

Research Article

The Use of Synthetic Data to Facilitate Eye Segmentation Using Deeplabv3+

Melih Öz^{1,*}, Taner Danışman¹, Melih Günay¹, Esra Zekiye Şanal², Özgür Duman² and Joseph William Ledet¹

¹Akdeniz University, Faculty of Engineering, Computer Engineering Department

melihoz@akdeniz.edu.tr; tdanisman@akdeniz.edu.tr; mgunay@akdeniz.edu.tr; josephledet@akdeniz.edu.tr

²Akdeniz University, School of Medicine, Department of Pediatric Neurology

esraguzey@akdeniz.edu.tr; oduman@akdeniz.edu.tr

*Correspondance: melihoz@akdeniz.edu.tr

Received: 8th January 2021; Accepted: 26th March 2021; Published: 1st July 2021

Abstract: The human eye contains valuable information about an individual's identity and health. Therefore, segmenting the eye into distinct regions is an essential step towards gathering this useful information precisely. The main challenges in segmenting the human eye include low light conditions, reflections on the eye, variations in the eyelid, and head positions that make an eye image hard to segment. For this reason, there is a need for deep neural networks, which are preferred due to their success in segmentation problems. However, deep neural networks need a large amount of manually annotated data to be trained. Manual annotation is a labor-intensive task, and to tackle this problem, we used data augmentation methods to improve synthetic data. In this paper, we detail the exploration of the scenario, which, with limited data, whether performance can be enhanced using similar context data with image augmentation methods. Our training and test set consists of 3D synthetic eye images generated from the UnityEyes application and manually annotated real-life eye images, respectively. We examined the effect of using synthetic eye images with the Deeplabv3+ network in different conditions using image augmentation methods on the synthetic data. According to our experiments, the network trained with processed synthetic images beside real-life images produced better mIoU results than the network, which only trained with real-life images in the Base dataset. We also observed mIoU increase in the test set we created from MICHE II competition images.

Keywords: Deep Neural Networks; Eye Segmentation; Image Augmentation; Synthetic Data

1. Introduction

The eyes are part of the human body that are responsible for visual inputs. Additionally, the eyes show signs about the wellness of a human. For example, redness of the sclera can be a sign of allergies or infections. The pupil might lose its brightness due to cataract disease. They can lose their shape due to genetics or accidents [1, 2]. The eyes are protected from outside harm by the cornea. Inside of the cornea, there are three main parts of the eye: the iris, colored part of the eye that has unique patterns for each person [3]; the sclera, the white, veiny, and protective part of the eye; and the pupil, the inner circle inside the eye. Due to their informative nature, the eyes have always been a point of interest. Visual inputs are the first step to visual understanding, a fundamental part of our continued survival. One of the ways we can extract eye information is called eye segmentation. Eye segmentation is a task used to get boundaries of the eye parts we want to explore. For example, if an eye image is annotated to have two parts as iris and background, this is commonly referred to as iris segmentation [4]. Figure 1 shows an example for eye region segmentation.



Figure 1. Eye regions and the segmentation mask.

Early iris segmentation methods commonly used the Gaussian filter to reduce noise in the image. This method then uses a circular edge detector to find the iris boundaries. The most popular circle detection algorithm is the circular Hough transformation, and one of the most known applications was proposed by Wildes [5]. Wildes also used the Gaussian filter to smooth the eye image. He applied an almost vertical edge detector to get the outer iris boundary and created the eye image's edge points. Then, he used the circular Hough transformation to get the outer iris boundary. The circular Hough transformation finds the best fitting circle relative to the initial point by comparing each edge point to the initial point. The potential circle contains most edge points returned. After finding the circle, the inner iris boundary was found by applying an edge detector without any orientation and applying the circular Hough transformation within the range of the first found circle. Eyelid boundaries are also calculated as two different parabolic arcs. Horizontal edge detection is used to find parabolic arcs. Figure 2, as an example, shows gradient-based edge detection results at different directions applied to the eye image.

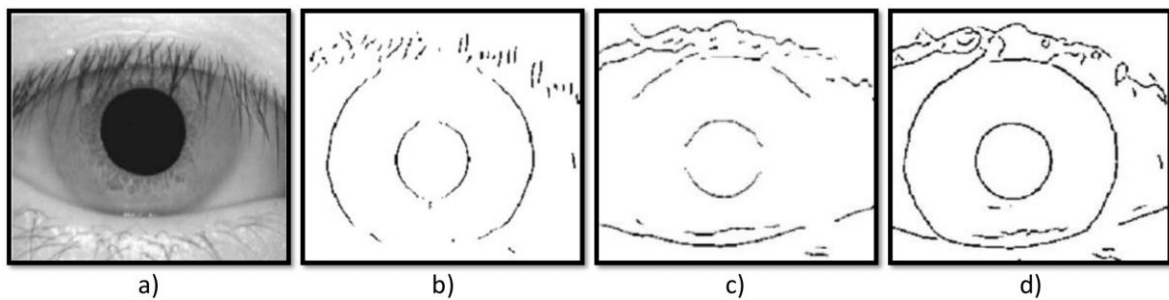


Figure 2. a) The eye image. b) result of vertical, c) horizontal, and d) without any orientation edge detector.

The problem with early iris segmentation methods is their conditions were often too restrictive, the requirement of near-infrared images and iris not being circular, repeated patterns that are near impossible to get when a 3-D model of the eye is considered [4]. However, in restricted and controlled conditions, early methods had a satisfactory accuracy rate [6]. The problem of the eye not being circular and concentric was solved with later research [7, 8].

When the sclera is extracted from an eye image, it is called sclera segmentation. Sclera information can also be used as a biometric [9] feature. Image processing-based sclera segmentation methods are somewhat similar to iris segmentation methods. The iris boundaries are detected using edge detection methods, and the sclera region is detected by color intensity differences since the sclera region is mostly white [9, 10]. These methods also work in restricted conditions. To deal with unconstrained eye segmentation, more powerful segmentation methods that adopt different inputs must be used. This is where Convolutional Neural Networks shine. Convolutional Neural Networks are successful in iris and sclera segmentation tasks [4, 11–13]. Although Convolutional Neural Networks' usage goes back several decades, they became used in standard segmentation methods in recent years in image segmentation tasks. Neural networks require input and output pairs to understand the problem and create the addressing filters to address a problem. For different images to be segmented, an adequate number of segmented input and output pairs need to be given to training the network. Thus, this creates a challenge compared to image processing methods, requiring specific calculations to do the task. There is a need for a massive amount of manually segmented data [14]. Even so, for broader and more accurate segmentation using deep neural networks, networks need to be trained with a massive amount of data. Therefore, finding a large quantity of data and creating quality segmentation masks with them is necessary to create a network for the problem [15].

Deep Convolutional Neural Networks became popular after the release of the ImageNet database and the introduction of the AlexNet [16]. The idea was to keep the segmentation data towards multiple layers of filters to contain as much information as possible. Nevertheless, while going in-depth with filters, some information was lost. This problem rose from mapping two layers directly [17]. To tackle this problem, the idea of learning the difference between layers instead of direct mapping was created. With that idea problem of losing information while going deeper into layers improved. Still, a more in-depth network, requires more computation power. To get more information in a layer, parallel convolutional operations with dimension reduction was proposed [18]. Later, that idea was improved by separable convolutions on every channel with dimension reduction. Ultimately, multiple contexts in a layer are compressed without increasing computational complexity [19]. While computing through the layers, dimensions shrink due to convolution operations, and to get the original dimensions, directly using up-sampling methods can lead to data loss. To tackle this problem, encoder-decoder networks were proposed [20]. In contrast, encoder structures work like regular networks without up-sampling operations. Decoder structures take low-level features and up-samples them by convolving with changeable de- coder layers. Using deep neural networks segmenting operation can be done by the filters created by the network structure [14].

This study used Google's Deeplabv3+ network, which acquired promising results in semantic segmentation task, using its separable convolution method with different rates [21]. Our study has shown that using a synthetic data set along with a real-life data set improved the eye segmentation results. The main contribution of the study is two-fold:

- First, we showed that synthetic images and real images improve overall segmentation accuracy compared to the model trained only on real-life images.
- Second, we contribute to the literature by creating a manually annotated eye segmentation dataset.

We present our materials, including the datasets used in the experiments and the methodology in Section 2. Section 3 provides details of the experiments done using both the synthetic and real-life dataset. Finally, Section 4 provides a discussion of the experimental results and concludes the study.

2. Methodology

Semantic segmentation is a task where every pixel of the image is assigned to pre-determined classes. For experimenting with real-world scenarios, images that were taken for the Akdeniz University Scientific Research Projects Coordination Unit Project ID: TTU-2018-3295 were used. Images using different angles, distances, and lighting conditions were used, making them good candidates for challenging image segmentation [22]. In our work, eye images are segmented into four regions: the sclera, iris, pupil, and the background, as shown in Fig 1. This segmentation is useful for tracking eye movements and pupillary responses. While keeping test data the same for all experiments, we explore the synthetic data's effect, image augmentation methods on synthetic data for our segmentation task. Since there is no similar work conducted or standard dataset available for this segmenting method, we will be exploring the improvements using stated methods. We used mIoU metric to measure the proposed method's performance due to its being a meaningful metric for evaluating segmentation quality [20].

For the data required in this study, the UnityEyes interface, which generates synthetic human eyes, was used [23]. With this interface, generating an automatically annotated dataset is possible. Another advantage of UnityEyes is that it adds a light reflection to the generated images, making it closer to real-life conditions. Thus, the image augmentation methods cause synthetic data to be similar to real-life data. When we consider image-to-image translation problems, it is hard to cover all possible test conditions; machine-generated images can help cover more.

We used Google's Deeplabv3+ neural network, as shown in Figure 3, for the segmentation task. Deeplabv3+, in addition to the previous work, adds depthwise separable convolution to get better results in benchmark datasets.

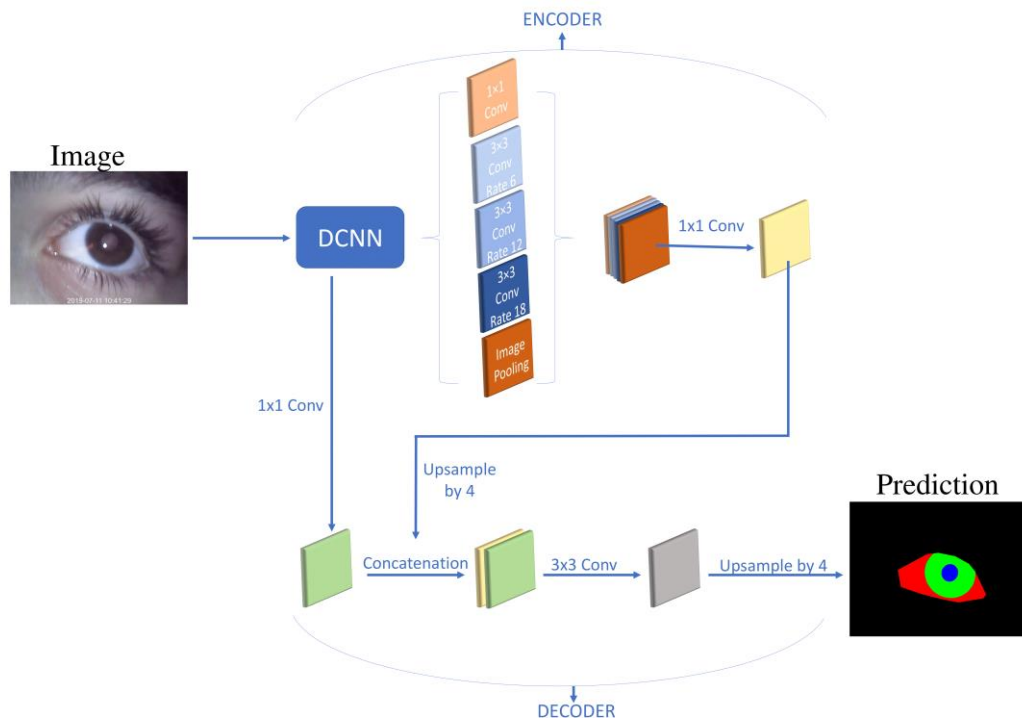


Figure 3. Deeplabv3+ network structure [21].

Deeplabv3+ uses depthwise separable convolution with different rates of atrous convolutions (1) to get the context [21]. This results in increased performance and speed compared to traditional convolution operation.

$$I_{out}[i] = \sum_k I_{in}[i + r \cdot k]f[k] \quad (1)$$

When we choose r value as 1, it becomes a traditional convolution operation. I_{out} is convolution output, f is the filter, and I_{in} is the input image. Depthwise separable convolution merges every channel of the layer with 1×1 convolution on every channel simultaneously, instead of filtering each channel separately and merging them later. The decoder first uses depthwise separable convolution to capture low-level features, then merge them with encoder results, using 3×3 convolution to polish results and upsample to get segmentation results [15].

2.1. Dataset

Our real-life dataset consists of 350 images that vary in distance, angle, and lightning. These images are manually annotated. The images have a resolution of 640×480 and have RGB channels. This set of images will be referred to as the base dataset. As synthetic data, 900 eye images were generated from the Unityeyes software. Synthetic data created from the Unityeyes already have sclera and iris annotations; therefore, only the iris region was annotated by us. A set of images generated from Unityeyes software will be referred to as the syn dataset. Syn dataset images rotated from 0 to 25 degrees in both directions from the center randomly. Their pixel values darkened from 0.5 to 0.9 of the original value. They are likewise blurred, and motion blurred in both directions with random kernel sizes from 1 to 15, which means some images stayed without any blurring. All these operations are applied to the images once.

With these augmentation methods applied to the synthetic dataset, a synthetic processed (synp) dataset was created. To create a challenging segmentation problem, real-life images were split into training and test sets having 280 and 70 images, respectively [23]. Synthetic images were used for training with different amounts for testing their effectiveness. Figure 4 shows sample images from our real-life dataset.



Figure 4. Sample images from the base dataset.

To test our network's generalization abilities, the MICHE II dataset (Mobile Iris Challenge Evaluation II) was used. Three hundred and twelve (312) images from the SamsungGalaxyS4 dataset within MICHE II that have 16:9 aspect ratio and scaled to 540×960 resolution were used alongside 148 images from the SamsungGalaxyTab2 dataset that have the same resolution as our dataset [24]. Images with their name ending with *.jpg were chosen for annotation to avoid favoritism and reduce the amount of annotation labor. These datasets will be referred to as Miche960 and Miche480, respectively. Table 1 shows the summary of the datasets used in this study.

Table 1. Dataset Attributes.

Dataset Name	Resolution	Number of Train Images	Number of Test Images
Base	640×480	280	70
syn	640×480	900	0
synp	640×480	900	0
Miche480	640×480	0	148
Miche960	540×960	0	312

2.2. Augmentation

While the real-life images of our dataset were captured indoors and have low lighting conditions, images created with UnityEyes software show outdoor properties. Therefore, to make them look more like our real-life data, some image augmentation methods were applied.

2.2.1. Brightness and Contrast

Brightness and Contrast of an image can be formulated as:

$$g(x) = \alpha f(x) + b \quad (2)$$

Where $f(x)$ represents pixel value at each channel of the image, $g(x)$ represents the output of the pixel values, α controls contrast, and b controls brightness respectively if we simplify the formula on the digital color space we get

$$g(x) = \alpha(f(x) - 128) + 128 + b \quad (3)$$

α contrast modifier and b means brightness modifier. By picking an α value between 0-1 we can reduce the contrast and values above one will increase it. Brightness is a more straightforward constant that can be added to every pixel value of the image [19].

2.2.2. Rotation

Rotation is a common image augmentation technique and new location of the rotated pixel can be represented as:

$$\begin{aligned} X_n &= \cos(\theta) * (X_{n-1} - X_0) + \sin(\theta) * (Y_{n-1} - Y_0) \\ Y_n &= \sin(\theta) * (X_{n-1} - X_0) + \cos(\theta) * (Y_{n-1} - Y_0) \end{aligned} \quad (4)$$

X_0 and Y_0 represent the anchor point of rotation X_{n-1} and Y_{n-1} represent the old location of the pixel value and X_n and Y_n become the new location of the pixel.

2.2.3. Resize

To simulate different camera to eye distance conditions resizing operations are used. For the resizing operation, the method of bicubic interpolation is used. Bicubic interpolation creates a surface for unlimited resizing by the formula:

$$f_{i(x,y)} = \sum_{\{i=0\}}^3 \sum_{\{j=0\}}^3 a_{\{i\}x\{j\}y}^i y^j \quad (5)$$

Equation (5) shows that a total of 16 coefficients (a_{ij}) need to be calculated to find the interpolated area. Four of the coefficients are calculated using horizontal derivatives, four of the coefficients are calculated using vertical derivatives, four of the coefficients are calculated using diagonal derivatives, and the remaining coefficients are calculated for corner intensity values [25].

2.2.4. Blur

Blur is a process to convolve the image with different kernel matrices to get a smoother image. Motion blur is a process to convolve the image with various kernels to give it a shaky look. It can be formulated as:

$$I_{b(x,y)} = I(x, y) * K(s, s) \quad (6)$$

In our study, kernels used for blurring was a matrix filled with ones. For vertical motion blur, a matrix filled with ones in the middle vertical row, for horizontal motion blur, a matrix filled with ones in the middle horizontal row was used [25].

2.3. Evaluation

Mean Intersection Over Union (IoU) metric was used as an evaluation metric to measure the proposed method's performance. IoU calculates the ratio of segmentation success by comparing correct segmentation results to the union of correct and incorrect segmentation results, as shown in (7).

$$IoU_c = \frac{Intersection}{Union} = \frac{PL_c \cap TL_c}{PL_c \cup TL_c} \quad (7)$$

Correct segmentation of the calculated class results are represented by true labels (TL_c); predicted segmentation results are represented by predicted labels (PL_c). Intersection of these labels represents correctly segmented pixels. The union of those labels represents correct segmentation pixels alongside miss-segmented pixels. With their division, we get the IoU value for a class. The total of each class IoU value is divided by the total number of classes to calculate the mIoU value.

3. Experiments

Several experiments were conducted under differing conditions. The real-life images for the experiment were always kept at 280 images for the training set and 70 images for the testing set. However, the size of the training set varies to examine the effect of the additional data while maintaining the real-life image factor the same. Synthetic images are tested as processed and unprocessed categories, as shown in Fig 5. Synthetic images processed beforehand were called synp, and the non-processed image set was called syn for distinction.

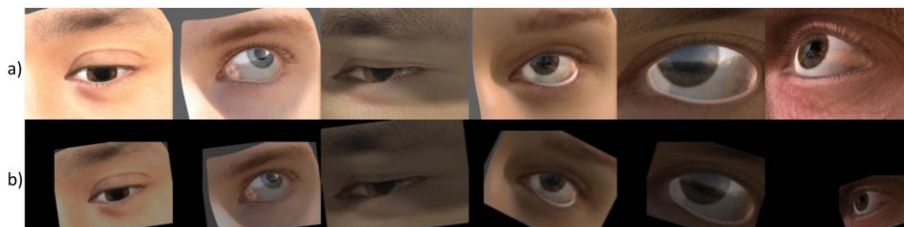


Figure 5. Synthetic data samples. a) Unprocessed. b) Processed.

Different weight multipliers to each class to increase segmentation performance. Since we have non-uniform eye regions, the multipliers are chosen in contrast to the region sizes in the images. The background class weights stayed the same, the sclera class weights multiplied by 10, the iris weights multiplied by 20, and the pupil weights multiplied by 30, empirically. All networks trained for 100,000 steps with the same settings: batch size was 10, crop size was 256×256, output stride was 16, the atrous convolution rates were 6, 12, 18, momentum Optimizer with 0.9 rate was used, the learning rate was 1×10^{-3} , and the decay rate was 1×10^{-3} for every 1000 steps. Thus, every 10,000 steps network was saved, and the best performing network was chosen.

The first experiment was conducted with no additional synthetic data to create a baseline. The network trained ten times with the mentioned settings to tackle the variance problem, and the best test result was chosen for comparison. In the second experiment, the network trained with base synthetic data and processed synthetic data alongside real-life images. The base results are shown in Table 2. Both networks performed worse than the base network, but some images that were segmented significantly better with synthetic data can be seen in Fig 6(a).

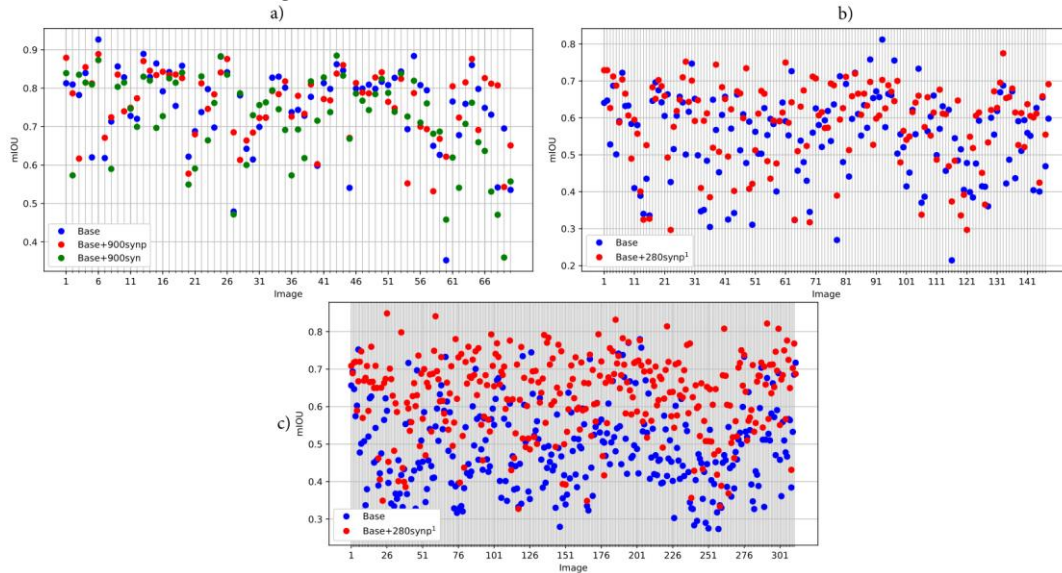


Figure 6. a) mIoU results per image of the Base, Base+900syn and Base+900synp networks on base dataset. b) mIoU results per image of the Base and Base+280synp¹ networks on Mische480 dataset. c) mIoU results per image of the Base and Base+280synp¹ network.

To explore the generalization abilities of the networks, a second set of experiments were conducted using images from MICHE II dataset. Those images were only used for testing and no training is done with them to get performances of Base and Base+280synp2 networks in different settings. One of the sets used for experimenting referred to as Mische480, which has 148 test images with a resolution of 640×480 pixel similar were added to our dataset. The second one is referred to as Mische960 and it has 540×960 pixel resolution. For both datasets, Base+280synp1 network consistently performed better. Results for per class mIoU can be seen at Fig 6(b), and Fig 6(c).

Table 2. Base, Mische480, and Mische960 dataset mIoU results of different networks.

Dataset	Network	Total	Sclera	Iris	Pupil	Background
Base	Base	73.0	63.5	74.7	56.8	97.3
Base	Base+900syn	68.9	57.2	67.2	54.4	96.6
Base	Base+900synp	71.8	64.8	73.0	51.7	97.6
Base	Base+280synp ¹	75.4	68.0	73.8	62.1	97.7
Base	Base+280synp ²	73.4	60.6	74.6	62.2	97.0
Base	Base+280synp ³	71.6	63.1	71.5	54.0	97.8
Miche480	Base	53.9	38.8	49.9	29.2	97.8
Miche480	Base+900syn	58.6	38.1	59.7	38.5	98.9
Miche480	Base+900synp	61.3	42.8	63.1	41.2	98.2
Miche480	Base+280synp ¹	58.1	44.2	58.8	30.7	99.1
Miche480	Base+280synp ²	55.6	35.2	53.9	36.8	96.3
Miche480	Base+280synp ³	56.2	37.1	46.0	44.4	97.4
Miche960	Base	48.0	29.4	31.9	32.4	98.3
Miche960	Base+900syn	58.7	42.6	61.0	32.1	98.8
Miche960	Base+900synp	60.6	44.2	61.1	38.4	96.9
Miche960	Base+280synp ¹	62.4	45.6	65.3	39.6	99.1
Miche960	Base+280synp ²	57.4	36.5	56.3	39.7	97.1
Miche960	Base+280synp ³	57.4	37.8	47.4	46.7	97.7

Therefore, we reduced the number of synthetic images on training data with a one-to-one ratio of the base training set to train more balanced networks. The synthetic images that were used in that setup were generated randomly with overlapping images. The best mIoU result and the networks trained with the same protocol can be seen in Table Error! Reference source not found..

Miche480 and Miche960 datasets were used to test the generalizing abilities of the networks. Since they were not used in any training and have uncontrolled conditions, they made a reliable testing ground for measuring the synthetic data's effect. Since the Miche480 dataset has the same resolution as the Base and synthetic dataset, test performance is expected to be higher than the Miche960 dataset, but Table **Error! Reference source not found.** shows that performances almost identical. However, this does not apply to the Base network, which performed significantly better in the Miche480 test. This was expected due to it being only trained with the Base dataset.

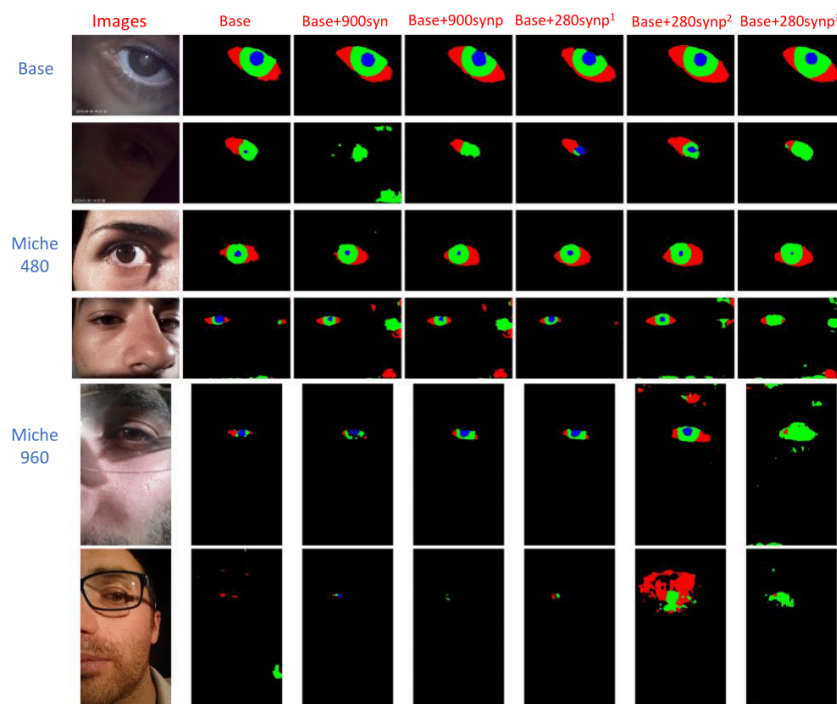
4. Discussion and Conclusion

This study explored the eye segmentation in semi-unrestricted conditions and increased the network's performance using synthetic eye data. The variety of images in this study makes eye segmentation challenging as possible reflections can be incorrectly segmented, such as the pupil. Similarly, some images do not contain a visible pupil due to lack of illumination or occlusions by the eyelid.

We showed that segmentation accuracy increases when external data with a structure similar to the original data is used. If left unsupervised, that data might reduce network performance unless they are almost identical or carefully picked. Fig 7 shows that different data used for training changes the way the images are segmented. When we consider the test results obtained in the Base dataset, our judgment on the augmentation methods and analyses on the dataset were proven to be effective since the Base+900syn network had no improvement in the single class but Base+900synp performed better than the Base network in the sclera class. Table **Error! Reference source not found.** shows that for every class, there can be a significant increase in mIoU using the processed synthetic data in addition to real-life data.

The base test has shown that when working on a limited dataset, the dataset's balance can be the deciding factor of the network's performance. Networks trained with the same amount of Base+ synp dataset performed better than Base and unbalanced networks. Some of it can be due to network variance, but consecutive training of the networks showed a consistent improvement in network performance. We obtained 2.4% increase in mIoU at the Base dataset.

Synthetic data showed a consistent and decent improvement on Miche tests. We obtained a 7.4% and 14.4% increase in mIoU using the Miche480 and Miche960 datasets, respectively. There can be further improvements in network performance with better adjustments to the data. For the reproducibility of the results, the dataset and annotations we used for this study are published at GitHub¹.



¹ <https://github.com/melihoz/eyedataset>

Figure 7. Sample images used in testing and the segmentation masks created by the networks compared.

Acknowledgements

This work was supported by the Scientific Research Projects Coordination Unit of Akdeniz University Project Number: TTU 2018-3295.

References

- [1] Meimei Yang, Ji Jiang Yang, Qinyan Zhang, Yu Niu and Jianqiang Li, "Classification of retinal image for automatic cataract detection", in *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services, Healthcom 2013*, Published by IEEE, 2013, pp. 674–679, ISBN:9781467358019, DOI: 10.1109/HealthCom.2013.6720761, Available: <https://ieeexplore.ieee.org/document/6720761>
- [2] Melih Gunay, Evgin Goceri and Taner Danisman, "Automated detection of adenoviral conjunctivitis disease from facial images using machine learning", in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Published by IEEE, 2015, pp. 1204–1209, DOI: 10.1109/ICMLA.2015.232.
- [3] John Daugman, "Biometric personal identification system based on iris analysis", Mar. 1 1994, US Patent 5,291,560, Available: <https://patents.google.com/patent/US5291560A>.
- [4] Shabab Bazrafkan, Shejin Thavalengal and Peter Corcoran, "An end to end Deep Neural Network for iris segmentation in unconstrained scenarios", *Neural Networks*, vol. 106, pp.79–95, 2018, ISSN:0893-6080, DOI: <https://doi.org/10.1016/j.neunet.2018.06.011>.
- [5] Richard P. Wildes, "Iris recognition: An emerging biometric technology", *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, 1997, Print ISSN: 0018-9219, Electronic ISSN: 1558-2256, DOI: 10.1109/5.628669, Available: <http://ieeexplore.ieee.org/document/628669/>.
- [6] James R. Matey, Oleg Naroditsky, Keith Hanna, Ray Kolczynski, Dominick J. Loiacono et al. "Iris on the move: Acquisition of images for iris recognition in less constrained environments", *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1936–1946, 2006, ISSN:00189219, DOI: 10.1109/JPROC.2006.884091.
- [7] John Daugman, "New Methods in Iris Recognition", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1167–1175, 2007, DOI: 10.1109/TSMCB.2007.903540.
- [8] Samir Shah and Arun Ross, "Iris Segmentation Using Geodesic Active Contours", *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 824–836, 2009, DOI: 10.1109/TIFS.2009.2033225.
- [9] Zhi Zhou, Eliza Yingzi Du, N. Luke Thomas and Edward J. Delp, "A New Human Identification Method: Sclera Recognition", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 571–583, 2012, DOI: 10.1109/TSMCA.2011.2170416.
- [10] S. Alkassar, W. L. Woo, S. S. Dlay and J. A. Chambers, "Robust Sclera Recognition System with Novel Sclera Segmentation and Validation Techniques", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 474–486, mar 2017, ISSN:10834427, DOI: 10.1109/TSMC.2015.2505649.
- [11] Diego R. Lucio, Rayson Larooca, Evair Severo, Alescu S. Britto and David Menotti, "Fully Convolutional Networks and Generative Adversarial Networks Applied to Sclera Segmentation", in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Published by IEEE, 2018, pp. 1–7, DOI: 10.1109/BTAS.2018.8698597.
- [12] Muhammad Arsalan, Hyung Gil Hong, Rizwan Ali Naqvi, Min Beom Lee, Min Cheol Kim et al. "Deep Learning-Based Iris Segmentation for Iris Recognition in Visible Light Environment", *Symmetry*, vol. 9, no. 11, p. 263, November 2017, ISSN:2073-8994, DOI: 10.3390/sym9110263.
- [13] Nianfeng Liu, Haiqing Li, Man Zhang, Jing Liu, Zhenan Sun and Tieniu Tan, "Accurate iris segmentation in non-cooperative environments using fully convolutional networks", in *2016 International Conference on Biometrics, ICB 2016*, Published by IEEE, August 2016, ISBN:9781509018697, DOI: 10.1109/ICB.2016.7550055.
- [14] Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, 1st ed. Cambridge, USA: MIT Press, 2016, Available: <http://www.deeplearningbook.org>.
- [15] Taner Danışman, "Segmentation of Portrait Images Using A Deep Residual Network Architecture", *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, vol. 22, pp. 569 – 580, 2020, ISSN:1302-9304, E-ISSN:2547-958X, DOI: 10.21205/deufmd.2020226523.
- [16] Jia Deng, Wei Dong, Richard Socher, Li-Jia. Li, Kai Li et al. "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Published by 2009, pp. 248– 255, DOI: 10.1109/CVPR.2009.5206848.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, Published by IEEE Computer Society, December 2016, pp. 770–778, ISBN:9781467388504, ISSN:10636919, DOI: 10.1109/CVPR.2016.90.

- [18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov *et al.* "Going deeper with convolutions", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June- 2015, Published by IEEE Computer Society, October 2015, pp. 1–9, ISBN:9781467369640, ISSN:10636919, DOI: 10.1109/CVPR.2015.7298594.
- [19] François Chollet, "Xception: Deep learning with depthwise separable convolutions", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, Published by IEEE, November 2017, pp. 1800–1807, ISBN:9781538604571, DOI: 10.1109/CVPR.2017.195.
- [20] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017, DOI: 10.1109/TPAMI.2016.2644615
- [21] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff and Hartwig Adam, "Encoder- Decoder with Atrous Separable Convolution for Semantic Image Segmentation", in *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Eds., Published by Cham: Springer International Publishing, 2018, pp. 833–851, ISBN:978-3-030-01234-2.
- [22] Farmanullah Jan, "Segmentation and localization schemes for non-ideal iris biometric systems", pp. 192– 212, April 2017, ISSN:01651684, DOI: 10.1016/j.sigpro.2016.11.007.
- [23] Erroll Wood, Tadas Baltrusaitis, Louis-Philippe Morency, Peter Robinson and Andreas Bulling, "Learning an appearance-based gaze estimator from one million synthesised images", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, Published by 2016, DOI: 10.1145/2857491.2857492.
- [24] Maria De Marsico, Michele Nappi and Hugo Proença, "Results from MICHE II – Mobile Iris CHallenge Evaluation II", *Pattern Recognition Letters*, vol. 91, pp. 3–10, 2017, Mobile Iris CHallenge Evaluation (MICHE-II), ISSN:0167-8655, DOI: 10.1016/j.patrec.2016.12.013.
- [25] Anil K. Jain, *Fundamentals of Digital Image Processing*, 1st ed. New Jersey, USA: Prentice Hall, ISBN:9780133361650, 1989, Available: <https://books.google.com.tr/books?id=GANSAAAAMAAJ>.



© 2021 by the author(s). Published by Annals of Emerging Technologies in Computing (AETiC), under the terms and conditions of the Creative Commons Attribution (CC BY) license which can be accessed at <http://creativecommons.org/licenses/by/4.0>.