RETINAL BLOOD VESSEL SEGMENTATION USING TRANSFER LEARNING ON UNET

Ramazan KARTAL[1], Sema KOÇ KAYHAN [2]

[1] Gaziantep University, Engineering Faculty, https://orcid.org/0000-0003-2454-0901

[2] Gaziantep University, Engineering Faculty, https://orcid.org/0000-0002-8129-7672

ABSTRACT

With the aid of morphological features of the eye, such as thickness and tortuosity, segmentation of the retinal blood vessels plays a significant role in the diagnosis of illnesses like diabetes and hypertension. This process needs a lot of time with trained people and even highly experienced ophthalmologists can have different conclusions on an example when doing it manually. With the recent advancements in Deep Learning, automated retinal blood vessel segmentation showed improvements. In this project, we used UNET, a Convolutional Neural Network-based architecture specifically designed for medical images, by designing the network with the Residual Network approach which has proven high efficiency on the very deep neural networks. We developed our method with Transfer Learning, a technique that launches the weights from the previously trained network instead of starting the learning with random values, we also used Data Augmentation since the publicly available DRIVE dataset contains only 20 training and 20 test images. We applied our model to several state-of-the-art pre-trained network architectures such as VGG, Res Net, and Efficient Net. Consequently, we reached a high efficiency and the outcomes far more than a professional ophthalmologist's manual way. We reached 96.84% accuracy with VGG19 architecture by freezing the first 12 layers of the network. With this work, we have shown that transfer learning has promising results. With the new improvements in transfer learning, it will not only decrease the time for training but also help to solve health problems like blindness of the patient on a global scale.

Key Words: Deep Learning, Transfer Learning, UNET, Retinal Blood Vessel, Segmentation

1. INTRODUCTION

In the last several years, we have seen investment and research in medical works going higher and higher. It is a very important area of work since it is directly related to humans. Even a small improvement in this area could make people life's better or even maybe it could save someone's life. Deep learning has shown great success in many areas, one of them is medical imaging. Especially in 2012, Krizhevsky et al. applied their model to the IMAGENET classification competition. Their model showed great performance and Deep Learning once again attracted by many researchers[1]. Their model was based on CNN-Convolutional Neural Network[2].
A lot of illnesses can be diagnosed by looking at the pictures of eye blood vessels, such as age-related macular degeneration, diabetic retinopathy, glaucoma, and hypertension[3]. Having an early detection system plays a significant role for the doctors to protect the patient's health and apply appropriate therapy. All of the blood vessels in the body are influenced by the chronic disease known as diabetes. Only in the eye can specific blood arteries be used, seen, and evaluated directly[4]. According to WHO around 3.9 million people have diabetic retinopathy problems globally, also diabetes is the main factor of avoidable blindness in North America[5].

Fraz et al. Divided the retinal vessel segmentation algorithms into 6 categories which are: (1) Pattern recognition, (2) Vessel tracking, (3) Matched filtering, (4) Mathematical morphology, (5) Multiscale approaches, (6) Model-based approach[6].
Emary et al. presented an automated method for segmenting retinal blood vessels based on fuzzy c-means clustering and artificial bee colony optimization. The DRIVE database had an accuracy of 0.939. The STARE database had an accuracy of 0.947 [7].
In order to combine two effective classifiers, Wang et al. employed a convolutional neural network to serve as a trainable hierarchical feature extractor and a random forest to serve as a trainable classifier[8].
Fu et al. combined a regular 7-layer CNN with a conditional random field and reconfigured it as a recurrent neural network to represent long-range pixel interactions[9].
For the segmentation of blood vessels, Jiang et al. suggested a fully convolutional network with transfer learning[10].

1.1 Transfer Learning and UNET

In this work, we build our model on two subjects, Transfer Learning and UNET. The DRIVE dataset does not have enough samples, it only contains 20 training and 20 test images, due to that using transfer learning could be simplified and increase the retinal blood vessel segmentation task. Transfer learning is a method of machine learning where a model is developed for one task and is used as the foundation for a model on another. Yosinski et al. demonstrated that transferring features can really be preferable to random initialization even when there is a significant difference between the targeted dataset and the original data[11].
In transfer learning, there are 3 approaches:

1- Fine-tuning: Freeze some layers and let the model learn the remaining layers.
2- Freeze all layers: Use as a feature extractor
3- Retrain the whole network: take the pre-trained network as a reference and update the weights while learning.

We took into consideration the fine-tuning and retrain the whole network approaches, as the DRIVE dataset and the source dataset are quite different from each other.
We used 3 pre-trained network architectures in this work:

1- VGG19
2- ResNet
3- EfficientNet

In order to overcome the vanishing/exploding gradient problem which happens when we have a very deep network, we used the Residual Block approach proposed by He et al.[12][13]

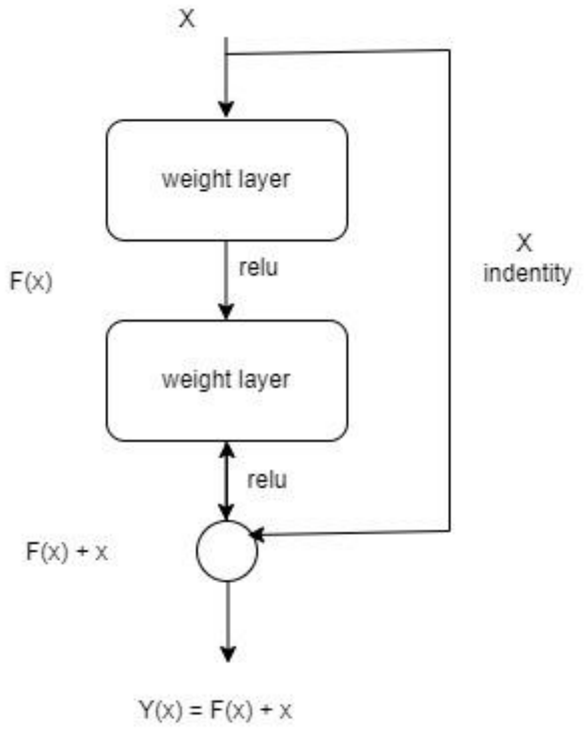


Figure 1. Residual Block

U-NET was first created especially for the tagging and categorization of medical photographs. Since it has demonstrated such strong performance that it is being used in several additional contexts[14]. It can be considered as a network of encoders and decoders. In the encoder part, the network is trying to understand "what" is inside the image, for the decoder part, the network is trying to identify "where" this object is in the image.

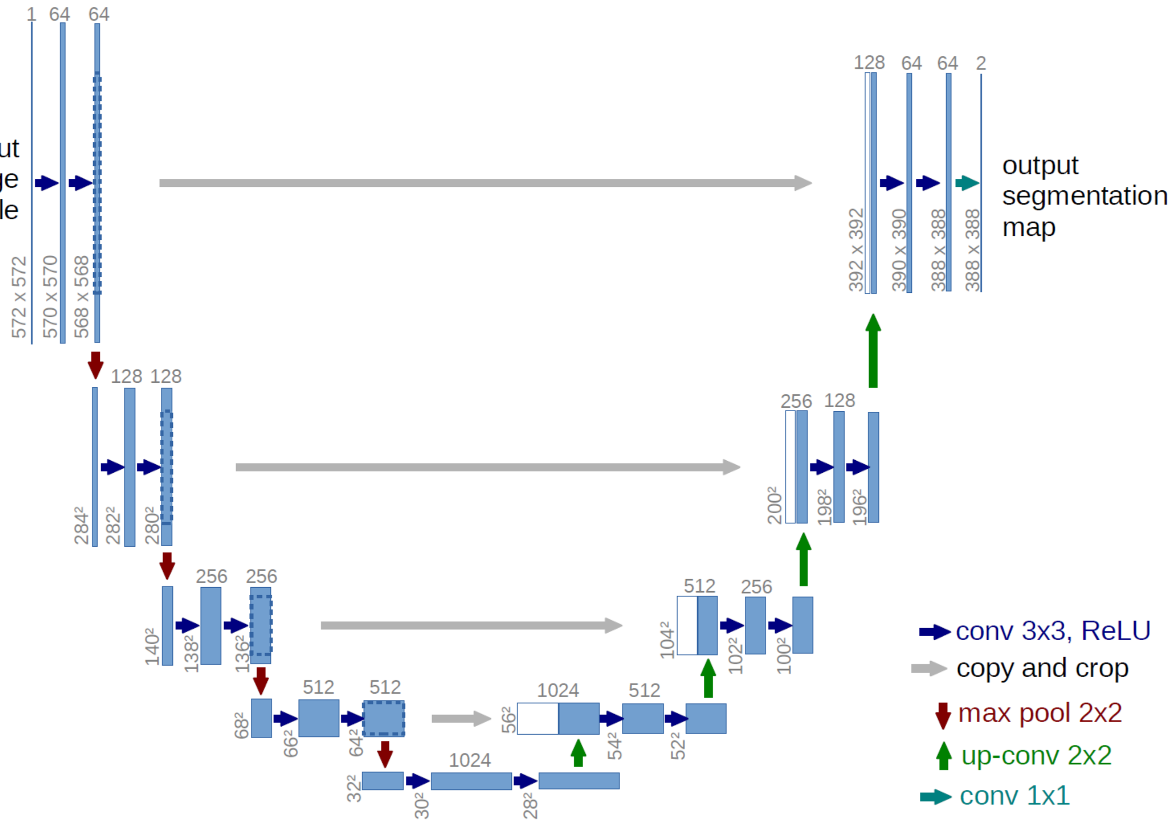


Figure 1: UNET

2. METHODOLOGY

2.1 Dataset

The DRIVE database images are 565 × 584 pixels with 8 bits per color channel. The library includes binary pictures with results of manual segmentation that were tested against the output of the vessel segmentation methods.

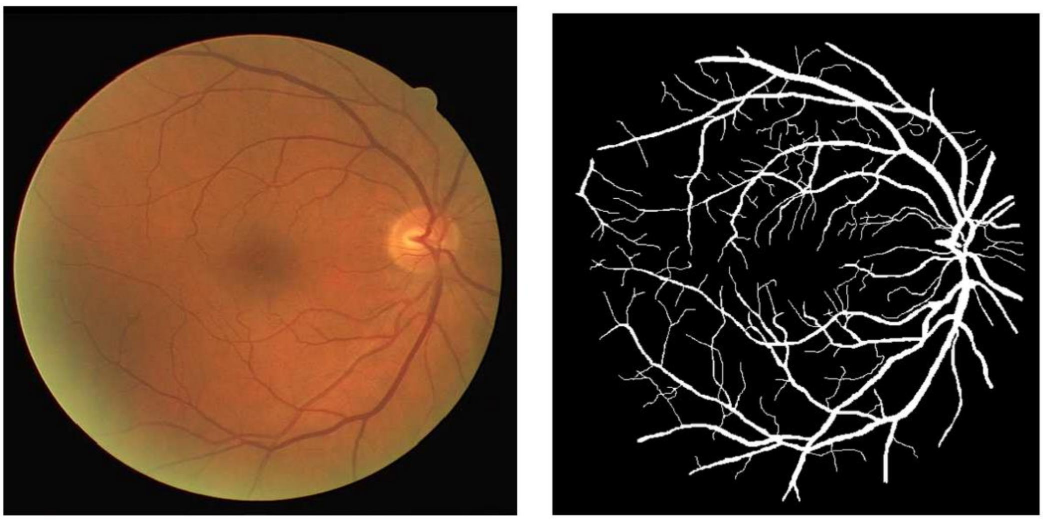


Figure 3. Image (a) and mask(b) from DRIVE Dataset

2.2 Preprocessing

Using preprocessing technique is a crucial step to enhancing the quality of data in deep learning, especially when having a limited number of samples as we have here. We applied some augmentation methods willing to get more data and have a more robust model against overfitting. We utilized the Keras image preprocessing tool. Below are the metrics with the corresponding values:

1- Rotation range = 0.40
2- Width shift range= 0.1
3- Height shift range= 0.1
4- Shear range= 0.2
5- Zoom range= 0.2
6- Horizontal flip= True
7- Fill mode= Reflect

2.3 Method

UNET was the backbone of our model. We separated our job into two parts:

The pre-trained network was loaded in the first section before being connected to UNET at specific layers. By doing so, the model's weights won't be randomly started; instead, we'll start with what we've learned. We didn't code any UNET components until the bottleneck's end when max-pooling happens four times. We just took the structure of the pre-trained network. As a result, we had various network topologies and, as would be expected, various quantities of parameters from each pre-trained network. For the avoidance of doubt, we linked each output of the pre-trained network—block 01, block 02, etc.—to its equivalent output of UNET.
We started coding the UNET architecture in the second section of the model. With the residual blocks, we coded the second portion of the network. The implementation of residual blocks was utilized in version 2. Since our networks were quite deep, we wanted to avoid the vanishing gradient problem for that reason. In contrast to what the original UNET recommended, we employed three convolution layers in this expansion path. We sought to boost the effectiveness of the network.

Due to the large pixel density (560x560x3), we limited the batch size to 2 during training, and we used the optimizer Adam with 300 epochs. We had a dropout rate of 0.3%. The loss was binary cross-entropy. Additionally, we utilized TensorFlow's EarlyStopping and ReduceLROnPlateau callbacks, which cause training to end if validation accuracy does not increase after 100 iterations. ReduceLROnPlateau is used to lower the learning rate if, after 10 iterations the validation accuracy does not improve.

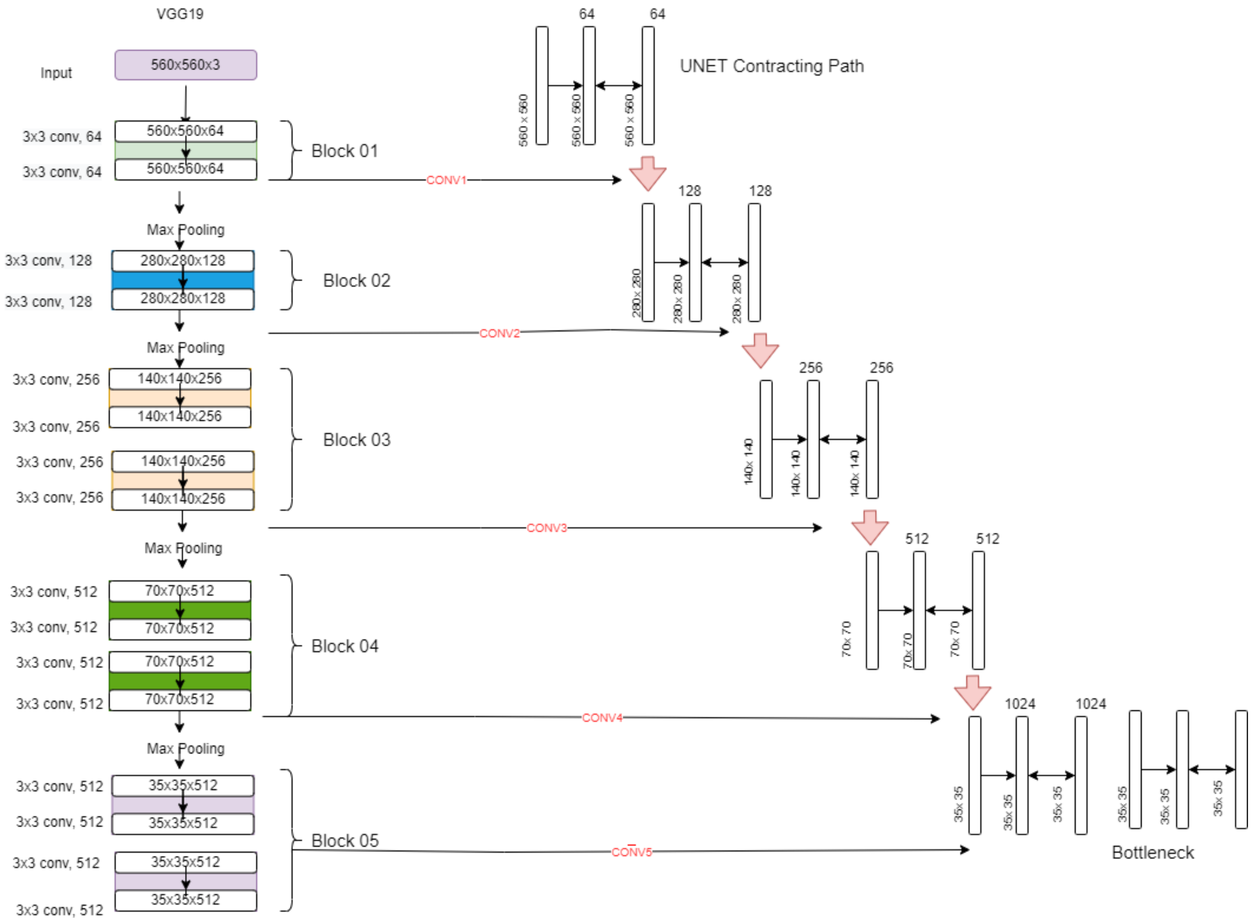


Figure 4. Proposed model connections

3. RESULTS AND CONCLUSIONS

To be able to measure the model's performance and its forecast, Loss and Accuracy graphs are used in this work. In accordance with how frequently the prediction deviates from the actual value, a loss function takes into account the probability or volatility of a forecast. While learning, the loss is also utilized to find the optimal model hyperparameters. In order to figure out how accurate the model the Accuracy comes into play. Following Loss and Accuracy, we also monitored Precision, AUC, Sensitivity, and Specificity.

By freezing the first 12 layers and allowing the remaining layers to continue learning, we were able to achieve the greatest accuracy of **%96,84** on the VGG19 network with a 560x560 pixel size, out of 12 training exercises using 3 different pre-trained networks which were ResNet, VGG, and EfficientNet.

Below there are some plotted results for the DRIVE dataset for the test data. The first column in the figure is the original image, the second one is the ground truth, and the third column is our model's prediction.

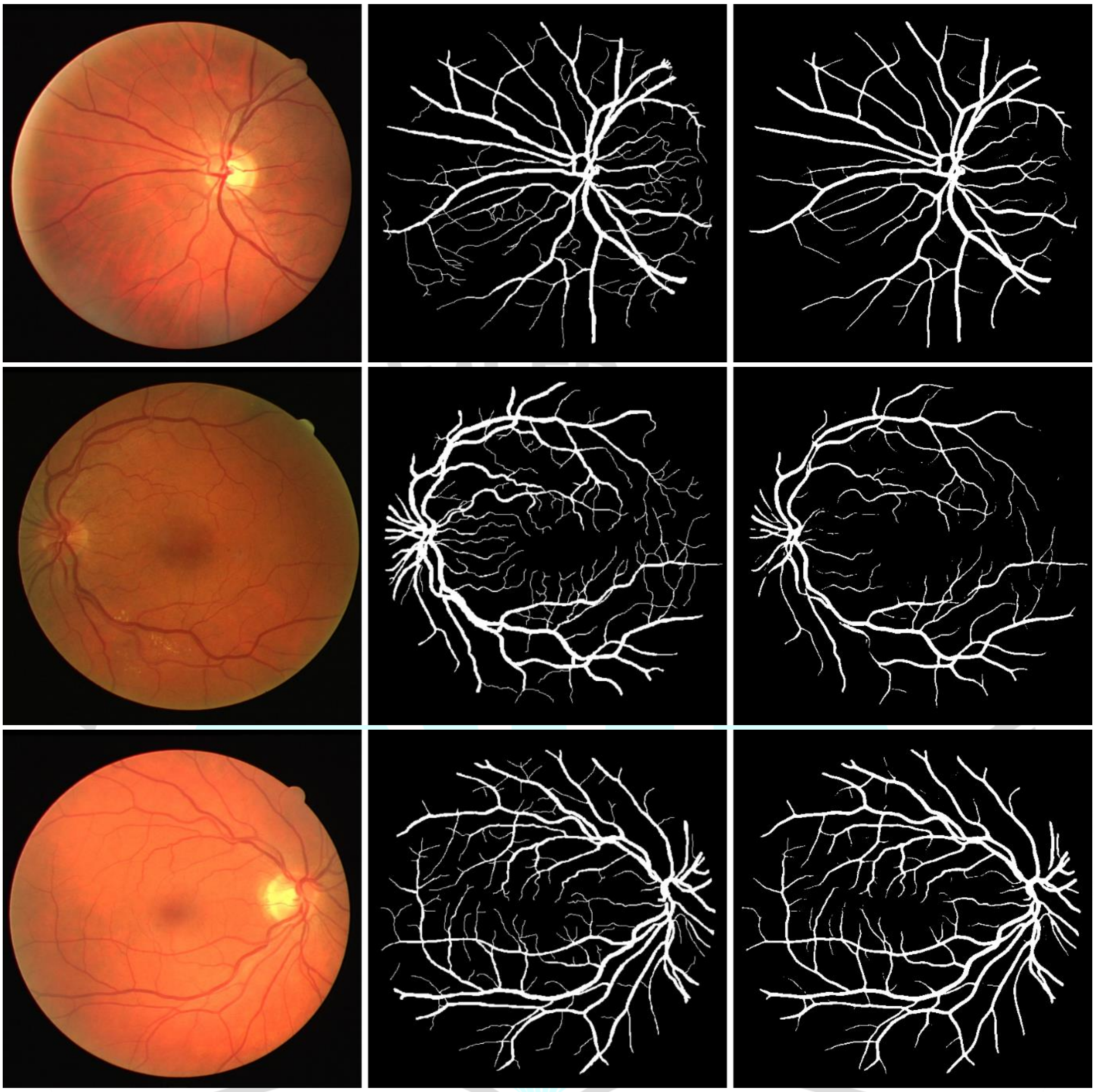
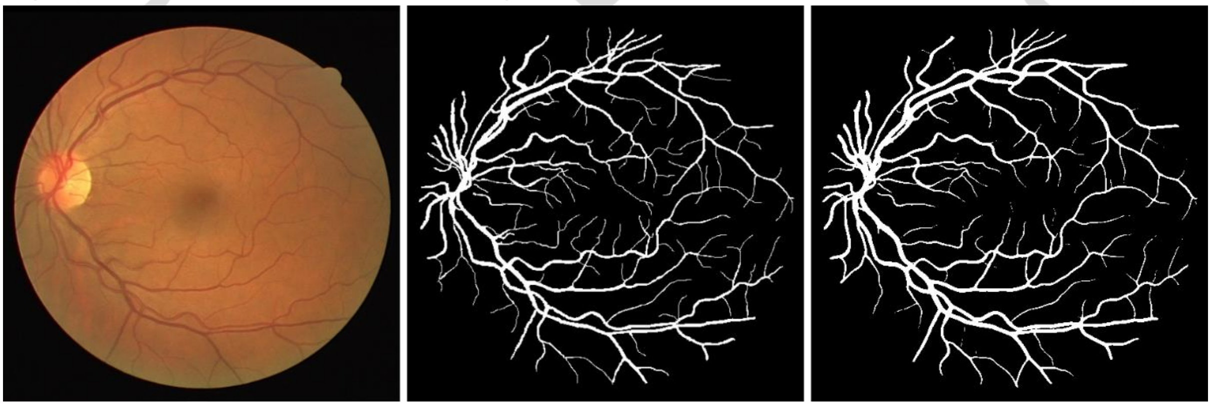


Figure 5. Results - (a) shows input image and (b) Ground Truth (c) shows the predicted output



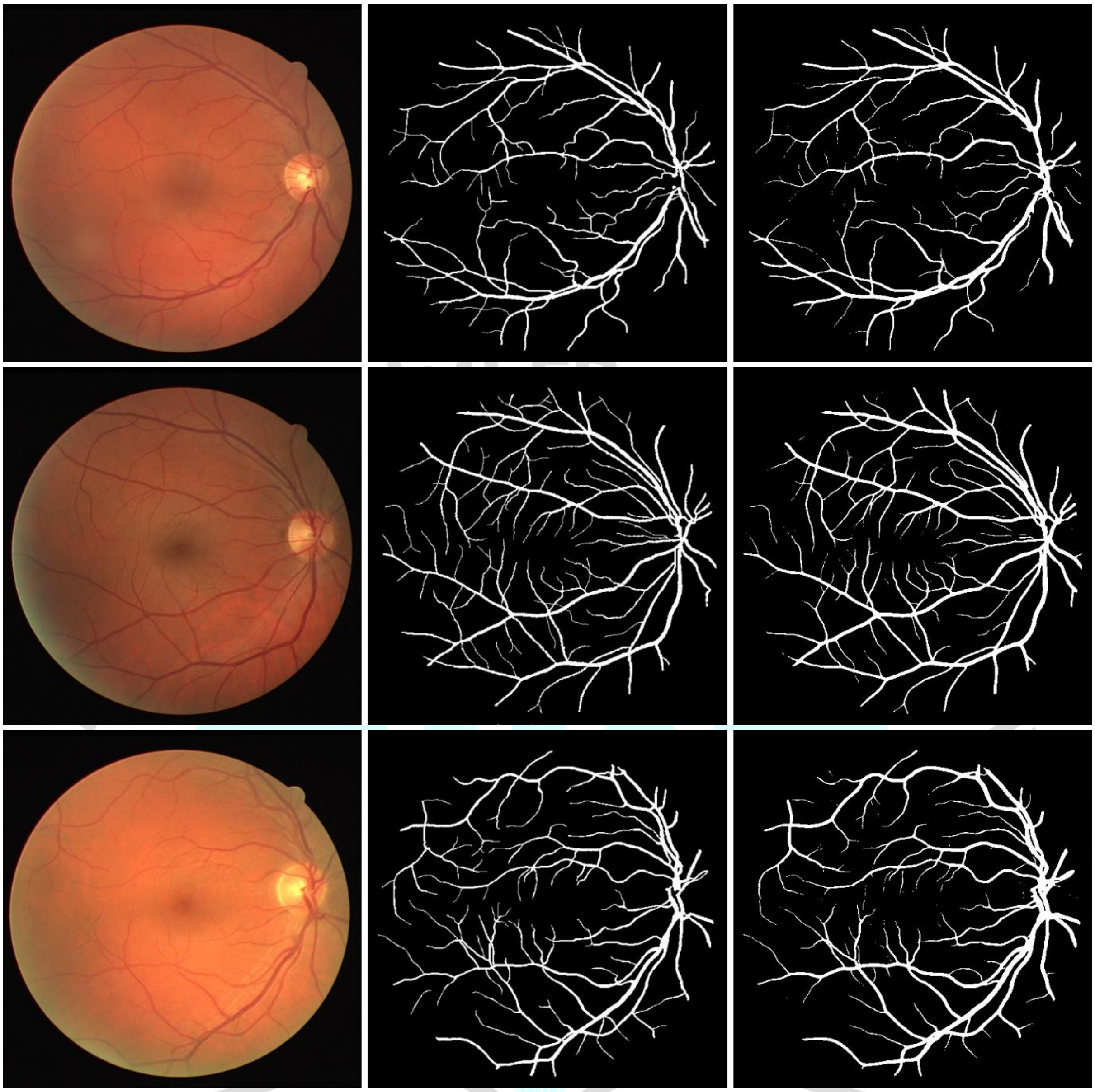
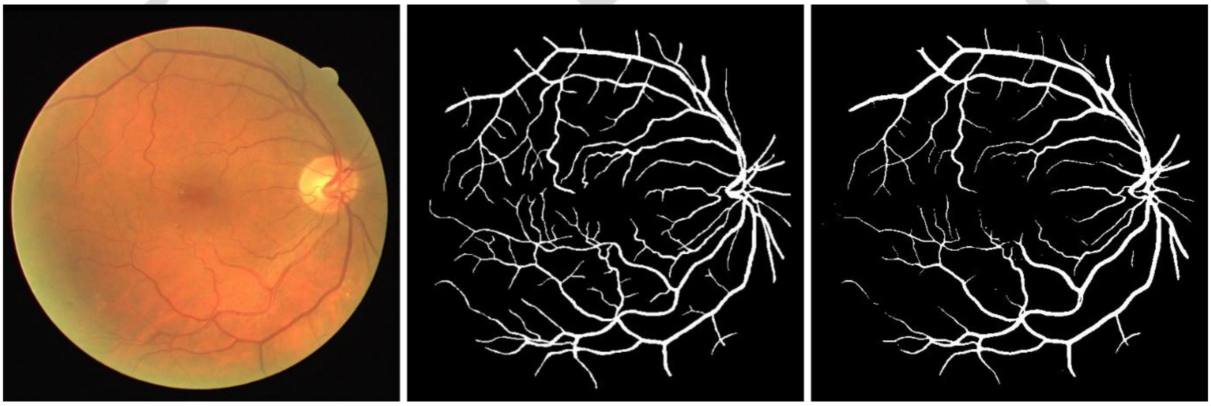


Figure 6. Results - (a) shows input image and (b) Ground Truth (c) shows the predicted output



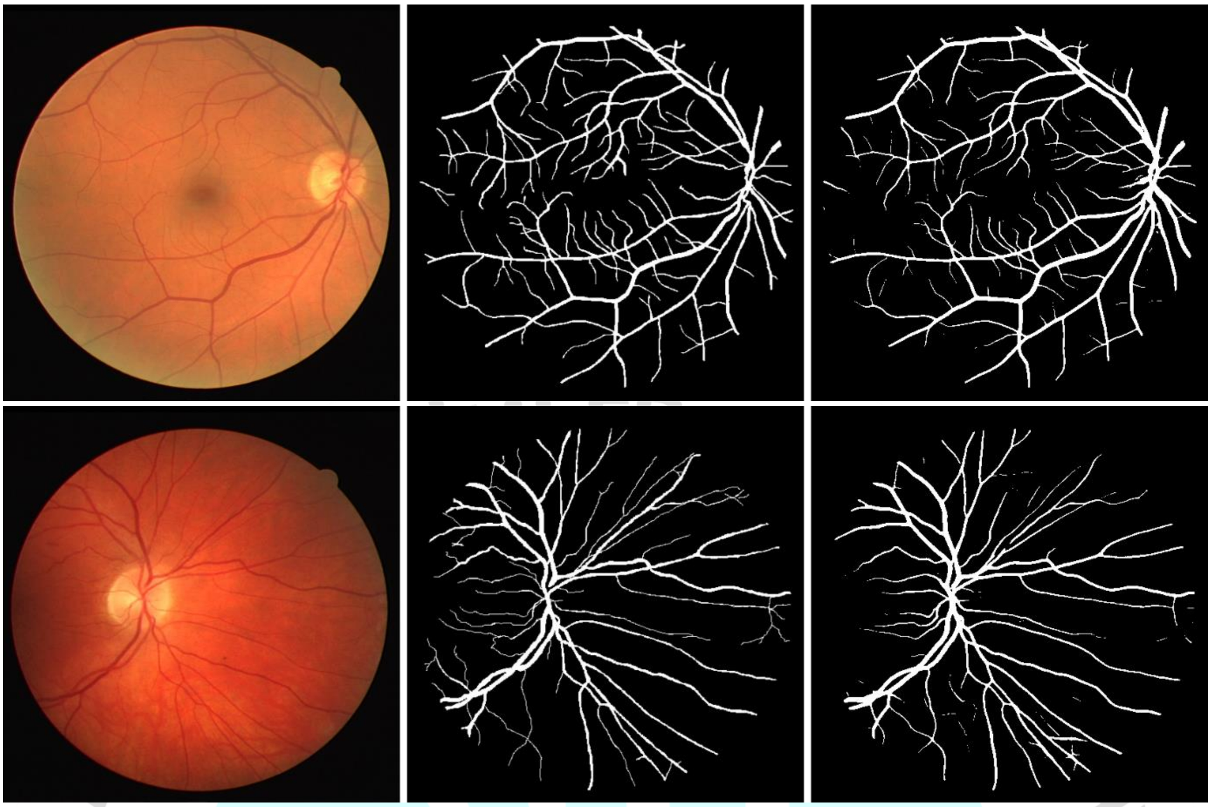


Figure 7. Results - (a) shows input image and (b) Ground Truth (c) shows the predicted output

It is time to discuss the results for every network we trained.

VGG19

Table 1. VGG Results

Network	**VGG19**	**VGG19[0:12]**	**VGG19[0:12]**	**VGG19**
Shape	560	560	256	256
Ori Params	20,024,384	17,698,816	17,698,816	20,024,384
Model Params	42,748,033	40,422,465	40,422,465	42,748,033
Ori Layers	22	22	22	22
Model Layers	102	102	102	102
Time(sec)	20	19	10	10
Epoch Stopped	267	300	300	300
Loss	0.8619	**0.0811**	0.8979	0.0904
Accuracy	0.9674	**0.9684**	0.9678	0.9667
AUC	0.971	**0.984**	0.980	0.978
Precision	0.837	0.866	0.851	**0.877**
Sensitivity	**0.777**	0.754	0.756	0.707
Specificity	0.986	0.989	0.988	**0.991**

We tested our model on two different sizes for every network, 256x256 and 560x560 pixels, in this way we aimed to understand the relationship between time and accuracy. We saw that even though the time for the 256 pixels(10-sec) and 560 pixels(20-sec) is doubled, the accuracy did not change dramatically. Original VGG19 has 19 layers whereas ours has 102 layers after we combined it with UNET.

With the VGG19 pre-trained network, we had an accuracy of **0.9684, 0.0811** Loss, and **0.984** AUC, as the best results by freezing the first 12 layers of VGG19 and 560x560 shape. For the Precision, 256x256 shape with all parameters trainable gave the best result, which was **0.877.** 560x560 shape, by training the whole network, we had a **0.777** Sensitivity result. Finally, by using a 256x256 shape with training the whole network, we had the **0.991** best result for Specificity.

EfficientNet

Table 2. EfficientNet Results

Network	**EfficientNet B0**	**EfficientNet B0**	**EfficientNet B7**	**EfficientNet B7**
Shape	560	256	560	256
Ori Params	4,007,548	4,007,548	63,786,960	63,786,960
Model Params	22,805,921	22,805,921	39,145,945	39,145,945
Ori Layers	237	237	813	813
Model Layers	246	246	642	642
Time(sec)	16	7	30	14
Epoch Stopped	216	158	219	232
Loss	0.1259	0.2164	0.2115	**0.1145**
Accuracy	**0.9639**	0.9549	0.9549	0.9612
AUC	0.960	0.942	0.942	**0.968**
Precision	0.874	0.880	0.880	**0.884**
Sensitivity	**0.685**	0.561	0.561	0.623
Specificity	0.991	**0.993**	**0.993**	0.992

In our experiment, we used two versions of EfficientNet, B0 and B7, with the shapes 256 and 560. EfficientNet had more layers than the other two pre-trained networks, 642 layers. B7 with the shape 256, gave the best results for Loss-**0.1145**, AUC-**0.968**, and Precision-**0.884**. B0, with the shape 560 on the other hand, had the best Accuracy-**0.9639**, and Sensitivity-**0.685.**

ResNet

Table 3. ResNet Results

Network	**ResNet50V2**	**ResNet50V2**	**ResNet152V2**	**ResNet152V2**
Shape	560	256	560	256
Ori Params	23,519,360	23,519,360	58,187,904	58,187,904
Model Params	26,918,273	26,918,273	61,918,273	61,918,273
Ori Layers	190	190	564	564
Model Layers	228	228	602	602
Time(sec)	15	9	25	16
Epoch Stopped	161	181	166	300
Loss	**0.088**	0.0975	0.0888	0.0921
Accuracy	0.9663	0.9653	**0.9666**	0.9663
AUC	0.978	0.973	**0.979**	**0.979**
Precision	0.862	**0.876**	0.86	0.847
Sensitivity	0.731	0.693	0.738	**0.739**
Specificity	0.989	**0.991**	0.989	0.988

ResNet152 V2 with the shape 560, performed the best accuracy, **0.9666,** and AUC **0.979.** ResNet50 V2 had the best Loss value, **0.088** with the shape 560.

In this paper, we used the U-NET architecture with transfer learning to segment the eye blood vessels. When the test data is different from the training data, and also if there are not enough test samples, fine-tuning might be the best approach as we showed. In the first layers of the Convolutional Neural Networks, more generic features are learned, so freezing them will not have a strong effect. Moreover, this will decrease the training time.

For the future experiment, working with a network that is specifically trained for segmentation or blood vessels, surely will increase the accuracy. Also, collecting more data on eye blood vessels will have a big impact on this.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, pp. 84–90, 2012.

[2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[3] A. Krestanova, J. A. N. Kubicek, and M. Penhaker, "Recent Techniques and Trends for Retinal Blood Vessel Extraction and Tortuosity Evaluation : A Comprehensive Review," *IEEE Access*, vol. 8, pp. 197787–197816, 2020.

[4] E. Y. K. Ng, U. R. Acharya, A. Campilho, and J. S. Suri, *Image Analysis and Modeling Ophthalmology*. CRC Press Taylor & Francis Group, FLL, 2014.

[5] WHO, "Blindness and vision impairment: Refractive errors," *World Health Organization*, 2021. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment%0Ahttps://www.who.int/news-room/q-a-detail/blindness-and-vision-impairment-refractive-errors.

[6] M. M. Fraz *et al.*, "Blood vessel segmentation methodologies in retinal images - A survey," *Comput. Methods Programs Biomed.*, vol. 108, no. 1, pp. 407–433, 2012.

[7] E. Emary, H. M. Zawbaa, A. E. Hassanien, G. Schaefer, and A. T. Azar, "Retinal blood vessel segmentation using bee colony optimization and pattern search," *Neural Networks(IJNN)*, pp. 1001–1006, 2014.

[8] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng, and G. Yang, "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," *Neurocomputing*, vol. 149, pp. 708–717, 2015.

[9] H. Fu, Y. Xu, D. W. K. Wong, and J. Liu, "Retinal vessel segmentation via deep learning network and fully-connected conditional random fields," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2016-June, pp. 698–701, 2016, doi: 10.1109/ISBI.2016.7493362.

[10] Z. Jiang, H. Zhang, Y. Wang, and S. Ko, "Retinal blood vessel segmentation using fully convolutional network with transfer learning," *Comput. Med. Imaging Graph.*, vol. 68, pp. 1–15, 2018.

[11] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3320–3328, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks,"

ECCV 2016. Lect. Notes Comput. Sci., vol. 9908 LNCS, pp. 630–645, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.

[14] and T. B. Olaf Ronneberger, Philipp Fischer, "UNet: Convolutional Networks for Biomedical Image Segmentation," *Med. Image Comput. Comput. Interv. – MICCAI 2015*, vol. 9351, 2015.