eyelid, and varied lighting. In these instances, the EyeSeg predictions are often close, some degradation exists in the confidence of boundaries, such as in the left side of the final image.

## 5 Conclusion

In this work, we introduce EyeSeg, a generalized method for few-shot segmentation with the use of an efficient encoder-decoder and customized total loss function. We apply EyeSeg to the Open Eye Dataset[9], a challenge for semantic segmentation of eye regions in images taken for VR/AR displays. Our method uses a combined loss function to reduce the impact of imbalanced class features often prevalent in real-world datasets. Additionally, several data augmentation techniques are applied to mitigate the limited amount of labeled data, as well as to accommodate for challenging categories of images within the dataset, such as makeup, glasses, and closed eyelids.

We demonstrate performance of EyeSeg against the baseline implementation on the challenge [9], outperforming by 10.5% mIOU. We also compare with recent related approaches, achieving state-of-the-art performance while maintaining a lightweight design for the capability of real-world use in AR/VR environments.

In future work, we aim to further optimize EyeSeg by reducing the number of parameters, increasing performance per mIOU, and addressing shortcomings on the outlier data, identified in Figure 6. Since the image data is a sequence of video frames an application of memory units such as an LSTM could have a positive impact on the accuracy of EyeSeg. While our data augmentation methods are beneficial to the performance of EyeSeg in low light or noisy environments, it is not entirely solved and could be addressed from the application of a memory unit or additional pre-processing techniques. We plan apply our approach to further domains and data which contains varying levels of class feature imbalances, and limited labeled data.

| Name | Type | Output Size |
|---|---|---|
| input | | $16 \times 640 \times 640$ |
| Encode Block 1.0 | | $32 \times 640 \times 640$ |
| Encode Block 1.1 | | $32 \times 640 \times 640$ |
| Encode Block 1.2 | | $32 \times 640 \times 640$ |
| Encode Block 1.3 | downsampling | $32 \times 320 \times 320$ |
| Encode Block 2.0 | | $32 \times 320 \times 320$ |
| Encode Block 2.1 | dilated (2×2) | $32 \times 320 \times 320$ |
| Encode Block 2.2 | dilated (4×4) | $32 \times 320 \times 320$ |
| Encode Block 2.3 | downsampling | $32 \times 160 \times 160$ |
| Encode Block 3.0 | | $32 \times 160 \times 160$ |
| Encode Block 3.1 | dilated (2×2) | $32 \times 160 \times 160$ |
| Encode Block 3.2 | dilated (4×4) | $32 \times 160 \times 160$ |
| Encode Block 3.3 | downsampling | $32 \times 80 \times 80$ |
| Encode Block 4.0 | | $32 \times 80 \times 80$ |
| Encode Block 4.1 | | $32 \times 80 \times 80$ |
| Encode Block 4.2 | | $32 \times 80 \times 80$ |
| Encode Block 4.3 | | $32 \times 80 \times 80$ |
| Decode Block 1.0 | | $32 \times 80 \times 80$ |
| Decode Block 1.1 | | $32 \times 80 \times 80$ |
| Decode Block 1.2 | | $32 \times 80 \times 80$ |
| Decode Block 1.3 | upsampling | $32 \times 160 \times 160$ |
| Decode Block 2.0 | residual connection | $64 \times 160 \times 160$ |
| Decode Block 2.1 | | $32 \times 160 \times 160$ |
| Decode Block 2.2 | | $32 \times 160 \times 160$ |
| Decode Block 2.3 | upsampling | $32 \times 320 \times 320$ |
| Decode Block 3.0 | residual connection | $64 \times 320 \times 320$ |
| Decode Block 3.1 | | $32 \times 320 \times 320$ |
| Decode Block 3.2 | | $32 \times 320 \times 320$ |
| Decode Block 3.3 | upsampling | $32 \times 640 \times 640$ |
| Decode Block 4.0 | residual connection | $64 \times 640 \times 640$ |
| Decode Block 4.1 | | $32 \times 640 \times 640$ |
| Decode Block 4.2 | | $32 \times 640 \times 640$ |
| Decode Block 4.3 | | $32 \times 640 \times 640$ |
| Output | | $4 \times 640 \times 640$ |

**Table 1.** Architecture of our proposed method. Output sizes are provided for input size of $640 \times 640 \times 1$.

| | mIOU | background | sclera | iris | pupil | #parameters |
|---|---|---|---|---|---|---|
| Baseline[9] | 0.84 | 0.971 | 0.674 | 0.835 | 0.835 | 40k |
| MinENet[11] | 0.91 | 0.99 | 0.83 | 0.93 | 0.89 | 222k |
| RITNet[3] | 0.93 | 0.99 | 0.87 | 0.95 | 0.915 | 250k |
| **EyeSeg** | **0.945** | 0.99 | 0.89 | 0.95 | 0.95 | 190k |

**Table 2.** Comparison of semantic segmentation approaches on the OpenEDS dataset, as of submission. The Baseline is the model provided in the OpenEDS Semantic Segmentation Challenge 2020.

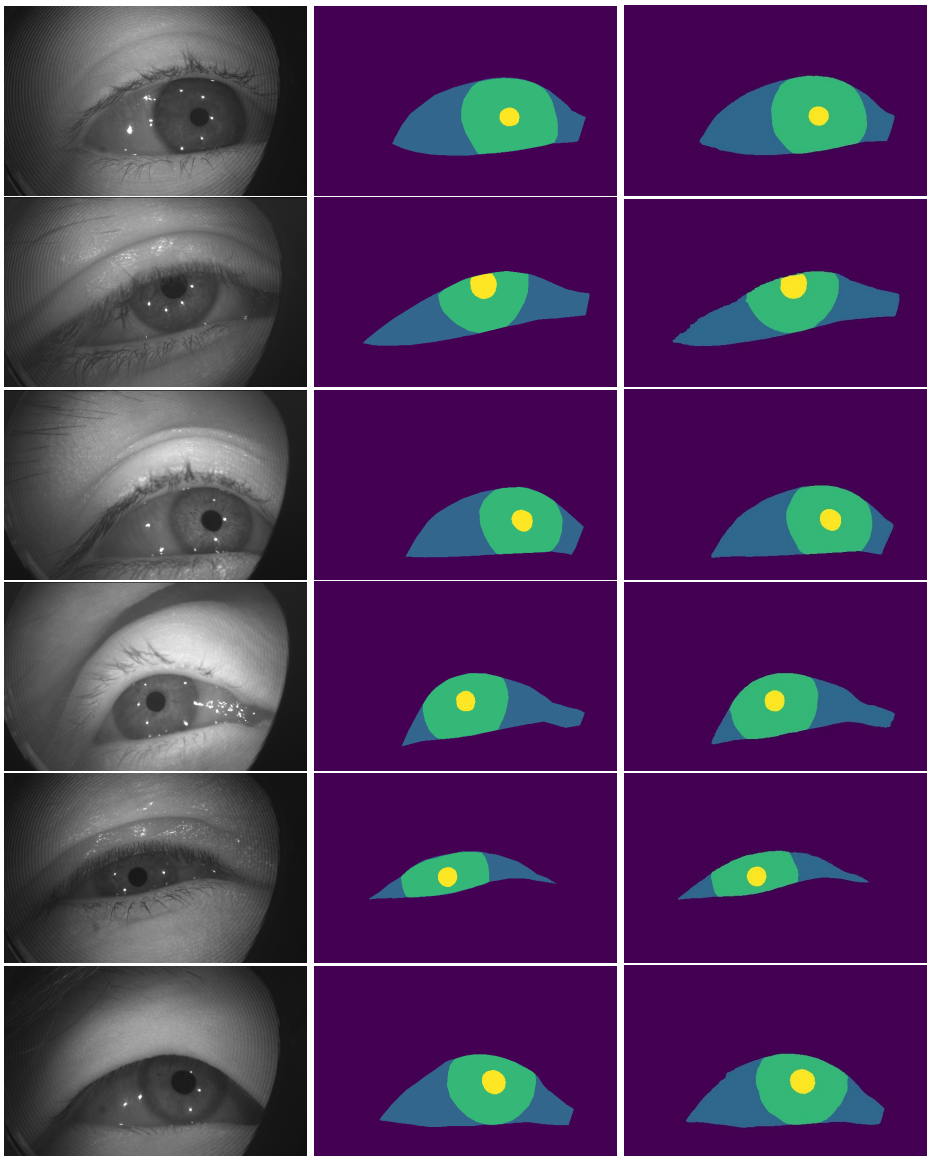

**Fig. 5.** Results from EyeSeg without any samples that include low visibility or reflections. From Left: Original input image, Ground truth or target value, predictions.

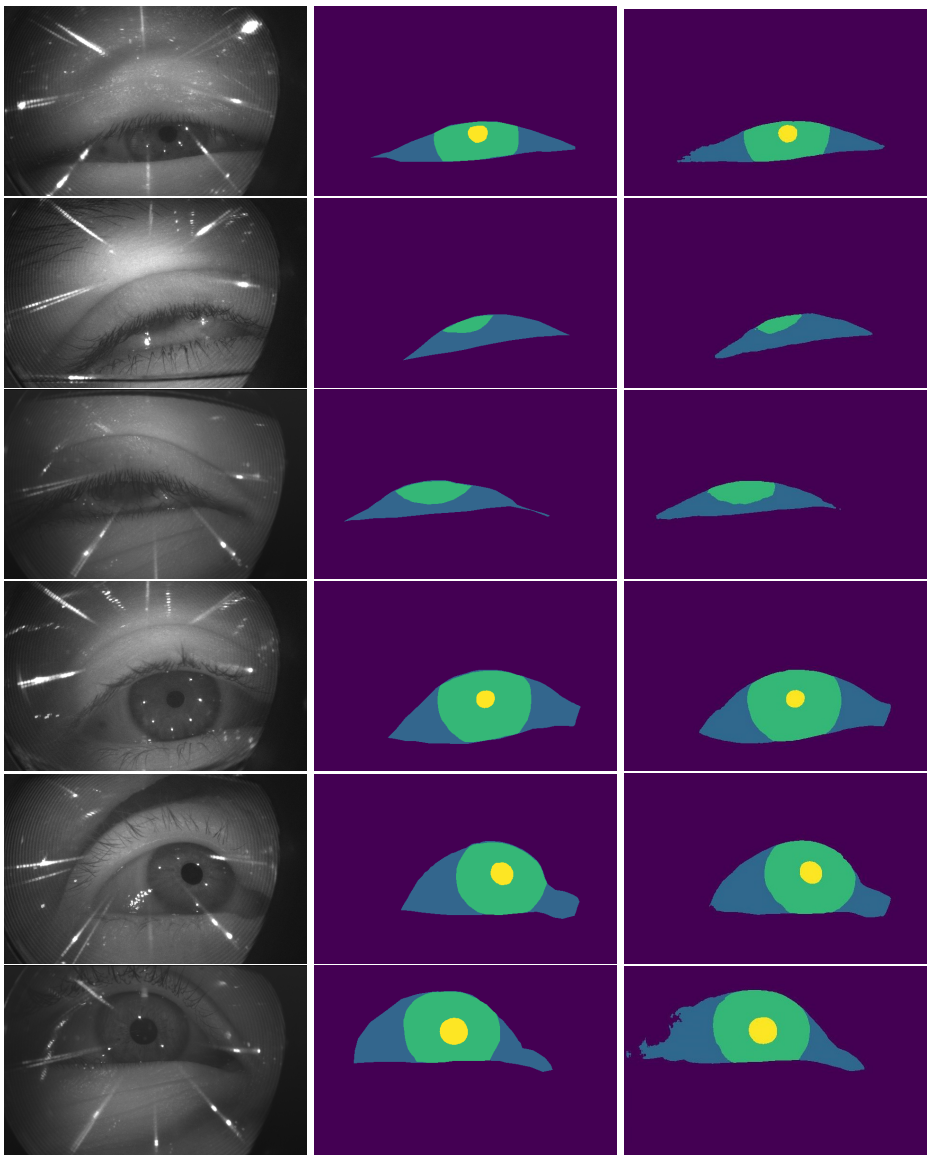

**Fig. 6.** Results from EyeSeg that posed more of a challenge including low light and reflections. From Left: Original input image, Ground truth or target value, predictions.

## References


1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
2. Boutros, F., Damer, N., Kirchbuchner, F., Kuijper, A.: Eye-mms: Miniature multi-scale segmentation network of key eye-regions in embedded applications. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
3. Chaudhary, A.K., Kothari, R., Acharya, M., Dangi, S., Nair, N., Bailey, R., Kanan, C., Diaz, G., Pelz, J.B.: Ritnet: Real-time semantic segmentation of the eye for gaze tracking. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3698–3702 (2019)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP** (06 2016). https://doi.org/10.1109/TPAMI.2017.2699184
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: The European Conference on Computer Vision (ECCV) (September 2018)
6. Garbin, S.J., Shen, Y., Schuetz, I., Cavin, R., Hughes, G., Talathi, S.S.: Openeds: Open eye dataset. arXiv preprint arXiv:1905.03702 (2019)
7. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1026–1034 (Dec 2015). https://doi.org/10.1109/ICCV.2015.123
8. Kim, S.H., Lee, G.S., Yang, H.J., et al.: Eye semantic segmentation with a lightweight model. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3694–3697. IEEE (2019)
9. Palmero, C., Sharma, A., Behrendt, K., Krishnakumar, K., Komogortsev, O.V., Talathi, S.S.: Openeds2020: Open eyes dataset (2020)
10. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
11. Perry, J., Fernandez, A.: Minenet: A dilated cnn for semantic segmentation of eye features. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
12. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
13. Sudre, C., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. pp. 240–248 (09 2017). https://doi.org/10.1007/978-3-319-67558-9_28
14. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2517–2526 (2019)
15. Yadav, G.: Contrast limited adaptive histogram equalization based enhancement for real time video system (09 2014). https://doi.org/10.1109/ICACCI.2014.6968381

16. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR) (2016)
17. Zou, Y., Yu, Z., Vijaya Kumar, B.V.K., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 297–313. Springer International Publishing, Cham (2018)