REPUBLIC OF TURKEY

AKDENIZ UNIVERSITY

EYE SEGMENTATION USING DEEP NEURAL NETWORKS

Melih ÖZ

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

JANUARY 2021

ANTALYA

REPUBLIC OF TURKEY

AKDENIZ UNIVERSITY

EYE SEGMENTATION USING DEEP NEURAL NETWORKS

Melih ÖZ

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

JANUARY 2021

ANTALYA

REPUBLIC OF TURKEY

AKDENIZ UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

EYE SEGMENTATION USING DEEP NEURAL NETWORKS

Melih ÖZ

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

This thesis unanimously accepted by the jury on 15/01/2021.

Assist. Prof. Dr. Taner DANIŞMAN

Prof. Dr. Melih GÜNAY

Assist. Prof. Dr. Shahram TAHERI

ÖZET

DERİN SİNİR AĞLARI KULLANILARAK GÖZ BÖLÜTLEMESİ

Melih ÖZ

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Taner DANIŞMAN

Ocak 2021; 29 sayfa

Gözler insanlık tarihinin başından beri insanların odaklandığı noktalardan birisi olmuştur. Görsel girdilerden sorumlu olmasının yanı sıra, kişinin sağlığı hakkında bilgi taşıması, insanların duygusal durumunu anlama yönünde yardımcı olması gibi özelliklerinden dolayı pek çok araştırmada kullanılmaktadır. Bu çalışmada göz fotoğrafları sklera, iris, göz ve arkaplan olarak bölütlenmesi amaçlanmıştır. Bu amaca hizmet etmesi için derin sinir ağları kullanılmıştır.

Bu çalışma kapsamında, derin sinir ağlarını eğitebilmek için bir veri seti oluşturulmuştur ve bu veri seti araştırmacıların faydalanması için yayınlanmıştır. Veri setinde bulunan fotoğraflar farklı ışık, renk, mesefe ve yansıma özellikleri göstermektedir.

Bu çalışmada derin sinir ağları göz bölütleme konusu üzerinden kıyaslanmıştır. İlk olarak temel derin sinir ağları parametrelerini belirlemek için deneyler yapılmış ve elde edilen parametreler doğrultusunda derin sinir ağlarının performansları kıyaslanmıştır. Bu sinir ağları VGGNet, UNet, PSPNet, DeeplabV3+ ve HRNetV2 yapılarını içermektedir. Bu ağların genelleme yeteneği eğitim ve test setindeki başarıları ölçülmüştür. Veri setinin bir kısmını oluşturan sentetik datanın ve data çoğaltma tekniklerinin etkisi incelenmiştir, birbirlerine göre avantajları ve dezavantajları sıralanmıştır.

ANAHTAR KELİMELER: Derin Sinir Ağları, Göz Bölütlemesi

JÜRİ: Dr. Öğr. Üyesi Taner DANIŞMAN
Prof. Dr. Melih GÜNAY
Dr. Öğr. Üyesi. Shahram TAHERİ

ABSTRACT

EYE SEGMENTATION USING DEEP NEURAL NETWORKS

Melih ÖZ

MSc Thesis in Computer Engineering

Supervisor: Assist. Prof. Dr. Taner DANIŞMAN
January 2021; 29 pages

Eyes have been one of the focus points of people since the beginning of human history. In addition to being responsible for visual inputs, it is used in many research due to its characteristics such as carrying information about the person's health and helping people to understand their emotional state. In this study, it was aimed to segment the eye images into the sclera, iris, eye, and background. Deep neural networks have been used to aim this goal.

Within the scope of this study, a dataset created to train deep neural networks, and this dataset is published for the use of researchers. The photos in the dataset show different light, color, distance, and reflection characteristics. The dataset contains images from the MicheII dataset, synthetic images generated from the UnityEyes program, and the dataset we created from the videos taken for the BAP project TTU-2018-3295.

In this study, the performance of the different deep neural networks was compared on the subject of eye segmentation. Firstly, experiments were made to determine basic deep learning parameters. After that, performances of various deep neural networks were compared in line with the obtained data. These networks include VGGNet, UNet, PSPNet, DeeplabV3+, and HRNetV2 structure. Their success in the generalization performance was measured in the training and test set of these networks. Synthetic data that make up a part of the dataset and data augmentation techniques were examined. Advantages and disadvantages compared to each other.

KEYWORDS: Neural Networks, Eye Segmentation,

COMMITTEE: Assist. Prof. Dr. Taner DANIŞMAN
Prof. Dr. Melih GÜNAY
Assist. Prof. Dr. Shahram TAHERİ

ACKNOWLEDGEMENTS

First and foremost, I would like to express my very great appreciation to Assist. Prof. Dr. Taner DANIŞMAN for unwavering guidance and patience that cannot be underestimated during this study.

I wish to offer my special thanks to Prof. Dr. Melih GÜNAY who helped me with his unparalleled wisdom and experience throughout my graduate education.

I must extend my sincere thanks to the remaining dear professors of Akdeniz University Computer Science Education for their valuable contribution to my academic knowledge.

I am also grateful to my dear colleagues and friends Erdinç TÜRK, Taha Yigit ALKAN, and Caner SONGÜL for their support during this study.

Finally, I'm deeply indebted to my wonderful family for supporting me throughout my education life.

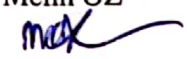
LIST OF CONTENTS

TEXT OF OATH

I declare that this study "EYE SEGMENTATION USING DEEP NEURAL NETWORKS", which I present as master thesis, is in accordance with the academic rules and ethical conduct. I also declare that i cited and referenced all material and results that are not original to this work.

15/01/2021

Melih ÖZ



ABBREVIATIONS

DNN : Deep Neural Network

mIoU : Mean Intersection Over Union

SGD : Stochastic Gradient Descent

PSPNet : Pyramid Scene Parsing Network

HRNet : High-Resolution Network

MicheII : Mobile Iris Challenge Evaluation II

GB : Gigabytes

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The Eyes are part of our body that are responsible for our visual inputs. Visual inputs are the first step to visual understanding that is a fundamental part of our survival and being. They are external organs that can be seen without any specialized tools that make them one of the first things you notice about a person. They are also unique to each individual whose color can be different from person to person. The main eye colors are brown, black, blue, and green. The colored part of the eye is called the iris that has unique patterns for each person. Therefore, it can be used as a personal identity. The white, veiny part of the eye is called the sclera. The sclera is responsible for eye structure and protecting the eye from any harm. The small inner circle inside of the iris is called the pupil. The pupil works like a regulator that adjusts how much light can enter the eye. With the information they contain such as iris patterns and the unique vein structure of the sclera, they can be used to identify a human (Daugman 1994). Also, they 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 the cataract disease. They can lose their shape due to genetics or accidents (Günay et al. 2015). One of the ways to obtain eye information is

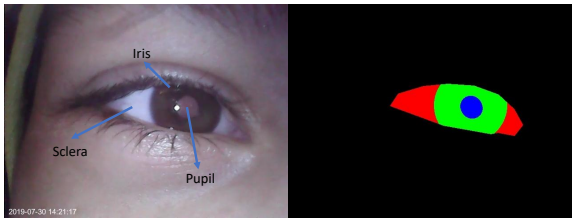


Figure 1.1. Eye regions

called eye segmentation. Segmentation is the task in which input images are annotated to parts referring to which objects are in that image. Eye segmentation is the task used to get boundaries of the parts we want to use. For example, sometimes only iris is required for the research. Therefore, the eye image is annotated to have two parts iris and background (Bazrafkan et al. 2018).

The most common use case of eye segmentation is iris recognition. Iris recognition systems use special hardware to capture eye image that is highly restricted and uses image processing methods to locate iris location. Main methods used for such task is called

Hough Circle Transformation and Daughman's Method. The main difference between those algorithms is Daughman's Method does not expect the iris to be perfectly circular. Therefore, it is more commonly used in iris recognition task (Daugman 1994).

The problem with image processing methods is the expected input of repeated patterns, that are near impossible to get when we consider 3d model of the eye. Therefore, there is a need for a model that adapts different patterns in inputs. This is where neural networks excel. Segmentation using neural networks is a common practice since it's being used in various problems such as segmenting the roads for autonomous driving cars, detecting faulty cells for disease detection, and even art style transformation (Chollet 2017). Neural networks require input, output pairs to understand the problem and create the required filters to address a solution for the problem. For different images to be segmented, an adequate amount of manually segmented input and output pairs need to be given to train the network. Thus, this creates a challenge compared to image processing methods which only require certain calculations to do the task (Goodfellow et al. 2016).

In order to satisfy training and testing data required for deep neural networks, several datasets are created. The first dataset contains images taken for BAP project TTU-2018-3295. The second dataset contains 900 synthetic images generated from the UnityEyes interface (Wood et al 2016). Alongside these datasets, 415 images from the MicheII challenge are also annotated (Marisco et al. 2017). Datasets contain images are in different angles, distances, and lighting conditions. Thus, that makes them a good candidate for comprehensive image segmentation. In addition to sparse images, the effect of the data augmentation methods is also tested. The model used for training also affects the results. In this thesis, various deep neural network models are evaluated using the datasets created. VGGNet (Simonyan and Zisserman 2015), UNet (Ronneberger et al. 2015), PSPNet (Zhao et al. 2017), HRNetV2 (Sun et al. 2019), and DeeplabV3+ (Chen et al. 2018) are used for testing how different structures affect the segmentation.

This thesis explores eye segmentation methods in the literature, DNN's in general, and eye segmentation in DNN's. Then the methods are used in this thesis. The following section is "Experiments and Discussion" where the results are shared, also comments on the results are made. The last part of this thesis is "Conclusion" where the thesis interpretations are made.

2. LITERATURE REVIEW

2.1. Eye Segmentation Methods

In this section eye segmentation and its sub-forms, iris and sclera segmentation methods explored.

2.1.1. Early Iris Segmentation Methods

The human eye has been a region of interest. However, the main interest has always been in iris recognition since iris being offered as an individual recognition method in 1923 by Burch (Bazrafkan et al. 2018). With the possibility of the eyes being used for identification, algorithm development for the task continued until John Daugman (1994) created an algorithm that can be used in commercial systems Fig 2.2 shows the workflow of the system. Daugman got a patent for his algorithm and that patent covers the method of iris recognition expired in 2005. Thus, this event accelerated the development of different iris segmentation methods (Bazrafkan et al. 2018).

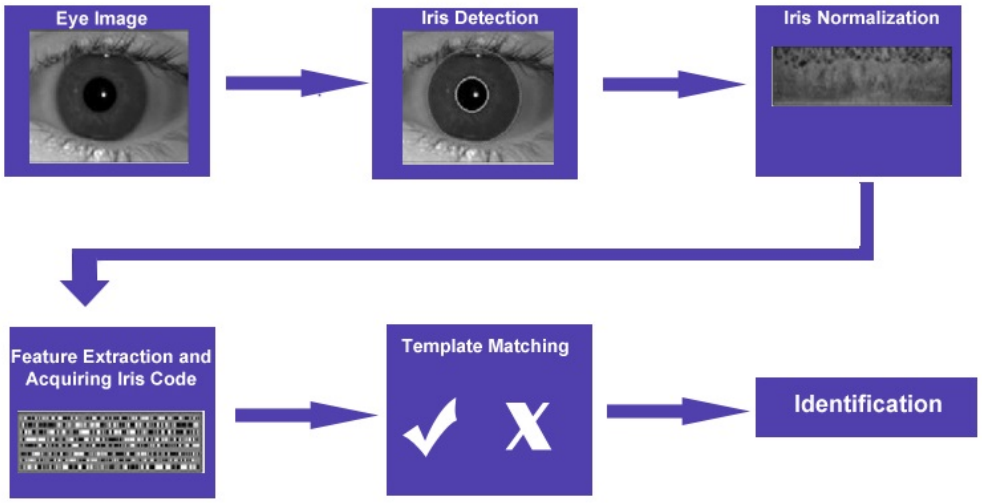


Figure 2.2. Iris recognition schematic

Early iris recognition systems had satisfactory rates of false accept rate about one in a million and false reject rate below one percent. Their shortcomings were user-contributed

constraints: the user needed to stay without any movement in a position and waited several seconds for the iris image to be captured. Constraints prevented the systems from being used with ease by disabled people and dropped the experience of users (Matey et al. 2006). Matey and friends developed a more mobile system that uses multiple cameras can capture iris when the subject moves along a certain path.

Iris recognition system developed by Daugman used Integro-Differential Operator (IDO) for iris segmentation, which is formulated as

$$max_{(r,x_0,y_0)} = \left| G_\sigma(r) \frac{\partial x}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2} ds \right| \tag{2.1}$$

Gaussian filter represented with $G_\sigma(r)$, σ is a ratio that is used to remove grainy look, reduce the effect of light reflections and obtain a smooth region in radius r. $I(x,y)$ represents pixel intensity values at the image domain. Search function moves circularly then defines a pixel wide circular anchor point, then it tries to find the circular region with the maximum amount of intensity change compared to first point x_0, y_0 for the pixels within a range of r in the image if it's not restricted. Maximum intensity change points are circular edges in this case, since these images are constrained eye images, the function returns either inner or outer iris boundary regarding on search radius defined before.

Other iris recognition systems use some circular edge detection as well. The most popular one is circular Hough transformation and one of the most known applications of the method was proposed by Wildes (1997). Wildes also first used the gaussian filter to smooth the eye image. In order to get the outer iris boundary, he first applied an almost vertical edge detector and created the edge map of the eye image. Then, he used circular Hough transformation to get the outer iris boundary. Circular Hough transformation formulated as

$$H_c(x_c, y_c, r_c) = \sum_{e=1}^{n} (x_e, y_e, x_c, y_c, r_c), \tag{2.2}$$

total number of edge points presented by n and this formula represents number of edge points (x_e, y_e) on circle defined as (x_c, y_c, r_c) and the potential circle with max edge points

calculated with votes:

$$H_c(x_e, y_e, x_c, y_c, r_c) = \begin{cases} 1, if(x_e, y_e, x_c, y_c, r_c) = 0 \\ 0, else \end{cases} \tag{2.3}$$

This formula represents that if a distance between (x_e, y_e) on the circle defined as (x_c, y_c, r_c) is 0 that counts as a vote, and the potential circle with most votes returned as the outer iris boundary. After finding the circle (x_c, y_c, r_c) the inner iris boundary found by applying an edge detector without any orientation and applying circular Hough transformation within range of r_c (Wildes 1997). Eyelid boundaries are also calculated as two different parabolic arcs. In order to find those arcs, horizontal edge detection is applied to the eye image. Fig 2.3 shows results of gradient-based edge detection with different directions applied to the eye image.

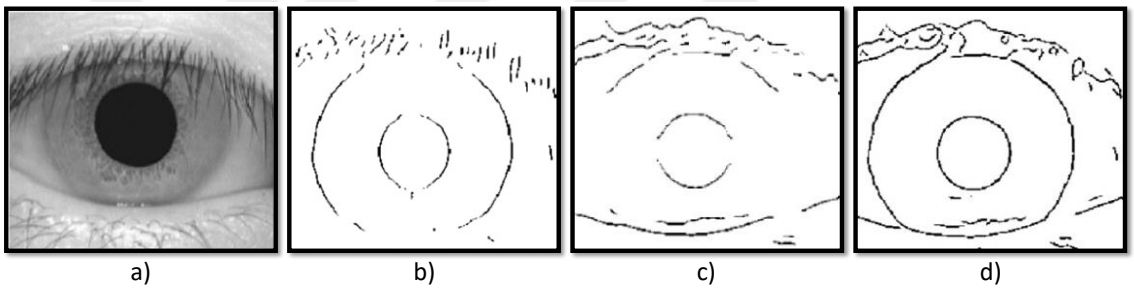


Figure 2.3. a) The eye image b) result of vertical, c) horizontal, and d) without any orientation edge detector applied to the image

The problem with early iris segmentation methods is their too strict conditions, the requirement of near-infrared images and iris not being circular, repeated patterns that near impossible to get when we consider 3d model of the eye. However, in restricted conditions, early methods had a satisfactory rate of accuracy (Matey et al. 2006). The problem of the eye not being circular and concentric is solved with future works (Daugman 2007; Shah et al. 2009).

2.1.2. Image Processing Based Sclera Segmentation Methods

When the sclera is extracted from an eye image, it's called sclera segmentation. Sclera information also can be used as a biometric (Zhou et al. 2011). Image processing based

sclera segmentation methods are partially 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 (Alkassar et al. 2017). These methods also work in restricted conditions.

2.1.3. Deep Neural Networks

For the segmentation task, there is a need for a model that adapts different patterns in inputs. This is where Deep Neural Networks (DNN) shines. Although the usage of DNN's goes back to several decades, they became standard in recent years in the image segmentation task.

Two events started the rise of Deep DNNs, firstly the release of the ImageNet database, and secondly the introduction of the AlexNet (Deng et al. 2009). The idea was to keep the segmentation data towards multiple layers of fully connected layers to contain as much information as possible. VGGNet structure also used this with multiple convolutions each layer to keep multiple levels of data. VGGNet was created for the image classification task, it was in the top five in ImageNet results when it was published. While going in-depth with filters, some information was lost. This problem rose from mapping two layers directly. To tackle this problem, the idea of learning the difference between layers instead of direct mapping created. With that idea, losing information problem while going deeper into layers improved (He et al. 2016). 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 (Badrinarayanan et al. 2017; Ronneberger et al. 2015). While encoder structures work like regular network without up-sampling operation. Decoder structures take low-level features and up-samples them by convolving with changeable decoder layers. UNet decoder structure concatenates corresponding level encoder outputs with up-sampling outputs to segmentation data. Thus, that enables recovery of the higher-level filter outputs. PSPNet introduced a decoder structure that keeps rich segmentation information thanks to different rates of pooling operations applied to the output of the encoder. Still more in-depth the network, more computation power is required. To get more information in a layer, parallel convolutional operations with dimension reduction were

proposed (Szegedy et al. 2015). Later, that idea improved by separable convolutions on every channel with dimension reduction. In the end, that way, multiple contexts in a layer are compressed without increasing computational complexity (Howard et al. 2017). DeeplabV3+ Structure combines parallel convolutional operations with dilated convolutions with different rates that encode different scale information with lesser computational power. A very recent HRNetV2 structure proposed the idea of maintaining high-resolution representations through the whole process. Unlike other DNNs HRNetV2 approach connects lower resolution convolution streams in parallel rather than in series. Thus, higher resolution information is kept during the process that outputs precise results.

2.1.4. Deep Learning Based Eye Segmentation Studies

With the rise of DNNs, multiple research on eye segmentation using DNNs are conducted. In this section starting from iris segmentation, segmentation models using DNNs are mentioned. Liu et al. (2016) proposed a network that includes three parallel convolution layers with different rates that concatenate at the last layer. Arsalan et al. (2017) proposed a two-stage model that uses image processing methods to get rough iris boundaries than for each pixel used VGGNet based DNN. Bazrafkan et al. (2017) used UNet like architecture to segment the iris region. These researches show that DNN architectures are applicable to solve the iris segmentation problem. Lucio et al. (2018) used Generative Adversarial Network (GAN), FCN, and SegNet in sclera segmentation. Proposed system had two layers one layer detected ROI used YOLO object detection algorithm and it's output fed to the neural network. Naqvi and Loh (2019) used ResNet based encoder-decoder network to segment sclera region. Rot et al. (2018) used SegNet based network to segment the eye into six distinct regions. Garbin et al. (2019) used Segnet based neural network to introduce their controlled eye dataset. Luo et al. (2020) segmented eye into the three regions iris, sclera, and background using SegNet based encoder-decoder network with GAN. SegNet network output given to pre-trained GAN encoder and discriminator, ground truth given to encoder, taking advantage of both structures.

3. MATERIAL AND METHOD

3.1. Optimizers

Gradient Descent based methods are used as optimizer in this study. Gradient descent is a method to minimize a function $J(\theta)$ calculated by a model's parameters by updating relative θ. The learning rate η determines the step size to reach the minimum. The idea of calculating loss function using the whole dataset is computationally too expensive for the deep learning algorithms. Thus, to solve this problem using a random sample from a set of data used to calculate the minimum is created. This method is called Stochastic Gradient Descent (SGD) used in this study to train the DNNs (Chollet 2017).

3.1.1. SGD With Momentum

SGD with Momentum usually referred to as Momentum optimizer, in addition to SGD uses momentum parameter to escape local minima by updating gradients with respect to previous gradients (Chollet 2017).

3.1.2. Adam

Name of the Adam derives from adaptive momentum estimation. Adam combines the properties of two optimization methods: the ability of AdaGrad to deal with sparse gradients, and the ability of RMSProp to deal with non-stationary objectives. Adam is using two-moment parameters with bias correction to update weight parameters (Kingma and Ba 2015).

3.2. Activation Functions

The activation functions are essential for DNNs, they are used to determine the output of the hidden layers of a network. They are used to make weights of the layers to be updated in a non-linear manner. Therefore, they are used to solve non-linear problems (Chollet 2017).

3.2.1. ReLU

Rectified Linear Unit (ReLU) function is faster learning activation function with good generalization properties (Goodfellow 2016). The ReLU activation function is a threshold operation to each input given that values less than zero are set to zero hence the ReLU is given by

$$f(x) = max(0, x) \begin{cases} X_i, \ if \ x_i \ \leq \ 0 \\ 0, \ else \end{cases} \tag{3.4}$$

This function removes the inputs values that are less than zero. Thus, forcing them to zero and solving the vanishing gradient problem which occurs in certain types of activation functions.

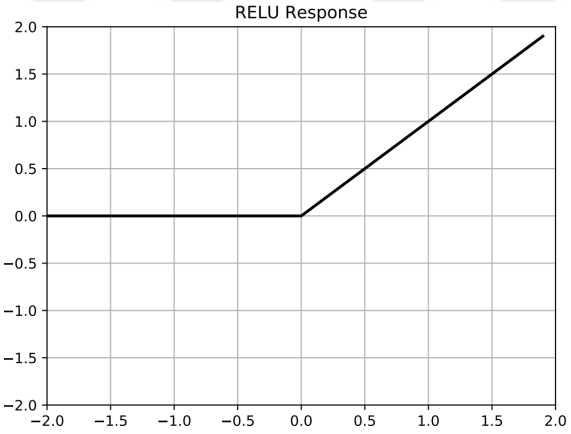


Figure 3.4. Relu response

3.2.2. Softmax Activation Function

The Softmax activation function is used to calculate the probability distribution for given inputs and number of classes. Calculated class probabilities for each class is between 0 and 1. The total of the probabilities for classes is equal to 1.

$$S(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{3.5}$$

The softmax function in image segmentation models used to calculate class of each pixel values in the input image and mostly used on the last layer of the network or blended in with loss function (Goodfellow 2016).

3.3. Models

In this section DNN models used for eye segmentation are briefly explained.

3.3.1. UNet

U-net is a decoder-encoder network. While encoder parts use 3×3 convolutions to extract features, decoder part concatenates deconvolution operation output with corresponding or encoder output. This decoder structure helps to preserve pattern information (Ronneberger et al. 2015). It doesn't have any fully connected layers which lower computational cost and can be trained with a small dataset.

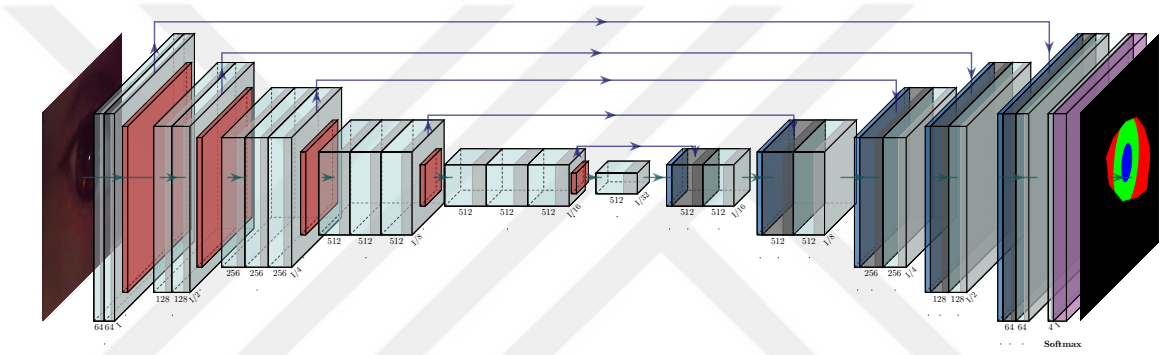


Figure 3.5. VGGNet encoder with UNet decoder

3.3.2. PSPNet

PSPNet takes features extracted from encoder structures. Same extracted output processed with different pooling rates to get rich contextual information. Then, each one corresponding to a pyramid level and processed by a 1x1 convolutional layer to reduce their dimensions. That enables the network to make use of information held on different sub-regions. The output of these layers upsampled and concatenated to acquire the final prediction (Zhao et al. 2016).

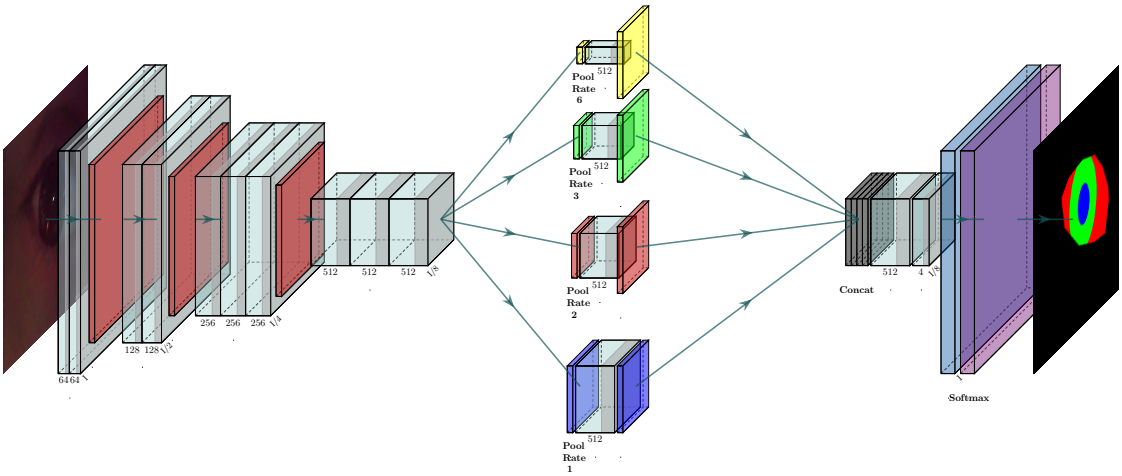


Figure 3.6. VGGNet encoder with PSPNet decoder

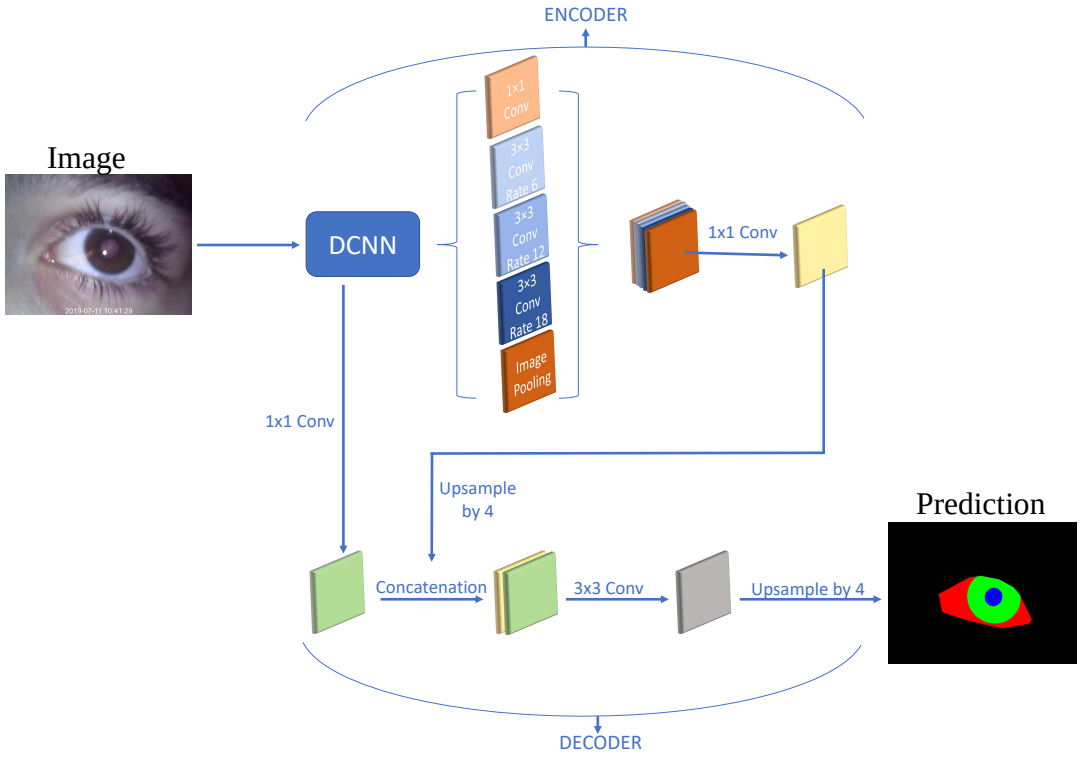


Figure 3.7. Deeplabv3+ network structure

3.3.3. HRNetV2

The unique feature of the HRNet is unlike encoder-decoder networks or fully connected neural networks keeps the high-resolution feature maps until the last layer while

extracting lower resolution feature maps alongside. Traditional convolution blocks use pooling operation after several convolution operations, HRNet uses pooling after each convolutional blocks as well, however, the output before pooling operation also joins another convolutional block. Modified version of HRNet used for segmentation is named HRNetV2 (Sun et al. 2019).

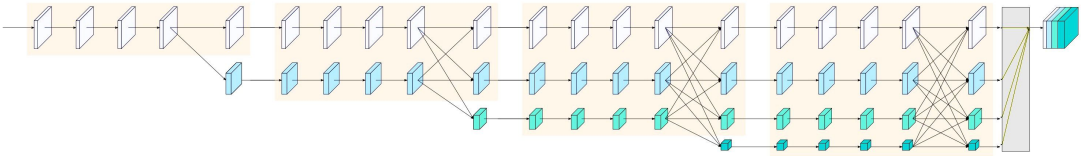


Figure 3.8. HRNet network structure (Sun et al. 2019)

3.3.4. DeeplabV3+

In this study, the segmentation task is done with Google's Deeplabv3+ neural network. Deeplabv3+, in addition to the previous work, adds depth-wise separable convolution to get better results in benchmark datasets. Deeplabv3+ uses depthwise separable convolution with different rates of atrous convolutions to get the context (Chen et al. 2018). This results in increased performance and speed compared to traditional convolution operation.

$$I_{out}[i] = \sum_{k} I_{in}[i + r \cdot k] f[k] \tag{3.6}$$

When we chose 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 3x3 convolution to polish results and upsample to get segmentation results (Chen et al. 2018).

3.4. Dataset

Our original dataset consists of 308 images that vary in distance, angle, and lightning. The images taken from low light videos, therefore, have lower quality. The images have a resolution of 640×480, have RGB channels, and are referred to as the Base dataset. The second dataset, 900 synthetic eye images are generated from the Unityeyes software used (Wood et al. 2016). The synthetic data created from the Unityeyes already have sclera and iris annotations; therefore, only the iris region is annotated by us. The third and last datasets were created from MICHEII (Mobile Iris Challenge Evaluation II) dataset. 143 images from the SamsungGalaxyTab2 dataset that have the same resolution as the Base and Syn datasets used. 312 images from the SamsungGalaxyS4 dataset that have 16:9 aspect ratio and scaled to 540×960 resolution are used (De Marisco et al. 2017). Images with their name ending with *1.jpg chosen for annotation to avoid favoritism and reduce the annotation labour these datasets referred to as the Miche480 and Miche640 respectively. All images are manually annotated with the annotation program created. To create a challenging segmentation problem, datasets are split into training and test set with around 4 to 1 ratio. Figure 3.9 show samples images from our real-life dataset. Table 3.1 shows the summary of the datasets used in this study. When all of the train and test datasets are combined it's referred to as All dataset.

Table 3.1. Dataset attributes

Dataset Name	Resolution	Number of Train Images	Number of Test Images
Base	640×480	248	60
Syn	640×480	715	181
Miche480	640×480	116	27
Miche960	540×960	250	62

3.5. Augmentation

In the scope of this thesis, the effect of augmentation methods is tested. In this section, these operations are explained.

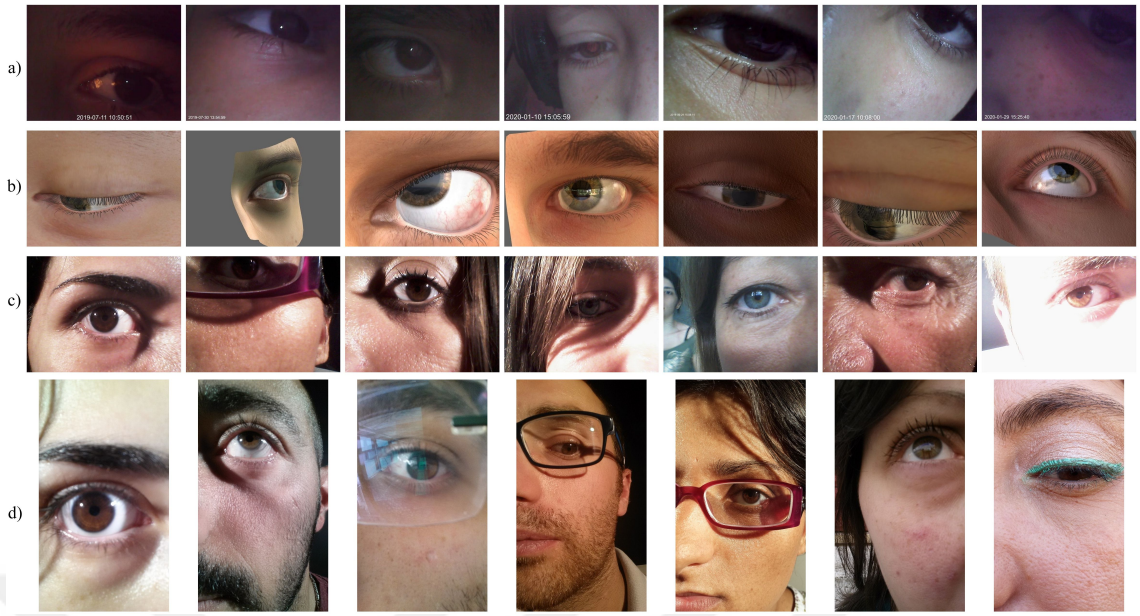


Figure 3.9. Sample images from the a) Base, b) Syn, c) Miche480, and d) Miche640 datasets

3.5.1. Rotation

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

$$\begin{aligned} x_n &= cos(\theta) * (X_{n-1} - X_0) + sin(\theta) * (Y_{n-1} - Y_0) \\ Y_n &= sin(\theta) * (X_{n-1} - X_0) + cos(\theta) * (Y_{n-1} - Y_0) \end{aligned} \tag{3.7}$$

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

3.5.2. Resize

In order to simulate different distance conditions resizing operation used. For resizing operation, method of bicubic interpolation used. Bicubic interpolation creates a surface for unlimited resizing by the formula.

$$f_i(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{3.8}$$

The equation (3.8) shows that 16 a_ij coefficients need to be calculated to find the interpolated area. Four of the coefficients are calculated with the horizontal derivatives, four of the coefficients are calculated with the vertical derivatives, four of the coefficients are calculated with the diagonal derivatives, and four of the coefficients are corner intensity values (Jain 1989).

3.5.3. Blur

Blur is a process to convolve the image with different kernel matrices in order 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 (3.9).

$$I_b(x, y) = I(x, y) * K(s, s) \tag{3.9}$$

In our study, kernels we used for blurring is a matrix filled with ones, for vertical motion blur a matrix filled with ones on the middle vertical row, and for horizontal motion blur a matrix filled with ones on the middle horizontal row (Jain 1989).

3.6. Evaluation

In this study mean Intersection Over Union(IoU) metric used as evaluation metric (3.10). IoU calculates the ratio of segmentation success by comparing correct segmentation results to correct segmentation results and incorrect segmentation results. Then the total of each class IoU averaged by the total class count to calculate mIoU (Danışman 2020).

$$IOU = \frac{TP}{TP + FP + FN} \tag{3.10}$$

Correct segmentation of the calculated class results presented by true positives TP and incorrect segmentations represented by false positives FP segmentations of the class from regions belong to other classes and false negatives FN represent the missed points from the classified class.

4. RESULTS AND DISCUSSION

In the scope of this thesis network performances compared using different datasets. Google Colab notebooks used for experiments and models trained using Keras-Tensorflow structure (Chollet and others 2015). While there can be differences between the notebooks GPU used for experiments mainly Nvidia Tesla K20 with 12GB memory used. In order to find optimal parameters for the networks, several experiments are conducted.

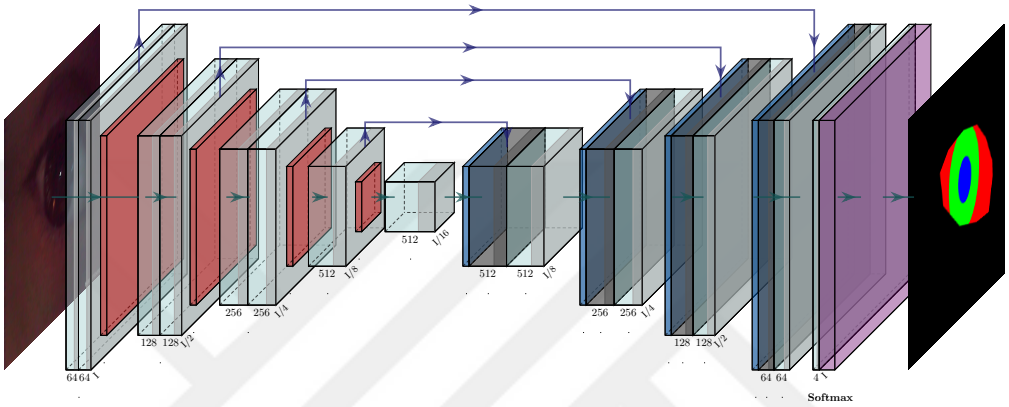


Figure 4.10. Initial network used for the experiments

4.1. Shuffling

Dataset shuffling means that each epoch batches created with a different order. To find out how much it effects the results experiment conducted with the Base dataset. Table 4.2 shows that shuffling makes a very significant change in generalization abilities. Therefore, the next experiments are always conducted with dataset shuffling.

Table 4.2. Effect of the shuffling

Dataset	Shuffle	Optimizer	Learning Rate	Train Loss	Train mIoU	Validation Loss	Validation mIoU
Base	False	Adam	1e-4	0.0065	0.9660	0.3034	0.6155
Base	True	Adam	1e-4	0.0072	0.9645	0.1528	0.7334

4.2. Optimizers

Using the All dataset Momentum and Adam optimizers compared with different learning rates. For this experiment ten sized batches are used, Table 4.3 shows results obtained with different settings. Adam's suggested learning rate of 0.0001 worked better compared to the momentum optimizer with different rates. In some cases, Adam performs better on the training set and Momentum is better at generalizing (Zhou et al. 2020). However, Momentum optimizer requires more tweaks on learning and momentum variables. Fig 4.11 shows clear difference between the optimizers. While Adam sometimes makes big leaps to escape local minima, Momentum optimizer shows smaller updates even with higher learning rate. Since it doesn't create any unfairness, the rest of the experiments conducted with Adam optimizer with the rate of 0.0001, the cross-entropy loss is used as a loss function, epoch size is 150 unless it's stated differently, and mIoU used as the evaluation metric.

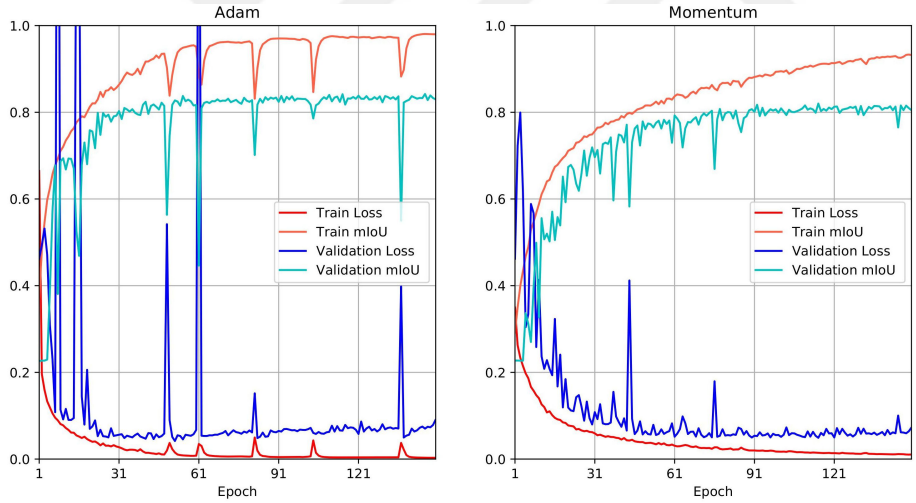


Figure 4.11. Adam optimizer with learning rate of 0.0001 compared to Momentum optimizer with learning rate of 0.01

4.3. Batch Size

One of the important factors that affect network performance is batch size. Parameter itself doesn't have an optimal number, it's performance varies with dataset size and learning rate. One good rule of thumb is not using a batch size of 1 because that limits

Table 4.3. Initial network results

Dataset	Optimizer	Learning Rate	Train Loss	Train mIoU	Validation Loss	Validation mIoU
All	Adam	1e-3	0.0035	0.9756	0.0953	0.8171
All	Adam	1e-4	0.0027	0.9817	0.0679	0.8381
All	Adam	1e-5	0.0065	0.9598	0.066	0.8048
All	Momentum	1e-2	0.0062	0.9600	0.0644	0.8201
All	Momentum	1e-3	0.0104	0.9334	0.0608	0.8161
All	Momentum	1e-4	0.0215	0.9100	0.0810	0.706

networks generalization abilities. Due to memory limitations, DeeplabV3+ and HRNetV2 can work with batch sizes up to 5. Keeping that in mind, several batch sizes are tested with the All dataset. Table 4.4 shows that the batch size of 1 makes the network more likely to overfit and should be avoided. A batch size of five seems to be a reasonable choice for performance-wise, also it adds consistency to the network comparisons.

Table 4.4. Batch size results

Dataset	Batch Size	Train Loss	Train mIoU	Validation Loss	Validation mIoU
All	1	0.0074	0.9628	0.2	0.7535
All	2	0.0023	0.9841	0.082	0.8371
All	5	0.0036	0.9745	0.0535	0.8420
All	10	0.0027	0.9817	0.0679	0.8381

The second part of the batch size experiment is while testing the batch size effect on each part of the dataset, also creates the standard results for comparison of the next experiments. Performance of each network shown at Table 4.5. While the batch size of 2 shows a slight improvement in sub-datasets only the Miche480 results can be considered as an improvement since other differences can be due to the weight initialization.

Table 4.5. Batch size results subsets

Dataset	Batch Size	Train Loss	Train mIoU	Validation Loss	Validation mIoU
Base	2	0.0072	0.9645	0.1528	0.7334
Base	5	0.0068	0.9645	0.1423	0.7266
Miche480	2	0.0024	0.964	0.0318	0.8206
Miche480	5	0.0024	0.9651	0.0342	0.7745
Miche640	2	0.0011	0.9759	0.0471	0.7993
Miche640	5	0.0016	0.9774	0.0365	0.7765
Syn	2	0.0016	0.9685	0.0365	0.9311
Syn	5	0.0037	0.9848	0.0386	0.9285

4.4. Augmentation

Augmentation is a good way to prevent overfitting (Chollet 2017). In order to see the effect of the augmentation operations the network trained with the following settings, images are rotated in both directions up to 20 degrees, moved in any direction up to 5%, sheared 5%, zoomed in or out up to 5%, and horizontally flipped. This operation is called Augmin in the Table 4.6, Augmid and Augmax also shares same properties but multiplied by two and four respectively.

Table 4.6. Augmentation results

Dataset	Augmentation	Validation Loss	Validation mIoU
All	Augmin	0.0321	0.8422
All	Augmid	0.031	0.83547
All	Augmax	0.0334	0.8355

4.5. Different Network Structures

Within the scope of this study full VGGNet encoder with UNet decoder as shown in Fig 3.5, partial VGGNet encoder with PSPNet decoder as shown in Fig 3.6, DeeplabV3+ ,and

HRNetV2 structures tested with the min augmentation settings. Results at Fig 4.7 suggest that all deeper network performed similarly expect VGGPSPNet which performed similar to the initial network which is expected when it's depth regarded.

Table 4.7. Dataset Results of the networks trained with All dataset per-class

Dataset	Network Name	Train Loss	Train mIoU	Validation Loss	Validation mIoU
All	VGGPSPNet	0.0243	0.8778	0.0334	0.839
All	VGGUNet	0.012	0.9355	0.0323	0.8521
All	DeeplabV3+	0.0077	0.9437	0.0359	0.8682
All	HRNetV2	0.7488	0.9133	0.7524	0.8689

In the second step of the network experiment, these network's sub-dataset performance and each class performance tested. Table 4.8 shows those results. When compared to the results of Table 4.5 and Table 4.4, while PSPNet shows similar results with the initial network other networks shown improvement in the All and sub-datasets. Since these networks are also deeper than the initial network these might be the effect of network structure. Syn dataset results seem similar that suggests, a dataset produces more consistency with the sample size. However, other datasets especially the Base dataset shows significant improvements when trained with the other datasets.

In order to expand the generalization abilities of the networks, they are trained with cross subsets that contain all images except one set. For example, the CrossBase train set contains all other subsets, and the test set contains Base dataset images. Results are given in Table 4.9. These results are shown that in the case of eye segmentation, DeeplabV3+ has the best ability to learn the training data and HRNetV2 has the best ability of generalization. Table 4.10 shows each network's test set performance on each the class. Even in the cross setting the background class shown high mIoU results, which means the networks can extract the eye location with ease.

Table 4.8. Test set results of the networks trained with All dataset per-class

Dataset	Network Name	Total mIoU	Background mIoU	Sclera mIoU	Iris mIoU	Pupil mIoU
All	VGGPSPNet	0.839	0.9914	0.8	0.8292	0.7347
All	VGGUNet	0.8521	0.991	0.8157	0.8455	0.7561
All	DeeplabV3+	0.8682	0.9929	0.8353	0.8597	0.7844
All	HRNetV2	0.8689	0.9929	0.8399	0.8617	0.7803
Base	VGGPSPNet	0.7182	0.9786	0.6757	0.7243	0.494
Base	VGGUNet	0.7510	0.9809	0.716	0.7526	0.5547
Base	DeeplabV3+	0.7752	0.9827	0.7275	0.7731	0.6177
Base	HRNetV2	0.7753	0.9815	0.7288	0.7699	0.6213
Miche480	VGGPSPNet	0.7757	0.992	0.6756	0.81	0.6254
Miche480	VGGUNet	0.7822	0.9929	0.7	0.8339	0.6019
Miche480	DeeplabV3+	0.8015	0.9934	0.7151	0.8423	0.655
Miche480	HRNetV2	0.8244	0.9951	0.7435	0.8748	0.6842
Miche640	VGGPSPNet	0.7428	0.9948	0.6491	0.8004	0.5175
Miche640	VGGUNet	0.7753	0.9952	0.69	0.8372	0.5737
Miche640	DeeplabV3+	0.7967	0.996	0.7206	0.8565	0.6083
Miche640	HRNetV2	0.783	0.9959	0.7231	0.8458	0.5561
Syn	VGGPSPNet	0.9212	0.9945	0.9110	0.8765	0.9025
Syn	VGGUNet	0.9222	0.9927	0.9088	0.8805	0.9071
Syn	DeeplabV3+	0.9333	0.9952	0.928	0.892	0.9181
Syn	HRNetV2	0.9357	0.9953	0.9308	0.8954	0.9213

Table 4.9. Different network results cross-datasets

Dataset	Network Name	Train Loss	Train mIoU	Validation Loss	Validation mIoU
CrossBase	VGGPSPNet	0.0272	0.8778	0.3876	0.412
CrossBase	VGGUNet	0.0152	0.9273	0.3787	0.4264
CrossBase	DeeplabV3+	0.0093	0.9283	0.4469	0.4495
CrossBase	HRNetV2	0.7546	0.9046	0.8706	0.4007
CrossMiche480	VGGPSPNet	0.0253	0.8672	0.0365	0.6606
CrossMiche480	VGGUNet	0.0169	0.91	0.0476	0.6788
CrossMiche480	DeeplabV3+	0.0092	0.9416	0.0467	0.6466
CrossMiche480	HRNetV2	0.7523	0.8871	0.7527	0.7472
CrossMiche960	VGGPSPNet	0.022	0.9097	0.0521	0.5887
CrossMiche960	VGGUNet	0.0156	0.9306	0.0509	0.5839
CrossMiche960	DeeplabV3+	0.0089	0.9484	0.0494	0.6042
CrossMiche960	HRNetV2	0.7564	0.9010	0.7633	0.5974
CrossSyn	VGGPSPNet	0.0453	0.7016	0.2527	0.5172
CrossSyn	VGGUNet	0.0244	0.8397	0.2976	0.5224
CrossSyn	DeeplabV3+	0.0102	0.8992	0.2818	0.5529
CrossSyn	HRNetV2	0.7708	0.8158	0.8106	0.6033

Table 4.10. Different network results of the cross-datasets per-class

Dataset	Network Name	Background mIoU	Sclera mIoU	Iris mIoU	Pupil mIoU
CrossBase	VGGPSPNet	0.9247	0.353	0.3376	0.0812
CrossBase	VGGUNet	0.9352	0.361	0.3406	0.0657
CrossBase	DeeplabV3+	0.9383	0.3551	0.3617	0.1428
CrossBase	HRNetV2	0.884	0.2846	0.2997	0.1337
CrossMiche480	VGGPSPNet	0.9901	0.6099	0.7331	0.2886
CrossMiche480	VGGUNet	0.9896	0.5898	0.7163	0.4164
CrossMiche480	DeeplabV3+	0.9907	0.6007	0.6787	0.3162
CrossMiche480	HRNetV2	0.9932	0.6826	0.8058	0.5074
CrossMiche960	VGGPSPNet	0.9893	0.4397	0.5521	0.3643
CrossMiche960	VGGUNet	0.9891	0.4219	0.5664	0.355
CrossMiche960	DeeplabV3+	0.989	0.4698	0.6516	0.3019
CrossMiche960	HRNetV2	0.9876	0.457	0.6411	0.2995
CrossSyn	VGGPSPNet	0.9413	0.436	0.4213	0.2641
CrossSyn	VGGUNet	0.9473	0.4546	0.4219	0.2659
CrossSyn	DeeplabV3+	0.9639	0.6028	0.455	0.1897
CrossSyn	HRNetV2	0.9637	0.57	0.4253	0.4541

5. CONCLUSION

Within the scope of this thesis eye segmentation problem analyzed. Image processing based iris and sclera segmentation methods are examined, their advantages are can be summarized as:

- Reliable on controlled settings
- Doesn't require heavy computational power like DNNs

Their shortcomings are

- Requires specialized equipment like IR cameras
- Images needs to be taken stationary
- They don't work as reliable when there are outside factors like reflections
- There aren't any image processing based methods can effectively segment the sclera, iris, and pupil regions together.

Therefore, to tackle these shortcomings this thesis suggested the idea of using DNNs. In this thesis, the eye segmentation problem explored using structurally different DNNs. The requirement of the data for this task is fulfilled by manually segmenting 1660 images 1150 of them being unique and rest is being from MicheII dataset. Thus, 1660 image masks and 1150 images are shared with researchers at:
https://github.com/melihoz/eyedataset.

Experiments on the DNNs shown several outcomes:

- Shuffling significantly increases a network's generalization abilities.
- Adam optimizer works efficiently without the need of tweaking recommended settings.
- Batch size needs to be above 1 to increase networks generalization abilities. However, the ideal batch size depends on the dataset size.
- Augmentation operations observed to not to have a negative effect on the eye segmentation if parameters are set low.

- Deeper and more complex network structures observed to bring improvements on the segmentation quality. However dataset content seems to be more impact-full.
- In the case of additional datasets trained with the dataset we want to segment. Effe

  of the additional datasets never observed to be harmful. In some cases, it observed to improve results significantly.
- In the eye segmentation case, when we use a network trained with a similar context data it's observed that network can still identify outer eye boundaries with high performance.

With the results obtained in this thesis, it's concluded that DNNs can be used in the eye segmentation problem with significant performance in loosely controlled situations. Thus, this study also shows that data has more effect than the network used for segmentation. Also, the roadmap of this thesis can be applied to any DNN based case studies.

6. REFERENCES

Alkassar, S., Woo, W.L., Dlay, S.S. and Chambers, J.A. 2017. Robust Sclera Recognition System with Novel Sclera Segmentation and Validation Techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(3):474–486.

Arsalan, M., Hong, H.G., Naqvi, R.A., Lee, M.B., Kim, M.C., Kim, D.S., Kim, C.S. and Park, K.R. 2017. Deep Learning-Based Iris Segmentation for Iris Recognition in Visible Light Environment. *Symmetry*, 9(11):263, http://www.mdpi.com/2073-8994/9/11/263.

Badrinarayanan, V., Kendall, A. and Cipolla, R. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

Bazrafkan, S., Thavalengal, S. and Corcoran, P. 2018. An end to end Deep Neural Network for iris segmentation in unconstrained scenarios. *Neural Networks*, 106:79–95, https://doi.org/10.1016/j.neunet.2018.06.011.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818. 2018.

Chollet, F. 2017. Deep learning with Python. Manning Publications Company.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-January, pp. 1800–1807. Institute of Electrical and Electronics Engineers Inc, 2017.

Chollet, F. et al. 2015. Keras. GitHub, https://github.com/fchollet/keras.

Danişman, T. 2020. Segmentation of Portrait Images Using A Deep Residual Network Architecture. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 22(65):569–580.

Daugman, J.G., 1994. Biometric personal identification system based on iris analysis, uS Patent 5,291,560.

De Marsico, M., Nappi, M. and Proença, H. 2017. Results from miche ii–mobile iris challenge evaluation ii. *Pattern Recognition Letters*, 91:3–10.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE, 2009.

Garbin, S.J., Shen, Y., Schuetz, I., Cavin, R., Hughes, G. and Talathi, S.S. 2019. Openeds: Open eye dataset. *arXiv preprint arXiv:190503702*.

Goodfellow, I., Bengio, Y. and Courville, A. 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Gunay, M., Goceri, E. and Danisman, T. 2015. Automated detection of adenoviral conjunctivitis disease from facial images using machine learning. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). pp. 1204–1209. IEEE, 2015.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:170404861*.

Jain, A.K. 1989. Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice Hall.

Liu, N., Li, H., Zhang, M., Liu, J., Sun, Z. and Tan, T. 2016. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In: 2016 International Conference on Biometrics, ICB 2016. Institute of Electrical and Electronics Engineers Inc, 2016.

Lucio, D.R., Laroca, R., Severo, E., Britto, A.S. and Menotti, D. 2018. Fully convolutional networks and generative adversarial networks applied to sclera segmentation. In: 2018 IEEE 9th International Conference on Biometrics Theory, Applica-

tions and Systems, BTAS 2018. Institute of Electrical and Electronics Engineers Inc, 2018.

Luo, B., Shen, J., Cheng, S., Wang, Y. and Pantic, M. 2020. Shape constrained network for eye segmentation in the wild. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 1952–1960. 2020.

Matey, J.R., Naroditsky, O., Hanna, K., Kolczynski, R.A., Loiacono, D.J., Mangru, S., Tinker, M., Zappia, T.M. and Zhao, W.Y. 2006. Iris on the move: Acquisition of images for iris recognition in less constrained environments. *Proceedings of the IEEE*, 94(11):1936–1946.

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

Ronneberger, O., Fischer, P. and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer, 2015.

Rot, P., Emeršič, Ž., Struc, V. and Peer, P. 2018. Deep multi-class eye segmentation for ocular biometrics. In: 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI). pp. 1–8. IEEE, 2018.

Shah, S. and Ross, A. 2009. Iris segmentation using geodesic active contours. *IEEE Transactions on Information Forensics and Security*, 4(4):824–836.

Simonyan, K. and Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9. 2015.

Wildes, R.P. 1997. Iris recognition: An emerging biometrie technology. *Proceedings of the IEEE*, 85(9):1348–1363.

Wood, E., Baltrušaitis, T., Morency, L.P., Robinson, P. and Bulling, A. 2016. Learning an appearance-based gaze estimator from one million synthesised images. In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications. pp. 131–138. 2016.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. 2017. Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890. 2017.

Zhou, P., Feng, J., Ma, C., Xiong, C., HOI, S. et al. 2020. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *arXiv preprint arXiv:201005627*.

Zhou, Z., Du, E.Y., Thomas, N.L. and Delp, E.J. 2011. A new human identification method: Sclera recognition. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(3):571–583.

CURRICULUM VITAE

Melih ÖZ

melihoz@akdeniz.edu.tr

EDUCATION

Master of Science 2017-2021	Akdeniz University Institute of Natural and Applied Sciences, Computer Engineering, Antalya, Turkey
Bachelor of Science 2008-2014	Akdeniz University Faculty of Engineering, Electrical Electronics Engineering, Antalya, Turkey

WORK EXPERIENCE

Research Assistant 10/2017-Present	Akdeniz University Faculty of Engineering, Computer Engineering, Antalya, Turkey

PUBLICATIONS

1- Akçay, H. G., Kabasakal, B., Aksu, D., Demir, N., Öz, M., and Erdoğan, A. 2020. Automated bird counting with deep learning for regional bird distribution mapping. *Animals*, *10*(7), 1207.