KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRIC-ELECTRONIC ENGINEERING

DATA AUGMENTATION IN RETINAL FUNDUS IMAGES SEGMENTATION USING DEEP LEARNING

MASTER THESIS

Nagat MASUED

JULY 2019
TRABZON

KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRIC-ELECTRONIC ENGINEERING

DATA AUGMENTATION IN RETINAL FUNDUS IMAGES SEGMENTATION USING DEEP LEARNING

Nagat MASUED

This thesis is accepted to give the degree of

"MASTER OF SCIENCE"

By

The Graduate School of Natural and Applied Sciences at
Karadeniz Technical University

The Date of Submission : **21 / 05 /2019**

The Date of Examination : **21 / 06 /2019**

Supervisor : **Assistant.Prof.Dr. Mehmet ÖZTÜRK**

Trabzon 2019

KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

Department of Electrical & Electronics Engineering

Nagat MASUED

DATA AUGMENTATION IN RETINAL FUNDUS IMAGES SEGMENTATION USING DEEP LEARNING

Has been accepted as a thesis of

MASTER OF SCIENCE

after the Examination by the Jury Assigned by the Administrative Board of the Graduate School of Natural and Applied Sciences with the Decision Number 1806 dated 28 / 05 / 2019

Approved By

Chairman	:	Associate Prof. Dr. Nur Hüseyin KAPLAN
Member	:	Associate Prof. Dr. Önder AYDEMİR
Member	:	Assistant Prof. Dr. Mehmet ÖZTÜRK

Prof. Dr. Asim KADIOĞLU

Director of Graduate School

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisor, Assistant Prof. Dr. Mehmet Öztürk for the continuous support of my MSc study and related research, for his patience, motivation, and immense knowledge.

I must express my very profound gratitude to my family, and especially my sister Mariam without her help and support was impossible to complete my masters today.

Last but not the least I would like to thank T.C. Başbakanlık, Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı (YTB), who provided me an opportunity to join their Turkey Scholarships and Karadeniz Technical University to provide me an opportunity to do my masters from here.

Nagat MASUED

Trabzon 2019

DECLARATION

I, the undersigned, declare that this study entitled "Data Augmentation in Retinal Fundus Images Segmentation using Deep Learning", completed under the supervision of Assistant Prof. Dr. Mehmet ÖZTÜRK, is my original work and has not been presented for a degree in any other university and that all sources of materials used for the study have been duly acknowledged. 21/06/2019

Nagat MASUED

TABLE OF CONTENTS

Master Thesis

SUMMARY

DATA AUGMENTATION IN RETINAL FUNDUS IMAGES SEGMENTATION USING DEEP LEARNING

Nagat MASUED

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Department of Electrical & Electronics Engineering
Supervisor: Assistant Prof. Dr. Mehmet ÖZTÜRK
2019, 69 Pages

Image segmentation is an important tool for medical images where the images are separated based on the type of tissue and organs to further visualization and diagnosis.

In ophthalmology domain, retinal blood vessel segmentation is one of the substantial tasks due to the helpful information from the segmentation result, which provides diagnosis and monitoring of eye diseases such as diabetic hypertension glaucoma and retinopathy. Over the last few years, deep learning based methods accomplished a state-of-art performance in most of computer vision tasks. Most of these methods are supervised and require large amount of labeled data to be able to train the models.

This project investigates the retinal vessel segmentation using deep learning approach. To overcome the scarcity of the data in this domain, the thesis presents a method of data augmentation based on Generative Adversarial Networks (GANs) to generate more training data for sufficient learning.

Key Words: Retinal vessel Segmentation, Deep Learning, Generative Adversarial Networks (GAN), Data Augmentation, Medical Image Analysis

Yüksek Lisans Tezi

ÖZET

DERİN ÖĞRENME KULLANARAK RETİNA FUNDUS GÖRÜNTÜLERİNİN SEGMENTASYONU İÇİN VERİ ARTIRMA

Nagat NASUED

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği
Danışman: Dr. Öğr. Üyesi Mehmet ÖZTÜRK
2019, 69 Sayfa

GGörüntü segmentasyonu, görüntülerin daha iyi görselleştirilmesi ve tanı konulması için doku ve organların türüne göre ayrıldığı tıbbi görüntüler için önemli bir araçtır. Oftalmoloji alanında, retinal kan damarı segmentasyonu, diyabetik hipertansiyon glokomu ve retinopati gibi göz hastalıklarının teşhisi ve izlenmesini sağlayan faydalı bilgilerinden dolayı önemli görevlerden biridir. Son birkaç yılda, derin öğrenme temelli yöntemler, bilgisayarla görme görevlerinin çoğunda üstün bir performans sergilemiştir.

Bu yöntemlerin çoğu danışmanlı öğrenmeye dayalıdır ve modelleri eğitmek için büyük miktarda etiketlenmiş veri gerektirir. Bu tez, derin öğrenme yaklaşımı kullanarak retinal damar segmentasyonunu araştırmaktadır. Bu alandaki verilerin azlığının üstesinden gelmek için, tez, daha fazla eğitim verisi üretmek üzere, Üretken Çekişmeli Ağlara (Generative Adversarial Networks - GANs) dayanan bir veri büyütme yöntemi sunmaktadır.

Anahtır Kelimeler: Retina damarlarının segmentasyonu, Derin öğrenme, Üretken Olumlu Değişken Ağlar, Veri büyütme, tıbbi görüntü analizi.

LIST OF FIGURES

,

LIST OF TABLES

Page No

LIST OF SYMBOLS AND ABBREVIATIONS

Adam	Central Processing Unit
Adam	Adaptive Moment Estimation
CNN	Convolutional Neural Network
DCGAN	Deep Convolutional Generative Adversarial Network
FCNN	Fully-Convolutional Neural Network
GAN	Generative Adversarial Network
Leaky ReLU	Leaky Rectified Linear Unit
ReLU	Rectified Linear Unit
tahn	Hyperbolic Tangent
WGAN	Wasserstein Generative Adversarial Network
SGD	Stochastic Gradient Descent
FA	Fluorescein Angiography
VGG	Visual Geometry Group
DRIVE	Digital Retinal Images for Vessel Extraction
STARE	STructured Analysis of the Retina

1. INTRODUCTION

vessel segmentation in fundus images is an essential stage for retinal diseases diagnosis as well as a helpful step in understanding the underlying mechanisms of eye diseases especially, diabetic which is one of the most leading causes of blindness. Therefore, early diagnosis of these diseases can be an important factor in preventing or reducing the burden of disease. One of the approaches which have been deployed is machine learning. Machine learning applications have been widely used in our daily life at all levels from automatic speech recognition [1] to face detection [2], and medical diagnosis [3]. Deep learning, in particular, currently is extensively applied in the medical domain. It is a supervised approach which means it requires large training set of labeled data, (ground-truth), in order to make a reliable prediction. However, due to the strenuous of manual labeling for medical images, only a small dataset of labeled vessels in fundus images is available. STARE [4] and DRIVE [5] fundus image benchmarks are notable examples of publicly available dataset that have be utilized in this task.

1.1. Objectives

The main objective of this project is to describe a deep learning approach for vessels segmentation for retinal fundus image and improve the result achieved from other approaches. There are four main goals that the project aims to address:

1. Examine the application of deep neural networks in the semantic segmentation task.

2. Expand the dataset by generating synthetic data using different approaches such as traditional augmentation and Generative Adversarial Network (GANs).

3. Test the potential of improving the output of the segmentation system by using synthetic data.

4. Assessing the degree of contribution a joint GAN and deep learning models may

have in handling the representation of the images and improving the segmentation task.

1.2. Challenges

To address the aim of this project a number of challenges need to be highlighted.

- Lack of annotated dataset due to laborious natural of manually labeling. This formed a barrier and limited the model's abilities to produce good results as learning semantic interpretation in deep learning is strongly tied to the number of training examples.
- Inconsistency in some datasets due to variabilities in clinical settings acquisition protocols for medical images.
- Resource availability and computation cost.In addition to data scarcity, a limited resource such as GPU power has placed some limitation on the training.

1.3. Contributions

In this project we proposed a supervised deep learning pipeline, our approach is based on a synthetic fundus image dataset and combine this dataset with our training dataset (labeled dataset) to fill the requirements of deep learning of large training dataset and enhance the performance of existing supervised/deep learning method.

1.4. Retinal Vasculature

Retinal consist of layers of tissues ans vessels. Vessels are portion of the transmission system to supply retinal tissues with oxygen and nutrition. The cornea and the human eye's crystalline lens are equivalent to a camera's lens and the eye's iris works like a camera's diaphragm that monitoring the amount of light that reaches the retina [6]. The light is converted into electrical signals in the retina to form as a vision in the brain [6]. The general structure of human retinal is shown in Figure (1.3).

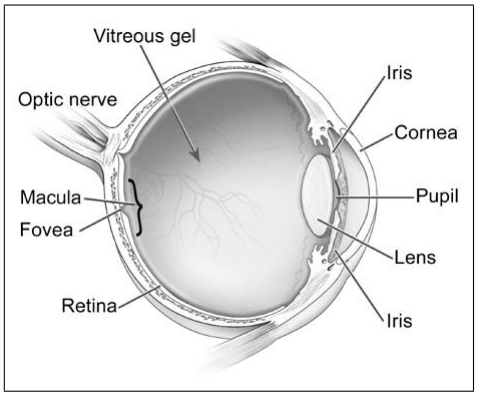
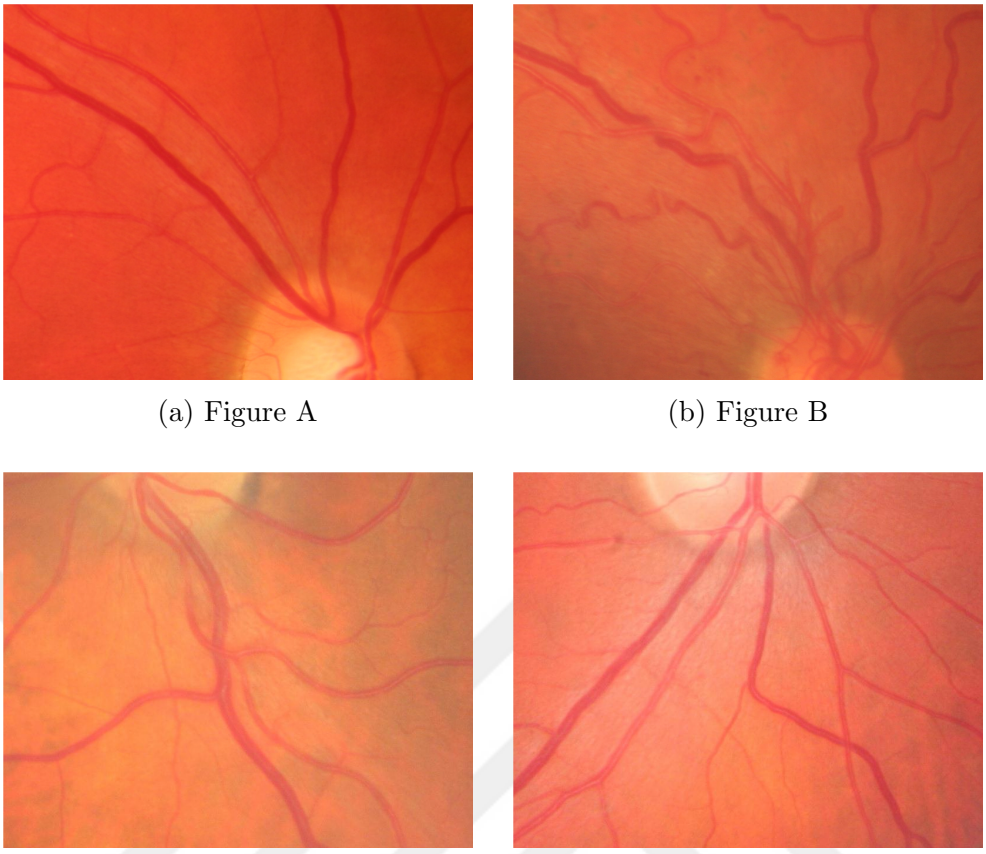


Figure 1.1. Human eye structure [5]

From a medical perspective, a different type of diseases appears in the retina. For example, ischemia and new blood vessels growing can be caused by diabetic retinopathy. Moreover, changes in retinal structure can be a sign to heart diseases such as stroke [7,8]. In fact, diabetes is one of the most recognizable diseases through the retina and one of the blindness common causes [6]. Roughly there are around 150 to 200 millions people with diabetes globally. Glaucoma, on the other hand, can also cause a visual loss by damaging the optic nerve. Early detection of Glaucoma can minimize the effect of the disease [9].

Other diseases that connected to the retina are Neovascularization and Cardiovascular. Neovascularization or macular rupture is one of the most vision-threatening diseases significantly causes deterioration and damage in the macula which affects the central vision [10]. Cardiovascular is another disease that can be manifested in the retina. This kind of disease can cause widening in the veins and thin in the arteries which in turn can increase the risk of stroke.choroidal infarcts is another sign of cardiovascular diseases which can be manifested in the retina [6]. An illustration different disease conditions in retinal vessels is shown in Figure (1.2). Image A shows a healthy retinal vasculature while image B demonstrates a diabetic retinopathy case of retinal vasculature with venous beading. Image C, compared to healthy vasculature in D, illustrates vasculature with narrow arteries. The occurrence of arterial narrowing can be a sign of hypertension.



(a) Figure A (b) Figure B

(c) Figure C (d) Figure D

Figure 1.2. Condition of diseases in retinal vasculature

1.5. Imaging Modalities

This section details the basic background on image modalities for segmentation system. Fluorescein angiography and optical coherence tomography angiography are firstly discussed since they represent the common format of medical imaging which is widely used in deep learning research.

1.5.1. Fluorescein Angiography

When the interior surface of the eye is capture by the a narrow-band camera this type of images called fundus photography, Fluorescein Angiography (FA) format was firstly introduced in 1961 and have been used since then. FA modality is an invasive imaging method where the circulation system injected by sodium fluorescein [11]. The drawback of FA method is the risk of the side effects from the fluorescent dye injection.

in spite of the fact that it produces a good 2D illustration of the retina, it could mix up superficial and deeper retinal networks. Figure (1.3) shows an example of FA image from DRIVE dataset.

Figure 1.3. An example of FA images DRIVE dataset [5]

1.5.2. Optical Coherence Tomography Angiography

To overcome the drawback of FA, optical coherence tomography (OCT) is presented. OCT is a noninvasive imaging technique captures the tissue using a long-wavelength light with the micrometer-scale resolution[12]. OCT is the most used imaging technology in ophthalmology, due to its capability to envisage the context in the retina in 3D. Figure (1.4)shows an example of image capture by OCT modality.

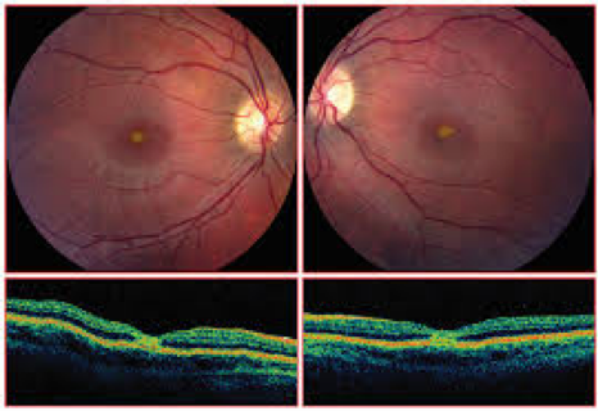


Figure 1.4. An example OCTA image [13]

1.6. Machine Learning

In the digital society, the need for a machine which is capable to make decisions from large amounts of data has been increased. This machine can deal with both the interpretation and the decision making, instead of manually setting up the interpretation. While machine learning methods may require human interaction, from data pre-processing to tuning. Deep learning which is a subset of machine learning, supervised learning, automatically learns representations and internal structure of raw input data by using stacked processing and convolution layers when dealing with images. In the following subsections, the basics of Artificial Neural Networks are explained, starting with the building blocks, the concepts and leading up to Convolutional Neural Networks (CNN) which is the common architecture for image processing.

1.6.1. Artificial Neural Networks

Artificial neural networks consist of neurons organized in a network structure. The neural network mimics the neurons association of the brain biological neurons. The ANN usually takes multi-inputs and produce output. ANN can be deployed for both regression and classification tasks. For example in medical classification settings, an image of a human organ can be fed to the network as input pixels and the network classifies the organ type.

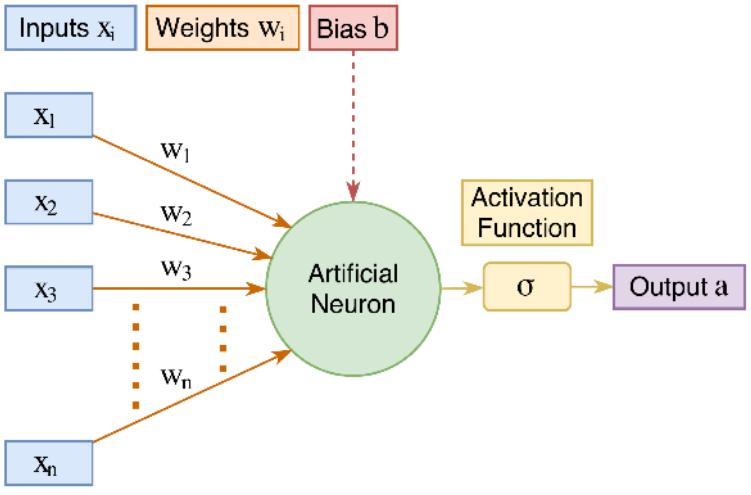


Figure 1.5. Schematic of an artificial neuron

The fundamental concept of ANN neuron is to receive multiple inputs $x_1, x_2, ...$ and compute a weighted sum z for these inputs using the weights $w_1, w_2, ...$ so each layer of the network has its own unique weight values. Additionally a b bais value is added to the weighted sum z to adjust the output. Equation (1.1) shows the weighted sum in a basic ANN, while equation (1.2) illustrates that the output of the ANN usually goes through some nonlinear activation function θ.

$$z = f(x, w, b) = \sum_{i=0}^{N} w_i x_i + b \tag{1.1}$$

Which can be reformulated as a vector:

$$y = \theta(z) \tag{1.2}$$

Figure (1.5) illustrates the structure of the artificial neuron. Through the choice of weights and bias, the artificial neuron can be trained to approximate a function given the inputs x.

Furthermore, the network can have any number of hidden layers. If ANN has one hidden layer, the network classified as a simple or shallow neural network and it called deep neural networks if the number of hidden layers is more than one. The diagram in Figure (1.6) shows a shallow neural network with one input layer, one hidden layer, and an output layer. This type of network is called a fully connected layer where each layer contains a various number of neurons that receive the inputs from all neurons in the previous layer and send output to all the neurons in the next layer.

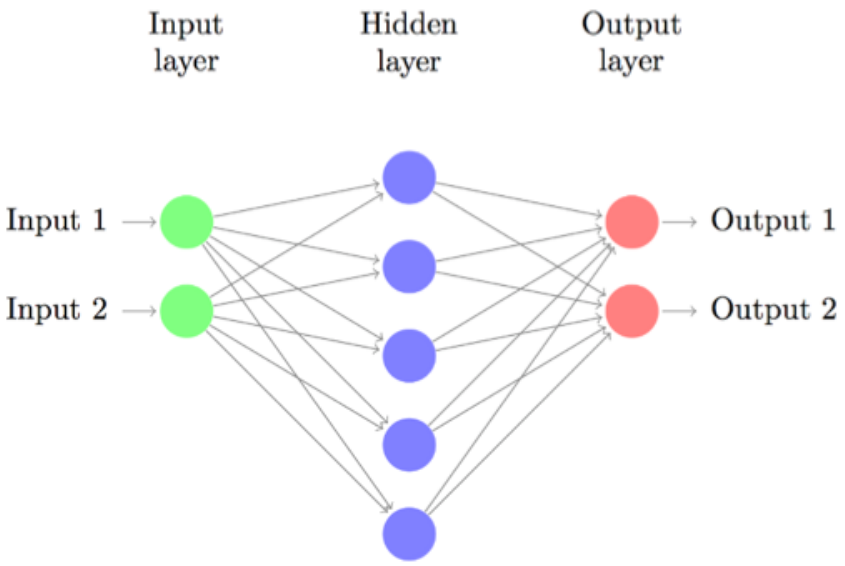


Figure 1.6. Neural network with one hidden layer

1.7. Optimizers

Optimizers are a fundamental part of the neural network, understanding how they work would help us to choose the right optimizer for our model. The aim of optimizing ANN is to find the optimal weights and biases automatically, so the network finds the target output y for the input x. For the function to achieve this goal, it is essential to outline a metric for how well the network approximates the output, this metric called *cost function* or *loss function* $j(\theta)$ where θ represent the weights and biases.
Given a set of N training examples $xt = [xt1; xt2; ..; xn]$ and corresponding targets $y = [y1; y2; ...; yn]$, $j(\theta)$ is typically computed as the average of the per-example loss function as shown in equation (1.3).

$$j(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(a(xT_{i;}) ; y_i) \tag{1.3}$$

where L is given by:

$$L = ||a(x_i; (\theta) {y_i}^2|| \tag{1.4}$$

Since the goal of the loss function is to evaluate how good the network can predicate

the output by depreciating the difference between the predicted output by the network and the ground-truth. This is usually done using a variation of the gradient descent algorithm.

1.7.1. Gradient Descent

The aim of gradient descent is finding the local minimum of $J(\theta)$ with respect to it is parameters (weights and biases)) and updating the parameters by taking proportional steps to the opposite direction of the gradient at the present point . For each training example x_{Ti} with the corresponding target output yi, the narrowest descent direction is given by calculating the per-example loss function's negative gradient with respect to the parameters in x_{Ti} and yi position:

$$-g\theta_i = \Delta_\theta L(\alpha(x_{T_i}; \theta), y_i) \tag{1.5}$$

The ultimate gradient for the loss function $J(\theta)$ given by calculating the average of all gradients over the whole training set x_i:

$$-g\theta = \frac{1}{N} \sum i = 1N - g\theta \tag{1.6}$$

Depending on the initialization of the parameters, it is possible for gradient descent to find the global minimum of $g(\theta)$, however, this is not guaranteed, unless $g(\theta)$ is convex [14].Additionally, it is necessary to perceive that for gradient descent to converge, the loss function needs to be smooth and provides gradients everywhere. This is also the reason why the chosen loss function is typically different from the actual objective, and a surrogate loss function is used instead.

Backpropagation and Gradient Descent. The optimization is a complicated task in deep neural networks because of the high number of parameters such as weights and baises . The backpropagation algorithm is presented to overcome this problem [15]. Backpropagation is determined by calculating the derivative of the loss function with respect to network parameters in the network so the number of the equations and

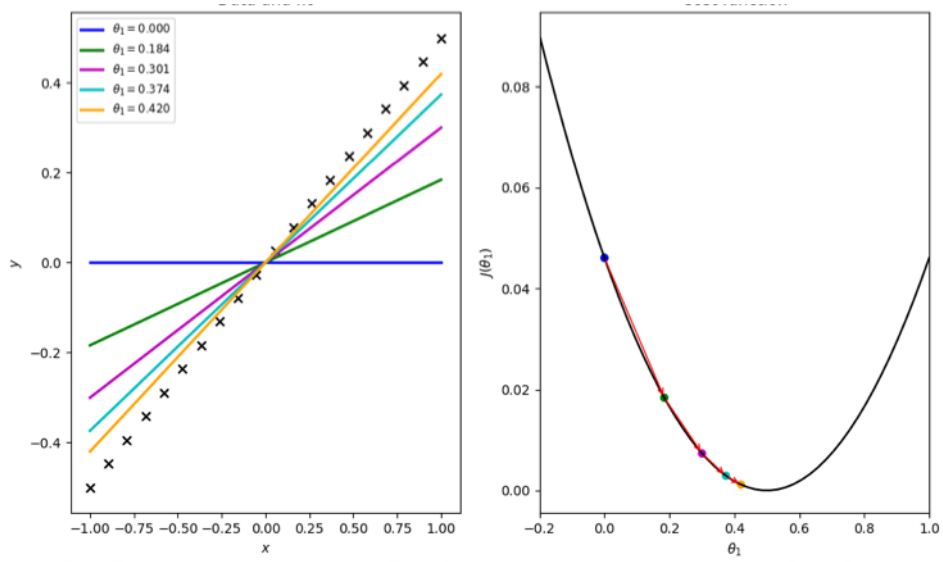


Figure 1.7. Gradient descent on a 1-dimensional loss function showing how the method iteratively gets closer to the global minimum

unknowns remain unchanged . Backpropagation on a multi-layer network is performed by applying the chain rule to find the change in output [16]. Additionally, backpropagation and gradient descent is utilized together to reduce the loss function (weights, biases and other learning parameters). There is a variation of gradient descent and one of the commonly used types is the stochastic gradient descent.

Stochastic gradient descent (SGD) where the update is done by calculating the weight for each training data and update the new weights directly.
Mini-batch stochastic gradient descent is an extension of gradient descent, where the updates are performed based on the average gradient $g(\theta)$ of several randomly selected training samples. Even though gradient descent is a very powerful optimization technique, sometimes it fails in choosing the best learning rate for the training dataset in some cases. On one hand, if the learning rate is too large for the dataset, the training might oscillate, not converge, or skip over relevant local minima. On the other hand, if the learning rate is selected to be too small, it significantly delays the convergence process. To overcome this problem, an adaptive learning rate optimizer is used. So instead of finding a global learning rate, every parameter in the training has its own learning. Most commonly used adaptive learning optimizers are Adaptive Moment Estimation *Adam* [17] and *RMSProp* [18].

1.7.2. RMSProp

The RMSProp optimizer leverages the magnitude of current gradients to normalize the gradients. Therefore, the algorithm can take larger steps toward the gradient to make the training faster. The difference between RMSprop and gradient descent is on how the gradients are calculated. Equation (1.7) shows the calculation for the RMSProp optimizer.

$$G_{ii} = \gamma G_{ii} + (1 + \gamma)(\frac{m}{i = 1} \sum_{i=1} m g(\theta_i)) \tag{1.7}$$

where G G is a matrix containing the sum of squares of the past gradients with respect to all parameters along its diagonal, g is a gradient and m is the number of examples in minibatch.

The decay rate γ is now an bonus hyperparameter with a default value of 0.9. The algorithm still adjusts the learning rate of each parameter based on the magnitude of its gradients, however, the moving average prevents updates from getting smaller [11].

1.7.3. Adaptive Moment Estimation

Adam algorithm finds an individual learning rate for each parameter using the strengths of adaptive learning techniques. It can be considered a combination of both RMSProp and SGD. It uses the SGD to keep the learning rate of each trainable parameter in the equilibrium state. Similarly to RMSProp, Adam uses the momentum by adding the small values of the previous gradients to the new gradients to accelerate the gradients vectors in the right directions.
The equation formulas for Adam optimizer are giving in the following equations:

$$m = \beta_1 m + (1 + \beta_1)(\frac{m}{i = 1} \sum_{i=1}^{m} g\theta_i) \tag{1.8}$$

$$v = v\beta_2 + (1 + \beta_2)(\frac{m}{i = 1}\sum_{i=1}^{m} g\theta_i) \tag{1.9}$$

m and v are the estimates of the moment of the gradients. The Adam loss for the updated parameters is given by:

$$\theta = \theta - \eta\frac{v}{\sqrt{v} + \epsilon}m \tag{1.10}$$

Recommended values by the authors [19] for the parameters of Adam are $=1.10^{(-8)}$, β_1=0.99 and β_2=0.99

1.8. Activation Functions

Activation functions are nonlinear transformations implemented to the weighted sum of the input. [20] shows that using a non-linear transformation neural networks are global scale since it is able to approximate any computable function. There is a number of choices for activation functions which will be summarized with their properties in the next section.

1.8.1. Sigmoid Function

Sigmoid function changes the range of the output from $(-\infty,\infty)$ to (0,1) [21] as shown in Figure (1.8). Due to its biologically inspired interpretation where the 0 represent the neuron that does not work at all, and 1 represents a neuron working at maximum frequency [22]. Sigmoid function is widely used in deep learning tasks despite its disadvantages. One of its notable drawbacks is that the function saturates and therefore it provides only gradients very close to zero in these regions [23].

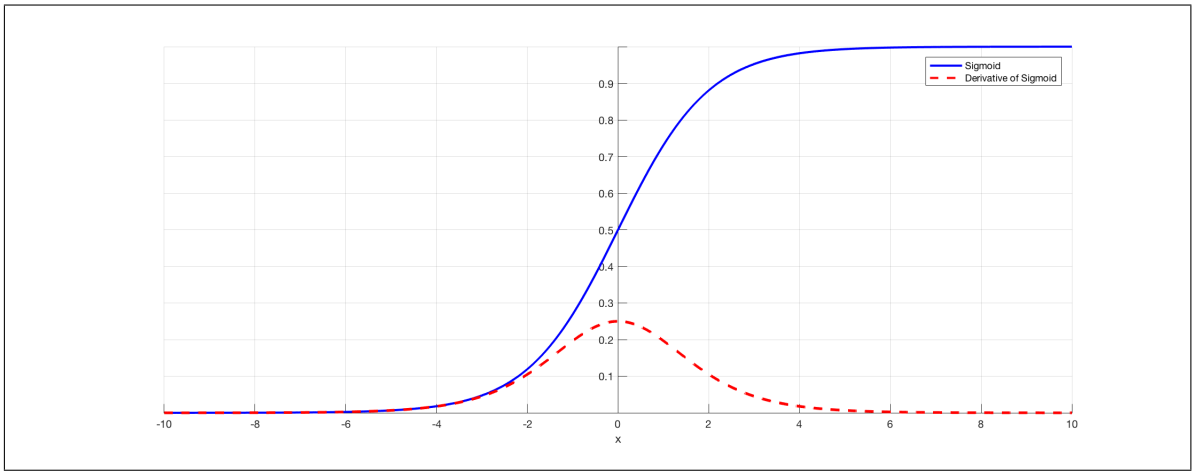


Figure 1.8. Sigmoid function for input range x=[-10,10]

The mathematical formula for the sigmoid function is giving by:

$$\varphi = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{1.11}$$

1.8.2. Hyperbolic Tangent

Hyperbolic tangent is another type of activation function. It can be considered as the scaled version of sigmoid function where the output is scaled between (-1,1) [22]. To generate the scaled output, the output has to be passed to a *tanh* function as shown in equation (1.12).

$$\varphi = tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{1.12}$$

This means that the function is centered around 0 and does not have the drawback that sigmoid has. As such, hyperbolic tangent is preferred to be used over sigmoid function [21]. The function is shown in Figure (1.9)

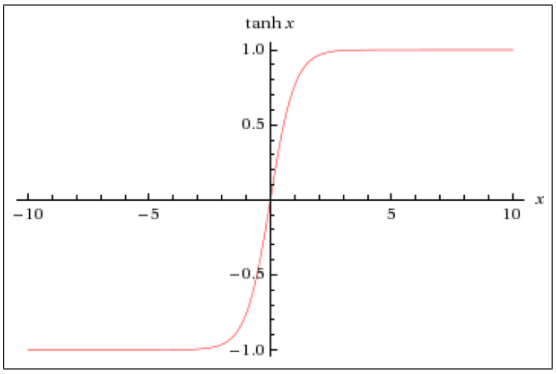


Figure 1.9. Hyperbolic function for input range [-10,10]

The disadvantage of these functions is their behavior in hidden layers since their derivatives have a tendency to produce poor gradients when a neuron approaches saturation (function output comes very close to the upper or lower bounds of the output range) [14].

1.8.3. Rectified Linear Unit

AAnother activation function which presents good results and has low computation cost for deep neural networks, this function is called the rectified linear unit (ReLU) [24]. ReLU is defined by the following equation:

$$ReLU(x) = max(0, x) \tag{1.13}$$

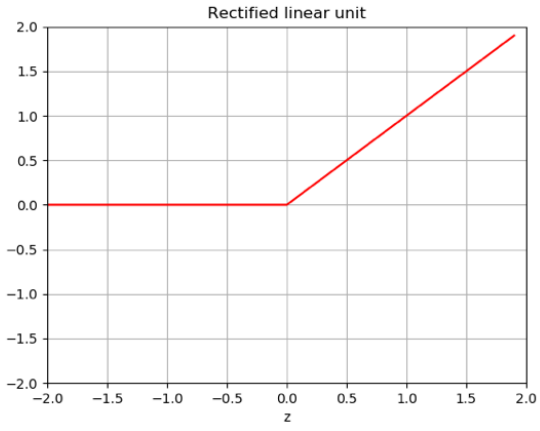


Figure 1.10. ReLU function

ReLU returns zero value for negative values and linear output for all positive values. Similar to sigmoid, ReLU a biological inspired function, it is efficient and faster and as such, it is mostly used for CNNs since it is being differentiable everywhere except 0, so it is easy to calculate the gradient descent beside its effect in speeding up the processing step. Figure (1.10) shows the ReLU function.

1.8.4. Leaky ReLU

Leaky ReLU is another version of ReLU to overcome *"the dying Relu"* " phenomenon by having a small slope in the negative range as shown in Figure (1.11).

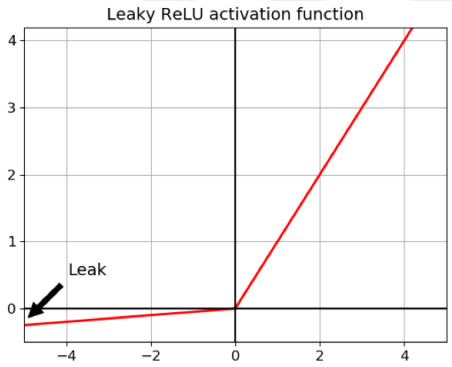


Figure 1.11. Leaky ReLU activation function

1.8.5. Softmax Activation

Softmax activation function calculates the probability distribution over some events. The probabilities sum is equal to 1. As such this function is commonly used when dealing with classification tasks, in which the probability of each class is calculated and the largest one represents the right class. In such a task, the softmax function is utilized as the ultimate output layer of the network. The function is defined by:

$$softmax(z) = \frac{e^{x}}{\sum_{n} e^{zn}} \tag{1.14}$$

Where z is a vector of the inputs to the output layer and n indexes the number of events.

1.9. Loss Function

Loss functions is another important concept in ANN. It is used to calculate the difference among the predicted output by the model and the real output (groundtruth), by decreasing the value of loss function the model performance increases. It is important to choose the correct loss function for each task. In segmentation tasks, for instance, there are two widely used loss functions [3]:

- Binary Cross Entropy (BCE) is a measure of the negative log likelihood. It quantifies the difference between two probability distributions. The mathematical equation (1.15) shows the computation for the BCE loss function.

$$l_{BCE} = \sum_{n=1}^{N} \widetilde{y}_n log(y_n) + (1 - \widetilde{y}_n).log(1 - y_n) \tag{1.15}$$

- Dice Loss Functionis another frequently utilized function in segmentation tasks. The function showed good results in imbalanced segmentation tasks [25] by resolving the class unevenness. It does so by totally ignoring true negative predictions. The function is calculated as:

$$dice\_loss\_function = 1 - DSC(y', y) \tag{1.16}$$

Where y : ground truth and y': predicated output.

1.10. Convolutional Neural Networks

Convolutional neural networks have the same building blocks as fully connected neural networks such as weights, bias, loss function, and other previously presented elements. In addition to these, unique elements that are specific to CNN used such as convolutional filters.

CNNs have been around in the 1980s [26] but it was not utilized widely as it was not impractical for real-world applications due to two reasons: the small amount of the training data and the demand for long training time because of process parallelization. Recently, researchers have overcome the first problem with the increase of

user-generated content such as photographs/data being uploaded on the web. As for the second drawback, the advances in GPU technology allowing faster training time have helped to accelerate the research in this area.

The difference between CNNs and fully connected neural networks is the connections between weights and output. A typical CNN is made up of several layers connected by weights acting as convolutional filter kernels. A convolution of 2D function I by a convolution kernel k identified as:

$$(k * I)(x, y) = \sum_i \sum_j I(x - i, y - j)K(i, j) \tag{1.17}$$

The the filters are usually smaller in size-wise compared to the image, meaning that they are applied locally to the image. They have then shifted a set distance in the image. The pixel output is the result of a dot product between the filter and a local neighborhood of the corresponding voxel in the input image as shown in figure (1.12).

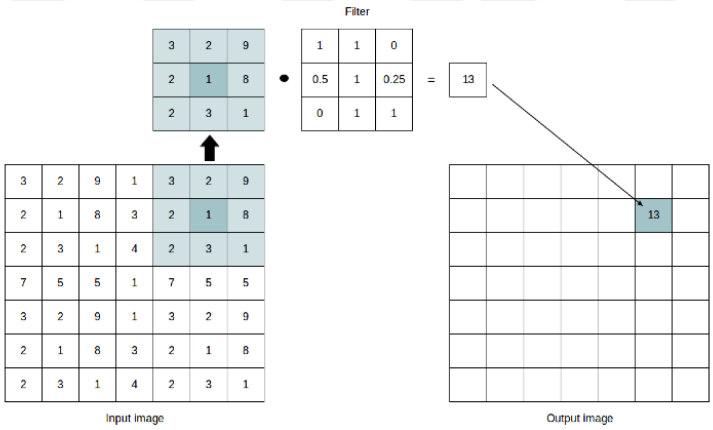


Figure 1.12. Visualization of how the output is formed in a CNN. A filter is applied to a local neighborhood in the image through dot product. The result is a single value at the center of the corresponding position of the neighborhood in the output image. The filter is applied to the entire input image in this way.

The number of parameters is reduced after each layer in CNN . Each feature map has its own unique set of weights that are applied to every spatial location of the input. Feature Map input features to hidden units to form new features feed to the next layer. The number of weights for each Feature Map [FM] is thus only dependent on the kernel

size and the number of channels in the input to the layer. Shared parameters also introduce shift-invariance, meaning that any given feature map will detect if a certain feature is present anywhere in the input.

1.10.1. Pooling Layer

PPooling layer is one of the differences between ANN and CNN. This layer is also known as downsampling since it works on reducing the dimensions of the layer. Training on a high number of features can lead to several problems like overfitting, therefor pooling is used to boost the effective receptive field of outputs with respect to the input. The pooling layer usually comes after ReLU. The most used pooling method is Maxpooling. The method depends on applying a filter (size 2x2) on the local region where the pooling is applied. This is particularly critical since this module allows to propagate the error with the highest responses so that it decreases the chance of vanishing gradient. Another type of pooling is average pooling which is applied by dividing the data into regions and calculating the average value for each part. Figure (1.13) shows an example of pooling task.

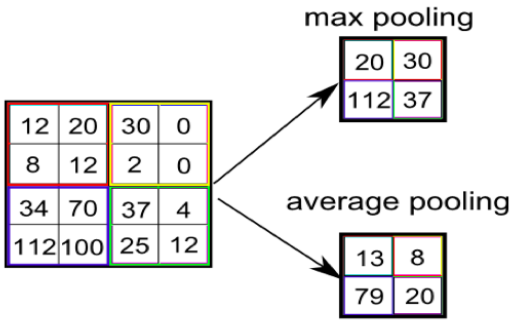


Figure 1.13. Example max pooling and average pooling

1.10.2. Padding

CNN as the name suggests it depends on convolution every time we convolve the image with the filter, the size of the image shrinks beside throwing away a lot of information on the edge. Researches have overcome this problem by symmetrically adding zeroes to the borders of the input matrix allowing the filter to be performed

on all the borders. This method is called zero padding [27]. Figure (1.14) shows an example of zero padding.

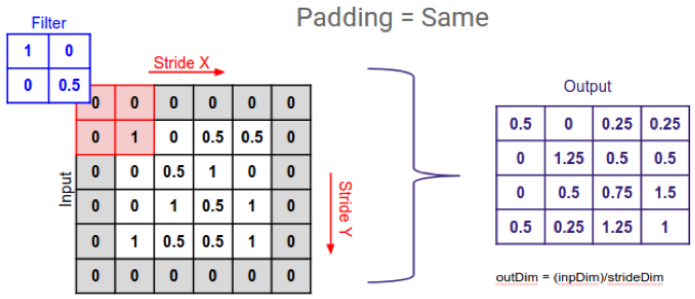


Figure 1.14. Example of zero padding

1.10.3. Overfitting

This term refers to when a model is overtrained on the training data that it is not able to generalize well for the validation and test sets. While producing a good result on training data 99% to 100% but it has low performance on the test data. In this case, the model might have been tuned too well to the specific samples of the training set, rather than learning universal features that can be generalized to unseen data. This performance is very undesirable and there are methods to overcome this problem such as drop-out layers and data augmentation.

Drop-out layers are a way to combat the overfitting by simply drop out random weights and their connections in each training iteration by setting them to zero or ignoring them as shown in Figure (1.15). This step helps the network to constantly learn how to process the data properly from a new set of active neurons in each training iteration, which has significantly decreased the risk of overfitting.

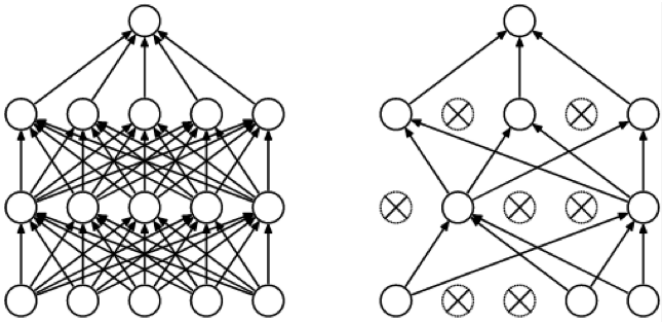


Figure 1.15. Application of drop-out in neural networks. Left: a neural network. Right: a Dropout applied on the network

1.10.4. Deep Supervision

In supervised learning applications, the parameters are updated based on the difference or the loss between the ground truth of the model and the network output. The idea of deep supervision was proposed by Lee.et.al [28] by making the network produces applicable outputs and compute the loss for each output layer then the additional layer is added to produce output shape like the desired output.

1.10.5. Transfer Learning

In deep learning applications finding a large amount of data to train the model is a critical step. The method of utilizing a pre-trained model on a new model with the new dataset called Transfer learning. This pre-trained model will act as a feature extractor by keeping the weights of all other layers unchanged then eliminate the last layer of the trained network and replace it with a different layer specific to the problem space e.g classifier or regressor. This type of training has accelerated the work that involves data scarcity, especially in certain domains such as medicine [29].

1.11. Generative Adversarial Networks

Generative Adversarial Networks (GAN) was introduced by Goodfellow.et.al [30], GAN belongs to generative models, which means that they are capable of generating new images from the input images. Generative models made up of two sub-models. The

first sub-model the generator denoted as G and the second is discriminator D as shown in Figure (1.16). The generator generates a new image by taking in random numbers and proceeds an image corresponding to the input image (x). This image is fed to the discriminator alongside with the real input image, the discriminator evaluates them and decides whether this generated data is real or synthetic by returning a probability value between 0 and 1 where 0 means the image is fake and 1 the image is real. This implies that the generator and discriminator are in a feedback loop and compete against each other. Hence, G is trained to reduce the probability of D identifying the generated images as synthetic while D is trained to maximize the probability of D is correct. The generator and discriminator are defined by equation (1.18):

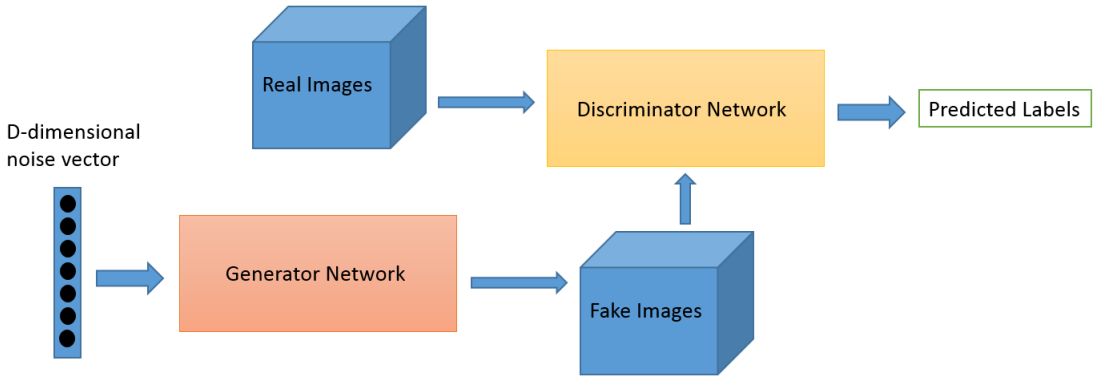


Figure 1.16. GAN Deep Learning Architectures

$$L_G(G_\theta) = -\sum_I \log(D_\gamma(G_\theta(y_i, z_i), y_i) + \lambda||x_i - G_\theta(y_i, z_i)||_1 \tag{1.18}$$

Where G(z) is the generated data and D(x) is the probability of the real input data x. While discriminator D is learned by maximizing its loss function L_D which is given by:

$$L_D(D_\gamma) = \sum_I D_\gamma(x_i + y_i) + \log(1 - (D_\gamma(G_\theta(y_i, z_i), y_i)) \tag{1.19}$$

The competition between the generator and discriminator is tenuous which means if

one of them overpower the other this can lead to network collapse, if the discriminator is performing in a good way by returning probabilities value close to 0 or 1. Therefore, the generator will struggle to trick the discriminator and read the gradient. On the other hand, if the generator is performing very well this will lead to weaknesses in the discriminator which will result in a false negative. Therefore the two sub-models must have a parallel "skill level" [30].

Figure 1.17. Example of samples generated by Generative Adversarial Networks

GANs are now a very dynamic topic of research and there have been numerous types of GAN implementation Figure(1.17) shows an example of samples generated by Generative Adversarial Networks (GAN) after training on two datasets: MNIST and TFD. Some of the important ones that are actively being used currently are mentioned in the next section.

1.11.1. Deep Convolutional Generative Adversarial Networks

Deep Convolutional Generative Adversarial (DCGAN) Networks is one of the most discussed implementations of GANs. Radford.et.al [31] proposed DGAN to boost the quality of generated images and the training stability by integrated GAN and (FCNN) architectures. The resulting architecture is shown in Figure (1.18).

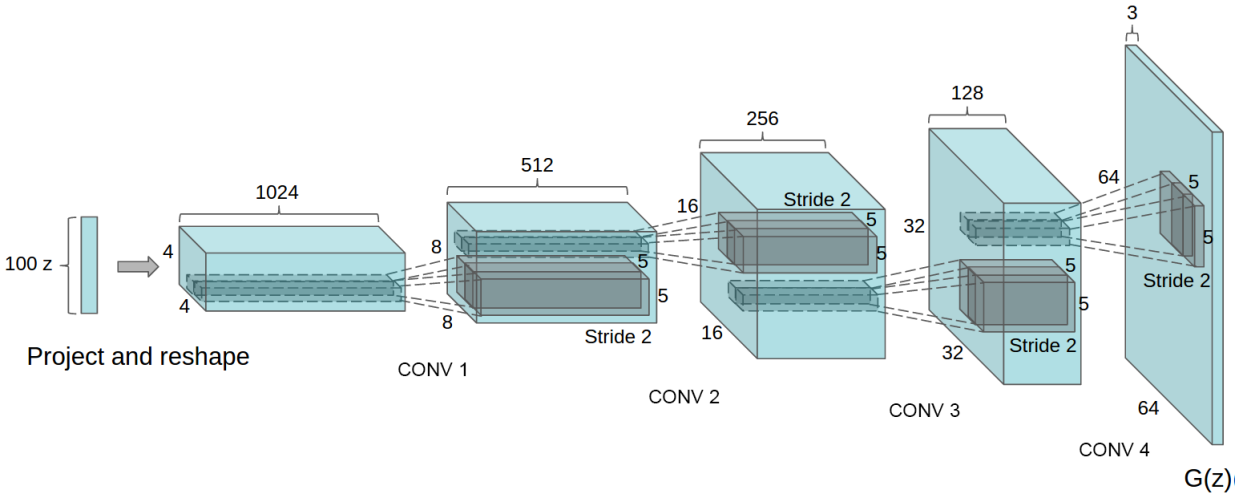


Figure 1.18. DGAN fully convolutional generator architecture

Some changing on FCNN has been introduced in [6] to enhance the stability of training by changing pooling layers to stride and to make the generator and discriminators fully connected fractionally stridden is used instead of up-sampling.
Additionally, adding activation functions and batch normalization to FCNNs's layers; but the last layer of the generator and the first layer of the discriminator [32] shows a significant improvement in the learning stability.

1.11.2. StackGAN

The main idea of StackGAN is to generate high quality images from text descriptions. The network consists of two stages [15]. In stage 1, the fundamental colors and shape of the images based on the text information are sketched, which produce low-resolution images. In the following stage, text description and images from the first stage are taken as inputs yielding to produce high-resolution images. Example of images generated by StackGAN is shown in figure (1.19).

Figure 1.19. Example of images generated by StakGAN

1.11.3. SimGAN

The application of SimGAN is linked to the approach followed in this project. The aim of SimGAN is to mimic simulated image data which their labels are given [24] and to improve the quality of the generated image data while maintaining annotation information without human interaction.

SimGAN achieved a state-of-the-art result in classification, SimGAN architecture is represented in Figure (1.20)

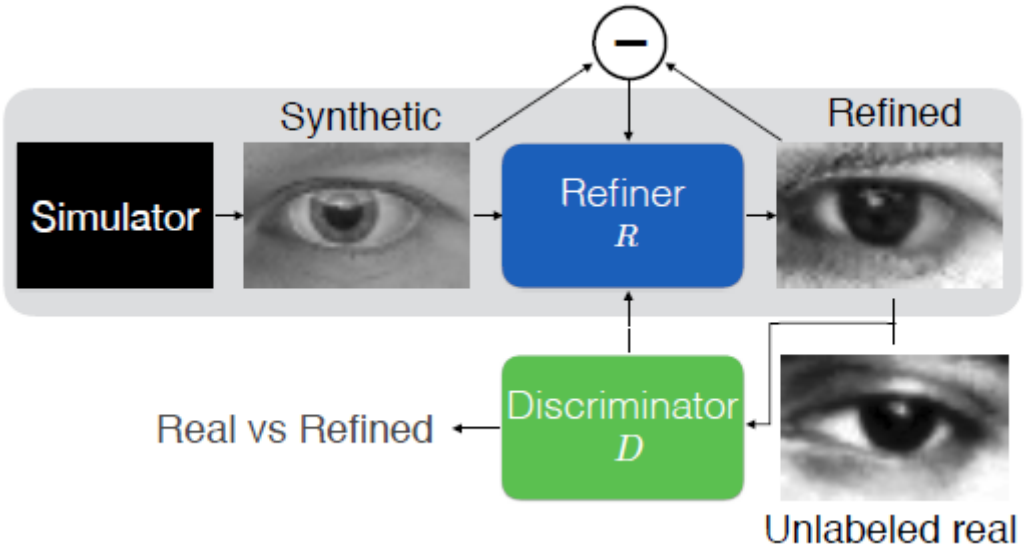


Figure 1.20. Example of images generated by SIMGAN

The major modification of SimGAN compared to GAN is finding a solution of artifacts in the network which over-emphasize specific image characteristics to deceive the discriminator network. SimGAN overcomes this problem by using local adversarial

loss. The discriminator has been designed to divide the output image to patches of probability map, then the local adversarial loss function calculated the loss by summing the cross-entropy losses over the local patches. The second modification is using the self-regularization loss to reduce the feature transformation of the real and synthetic images. Where feature transformation is an algorithm that uses existing features to create new features. These new features may not have the same interpretation as the original features, but they may have more discriminatory power in a different space than the original space.

1.11.4. Wasserstein Generative Adversarial Networks

The problem with traditional GANs is the loss function which does not correlate with image quality, therefore, an inspection of the images samples is required to check the quality of the generated images. Additionally, there is no guarantee for convergence, therefore, there is no indication when the training should be stopped. Wasserstein GAN uses Wasserstein distance [33] which calculate the distance between the real groundtruth distribution p_R and the model output distribution p_G using an informative and interpretable loss function that gives convergence correspond to image quality.

1.12. Evaluation Metrics

Several matrices can be used to evaluate how the model operates on the segmentation task. As for each input, the network predicts the class of each pixel, afterward the output parallel against the ground truth. There are four possible results for this comparison:

TP: True Positive, predicted to be a positive sample.

FP: False Positive, incorrectly predicted to be positive.

TN: True Negative, predicted to be a negative sample.

FN: False Negative, incorrectly predicted to be negative.

When the number of classes is more than two classes "multi-class classification", the occurrences of the true positive, the true negative, etc must be calculated separately for each foreground class. Evaluation metrics are built on a number of measurements

of the different consequences, each giving different insights into the performance of the prediction model. Some of the measurements are accuracy, recall, and specificity:

1.12.1. Accuracy

Accuracy is calculated as the number of the true predictions divided by the number of the data in the dataset. Accuracy ranges between 0 and 1. Value of 1 indicates a perfect prediction and 0 means the worst prediction. Accuracy is determined by the following equation:

$$Accuracy(\hat{y}, y) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.20}$$

Where $\hat{y}$ is the prediction result of the ground truth y. For classification on tasks where the number of the elements in the positive and negative classes are equal, the performance of accuracy will be meaningless.

1.12.2. Sensitivity or Recall

Sensitivity or Recall is determined as the number of true positive predictions divided by the complete number of positives.

$$Sensitivity = \frac{TP}{TP + FP} \tag{1.21}$$

Sensitivity is a measure of the rate of pixels classified as positive actually being true positives. However sensitivity alone can not give an insight of the model validation, as the model can achieve high sensitivity by classify all the pixel as foreground. Therefore,

To have a high degree of certainty about the model performance a negative prediction must be taken.

1.12.3. Specificity

Specificity measures the ratio of correctly classified negatives. It can be defined as the sensitivity for the background pixels and it is determined as the number of right negative predictions divided by the total number of negatives.

$$Specificity = \frac{TN}{TN + FN} \tag{1.22}$$

1.12.4. Dice Similarity Coefficient

Dice Similarity Coefficient (DSC) is a similarity measure commonly used in segmentation tasks, specially in medical image segmentation [34], as the accuracy is biased on background labels, specificity and sensitivity do not give a clear insight of the model validation. Dice measures the similarity between two pixel by computing the ratio of correctly classified foreground features to all positive predictions plus false negatives as:

$$DSC(\tilde{y}) = \frac{2TP}{2TP + FP + FN} \tag{1.23}$$

Using DSC as a loss function can lead to better gradients and increase the performance by motivating the model to predict some elements as positive [35].

1.13. Medical Image Segmentation Methods

Image segmentation is known as dividing an image area into homogeneous, constituent, non overlapping regions to detect boundaries [11]. Segmentation is one of the essential techniques of medical image processing. The importance of segmentation in medical images came from the fact that it provides noninvasive information about human organs. As such a various application of segmentation used in the medical do-

main such as simulation of surgeries, tumor detection, detection of vessels [36], heart segmentation diagnosis [37], and localization of pathology [38] and computer-guided surgery [39]. On the other hand, the low contrast of medical images can leads to noise or signal variation in the images and make the segmentation task challenging [40].
To overcome these challenges, multiple techniques have been developed. The commonly used methods are thresholding, region growing, deformable models, clustering, Markov random field (MRF) models, classes, artificial neural networks, and atlas-guided approaches [41,42] . There is no technique work for all application, most of these techniques used composed for accurate segmentation.

1.13.1. Region-based Methods

- Thresholding

  It is the simplest segmentation method and considered as first generation technique. Thresholding approach is based on partitioning the image intensities into binary values depending on the thresholding level. The thresholding level is a histogram value that partitioning the intensity into background and foreground [43]. The thresholding can be settled for the inter-image (global thresholding) or modify depending on pixel location as shown in Figure (1.21). Due to its limitation such as spatial characteristics of an image is not taking into account, thresholding is susceptible to noise and intensity, which often happen in medical images. Despite that, currently, thresholding is used as one of the main steps in image preprocessing.

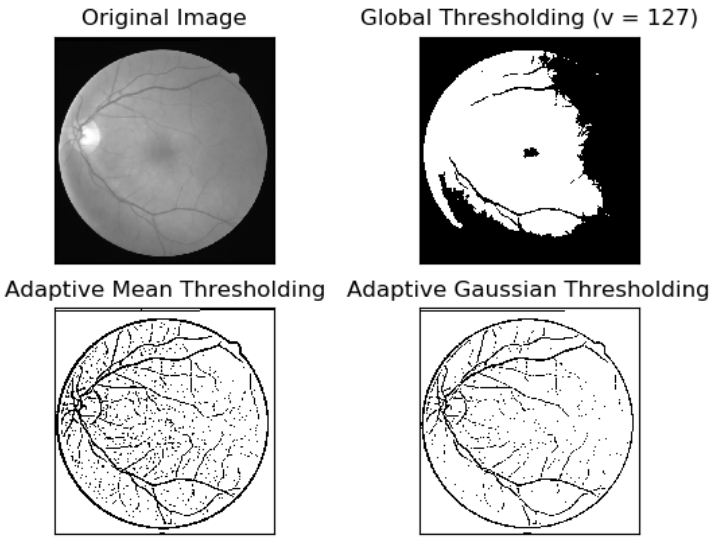


Figure 1.21. Four types of thresholding applied on image from DRIVE dataset

- Region Growing

  Region growing is an interactive segmentation technique [44]. This approach requires one seed point to start the process of growing around the region based on its homogeneity properties and according to neighboring pixels [45]. Region growing approach is sensitive to noise and requires a manual setup for seed point for every region. Region growing is usually used in conjunction with other segmentation methods to define the desired regions. Edge Detection is another region based method. This technique is based on the region gray level and identifies the outlines of the image where image brightness changes sharply. Currently, this technique has been used as a basic tool for segmentation. There are multiple edge detection filters like Roberts edge detection, canny edge detection, Prewitt Edge Detection [46]. Figure (1.22) shows an application of Canny edge detection.

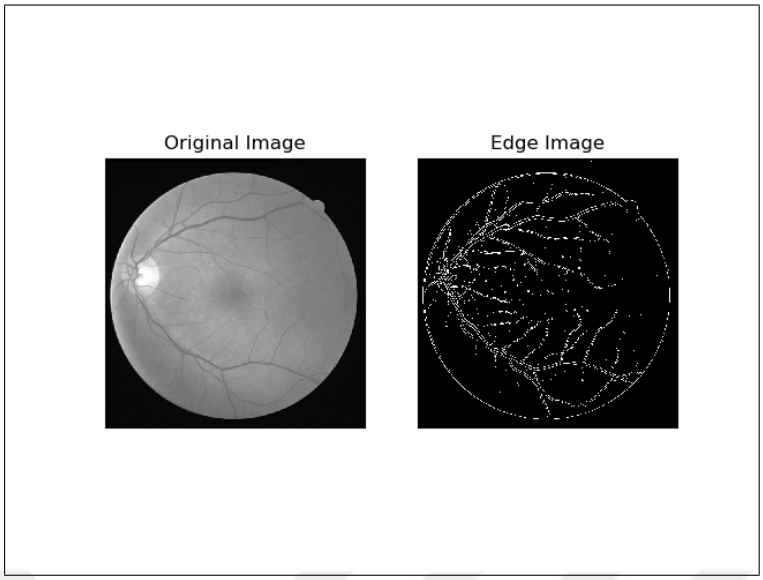


Figure 1.22. Canny edge detection applied on image from DRIVE dataset

1.13.2. Pattern Recognition Methods

Pattern recognition methods can be divided into two groups: classifiers, and clustering methods. Classification methods are supervised techniques that required training data and ground truth, while clustering does not need training data to segment the image.

1. Classification is a method seeks to disunite image intensities by utilizing training data, where the training data are manually segmented. The classifiers can be non-parametric classifiers or parametric classifiers. Non-parametric classifiers are not predicated on statistic structure and the common classifiers are KNN (K-nearest neighbor) and Pazrian window [47].

2. Clustering does not require data training. It is usually used when the classes are known in advance, and then similar pixel grouping in a cluster and train themselves on the obtained training data. Frequently used clustering algorithms are the fuzzy c-means algorithm and K-means algorithm [48], and the Expectation Maximization (EM) algorithm [49].

1.13.3. Active Contour Models

Active contour is widely used in image segmentation and it is defined as the use of energy forces and constraints for separating the vessels from the images based on the means of the two regions. Active contour models include gradient vector flow snake mode, snake model, and geometric or geodesic contours [50].
Active contour Models are applied to extract vessels in clinical images. It is the most used technique in scientific images segmentation because of it's far dominance over other segmentation techniques the maximum common techniques are deformable or snakes, the snakes model takes two shapes if the model is 2-D digital image it will be as a curve and if the images is 3-D the it will shaped as surface [51].It delineates region boundaries using closed parametric curves or surface that deforms beneath the influence of internal and outside forces to outlines the preferred segmented item edges in an image. The initial step to place a curved surface near the target boundary and then the inner forces maintain the smoothness all through the deformation, whilst the external forces drive the contour towards the target limitations.

1.13.4. Combination of Image Segmentation Methods

Accurate segmentation is one of the essential steps toward the right diagnosis in medical images. Due to the fast growth in medical image modalities, one segmentation method cannot provide an accurate segment. Therefore a combination of various segmentation methods is introduced in several approaches. Although those approaches bring more accurate results, the segmentation often becomes too complex and time-consuming. Baillard.et.al [52] presents combination of segmentation methods to segment brain structures by mingled Bayesian pixel classification and active contour models.

1.14. Related Work

The goal of vessel segmentation is to partition vessel pixels from non-vessel pixels in the image. Therefore segmentation is a pixel-wise method where the output image of

the segmentation operation is a labeled image detected by the segmentation algorithm. The goal is to predict an output pixel which has the correct labeled region. Due to its clinical importance, a lot of researches have been attached to the topic. These methods can be divided into two categories: model-based (unsupervised methods) and machine learning (supervised methods).

1.14.1. Model Based Approaches

Model-based approaches or unsupervised approaches do not require a training set for segmentation. The simplest form of model-based approaches is thresholding [53] where the pixels divided into foreground and background. If the input pixel is larger than a specific threshold, it belongs to the foreground class, else, it is background class. Generally, this approach only works for very easy tasks, where the regions of interest share a common pixel intensity range.

Other model-based approaches are Hessian-based techniques [36] or to incorporate the eigenvalues [36]. The problem with these techniques is that they can be tricky with irregularly shaped vessels. Another approach for unsupervised learning was proposed by Nguyen et al [54]. The approach is to filter the image to amplify and enhance the vessels images along lines with different direction. With input image contains green channel multiple filters are connected together to produce a response with enhanced vessel contrast. Based on isotropic un-decimated wavelet transform (IUWT) Bankhead.et.al [55] proposed an algorithm with binary image by calculating the threshold then applying post-processing techniques to fill the puncture in the vessel and eliminate the undesired objects.

Azzopardi.et.al.[56] proposed a method of vessels segmentation by using a combination of a shifted filter and responses filter (COSFIRE). By filtering the image and then using the responses filter to find the line patterns then combined the average of the lines and mark it as the vessels ending. Then, the optimal segmentation obtained by threshold.

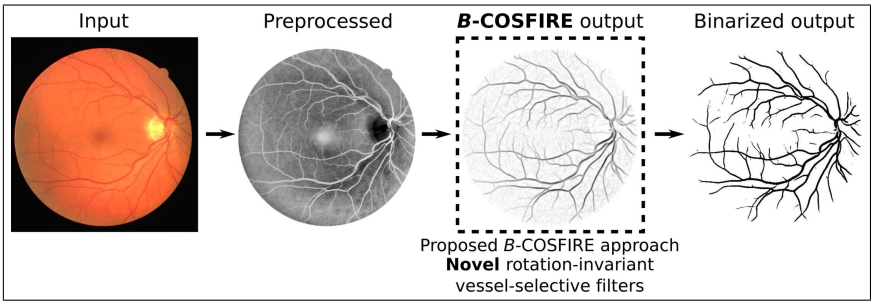


Figure 1.23. COSFIRE filter

1.14.2. Supervised Methods

Apart from unsupervised methods discussed above, supervised methods showed outstanding results that overcome human level in some application such as retinal fundus segmentation. Becker et.al [57] proposed an approach to classify retinal fundus where the classifiers learn automatically from images by optimizing a gradient boosting example of image segmented by this approach shown in Figure (1.24). Sofka et al. [58] extract the center-line of the vessels by using edge measures and multi-scale filtering. This approach focused on the segment of the narrow vessels by detecting low contrast parts of the image.

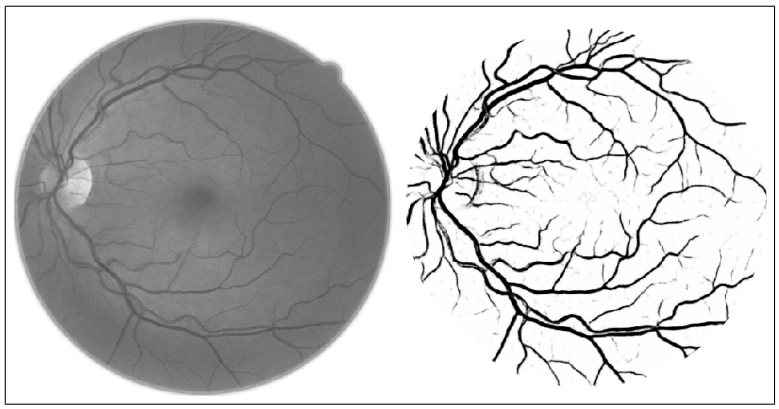


Figure 1.24. Example image and mask obtained by Kernel boost approach

Different classification algorithm proposed by Soares.et.al [59] which is based on using wavelet filter output as classification feature. The Gaussian mixture model (GMM),

the k-nearest least mean square error and the k-nearest neighbours (k-NN) used as classifiers. The disadvantages of this method are bad tolerance of mixed light around the vessels and the long training time.

We can not talk about deep learning approaches in segmentation without mentioning Unet which was proposed by Ronneberger et al [3]. Unet used to skip connection network specially for biomedical image segmentation and the limitation of labeled images. A variation of Unet architecutre was proposed by using a pre-trained network as encoder [60]. The network was adopted from visual geometry group (VGG) networks, which motivates this work to consider Unet network for the segmentation task. Figure (1.25) shows an illustration of the basic Unet network

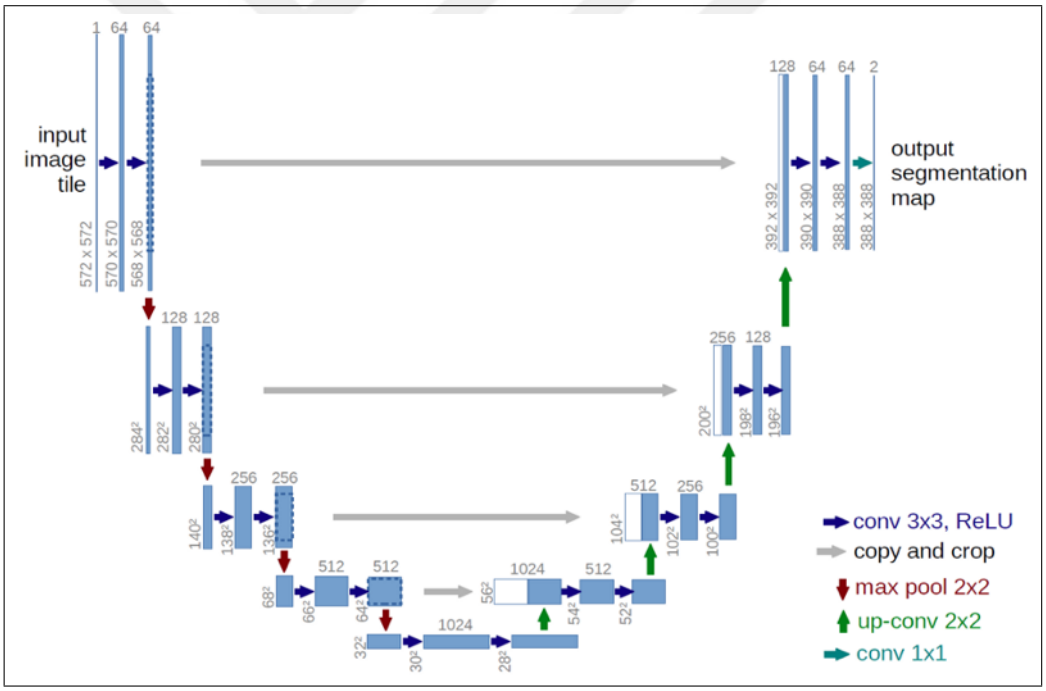


Figure 1.25. U-Net style encoder-decoder network architecture.

Maninis.et.al [61] used a deep (CNN). The network structure consists of two sets of a specialized layer and known as DRIU as shown in Figure (1.26). IIt is used to segment the optical disc and the retina. DRUI achieve a state of art results on DRIVE and STARE datasets by using area under the curve evaluation (AUC) 0.831 for STARE dataset and 0.822 for DRIVE dataset.

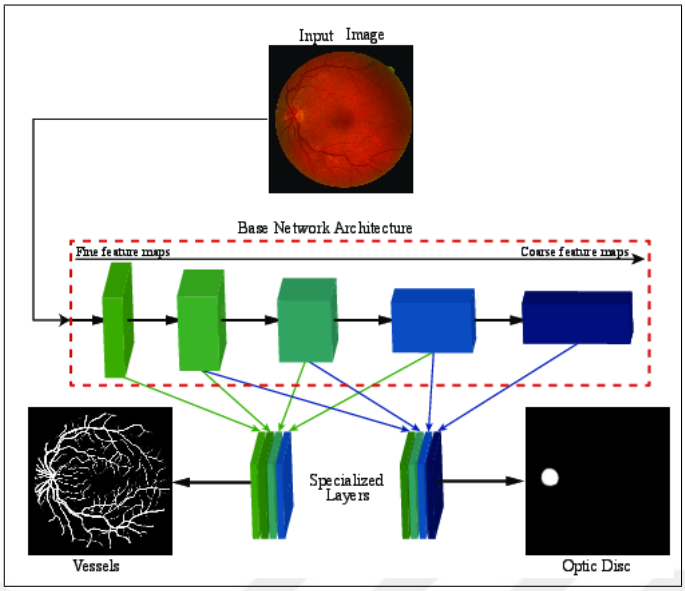


Figure 1.26. Base Network architecture of DRIU

In context of deep learning Maji.el.al [62] build a deep learning network containing twelve convolutional neural layers to split vessels from non-vessels pixels. The network did not achieve high accuracy (0.94) but the model was beneficial in terms of learning vessel structure from the data. Salem.et.al [63] proposed a novel enhancement on the k-mean clustering algorithm based on green channel intensity, where each pixel represented by a feature vector and local maxima of the largest eigenvalue and gradient magnitude. Sensitivity and Recall used to evaluate the performance resulting in sensitivity and specificity of 77% and 90% respectively on STARE dataset.

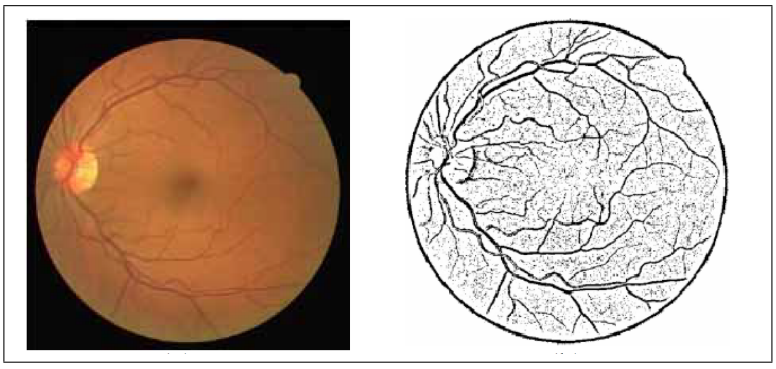


Figure 1.27. Image segmented by fuzzy c-mean clustering and genetic algorithm

Akhavan and Fae [64] proposed a two stages method for vessel tracking. The first step is tracking the vessel by detecting the center-line whereas the second stage was filling the gaps in the vessels using fuzzy c-mean. The accuracy of this method is 75% on DRIVE dataset and 77.6% on STARE dataset. Xie and Nie [64] proposed a novel approach based on genetic algorithm and fuzzy c-mean clustering. The approach starts with a pre-processing step to enhance the histogram equalization, then the image split into two layers. The first layer is the texture layers and the smoothing layer. An example of the result from this approach is shown in Figure (1.27)

2. DATA AND METHODS

In this chapter, we will present modified approach of [14] which is based on the state of ate Generative Adversarial Networks (GANs) by Ian Goodfellow.et.al [30].GAN architecture produces just images, without their corresponding groundtruth. In this project, we use a modified version of the GAN to generate synthesize images and their segmented pair. Two pipelines are proposed; first, we generate images and their masks using GAN and then using these images as training dataset beside the original images to satisfy the need for a large amount of the training data in deep learning networks. Furthermore, we present a standard data augmentation techniques and pre-processed techniques in this project.

2.1. Dataset

The segmentation methodology is evaluated using the benchmark DRIVE dataset as a training set. The DRIVE [5] contains 40 colored images and their corresponding segmented pair. Out of the 40 images, 20 images for training, and 20 for testing. Two sets of manually segmented masks are provided in the data-set and we decided to choose the first one as ground-truth to evaluate the model. Each image has size of 584×565 in tif format for training and gif format for the testing set. An example of image form the dataset can be seen in Figure (2.1). The images were taken by Canon CR5 non-mydriatic 3-CCD camera with a 45° field of view (FOV). The other dataset used for the evaluation is STARE dataset. STARE dataset [4] has 10 labeled fundus images of size 605×700, the images were captured using TopCon TRV-50 camera with 35° FOV.

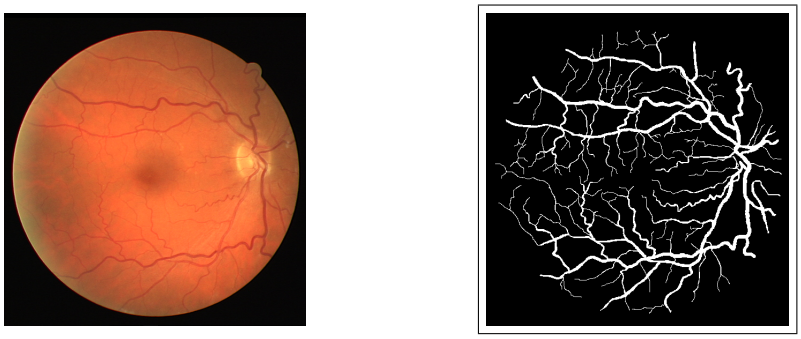


Figure 2.1. Example from DRIVE dataset. Left: an image entry. Right: a segmentation mask

2.2. Preprocessing

Generally, the retinal image has imperfections such as poor contrast and noise that need to be decreased or removed before classification is performed. In the center of the retinal images, there is a bright area called light reflex. For better segmentation result, it is helpful to get rid of this area by converting the image to the gray-scale as shown in Figure (2.2). The images in DRIVE are of size 584×565. As a first pre-processing step, we resize them to 512×512 via bicubic interpolation. The data generated using standard data augmentation and augmentation using GAN are resized to the same size.

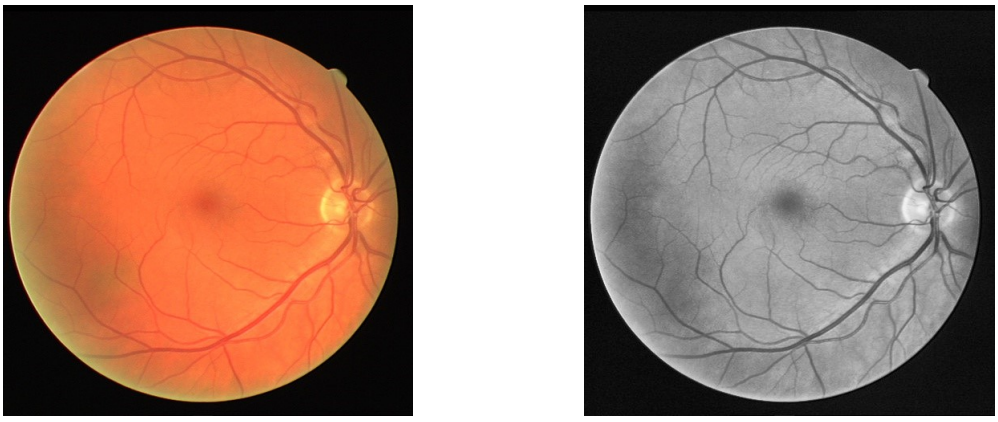


Figure 2.2. Left: Image before pre-processing. Right: After pre-processing

To keep the data consistent, images are normalized to $[-1, 1]$ followed by applying contrast-limited adaptive histogram equalization (HE), which is unlike the histogram

equalization, enhances the local edge definitions and the lightness on each region of an image.

2.3. Standard Data Augmentation

Data augmentation refers to different methods of extending the available data. By doing natural manipulations on the data, more variation can be represented which gives the network better conditions to learn important features. In addition to preventing the model from over-fitting on a training dataset. Given that our training dataset has only 20 images for training. In this task, it is crucial to extend the dataset so the deep learning model can learn the features without overfitting. Therefore data augmentation is applied. Mostly data augmentation is done online but in our method, we applied the standard augmentation online to speed up the training time.

2.3.1. Additive Noise

Adding noise to the training data from a specific distribution like Gaussian noise before feeding it to the deep learning model helps the model to tolerant uncorrelated in pixel values as shown in Figure (2.3). In this example a Gaussian noise of standard deviation of 0.3 added to the training image.

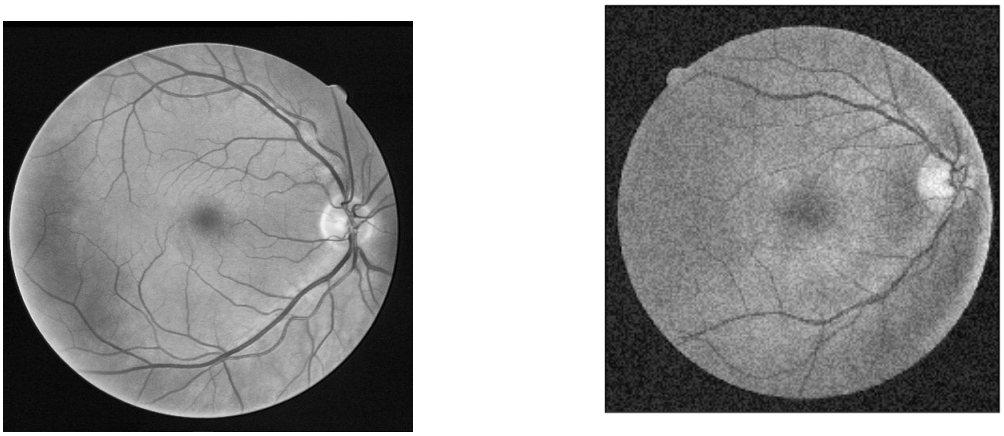


Figure 2.3. Example of additive Gaussian noise. Left: Base image Right: Image augmented with Gaussian noise

2.3.2. Random Rotation

Random rotation is a geometric coordinate transformation applied to image training set to extend the size of the set. In our task, we applied random rotation on the training images in every minibatch, where every image inside the minibatch is rotated randomly with the 45-degree angle as shown in Figure (2.4).

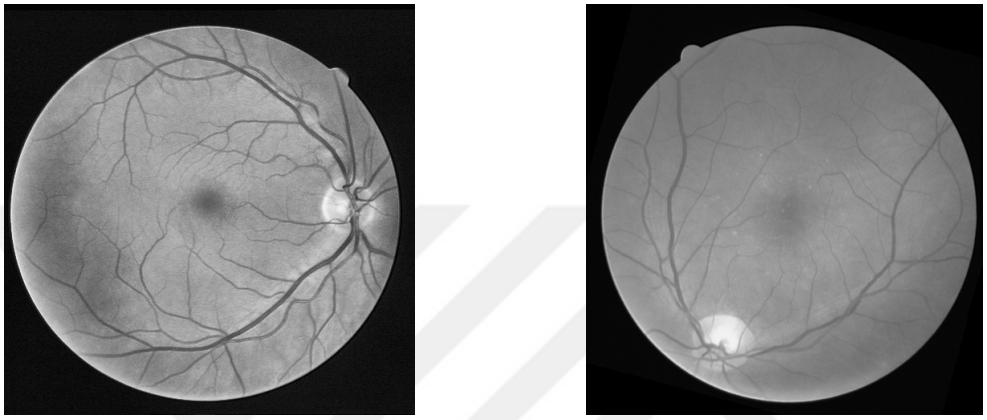


Figure 2.4. Example of random rotation for data augmentation. Left: Base image. Right: Image augmented with random rotation at angle 75

Random rotation can be a very useful technique for making the neural network invariant to rotation.

2.3.3. Intensity Shift

Intensity Shift resembles of a basic brightness transformation for the image. It is performing by producing element-wise addition to the image so that negative values result in darker images while the positive value leads to bright images. We add a positive value to the training images as shown in Figure (2.5).

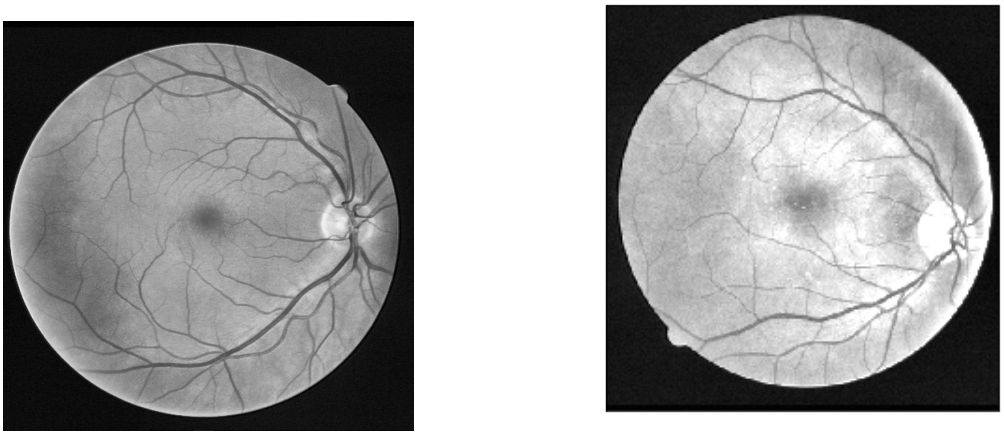


Figure 2.5. Example of intensity shift for data augmentation. Left: Base image. Right: Intensity shift

This intensity shift technique is particularly useful to improve generalization when varying brightness presented in the test data.

2.3.4. Random Cropping

In random cropping, each image is cropped to a smaller size. This technique is very useful when dealing with a small training dataset. Random cropping forces the network to generalize therefore the network will not only learn from specific features at specific positions. Another positive effect of cropping is saving memory by reducing the size of the image. Figure (2.6) shows an application of cropping applied to the training dataset.

Figure 2.6. Application of cropping on DRIVE dataset. Left: base image. Right: cropped image to 128×128

For the validation set, multi-cropping is applied by cropping the images multiple times so the whole image is covered by crops. The full-size image is assembled by emerging the cropped images.

2.3.5. Random Flip

Random flipping can be performed around the axis, horizontal or vertical. In this project, both of flipping are used on the training images as shown in Figure (2.7).

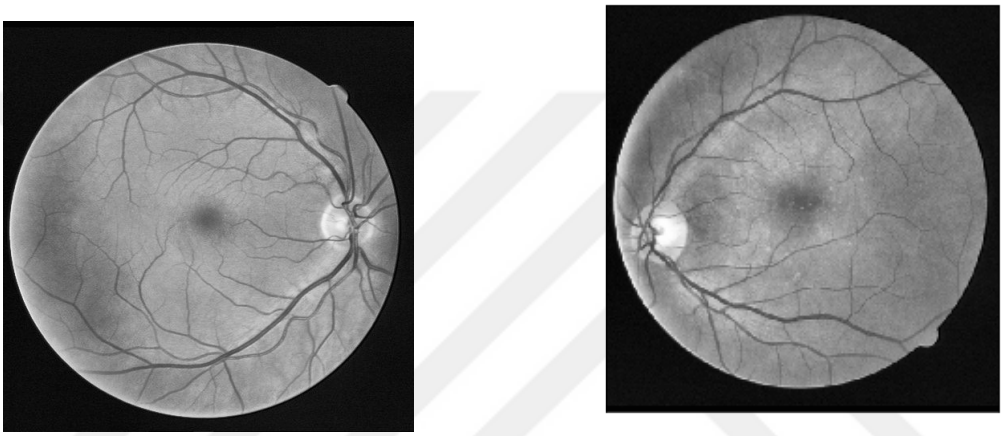


Figure 2.7. Application of cropping on DRIVE dataset. Left: base image Right: image flipped horizontally

Random flipping is a very useful technique when the test data contains natural images or faces.

2.3.6. Elastic Deformation

Elastic deformation is a very common and useful technique for medical images, as an adjusted spline transformation can improve the approximation of anatomical variation compared to very simple transformations[40]. An example of our implementation of elastic deformation on training dataset is shown in figure (2.8).

Deposit the fact that elastic deformation is useful, it should be used in a proper way since it is very computationally intensive and increases the training time when it is used in online augmentation.

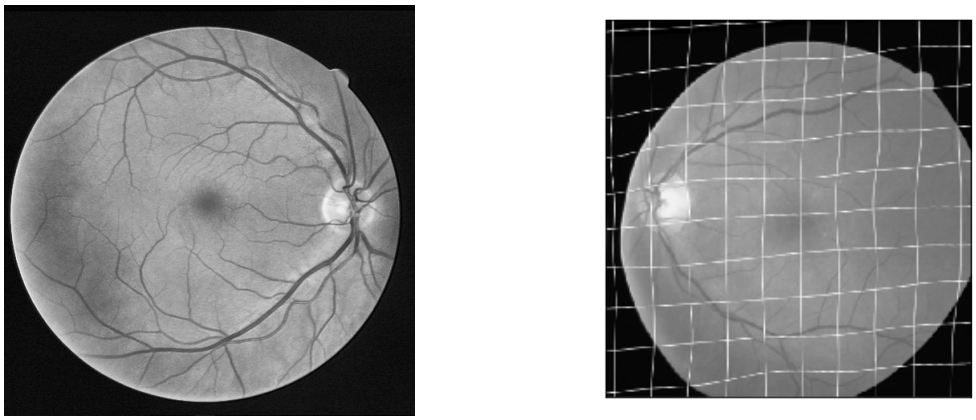


Figure 2.8. Application of elastic deformation on DRIVE dataset. Left: base image. Right: elastic deformation applied on the image

2.4. Data Augmentation Using GAN

In addition to using standard data augmentation techniques, we applied data augmentation using GAN based on the approach proposed by [14], as mentioned in section [2.8] GAN generates only train images without their corresponding masks. The adaptive approach in [14] is based on generating image-mask pair by modifying the generator to generate two images instead of one. This is done by changing the last convolution layer in the generator network. Likewise for the discriminator the layers need to be changed to accept two images, therefore, the first convolution layer is changed to a number of channels equal to the number of channels of the image-segmentation pair. Figure (2.9) GAN architecture for generation image-segmentation pairs.

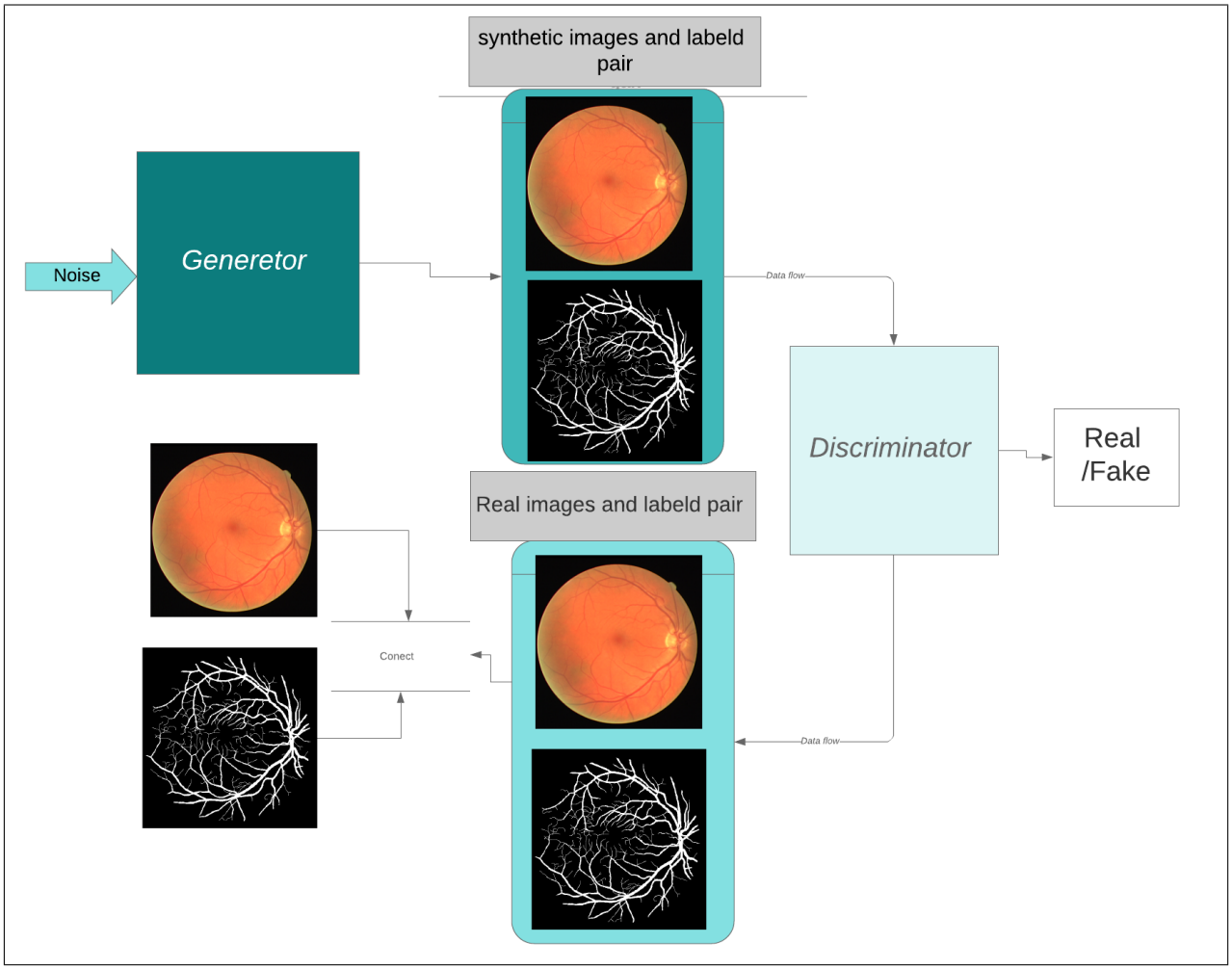


Figure 2.9. GAN Architecture for generation image-segmentation pairs

2.5. Network Implementation

The focus of this thesis is on data augmentation to increase the training datasets of medical images. In this section, a detailed description of the training process is included with choices of the loss function, and the optimizer.

2.5.1. Pytorch Framework

The programming in this project was carried by Pytorch [66], which is a deep learning framework whose name is inspired by the Torch deep learning framework, Pytorch offers two essential characteristics; dynamic computation graphs and imperative programming. The dynamic computation graphs are built at run-time and allow us to use

standard python statements. The second feature is imperative programming dynamic therefore there is no distinction separating defining the computation graph and editing for crucial programs.

2.5.2. GAN Implementation

In our experiment GAN is applied to generate more training images. It is programmed using WGAN training method unlike the original implementation in [67], we used two different optimizers for the generator and the discriminator. For the generator Adaptive Moment Estimation (Adam) is used whereas for the discriminator SGD has been used. Additionally, we increased the image resolution and dimensions to 512×512. To enhance GANs output, we added more layers to handle the high resolutions and also batch normalization was added to the network layers other than the last layer of the generator and the first layer of the discriminator. Detailed information about the generator and discriminator architecture of the proposed approach is presented in tables (2.1, 2.2). We train the generator twice the discriminator to reach faster convergence and balance the learning process. A Gaussian noise has been added to the input of the generator which has been beneficial to preserve a proper level of diversity since we have a small dataset. We trained the network for 3000 iterations and during the training process we kept a track of minimal loss for both generator and discriminator as well as the other hyperparameters. For the activation functions, after every convolution layer, a leaky ReLU is used, expect on the last layer where sigmoid activation function is used instead of Hyperbolic Tangent (tanh).

Table 2.1. Detailed information about the generator architecture of our GAN

DRIVE Database Generator Architecture							
Minibatch Size:8 , Optimizer: Adam (learning rate = 0.001) The input noise vector of dimensionality 512 was drawn from a zero-mean Gaussian distribution with unit variance.							
	Input Noise	Generator1	Generator 2	Generator 3	Generator4	Generator 5	Generator 6
	Fully-Connected	Strided Convolution	Strided Convolution	Strided Convolution	Strided Convolution	Strided Convolution	Strided Convolution
Input Dimension	[1x512]	[4x4x2048]	[8x8x1024]	[16x16x512]	[32x32x256]	[64x64x128]	[256x256x64]
Output Dimension	[4x4x2048]	[8x8x1024]	[16x16x512]	[64x64x128]	[128x128x64]	[256x256x64]	[512x512x2]
Number of Kernels	-	1024	512	256	128	64	2
Kernel Size	-	5	5	5	5	5	5
Stride	2	2	2	2	2	2	2
Padding	1	1	1	1	1	1	1
Activation	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Sigmoid
Batch Normalization	yes	yes	yes	yes	yes	yes	no

Table 2.2. Detailed information about the discriminator architecture of our GAN

DRIVE Database Discriminator Architecture							
Minibatch Size:8 , Optimizer: Adam (learning rate = 0.001) The input noise vector of dimensionality 512 was drawn from a zero-mean Gaussian distribution with unit variance.							
	Discriminator 1	Discriminator 2	Discriminator 3	Discriminator 4	Discriminator 5	Discriminator 6	output
	Convolution	Convolution	Convolution	Convolution	Convolution	Convolution	Convolution
Input Dimension	[512x512X2]	[128X128X32]	[64x64x64]	[32x32x128]	[16x16x256]	[8x8x512]	[4x4x1024]
Output Dimension	[128X128X32]	[64x64x64]	[32x32x128]	[16x16x256]	[8x8x512]	[4x4x1024]	1
Number of Kernels	32	64	128	256	512	1024	-
Kernel Size	5	5	5	5	5	5	-
Stride	2	2	2	2	2	2	-
Padding	2	2	2	2	2	2	-
Activation	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	-
Batch Normalization	no	yes	yes	yes	yes	yes	yes

2.5.3. Segmentation Implementation

After generating images and corresponding masks using GAN, we used the generated images and the original training images from DRIVE dataset for training and we train them using the modified version of UNet [3]. The UNet network consists of an encoder with 83X3 layers and to rebuild the image back the decoder initiated with 7 layers. The encoder partly used VGG11 [68], and pre-trained on ImageNet for image segmentation using VGG11 as encoder can increase the training compared to the original UNet. The architecture of the original $VGG11 + Unet$ is shown in figure (2.10).

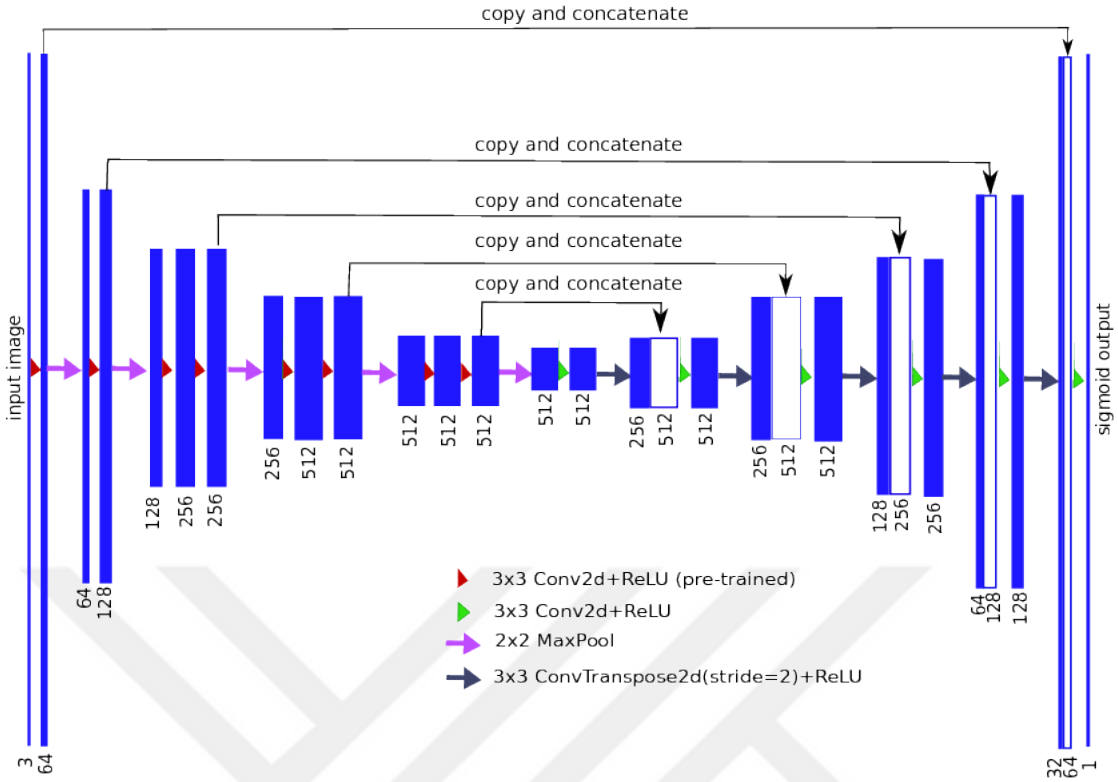


Figure 2.10. Diagram of Vgg11 pre-trained encoder for Unet

To improve the model numerical stability and increase the robustness of training, we added a batch normalization layer after each convolution which is a distinct to the original approach [68]. The architecture of the VGG+11 U-Net based segmentation network trained on DRIVE Database illustrated in figure (2.11) and described in details in table (2.3). To evaluate the segmentation performance, we used training dataset (generated images and the original images) and split it into training and validation set by $1:9$ ratio for validation and training. The model was trained for 500 epochs while keeping tracking of the training loss and validation loss, a Dice loss was used as loss function mentioned before and Adam optimizer with a learning rate of 0.001.

Table 2.3. DRIVE Database Segmentation Network Architecture

DRIVE Database Segmentation Network Architecture								
Minibatch Size:8 , Optimizer: Adam (learning rate = 0.001)								
	Encoder 1	Encoder 2	Encoder 3	Encoder Feature	Decoder1	Decoder2	Decoder3	Output
	Conv2D+ BatchNorm+ ReLU	Conv2D+ BatchNorm+ ReLU	Conv2D+ BatchNorm+ ReLU	Conv	TransposeConv2D+ BatchNorm+ ReLU (Concatenate)	TransposeConv2D+ BatchNorm+ ReLU (Concatenate)	TransposeConv2D+ BatchNorm+ ReLU (Concatenate)	11 Conv2D
Input Dimension	[512x512x1]	[256x256x64]	[64x64x64]	[32x32x64]	[32x32x64]	[128x128x64]	[256x256x64]	[512x512x64]
Output Dimension	[128x128x64]	[64x64x64]	[32x32x64]	[32x32x64]	[64x64x64]	[128x128x64]	[256x256x 64]	[512x512x2]
Number of Kernels	64	64	64	64	64	64	64	2
Kernel Size	3	3	3	3	3	3	3	3
Stride	2	2	2	2	2	2	2	2
Padding	1	1	1	1	1	1	1	1
Activation	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Sigmoid
Pooling	[2x2] Max Pooling	[2x2] Max Pooling	[2x2] Max Pooling	-	-	-	-	-

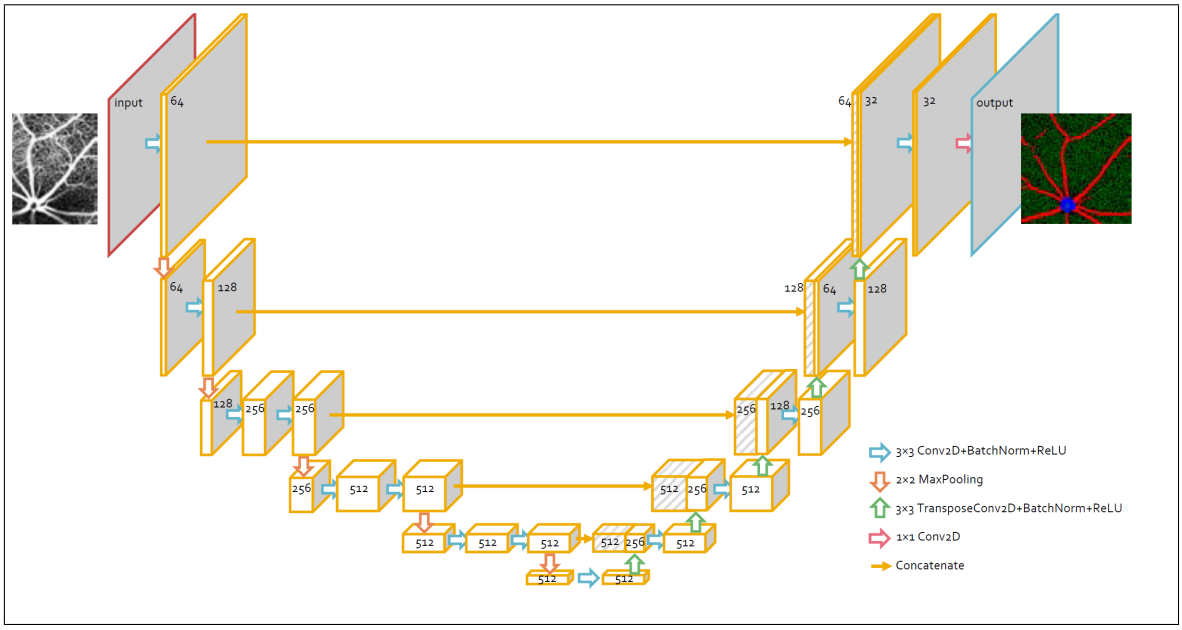


Figure 2.11. A Modified version of VGG11+Unet used in this thesis

3. RESULTS AND DISCUSSION

In this chapter the results of the proposed method are presented. Firstly, the result of the generated images using GAN is described, followed by the results of segmentation performance.

3.1. Run Time

The project were carried on a computer with CPU $i7$ core and with the free GPU 1xTesla $K80$ from Google Colab [1] with $12GB$ memory. Generating an image of a size 512×512 on GAN took 1.5 seconds and training one epoch took 339.7 seconds. GAN was trained for 3000 iterations which took approximately 16 hours to generate a realistic looking images.

3.2. GAN Network Training

In training GAN network, the generator learning rate was set to 10^3 and it was trained twice before training the discriminator, while the learning rate for the discriminator set at 10^{-4} as this balance the learning processes and the min max between the generator and the discriminator.

As previously mentioned the network was trained for 3000 epoch and we kept tracking the loss function while checking the images quality. In this process, we were able to extend the dataset size by 80 images with near realistic quality. Examples of the images trained on GAN are illustrated in Figure (3.1).The images in the Figure(3.1) represent images from our GAN which trained on the DRIVE Database. Even columns represent the generated segmentation masks or labels, while the odd columns represent the generated images.

[1] Google Colab: https://colab.research.google.com

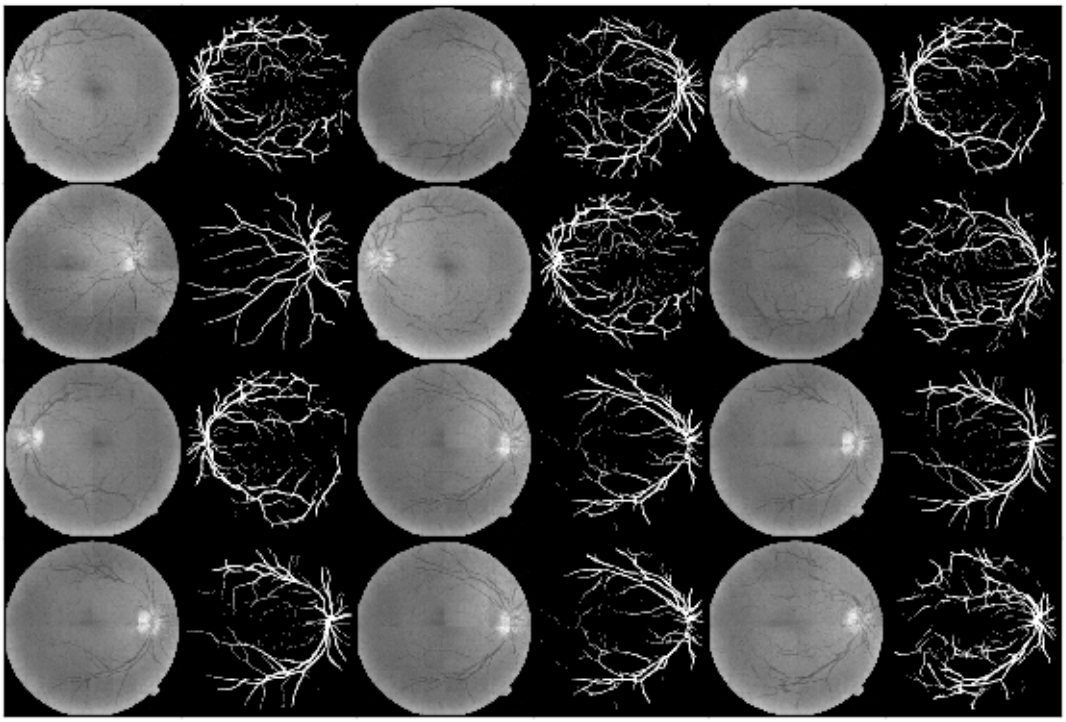


Figure 3.1. Images from our GAN trained on the DRIVE Database

3.3. Segmentation Process

We used the images generated from GAN and the original training images from DRIVE dataset as the training set. The images are down-sampled to 512×512 resolution with 8 minibatch size for every epoch. The dataset has been shuffled randomly and divided to training and validation dataset with 90 images for training and 10 images for validation. For testing, 20 images were used form DRIVE and 10 images for the STARE dataset. In figure (3.2) the evaluation setup is illustrated on the images generated from the GAN and the pre-trained encoder based on VGG+11 of Unet network.

To examine the impact of different data augmentation and GAN on the performance of the segmentation, different models have been trained with 500 epochs. The first model was trained without any form of data augmentation, the second model trained

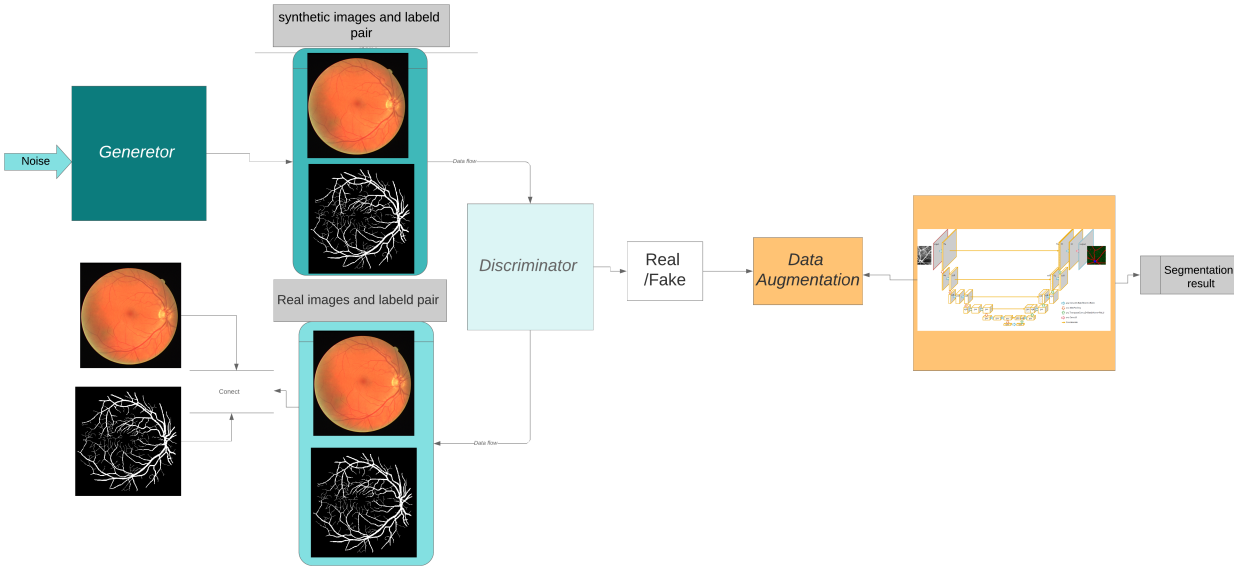


Figure 3.2. Evaluation setup for the segmentation

with standard data augmentation, while the last model trained with GAN generated data. Table (3.1) shows the accuracy and loss for the the training and the validation set of each of the three models. As it can be seen from the table, the data augmentation model shows an decrease in the loss in both training and validation and thus increase in the accuracy in both sets, comparing to the baseline model (*the model without any data augmentation*). GAN based model, on the other hand, shows the best performance among the three models, with a loss of 0.03 and 0.064 in the training and validation sets respectively. Setting an accuracy of 0.9806 and 0.9756 in training and validation sets respectively.

Table 3.1. The effect of different data augmentation methods on the accuracy of training and validation

Methods	Training Accuarcy	Training Loss	Validation Accuracy	Validation Loss
Without data augmentation	0.908	0.2	0.824	0.365
With standered data augmentation	0.956	0.165	0.871	0.323
GAN based data augmentation	0.9806	0.03	0.9736	0.064

Figure (3.4) shows the accuracy and the loss for training and validation for the model without any data augmentation while figure (3.3) shows the results of accuracy and loss with GAN- based data augmentation. As we can see from the plots, the GAN based segmentation model is stable while the non-GAN model starts to overfit the data. This behavior is expected as the training set was limited to 20 images. By increasing the training accuracy and decrease the validation accuracy we early stopped the model at 175 epoch after seeing the overfitting.

Figure (3.5) presents several exemplary segmented images from our method on the testing dataset from DRIVE [5]. Figure (3.6) shows the segmentation result on STARE dataset [4]. The figures present the baseline vessel structure images, ground truth images, and segmented images. From these examples, we conclude that the segmented images are able to define the vessels structures as well as the pattern from the groundtruth images as well.

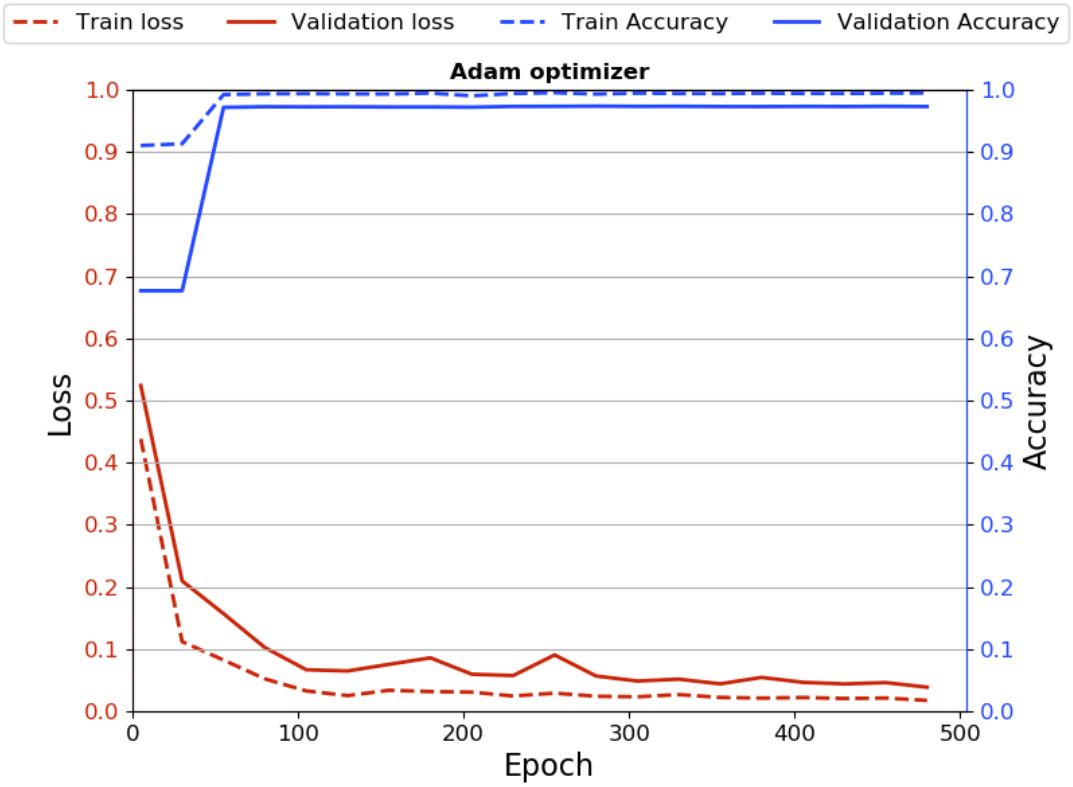


Figure 3.3. Segmentation performance for training on GAN-based augmentation

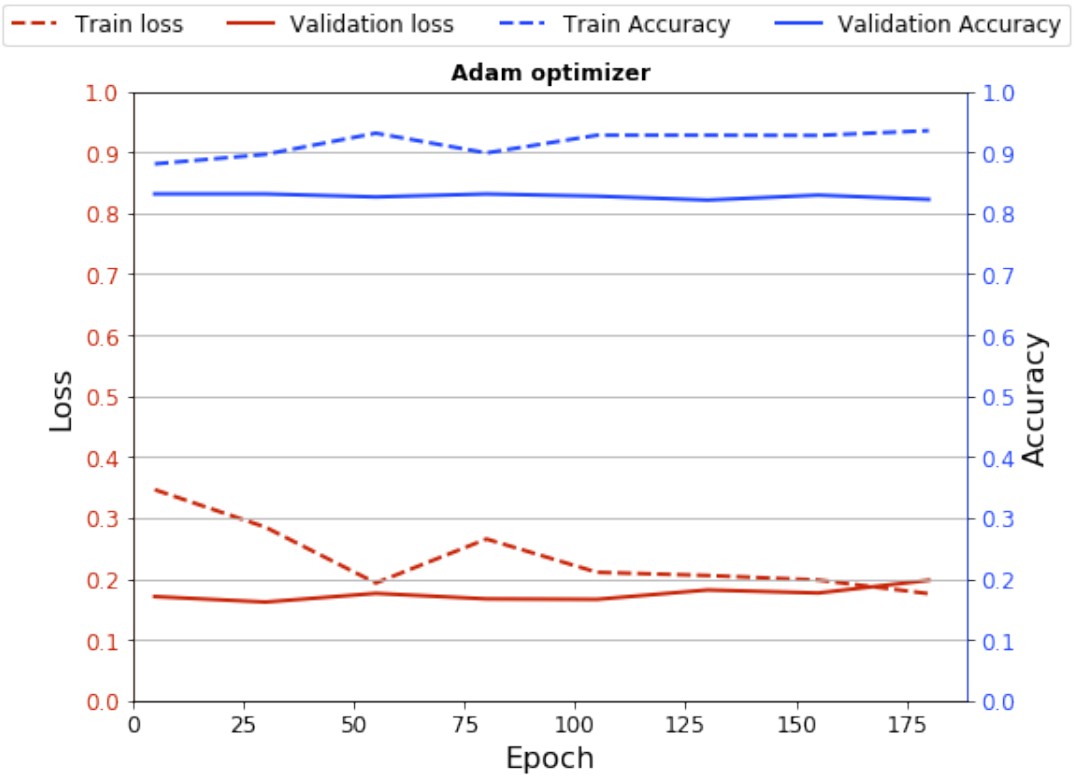


Figure 3.4. Segmentation performance for training on raw data

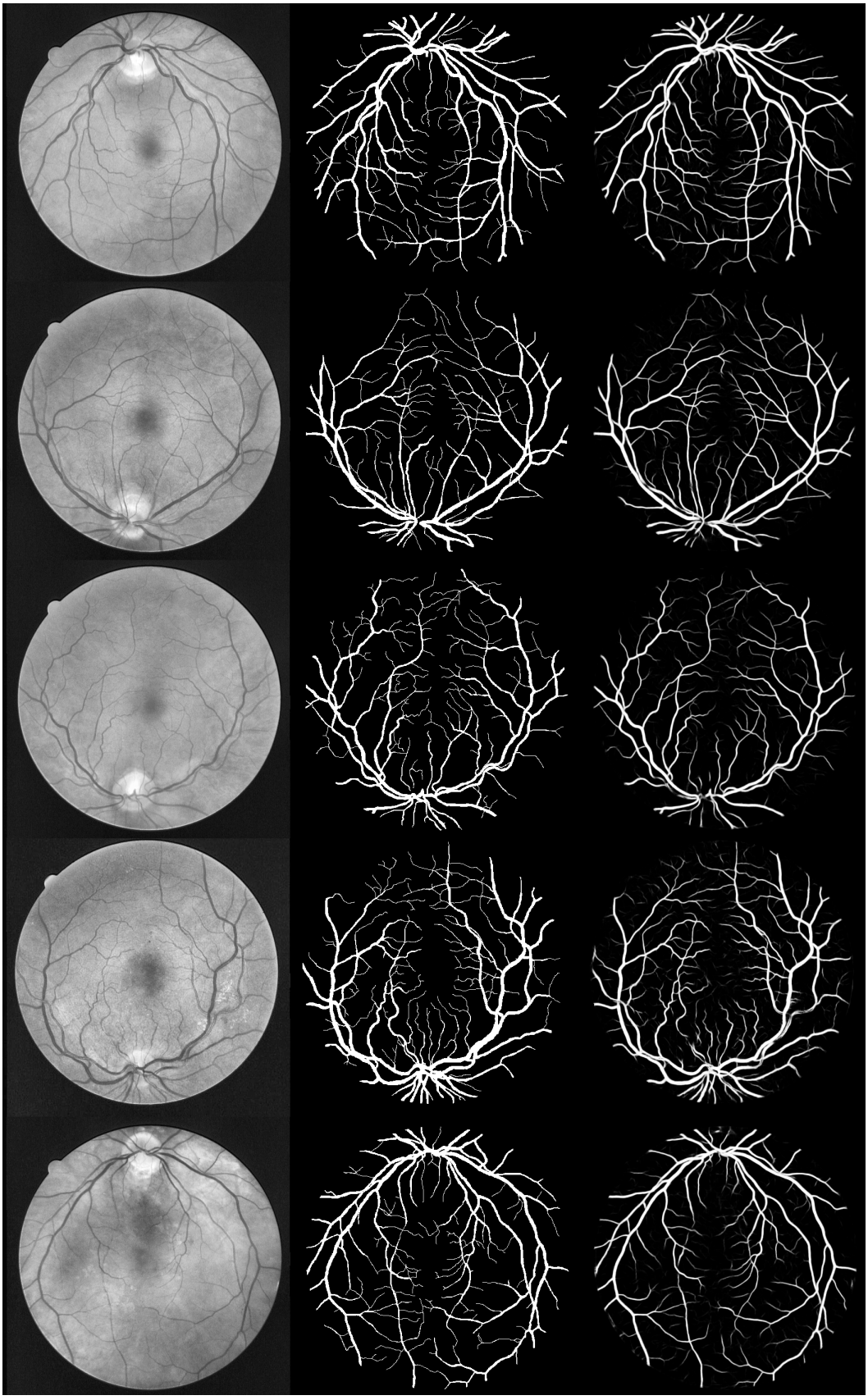


Figure 3.5. Sample of the segmented testing data from DRIVE dataset

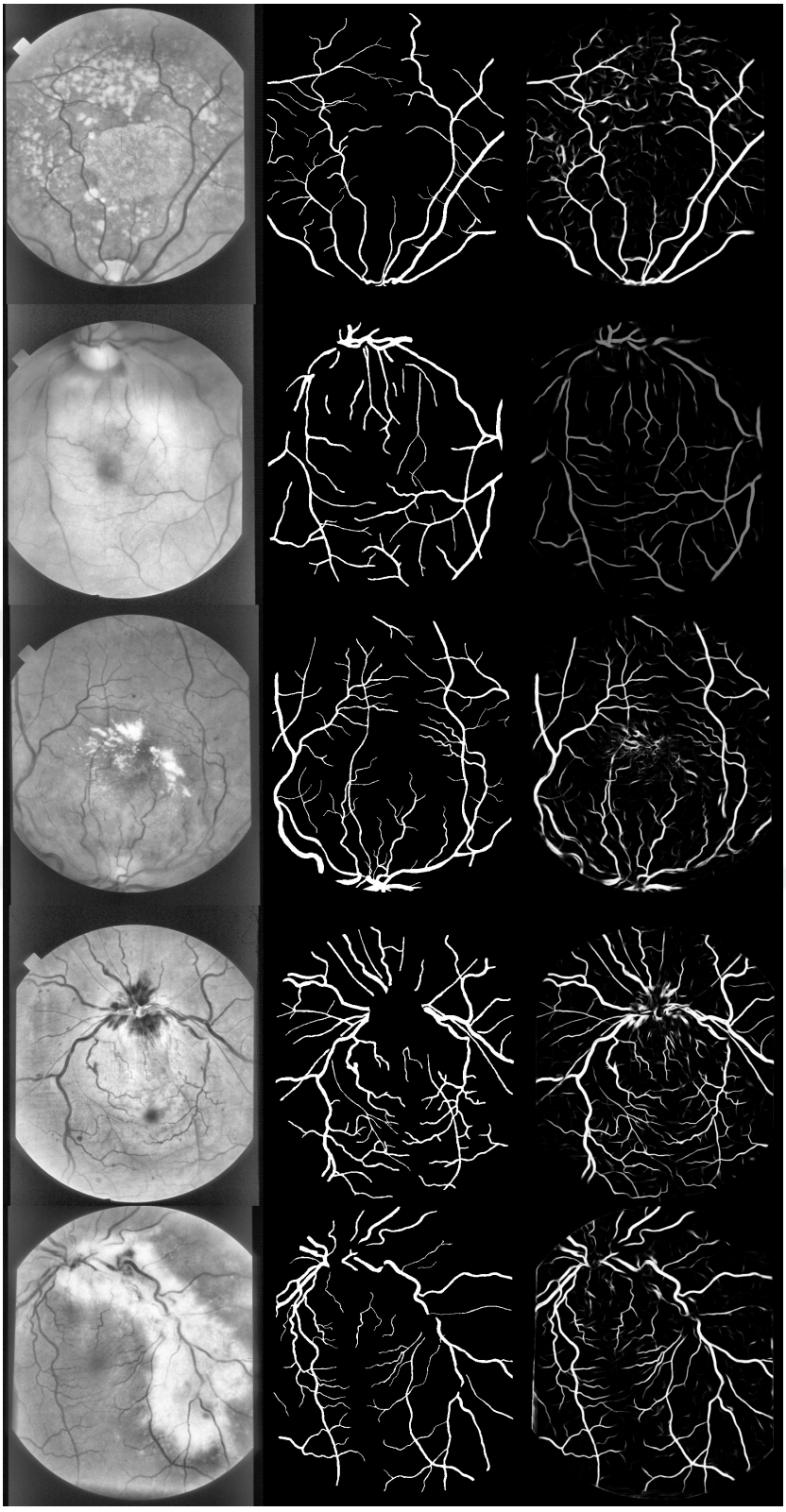


Figure 3.6. Sample of the segmented testing data from STARE dataset

Table (3.2) summarizes AUC for ROC and dice coefficients for our method and

other method applied on the same datasets; (DRIVE) and (STARE). Our method shows a good result in comparison with other methods, even if it did not out-preform the best result from (DRIU) [61]. However DRIU method provides a high probability value on the outline edge and on fine edges which can result over-segmentation and high false positive rate. While our method provides a suitable probability values to the ground truth by allowing more false negatives near the edges which tend to give low probability for the pixel in uncertain regions like small vessels or veins. Figures (3.7, 3.8) shows comparison of segmentation using our method and DRIU approach on DRIVE and STARE datasets respectively.

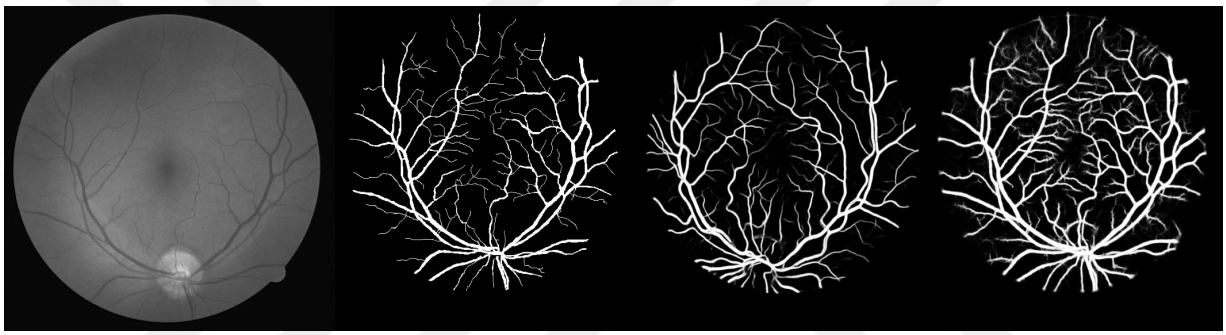


Figure 3.7. From left to right: Source image from DRIVE ,ground-truth, probability Map from our method, probability Map (segmented result) from DRIU

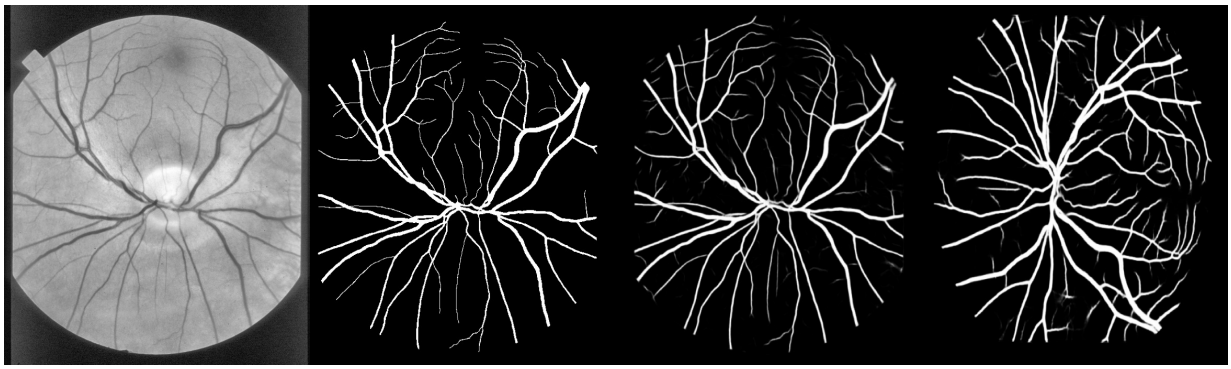


Figure 3.8. From left to right: Source image from STARE ,ground-truth, probability Map from our method, probability Map (segmented result) from DRIU

Table 3.2. Illustration of the proposed method with other present techniques on the basis of AUCROC and PR and Dice Score on testing data

	DRIVE			STARE		
Method	Dice Score	AUC ROC	AUC PR	Dice Score	AUC ROC	AUC PR
Our Method	0.819	0.978	0.898	0.830	0.9801	0.918
Unet	-	0.970	0.886	-	0.973	0.902
Kernel Boost	0.800	0.931	0.846	–	–	–
N4- Fields	0.805	0.968	0.885	-	-	-
DRIU	0.822	0.979	0.906	0.831	0.972	0.910
HED	0.796	0.969	0.877	0.805	0.976	0.888
Wavelet	0.762	0.943	0.814	0.774	0.969	0.843
Pixel-GAN	-	0.971	0.889	-	0.967	0.897
Image GAN	-	0.980	0.914	-	0.983	0.916
Patch GAN-1 (10x10)	-	0.970	0.889	-	0.976	0.903
Patch GAN-2 (80x80)	-	0.972	0.893	-	0.977	0.908

3.4. Discussion

This project investigated the segmentation performance of rential vessels using GAN and data augmentation. For GAN we modified the novel approach presented in [14], which has been successfully able to generate image-labels pair, unlike the traditional GAN which generates just images. Instead of using Tensorflow framework as in the original implementation of GAN, our approach was built from scratch using Pytorch framework. This approach allows us to increase the number of data in training dataset to be used in a supervised segmentation network.

3.5. Result Analysis

GAN: Our GAN performed well on generating 80 realistic images of the drive dataset and the segmentation performed well on extracting the vessels for the images. GAN based segmentation gives better segmentation results compared to the segmentation results using standard data augmentation or without any data augmentation.

U-net segmentation: The results show that all the models used in the experiments are able to create satisfactorily segmentation of the vessels, However, small vessels are often missing in our segmentation results

3.5.1. Quantitative Performance

The models are evaluated with the accuracy metric, reporting the F_1 scores (we report the average over the test set) as shown in table (3.3). According to the results, GAN-based model shows an improvement, for which the validation, test accuracy are 0.9735 and 0.978 respectively.

Table 3.3. Models Performance.

Model	**Validation Accuracy**	**Test Accuracy**
Baseline	0.824	0.834
Data Augmentation Model	0.871	0.885
GAN Model	0.9735	0.978

3.5.2. Qualitative Performance

To evaluate the quality of the segmented images, a qualitative analysis is performed. Examples of the segmented images are shown in figure (3.5) under section (3.2). The results of the VGG+11 Unet trained model after 500 epochs and 110 examples only. According to the result, the basic model is incapable of generating the images, whereas the GAN-based model is capable of capturing some key features in the source image and the ground-truth as in first example.

Figure (3.1) under section (3.2) shows examples of generated images of (GAN model) when an early stopping was performed in which the model was forced to stop after the 3000 epochs. The first example in the figure exemplifies a case of a generated image that is irrelevant to the problem domain, while the second one demonstrated a case of discontinuity. This quality of results can be attributed to the insufficient size of the dataset.

3.6. Limitations

As for the segmentation system, the GAN based model performed better than the basic model even though both models share the same parameters. The difference in the segmentation performance can be attributed to firstly the size of the dataset in the baseline model, and the secondly to the quality of the generated images in the standard data augmentation model. Based on the overall performance of the models, a couple of trends have been noticed:

- Repetitive patterns: Some GAN generated images tend to have repetitive patterns which cane be attributed to the limitation on the size of the dataset.
- Incorrect segmentation: Sometimes the segmented vessels tend to be incorrect and not presented in the image. This is likely related to the size of the trained dataset.
- Tendency to produce random vessels segmentation: The model shows tendency to generate and assume details. The explanation of this behaviour can be linked

to the size and the nature of the vessels in the original data sets.

- Small amount of data: One of the biggest limitations was the scarcity of data in the problem domain. With only 20 images in the available datasets, the application of deep learning will not consider significant. To partially overcome this limitation, a GAN was applied to extend the dataset. More groundtruth data will definitely improve the accuracy of the model.

- Time constrain: Another limitations was the lack of time left after finishing the implementation of the model. For computer vision tasks, learning the features is time demanding. As such models that achieved good performance are often trained for days with a high number of parameters.

- Limitation of resource: The lack of physical GPU facilities in the university forced the work to be implemented on Google Colab GPU which is free and limited service with very limited memory availability.

- Different directions of research approaches: All the above limitations have somehow imposed a different research direction in this thesis. The thesis was firstly designed to use Rotterdam Coronary Artery Algorithm Evaluation Framework, and subsequently preform the segmentation on 3D medical data. However, due to the lack of computing resources and the lack of research on the available dataset, mainly no sufficient information on extracting the ground truths in this dataset. All these factors have changed the direction of the research in the last few months and instead the DRIVE dataset has been used as a dataset.

4. CONCLUSION AND FUTURE WORK

4.1. Conclusion

In this project, we presented a hybrid approach for semantic segmentation of retinal fundus that combines U-net architecture with different data augmentation methods including GAN. The GAN model was trained with pairs of images and their corresponding masks to extend the dataset. The model generates a pair of image and its segmentation which used in addition to the original dataset to train the U-net for the segmentation task. Despite the improvement presented with the data augmentation, the quality of GAN generated images are still not realistic alike. Thus, we believe if this model is trained with larger datasets and more epochs, it could produce effective results. The idea of applying data augmentation approaches as GAN to extend the dataset deems efficient due to the practically of labelled images in the medical domain. However, the nature and the complexity of medical images seems to produce some difficulties in the capabilities of GAN to generalize to different image modalities. As such, this suggests a deep investigation and improvement to the data augmentation system could provide more scalable and robust results than using standard data augmentation methods.

4.2. Future Work

One improvement that could significantly enhance the model's performance and prevent the overfitting is to increase the size of the dataset. However, this addition usually comes with computation cost. As the model starts to learn features and relationships between the images and its segmentation, more training is likely to increase the model's ability to learn unique features. As such a future step to modify the model is to give it more training.

Improving data augmentation As data augmentation's contribution is significant

at this work, a next step would be to improve the augmentation methods with more sophisticated approaches that combine basic computer vision approaches with GAN. The quality of images generated by GAN showed a possibility for improvement where the vessels in the images appear disconnected. A post-processing steps can improve the quality. Another possible approach is to apply image-to-image translation GAN based approaches such as CycleGAN [69] which can be modified to generate image-segmentation pairs. Additionally, using different segmentation networks such as I2I-3D [70] to work on 2D images, since these networks showed good results in segmented small vessels.

5. REFERENCES

1. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, AR., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., and Sainath, T., Deep neural networks for acoustic modeling in speech recognition, IEEE Signal processing magazine, 29 (2012)

2. Taigman, Y., Yang, M., Ranzato, MA., Wolf, L., Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2014) 1701-1708.

3. 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, (2015) 234-241.

4. Hoover, A., Kouznetsova, V., Goldbaum, M., Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response, In Proceedings of the AMIA Symposium, (1998) 931.

5. Soares, JV., Leandro, JJ., Cesar, RM., Jelinek, HF., Cree, MJ. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification, IEEE Transactions on medical Imaging, 25,9 (2006) 1214-22.

6. Abràmoff, MD., Garvin, MK., Sonka, M., Retinal imaging and image analysis, IEEE reviews in biomedical engineering, 3 (2010) 169-208.

7. Hubbard, LD., Brothers, RJ., King, WN., Clegg, LX., Klein, R., Cooper, LS., Sharrett, AR., Davis, MD., Cai, J., Methods for evaluation of retinal microvascular abnormalities associated with hypertension/sclerosis in the atherosclerosis risk in communities study, AR in Communities Study Group et al. Ophthalmology, 106,12 (1999) 2269-80.

8. Wong, TY., Shankar, A., Klein, R., Klein, BE., Hubbard, LD. Prospective cohort study of retinal vessel diameters and risk of hypertension. bmj,329 (2004) 7457-79.

9. Tham, YC., Li, X., Wong, TY., Quigley, HA., Aung, T., Cheng, CY., Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis, Ophthalmology, 121,11 (2014) 2081-90.

10. Paulus, J., Jörg, M., Rüdiger, B., Joachim, H., and Georg, M., Automated quality assessment of retinal fundus photos, International journal of computer assisted radiology and surgery 5,6 (2010) 557-564.

11. Novotny, HR., and David, LA., A method of photographing fluorescence in circulating blood in the human retina. Circulation, 24,1 (1961) 82-86.

12. Rozantsev, A., Lepetit, V., Fua, P., On rendering synthetic images for training an object detector, Computer Vision and Image Understanding, 137 (2015) 24-37.

13. WWW.asrs.org/content/images/cms/solar-retinopathy.png/img-full.ImageHandler. May.2019

14. Neff, T., Payer, C., Stern, D., Urschler, M. Generative adversarial network based synthesis for supervised medical image segmentation, InProc. OAGM and ARW Joint Workshop, (2017) 140–145.

15. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, DN., Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, InProceedings of the IEEE International Conference on Computer Vision, (2017) 5907-5915.

16. Han, J., Pei, J., Kamber, M., Data mining: concepts and techniques, Elsevier, 09,6 (2011)

17. Kingma, DP., and Jimmy, Ba. Adam: A method for stochastic optimization, arXiv preprint, (2014) 1412-6980.

18. Hinton, G., Srivastava, N., and Swersky, K., Overview of mini-batch gradient descent, Neural Networks for Machine Learning, (2012) 575.

19. Kingma, DP., and Jimmy, Ba., Adam: A method for stochastic optimization, arXiv preprint, (2014) 1412-6980.

20. Karpathy, A. Cs231n convolutional neural networks for visual recognition, Neural networks, 1 (2016).

21. Saba, T., Rehman, A.,and Sulong, G., An intelligent approach to image denoising, Journal Of theoretical and Applied information technology, 17,2 (2010) 32-36.

22. Li, FF., Andrej, K., and Justin, J., Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition, (2017).

23. Beiu, V., Peperstraete, JA., Vandewalle, J., and Lauwereins, R., Closse approximations of sigmoid functions by sum of step for vlsi implementation of neural networks, Sci. Ann. Cuza Univ, 3 (1994) 5-34.

24. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R., Learning from simulated and unsupervised images through adversarial training, InProceedings of the IEEE conference on computer vision and pattern recognition, (2017) 2107-2116.

25. Cheng, L., Vishwanathan, SN., and Zhang, X. Consistent image analogies using semi-supervised learning, In2008 IEEE Conference on Computer Vision and Pattern Recognition, 23,7 (2008) 1-8.

26. Fukushima, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological cybernetics, 36,4 (1980) 193-202.

27. Dumoulin, V., Visin, F., A guide to convolution arithmetic for deep learning, arXiv preprint, (2016) 1603-7285.

28. Lee, CY., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., Deeply-supervised nets, InArtificial intelligence and statistics, (2015) 562-570.

29. Hutchinson, ML., Erin, A., Brenna, M., Gibbons, SP., Julia, L., and Bryce, M., Overcoming data scarcity with transfer learning, arXiv preprint, (2017) 1711.05099.

30. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., Generative adversarial nets, In Advances in neural information processing systems, (2014) 2672-2680.

31. Radford, A., Metz, L., and Chintala, S., Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint, 19,11 (2015)

32. Ioffe, S., and Christian, S., Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint, (2015) 1502.03167.

33. Kohonen, T., Self-organization and associative memory, Springer Science and Business Media, 8 (2012).

34. Zou, KH., Warfield, SK., Bharatha, A., Tempany, CM., Kaus, MR., Haker, SJ., Wells, III WM., Jolesz, FA., and Kikinis, R., Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports, Academic radiology, 11,2 (2004) 178-89.

35. Hashemi, SR., Seyed, SMS., Deniz, E., Sanjay, P,., Simon, K,. Warfield, SK., and Ali, G. ,Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection, IEEE, 7, (2018) 1721-1735.

36. Frangi, Alejandro F., et al. Multiscale vessel enhancement filtering, International conference on medical image computing and computer-assisted intervention, Springer, Berlin, Heidelberg, (1998).

37. Schaap, M., Metz, C., van Walsum, T., and Niessen, W., Rotterdam coronary artery algorithm evaluation framework, (2009).

38. Zhu, H., Chan, H.Y., Lam, F.K. and Lam, K.Y., Segmentation of pathology microscopic images, In Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Magnificent Milestones and Emerging Opportunities in Medical Engineering, 2, (1997) 580-581.

39. Troccaz, J., Baumann, M., Berkelman, P., Cinquin, P., Daanen, V., Leroy, A., Marchal, M., Payan, Y., Promayon, E., Voros, S. and Bart, S.,Medical image computing and computer-aided medical interventions applied to soft tissues: Work in progress in urology. Proceedings of the IEEE, 94,9 (2006) 1665-1677.

40. Despotović, I., Goossens, B. and Philips, W., MRI segmentation of the human brain: challenges, methods, and applications, Computational and mathematical methods in medicine, (2015).

41. Ren, S., He, K., Girshick, R. and Sun, J., Faster r-cnn: Towards real-time object detection with region proposal networks, In Advances in neural information processing systems, (2015) 91-99.

42. Mikolov, T., Karafiát, M., Burget, L., Černocký, J. and Khudanpur, S., Recurrent neural network based language model, In Eleventh annual conference of the international speech communication association, (2010).

43. Csurka, G., A comprehensive survey on domain adaptation for visual applications, Domain Adaptation in Computer Vision Applications, Springer, Cham, (2017) 1-35.

44. Ganin, Y. and Lempitsky, V., Unsupervised domain adaptation by backpropagation. arXiv preprint, (2014) 1409.7495.

45. Pham, D.L., Xu, C. and Prince, J.L., Current methods in medical image segmentation, Annual review of biomedical engineering, 2,1 (2000) 315-337.

46. Basu, M., Gaussian-based edge-detection methods-a survey, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 32,3 (2002) 252-260.

47. Gregor, K., Ivo, D., Alex, G., Danilo, JR,, and Daan Wierstra, Draw: A recurrent neural network for image generation, arXiv preprint, (2015) 1502.04623.

48. Bezdek, JC., Robert, E., and William, F., FCM: The fuzzy c-means clustering algorithm, Computers and Geosciences, 10,2-3 (1984) 191-203.

49. Bradley, PS., Usama, F., and Cory, R., Scaling EM (expectation-maximization) clustering to large databases, (1998).

50. Osher, S., and Nikos, P., Geometric level set methods in imaging, vision, and graphics, Springer Science and Business Media, (2003).

51. Mahesan, K.V., Bhargavi, S., and Jayadevappa, D., Segmentation of MR images using active contours: methods challenges and applications, International Journal of Innovative Research in Advanced Engineering, 2,4, (2017) 13-21.

52. Baillard, C., Pierre, H., and Christian, B., Segmentation of brain 3D MR images using level sets and dense registration, Medical image analysis, 5,3 (2001) 185-194.

53. Krause, J., Johnson, J., Krishna, R. and Fei-Fei, L., A hierarchical approach for generating descriptive image paragraphs, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2017) 317-325.

54. Menti, E., Bonaldi, L., Ballerini, L., Ruggeri, A. and Trucco, E., Automatic generation of synthetic retinal fundus images: Vascular network, In International Workshop on Simulation and Synthesis in Medical Imaging, Springer, Cham , (2016) 167-176.

55. Bankhead, P., Scholfield, C.N., McGeown, J.G. and Curtis, T.M., Fast retinal vessel detection and measurement using wavelets and edge location refinement, PloS one, 7,3 (2012).

56. Azzopardi, G., Strisciuglio, N., Vento, M. and Petkov, N., Trainable COSFIRE filters for vessel delineation with application to retinal images. Medical image analysis, 19,1 (2015) 46-57.

57. Becker, C.J., Rigamonti, R., Lepetit, V. and Fua, P., KernelBoost: Supervised Learning of Image Features For Classification, (2013).

58. Sofka, M., and Stewart, C.V., Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures, IEEE transactions on medical imaging, 25,12 (2006) 1531-1546.

59. Soares, J.V., Leandro, J.J., Cesar, R.M., Jelinek, H.F. and Cree, M.J., Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification, IEEE Transactions on medical Imaging, 25,9 (2006) 1214-1222.

60. Fiorini, S., Ballerini, L., Trucco, E. and Ruggeri, A., Automatic Generation of Synthetic Retinal Fundus Images, In Eurographics Italian Chapter Conference, (2014) 41-44.

61. Maninis, K.K., Pont-Tuset, J., Arbeláez, P. and Van Gool, Deep retinal image understanding. In International conference on medical image computing and computer-assisted intervention, Springer, Cham, (2016) 140-148.

62. Maji, D., Santara, A., Ghosh, S., Sheet, D. and Mitra, P., Deep neural network and random forest hybrid architecture for learning to detect retinal vessels in fundus images, In 2015 37th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, (2015) 3029-3032.

63. Salem, S.A., Salem, N.M. and Nandi, A.K., Segmentation of retinal blood vessels using a novel clustering algorithm, In 2006 14th European Signal Processing Conference, IEEE, (2006) 1-5.

64. Akhavan, R. and Faez, K., A novel retinal blood vessel segmentation algorithm using fuzzy segmentation, International Journal of Electrical and Computer Engineering, 4,4 (2014) 561.

65. Xie, S. and Nie, H., Retinal vascular image segmentation using genetic algorithm Plus FCM clustering, In 2013 Third International Conference on Intelligent System Design and Engineering Applications, IEEE, (2013) 1225-1228.

66. www.pytorch.com , 30.Feb.3019

67. He, K., Zhang, X., Ren, S. and Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, In Proceedings of the IEEE international conference on computer vision, (2015) 1026-1034.

68. Iglovikov, V. and Shvets, A., Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation, arXiv preprint, (2018) 1801.05746.

69. Zhu, J.Y., Park, T., Isola, P. and Efros, A.A., Unpaired image-to-image translation using cycle-consistent adversarial networks, In Proceedings of the IEEE international conference on computer vision, (2017) 2223-2232.

70. Merkow, J., Marsden, A., Kriegman, D. and Tu, Z., Dense volume-to-volume vascular boundary detection, In International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham, (2016) 371-379.

BIOGRAPHY

Nagat MASUED was born in Libya. She received her bachelor degree in Electronic Engineering (Telecommunication) from Benghazi university with First-Class Honours degree. she worked in Almadar Tele. company for one year. Currently she is doing MS in Electronic Engineering at Karadeniz Technical University Trabzon, Turkey under Turkish government scholarship program.