

COMP9517 Group Project Report

z5102903 Xiaowei Zhu
z5146117 Yue Wu
z5028039 Yuanyuan Kong
z5195745 Yingkun Tan
z5192332 Yubing Ding

Abstract—The goal of the subject is to separate four different retinal lesions and separate blood vessels. For the first task we apply SVM, VGG, U-NET methods. For the separation of blood vessels, ASF (traditional image processing) and U-NET are used.

Keywords—retinal lesions, blood vessels, U-NET, SVM, VGG, Fundus photo image processing

I. INTRODUCTION

In modern society, Diabetic Retinopathy is a common disease which leads to vision impairment and affects the major working population of the world [2]. According to Karst et al., different kinds of retinal lesions are found in about 50% of patients after 10 years diabetes, more lesions could present in about 77%-90% of patients suffer 15 or more years diabetes [1]. The incidence of these retinal lesions increases with the age and the duration of diabetes. However, if these diabetics can be diagnosed and receive timely standardized treatments, most of them can eliminate the danger of blindness.

In the past decades, with the rapid development of the medical image processing technology, medical image analysis has become one of the most important research directions in the field of clinical medicine [5]. During the clinical examination, the ophthalmologist may record visible findings in the eyes of the patient and then use all these findings to infer the medical details of the patient. The traditional Fundus Fluorescein Angiography (FFA) could cause some side effects “ranging from itching to a life-threatening emergency” because of the use of contrast agent [6]. The new medical image processing technology greatly strengthens the advantages of fundus examination, facilitates the patients with hypertension and diabetes. It also has the practical significance for the diagnosis and treatment of diabetes [7].

There are two aims of our project. Firstly, it is to segment four retinal lesions associated with diabetes based on iDRiD database. This database is from India. Images in this database are all in a pixel level and show some typical types of retinal lesions. These lesions are relatively clear which is really suitable to do the analysis of early retinal lesions associated with diabetic retinopathy [2]. Secondly, it is to segment the blood vessel based on DRIVE database. This database is from Holland. This database is made up of 40 randomly chosen retinal images from 400 diabetics ranging from 25 to 90 years of age. In these 40 images, “33 do not show any sign of diabetic retinopathy and 7 show signs of mild early diabetic retinopathy” [4]. As it is found that some diabetic retinopathy could lead to the growth of new blood vessels in the retina in the advanced stage, this database is really suitable to do the blood vessel segmentation [3]. Based on the research and implementation of this project, we tried to locate positions of these retinal lesions and determine the types of the lesions by analyzing retinal images. we learned various methods of image segmentation and understood the significance of this project for clinical medicine.

II. LITERATURE REVIEW

Before starting our project, we did some researches to learn some most frequently used methods in retinal lesion segmentation and blood vessel segmentation. Firstly, it is the SVM method for diabetic retinopathy detecting. This method is proposed by Malathi et al. in 2018 and generally based on another segmentation method called “shrinking edge-mark” [9]. In this method, the veins and other areas of the fundus are labelled separately, then some parameters obtained based on some formulas. For the last step, according to these parameters, using the RSVM algorithm to do the final classification and do the detection of lesions.

Secondly, it is the VGG method for diabetic retinopathy detecting. VGG is a new algorithm proposed in recent years, and it is praised by many people after it just presented. Based on the image the user wants to do the segmentation, this algorithm trains a mature model which has been trained many times again, and then make minor adjustments many times to achieve accurate segmentation of the target lesions [11].

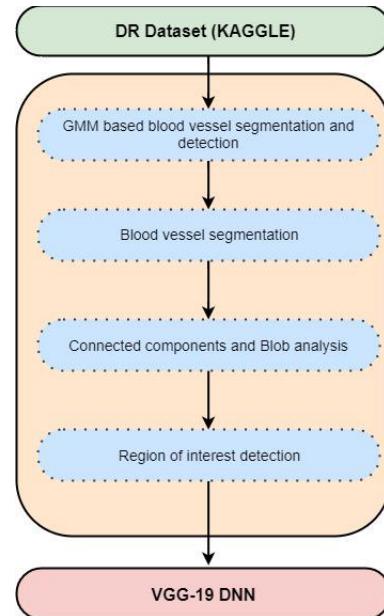


Fig. 1. Retinal blood vessel segmentation and detection [11]

Thirdly, it is the U-net method for diabetic retinopathy detecting and blood vessel segmentation. Although the U-net method is proposed in 2015, it is still a relatively mature algorithm in biomedical image segmentation and has implemented by many persons [5]. U-net method has two obvious features which are U-shaped structure and skip connection. As the semantics of a medical image is simple and the structure of a medical image is fixed, it is very suitable for U-shaped network do the semantic segmentation of medical images. However, there is one thing need to be mentioned is that the accuracy of U-net method sometimes could decrease because of the loss of detailed things. Moreover, as it is not

easy to generate appropriate subjects for a medical image database, there are not enough quantities of data in the database. The accuracy could also affect by the size of model and the quantity of parameters. In general, U-net has a good performance in many kinds of medical image segmentations.

Fourthly, it is a multiscale algorithm for blood vessel segmentation. This method at first uses Gaussian Pyramid to get the fundus image in green channel with different resolution levels. Then it uses Hessian Matrix to do the neighborhood analysis of each pixel to generate images of enhanced vessels. Finally, it does the fusion of analyzed images and obtain a relatively accurate segmentation result [13]. This method looks very simple, however, the actual result is less than satisfactory.

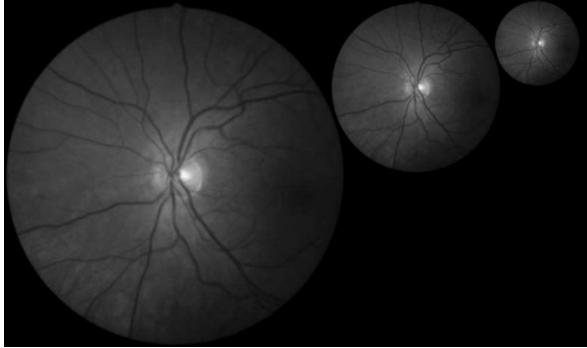


Fig. 2. The Gaussian resolution hierarchy generated from the green channel by decreasing resolution [13]

III. METHOD

Based on the research we did before, we tried to develop our own methods to solve these two segmentation tasks.

A. SVM

Given a set of linear separable training examples, a SVM model can find a clear gap that is as wide as possible. If the training examples are not linear separable, SVM can apply a kernel trick and map those example points to a higher dimensional space where they are linear separable. Compared to other models, SVM is relatively memory efficient and works relatively well when there is imbalance in the dataset.

B. VGG-19

Very Deep Convolutional Networks (VGGNet) [14] is invented by Visual Geometry Group from University of Oxford. It beats the GoogLeNet and won the localization task in ImageNet Large Scale Visual Recognition Challenges (ILSVRC) 2014. In this task, a pretrained VGG-19 model is introduced. The table below is from the original VGGNet paper and in the last column is the original structure of VGG-19.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv1-256	conv3-256 conv3-256	conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 3. VGGNet

This VGG-19 model has been trained on more than a million images from ImageNet database [15]. It is 19 layers deep which can classify images into 1000 categories based on the object presents. The size of input image should be 224*224 px.

C. U-Net

U-Net is a traditional and significant convolutional neural network, which was developed to resolve image segmentation in biomedical area. This network is combined by multi fully convolutional network. Usually, it is constructed by one up sampling part and one down sampling part with large number of feature channels. This channels allow the network to propagate information to the next higher resolution layers.

D. Adaptive Histogram Equalization (AHE)

Images from database DRIVE itself is unevenly illuminated and the contrast is poor. Histogram equalization can increase the global contrast. By doing this areas of lower contrast can gain a better local contrast and be more easy to segmented. Histogram Equalization achieve this by spreading out the most frequent intensity values. One drawback of this technique is that the calculated values are based on the global, which increases the noise of the background.

AHE is an enhanced version based on the original histogram equalization. For each individual small area, pixel histogram is calculated and this value is used to redistribute the pixel value. Therefore, it can improve the local contrast while enhancing the sharpness of the edges of the image.

A disadvantage of the AHE algorithm is that if the image block information in the rectangular region is relatively flat and the gray scale is close, the gray histogram is pointed, then excessive amplification noise may occur in the process of histogram equalization. CLAHE, a contrast-limited adaptive histogram equalization algorithm can effectively limit the noise amplification situation. In CLAHE, contrast clipping must be used for each small area. The CLAHE achieves the purpose of limiting the amplification by cropping the histogram with a predefined threshold before calculating the CDF(Cumulative Distribution Function). This limits the slope of the CDF and therefore limits the slope of the transformation

function. Cropped part was speeded evenly over the rest of the histogram. As shown below.

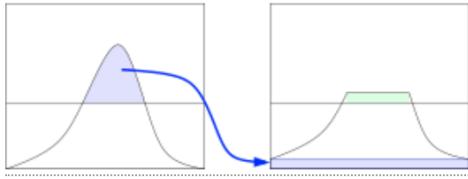


Fig. 4. Peak cutting in CLAHE

E. Morphological Processing

The main morphological processing techniques used are dilation, erosion, opening and closing. In general, erosion shrink images while dilation expands an image. An essential use of erosion is to move noise at the edge of the output image. Opening and closing are mainly used in alternate sequential filtering. We use alternate sequential filtering to smooth image. Smooth is obtained by the use of morphological opening and closing with increasing kernel size. Opening can smooth image contours while Closing tends to narrow smooth sections of contours, filling holes and gaps. Pixel value of background would not change a lot after smooth but pixel value of contours which has high contrast with background will be concentrated. Calculating the difference between origin image and smoothed image can show blood vessel trace. So, combining opening and closing can achieve edge detection, noise removal and background removal. Let $f(x, y)$ be a gray scale function and B be a binary structuring element.

$$\text{Opening: } f \odot B = (f \ominus B) \oplus (B).$$

$$\text{Closing: } f \odot B = (f \oplus B) \ominus (B).$$

$$\text{Dilation: } (f \oplus B)(x, y) = \max\{f(x-s, y-t) \mid (s, t) \in B\}.$$

$$\text{Erosion: } (f \ominus B)(x, y) = \min\{f(x+s, y+t) \mid (s, t) \in B\}.$$

Fig. 5. mathematical expressions for morphological method

Effect of opening and closing show as below.

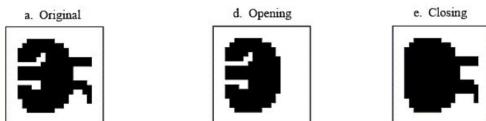


Fig. 6. Opening and closing image

F. Thresholding

Threshold can be used to binarize image and remove no needed details. It is important to choose a suitable threshold. If the value is too large, some unneeded details will be preserved. When the value is too small, a part that needs to be separated, such as a blood vessel, cannot be extracted.

G. Extra OpenCV Methods

- cv2.findContours can be used to find all contours and it return a list of contours. By iterate this list, all small contours which are not connected to blood vessel are removed.
- cv2.bitwise_and is an AND operation of binary data, that is, binary AND operation for each pixel value of an image. When we have or draw a mask, this function

can get the difference between the two images to remove unwanted noise.

H. Multiscale Method

As the multiscale algorithm mentioned in the research paper seems relatively simple, we decided to implement it. First of all, we generated two more images that their resolution levels are totally different from the original retinal fundus image by using the Gauss pyramid method. Secondly, we used the image with the highest resolution from the first step to generate the Hessian matrix of the 3×3 neighborhood around each pixel. Then we try to find the maximum and the minimum eigenvalues in each matrix and take these two values into a formula to calculate the new value of each pixel. However, it seems that there exists some errors when we tried to find the local maximum and minimum eigenvalues in each matrix. There are many noises shows on the output images. We tried a lot of modified methods to solve this problem, as they all failed, we finally determined to give up this algorithm.

$$P_{vessel} = 1 - a_l/a_h$$

Fig. 7. a) P_{vessel} is the result value for the given pixel b) a_l is the lower eigenvalue c) a_h is the higher eigenvalue [13]

IV. EXPERIMENTAL SETUP

Three approaches have been implemented in the lesion segmentation task. They are SVM, VGG-19 and U-Net.

A. SVM Classification

Run a binary classification on every pixel, to see if it belongs to the lesion class or background pixel. To do the classification on pixel, first derive a useful pixel-level feature vector for every pixel of the image. Pixel level features normally can be divided into two categories which are color and texture. Color spread is normally used as features to represent color and contrast, changes in intensities and GLCM are normally used to represent texture.

To get a pixel level feature, make a 5×5 window and scan through the image to do a convolution. For each window, calculate the average pixel intensities in RGB three channels and make that to be the feature vector of the center pixel in that window. Therefore in the end, every pixel has a 3×1 feature vector which represents the average pixel intensities in the neighbourhood.

If an image's size is 400×500 , then there are 200000 pixels in that image and then we have 200000 feature vectors as input to the classifier

To do the classification, SVM (Support Vector Machine) is used as the classifier. SVM works well for binary classification and when it's combined with a kernel method, non-linear data is mapped to a higher dimensional space where it's separable. Then, train the SVM model. Since one image contains a large number of pixels, not all the images are used in the training set to train the model since that will take too long. Five images are chosen from training set to train the model.

After training the model, the model is used to predict the class of each pixel for every image of the test set. Since there are four different types of lesions here, the above process is repeated for four times, once for each.

B. VGG-19

The task has four lesions that need to segment. The pretrained model presents to the transfer learning four times for each lesion. The original training images for each lesion are divided into two files, original images and annotations. Original images and annotations files are divided into two parts, training and validation. Below is the illustration of training data file structure.

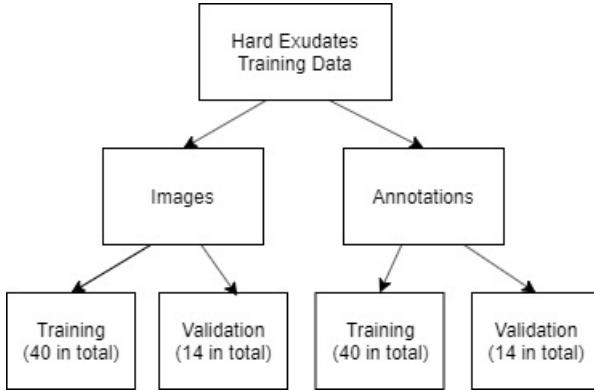


Fig. 8. File structure

The experiment is a transfer learning process, implements in python2 with TensorFlow and OpenCV. Numpy is introduced to modify TIF files and Scikit-learn packages are used for evaluation. The pretrained VGG-19 is adapted to train on our dataset. The learning rate is set to 10^{-4} to reduce large impact on the already adjusted weights of model. All input images have been resized into 224*224 px to be fit in the model. This code is modified from Shaw's work with extra features added. The original last convolution has been replaced by three new convolutional layers, followed by three ReLUs. Then the classified image is upscaled from 1*1*4096 to its original size 224*224.

There are 10000 rounds of training for each lesion, images have been shuffled and input 3 of them each round, then VGG-19 makes pixel wise classification for each input. Below is a comparison between using trained VGG-19 directly and result after 10000 times of training.

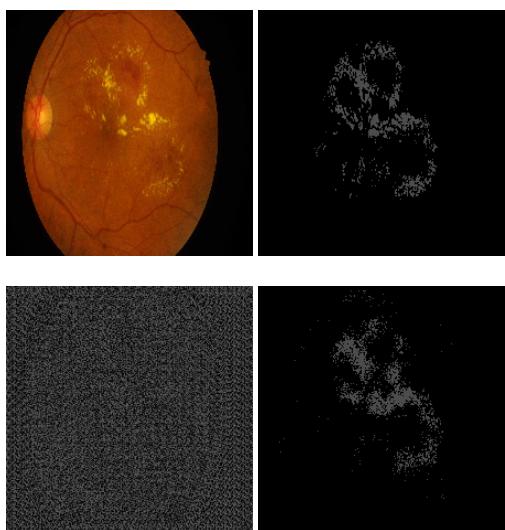


Fig. 9. a) Original image after resize b) Ground truth after resize c) Using trained VGG-19 directly d) Result for 10000 of training

The experiment makes significant improvement after training. Training takes approximately 2 hours for each lesion

and one trained model is saved every time after 5000 rounds of training. For testing, organize the testing images into the same structure as the training images, but put all images in the validation file and leave the training file empty. Change the last `tf.flag` into test mode, load the latest model and test data, the program would make a prediction on the test images based on the given model.

C. U-Net

1) Image Preprocessing

In this part, firstly, we pad the original image and every lesion image to 512*512. Simultaneously, we also pad the test image to 512*512 (Here we show the training and test image of hard exudates)

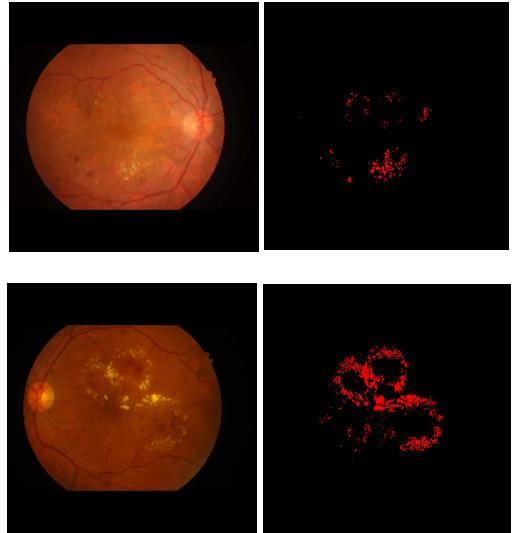


Fig. 10. Result after padding

After we pad all the images, we use Keras Image Data Generator to do data augmentation. We did these processing including Image rotation, Image width & height shift, Image zooming, Image horizontal flipping, Resizing image to 512*512, and Normalization.

In addition, we set up these parameters for Keras data generation method: `rotation_range = 0.2, width_shift_range = 0.05, height_shift_range = 0.05, shear_range = 0.05, zoom_range = 0.05, horizontal_flip = True, and fill_mode = 'nearest'`.

2) U-Net model implementation and training

Then we input the data to a U-Net model, the structure of our U-Net model is “1 => 64 => 128 => 256 => 512 => 1024 => 1024 => 512 => 256 => 128 => 64 => 64 => 1”.

We add a dropout and ReLU after every Convolution layer.

Finally, we add one dimensional convolution layer and a sigmoid activation after the convolution layer.

The learning rate is set as $3e-5$, loss function is binary crossentropy, steps for each training epoch is 500 and total epochs are 5. Then, we deploy our programming in Google Colab and did the training.

In the next step, we use the same logic to deal with the test images and then implement the prediction. However, the outcome is pretty bad.

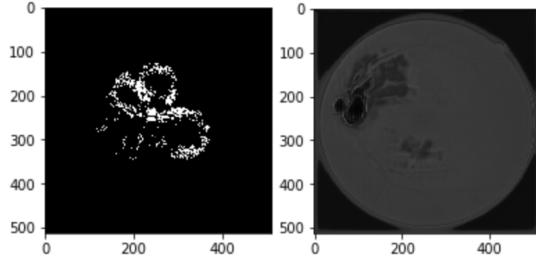


Fig. 11. Predicted result from U-Net

```

2019-11-18 01:40:42.740274: I tensorflow/stream_executor/platform/default/dso_loader.cc:141] Successfully opened dynamic library libcuda.so.1 - 1074.213mb/step - loss: 0.0336 - acc: 0.9935
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name tf.Summary is deprecated.

Epoch 00001: loss improved from Inf to 0.03358, saving model to vessel_unet.hdf5
Epoch 2/5
500/500 [=====] - 93s 186ms/step - loss: 5.9524e-04 - acc: 1.0000
Epoch 00002: loss improved from 0.03358 to 0.00060, saving model to vessel_unet.hdf5
Epoch 3/5
500/500 [=====] - 93s 185ms/step - loss: 2.0757e-04 - acc: 1.0000
Epoch 00003: loss improved from 0.00060 to 0.00021, saving model to vessel_unet.hdf5
Epoch 4/5
500/500 [=====] - 93s 185ms/step - loss: 1.0199e-04 - acc: 1.0000
Epoch 00004: loss improved from 0.00021 to 0.00010, saving model to vessel_unet.hdf5
Epoch 5/5
500/500 [=====] - 93s 185ms/step - loss: 5.8577e-05 - acc: 1.0000
Epoch 00005: loss improved from 0.00010 to 0.00006, saving model to vessel_unet.hdf5

```

Fig. 12. Training on Collab

We can see that the accuracy from training process is 1, which means overfitting. We many need to do more data preprocessing, such as image smoothing to generate more data to avoid overfitting. At the same time, we also need to change the loss function weight to make distinguish background and lesion efficiently.

We plan not to provide the evaluation data here due to the performance of this U-Net method is terrible compared to SVM and VGG.

Next comes to the segmentation of blood vessel, two methods have been introduced.

A. ASF

Workflow chart is shown below. Traditional methods about extract blood vessel consists of three part which are image preprocessing, blood vessel recognition and image postprocessing.

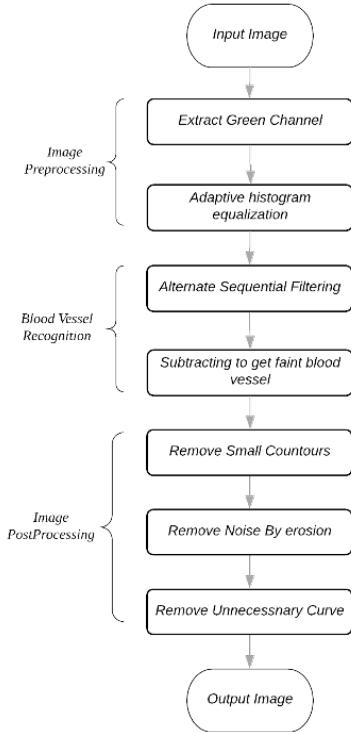


Fig. 13. Workflow of ASF

1) Image Preprocessing

We have tries both red and green channel picture. But the hemorrhages will affect the segmentation. So green channel was extracted because of better contrast and easy to segment.

After that, we apply CLAHE (Adaptive Histogram Equalization) to improve the local contrast. By doing this blood vessel can be better distributed on the histogram

The first picture in the picture below is the original picture, the second picture is the picture of the green channel, and the last picture is the picture after CLAHE.

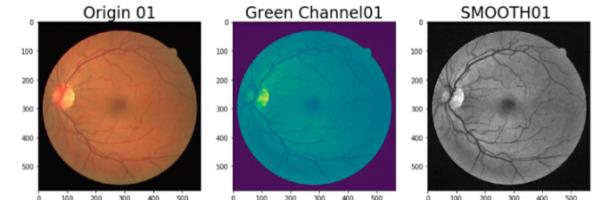


Fig. 14. Preprocessing in ASF

2) Blood Vessel Recognition

We apply ASF and it gives us an image with average intensity applied on it. By observing the histogram, we can see that the gray part of the values most are background and noise on image does not change much, most are still around 5000-6000. But the black part most are background and blood vessel are more concentrated. So, by subtracting the two figures, we can remove background and noise, get a faint trace of blood vessel.

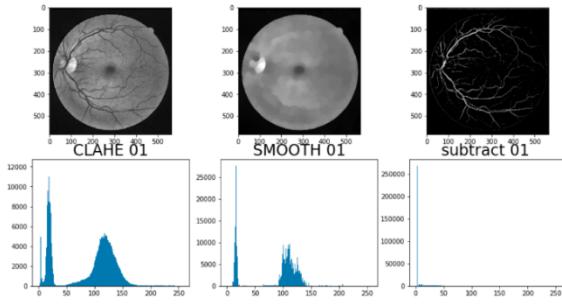


Fig. 15. Histogram for image after ASF

3) Image Postprocessing

Observing the picture we got from the previous step, there are still a lot of silky noises that are not connected to the blood vessels that we don't need. By using cv2.findContours, we can detect all contours and removes those small area contours. Then we binarize this image with a threshold T and get blood vessels segmented. Thresholding is used to remove unnecessary details.

The image below is the image obtained in the previous step, image with no small contours and the image after the binarize.

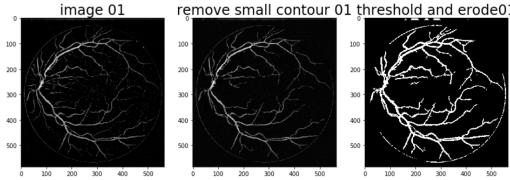


Fig. 16. PostProcessing for image after ASF

After these pretreatments, there are still unwanted curves that exist at the boundaries of the blood vessels. Since we already have a background mask, we can remove the curve by doing binary operations on the data of the two images as below.

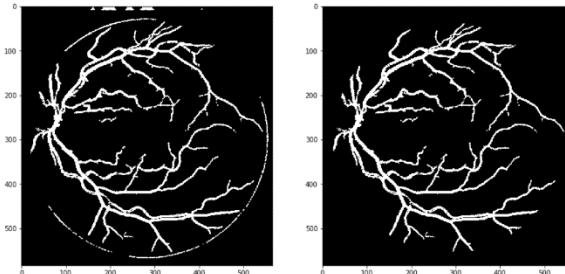


Fig. 17. Border remove for image after ASF

B. U-Net

1) Image Preprocessing

Firstly, we do the image data augmentation. The same logic as what we did in lesion segmentation in Task 1. We use the python package (ImageDataGenerator from Keras) to do the processes including image rotation, image width shift, image height shift, image zoom, image horizontal flip, image resizing, and image normalization

We have four different programming files. Data_generator.py is used to do the data generation for training data and testing data. Train_data.py is used to execute training process. Test_data.py is used to execute testing process. Une

t_model.py is used to store the U-Net model

2) Training and Blood Vessel Recognition

We also use a very traditional U-Net model here, but we change the input size to (512,512,1)

Here is the structure of our U-Net Model:

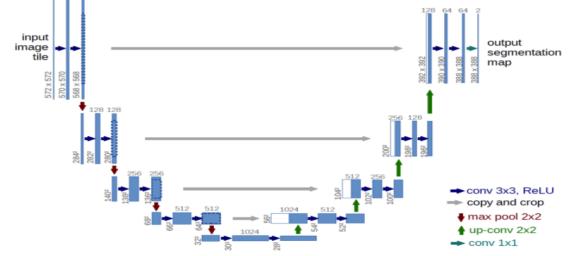


Fig. 18. U-Net Model

The upsampling structure is: $1 \Rightarrow 64 \Rightarrow 128 \Rightarrow 256 \Rightarrow 512 \Rightarrow 1024$. And the down sampling structure is: $1024 \Rightarrow 512 \Rightarrow 256 \Rightarrow 128 \Rightarrow 64 \Rightarrow 1$.

Every Convolution layer is followed by a ReLU activation. We also set a max pooling layer every two convolutional layers. In addition, we set a dropout with 0.5 drop rates between the up sampling and down sampling.

Here are the training process parameters, we set Learning rate: 3e-5, Loss Function: Binary Cross Entropy, epoch: 5 and Steps for each epoch: 600

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The n
Epoch 00001: loss improved from inf to 0.11103, saving model to vessel_unet_2.hdf5
Epoch 2/5
600/600 [=====] - 363s 605ms/step - loss: 0.0858 - acc: 0.9661
Epoch 00002: loss improved from 0.11103 to 0.08581, saving model to vessel_unet_2.hdf5
Epoch 3/5
600/600 [=====] - 363s 604ms/step - loss: 0.0815 - acc: 0.9675
Epoch 00003: loss improved from 0.08581 to 0.08154, saving model to vessel_unet_2.hdf5
Epoch 4/5
600/600 [=====] - 362s 603ms/step - loss: 0.0799 - acc: 0.9680
Epoch 00004: loss improved from 0.08154 to 0.07990, saving model to vessel_unet_2.hdf5
Epoch 5/5
600/600 [=====] - 363s 605ms/step - loss: 0.0782 - acc: 0.9685
Epoch 00005: loss improved from 0.07990 to 0.07817, saving model to vessel_unet_2.hdf5
```

Fig. 19. Training implementation on Colab

After the training, loss is improved from 0.07990 to 0.07817. And the training accuracy is 96.85%.

Then we store the training model to a HDF5 file.

In testing part, we also preprocess the test data following these rules. We deal with the Test original images and label images with the following these steps. Converting from BGR to GRAY, resizing image to (512,512), converting to float, normalization and reshaping to (512,512,1).

Generally, the aim of testing image processing is to make testing image satisfy the requirement of our training model.

3) Testing Image Prediction

We set up a new U-Net model and load the HDF5 file into our model, then do the prediction

V. RESULTS AND DISCUSSION

The evaluation and discussion of four lesions and blood vessel segmentation is included in this section.

Traditional accuracy calculation is not enough in these two tasks, since the targets are relatively too small compare to the background, precision and recall should reflect more. By combining and balancing the weight of precision and recall, F1 score makes better evaluations. Area under precision and recall curve (aucPR) is also introduced in the lesion segmentation task, since it is also relevant for unbalanced data.

A. SVM

A sample result for lesion segmentations and numerical evaluation is as following.

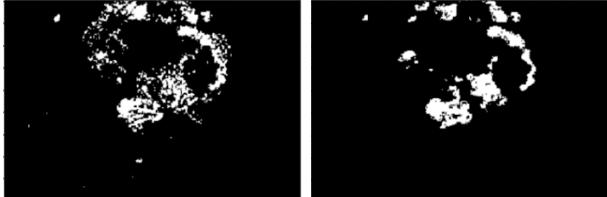


Fig. 20. (i) groundtruth for EX (ii) result from classification on training set

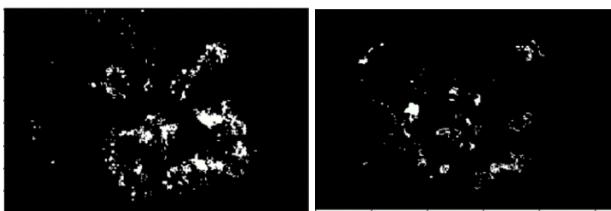


Fig. 21. (i) groundtruth for EX (ii) result from classification on test set

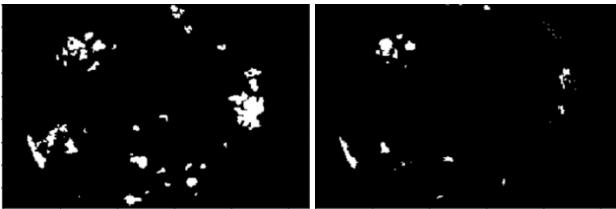


Fig. 22. (i) groundtruth for HE (ii) result from classification on training set

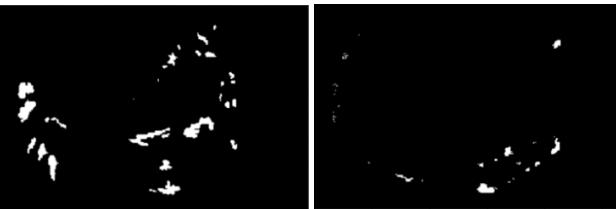


Fig. 23. (i) groundtruth for HE (ii) result from classification on test set

TABLE I. RESULT OF SVM

Lesions	Metrics			
	AUCPR	Recall	Precision	F1 score
EX	0.01	0.17	0.05	0.02
HA	0.05	0.01	0.01	0.01

Since this method failed on detecting SE and MA, so the result is not included. Overall, the result is not very good.

First, the feature is not meaningful enough so it's difficult to classify the two classes based on the average pixel intensities. For the future work, glcm should be considered as a possible feature. Also, radiomics can be used to extract a large amount of features as well. Second, since the two classes are not balanced, a weighted loss function should be used to apply more penalties to the class that's larger. Third, SVM doesn't perform well when the two classes are overlapping and not totally separable. The original images have some noises which can make the two classes not totally separable.

B. VGG-19

The following numerical result is obtained from comparison and evaluation between prediction and the ground truth.

TABLE II. RESULT OF VGG-19

Lesions	Metrics			
	AUCPR	Recall	Precision	F1 score
EX	0.19	0.17	0.21	0.17
HE	0.16	0.09	0.22	0.11
SE	0.03	0.01	0.06	0.02
MA	0.33	0.0	0.0	0.0

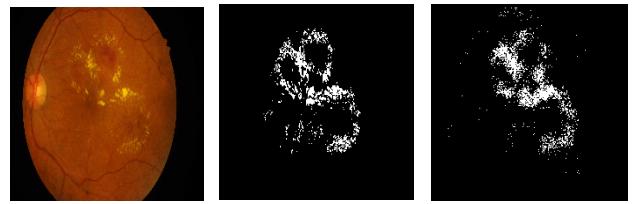


Fig. 24. a) Original image b) Ground Truth c) Segmentation result of hard exudates



Fig. 25. a) Original image b) Ground Truth c) Segmentation result of haemorrhages



Fig. 26. a) Original image b) Ground Truth c) Segmentation result of soft exudates

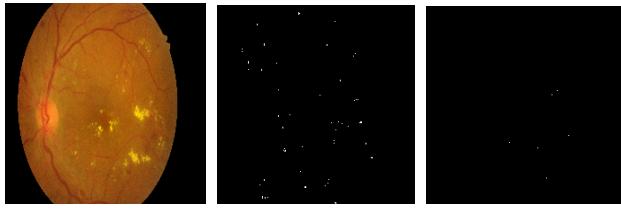


Fig. 27. a) Original image b) Ground Truth c) Segmentation result of microaneurysms

The model can make partial correct prediction on the hard exudates and haemorrhages but gives bad result on soft exudates and failed entirely on microaneurysms. There are several problems can cause this failure. Firstly, significant information is lost when resizing, especially the microaneurysms, since they are too small. Secondly, the pretrained VGG-19 is trained on a very different dataset which makes the transfer learning difficult. For more, there is no pre-processing done before training, so the lack of training data causes overfitting. The model has been trained on this task for 10000 times, the more times trained tend to give better and more detailed result sometimes as shown below.

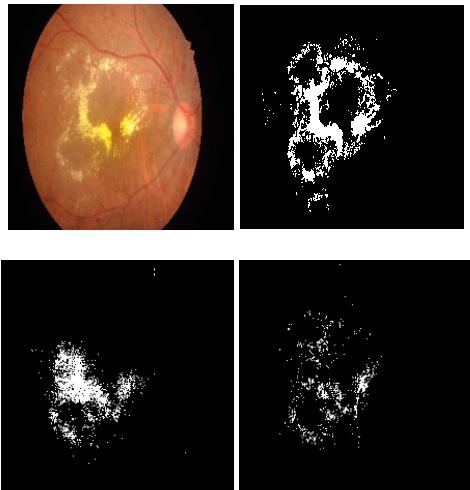


Fig. 28. a) Original image b) Ground truth c) Result after 10000 training d) Result after 15000 training

There is a training times exists that produces the best result, but this is not tested in this project because of the limitation of time.

C. Comparison

Below is a comparison between SVM and VGG-19 on hard exudates. Almost every attribute improved significantly but the overall performance of VGG-19 is still not satisfactory.

TABLE III. COMPARISON BETWEEN METHODS OF LESION SEGMENTATION

	SVM	VGG-19
aucPR	0.01	0.19
Precision	0.05	0.21
Recall	0.17	0.17
F1 score	0.02	0.17

Here comes to the part of blood vessel segmentation, we compare ASF and U-Net Deep Learning methods

A. ASF

The following images are the original image, label and true prediction values.

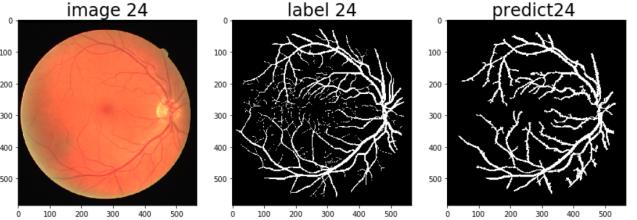


Fig. 29. Results for ASF method

As can be seen from the above figure, the general framework of the blood vessels can be identified from the diagram, but the smaller branches are missing. The reason for this is because the contrast difference of the original image is not obvious, and the small branches of the blood vessels cannot be well displayed after we subtract two images. And some branches are removed as small contours because they are not connected to blood vessel.

Accuracy	Precision	Recall	F1 score
0.940	0.711	0.638	0.673

From the data, due to the background also in calculation, the accuracy is high, but the recall and F1 score are not ideal. This means that the prediction is not accurate enough. We can then continue to tune the parameters of postprocessing and smooth for further experimentation.

B. U-NET

The results of our prediction are pretty good. In the image below, the first column is original retinal images, the second column is ground truth, the third column is the result that we get.

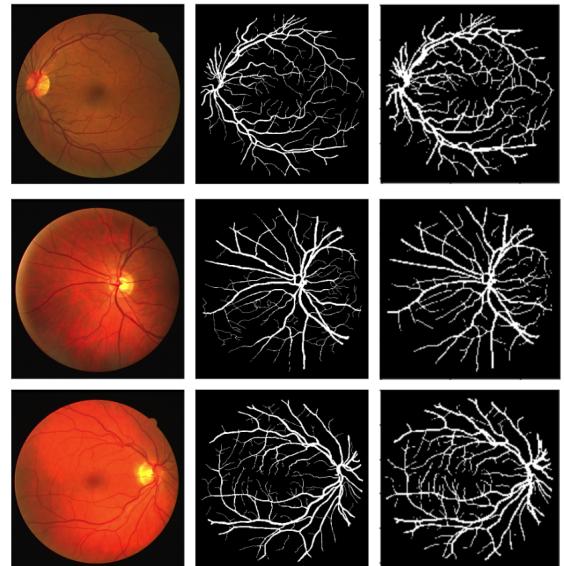


Fig. 30. Results for Task2 Vessel

It's very clear that, the predicted results from U-Net model almost cover every detail of blood vessel compared to the label we have.

We use metrics from Sklearn to evaluate our results from U-Net deep learning and get these answers.

TABLE IV. EVALUATION TABLE OF U-NET METHOD

Accuracy	95.2%
Precision	68.63%
Recall	91.6%
F1 Score	78.45%
AUCPR	80.5%

Above is the evaluation we got. This clearly show that the U-Net deep learning segmentation method is very accurate and efficient.

C. Comparison

In this part we compare the blood vessel segmentation result from ASF and U-Net.

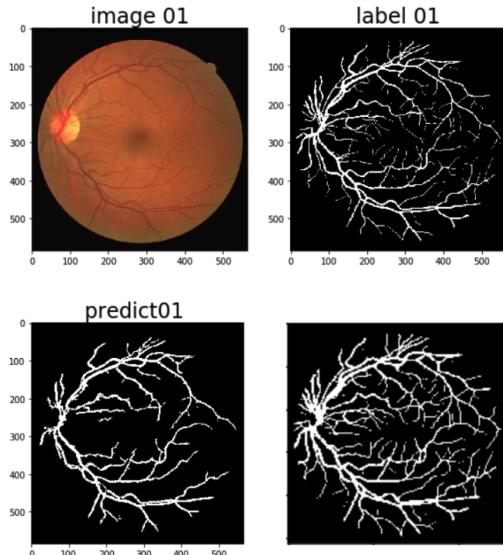


Fig. 31. images comparision for task2

It is very clear that that result from U-Net segmentation cover more details than the outcome from ASF method, and the similarity between U-Net prediction and ground truth is higher than the similarity between ASF prediction and ground truth.

TABLE V. COMPARISON BETWEEN METHODS OF BLOOD VESSEL SEGMENTATION

	ASF	U-net
Accuracy	0.94	0.95
Precision	0.71	0.67

Recall	0.64	0.92
F1 score	0.67	0.78

The comparison table shows that the precision of U-net is higher than ASF, which means U-Net take more area as the blood vessel characters. We can see this situation from the image below, the predicted result from U-net seems to be more noises.

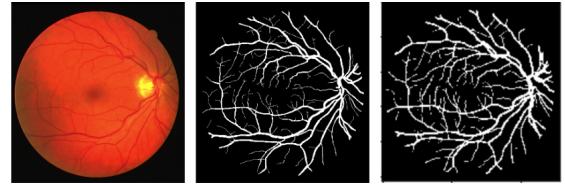


Fig. 32. Results of U-Net(For Blood Vessel)

U-net is better than ASF method. Especially for Recall and F1 score. As discussed before, the accuracy doesn't mean too much in this situation. Because the majority of image background is black which makes no difference for the accuracy.

VI. CONCLUSION

A. Lesion Segmentation

In the lesion segmentation, three method has been implemented with VGG-19 gives the relatively better result. Relatively, precision is higher than recall which means that the lesions that are identified as lesions are very likely to be lesions in the ground truth label, but the models are unable to identify a large region of lesions, treating them as the background. The overall performance is still not desirable and certain problems should be fixed in the future work. Preprocessing should be applied to the original images to enhance the contrast and data augmentation is certainly needed in VGG. With those techniques used, the outcome would be improved significantly.

B. Blood Vessel Segmentation

In task2, we successfully used two methods to separate the blood vessels. The ASF method is fast training and can obtain a basic vascular frame, but the results are not accurate enough. Based on the mechanical learning U-Net method, the training takes a long time, but the results are ideal and the details of the blood vessels are clearly visible. The first method parameter is relatively fixed. In the case of complex contrast and brightness in the fundus, the work does not work very well. However, the characteristics of the U-Net still extract the characteristics of the blood vessels.

CONTRIBUTION OF GROUP MEMBERS

Lesion Segmentation:

- 1) *SVM*: Yue Wu, Yingkun Tan, Xiaowei Zhu
- 2) *VGG*: Yubing Ding, Yingkun Tan, Yuanyuan Kong
- 3) *U-Net*: Xiaowei Zhu, Yue Wu, Yubing Ding

Blood Vessel Segmentation:

- 1) *ASF*: Yingkun Tan, Yubing Ding
- 2) *U-Net*: Xiaowei Zhu, Yuanyuan Kong
- 3) *Multiscale*: Yuanyuan Kong, Yue Wu

REFERENCES

- [1] S. G. Karst, J. Lammer, S. H. Radwan, H. Kwak, P. S. Silva, S. A. Burns, L. P. Aiello and J. K. Sun, "Characterization of In Vivo Retinal Lesions of Diabetic Retinopathy Using Adaptive Optics Scanning Laser Ophthalmoscopy," *International Journal of Endocrinology* (2018).J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [2] Indian Diabetic Retinopathy Image Dataset (IDRiD), 2018.
- [3] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, 2004, vol. 23, pp. 501-509.
- [4] A. Hoover, "STARE database." 1975.
- [5] G. Jiang, W. Qin, S. Zhou, C. Wang, "State of the Art in Medical Image Segmentation." 2015.
- [6] M. Andreucci, R. Solomon and A. Tasanarong, "Side Effects of Radiographic Contrast Media: Pathogenesis, Risk Factors, and Prevention," *BioMed Research International*, 2014.
- [7] S. Han, "The Analysis of Character for Blood Vessel of Ocular Fundus Images." 2010.
- [8] O. Ronneberger, P. Fischer, T. Brox "U-Net: Convolutional Networks for Biomedical Image Segmentation." 2015.
- [9] K. Malathi, R. Nedunceljan, "A recursive support vector machine (RSVM) algorithm to detect and classify diabetic retinopathy in fundus retina image", *Computational Life Sciences and Smarter Technological Advancement: Edition: II*, 2018.
- [10] N. Singh and L. Kaur, "A survey on blood vessel segmentation methods in retinal images." 2015.
- [11] M. Mateen, J. Wen, Nasrullah, S. Song and Z. Huang, "Fundus Image Classification Using VGG-19 Architecture with PCA and SVD", *Symmetry*, 2018.
- [12] X. Wang, W. Li, B. Miao, J. He, Z. Jiang, W. Xu, Z. Ji, G. Hong and Z. Shen, "Retina Blood Vessel Segmentation Using A U-Net Based Convolutional Neural Network." 2018.
- [13] Budai, Attila, G. Michelson, and J. Hornegger, "Multiscale Blood Vessel Segmentation in Retinal Fundus Images." In *Bildverarbeitung für die Medizin*, pp. 261-265. 2010.
- [14] K. Simonyan, A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION." 2018.
- [15] *ImageNet* <http://www.image-net.org>