Vessel Segmentation of Human Ophthalmoscopy Images by Using a Convolutional Neural Network

O. Akdag (0842508), T.P.A. Beishuizen (0791613), A.S.A. Eskelinen (1224333), J.H.A. Migchielsen (0495058) and L. van den Wildenberg (0844697)
Faculty of Biomedical Engineering, University of Technology Eindhoven, 5612 AZ Eindhoven,
The Netherlands

Abstract— **Determining the changes in retinal vasculature is crucial for diagnosis of diabetic retinopathy. To get information from the vessel geometry, the vessels from an ophthalmoscopy image have to be delineated, that is, segmented. Previously, the segmentation process has been done manually by labeling vessel and non-vessel pixels by hand, but that is time consuming, inefficient, prone to inter-observer variation. Altogether, it is a strenuous task especially in the clinical world where the amount of retinal image data is enormous. In order to track the changes in vasculature in time fast, automatic segmentation methods have been developed by using machine learning. In this study the segmentation was done by training a Convolutional Neural Network with retinal images from the DRIVE database. Various network parameters were studied and a comparison between two different Convolutional Neural Network was attempted. When comparing to manual segmentation, which was set as ground truth during the training of the network, the quality of the pictures were good enough to extract big vessels, but the smaller ones were lost in the image noise.**

I. INTRODUCTION

People with type I or type II diabetes have the chance of suffering from diabetic retinopathy. The vascular morphology changes in the retina for these people can eventually lead to blindness. Diabetic retinopathy is more likely to develop the longer a person has diabetes.[1] Ten percent of all diabetic patients have diabetic retinopathy, which is the primary cause of blindness in the Western world.[2] By scanning the retina in an early stage, it is possible to treat the person to prevent this eye condition. The screening for this eye disease can be facilitated by automating a part of the process.[3]

By scanning the retina, a photograph of the back of the eye is taken. The obtained image contains the blood vessels in the retina in color. Segmentation means separation and marking of certain regions of interest, and in case of retina this means delineation of the vessels. Segmentation of the vessels is still mostly done manually by different observers. The segmented images can then be used for analysis to determine certain morphological changes of the retina, such as changes in width of the vessels, color, reflectivity, tortuosity, abnormal branching or the occurrence of vessels of a certain width.[2]

Manual segmentation of the images is time consuming, inefficient and the resulting vessel segmentation may include some inter-observer differences. Deep learning is a promising machine learning tool to overcome these drawbacks. Deep learning can be used to train a program to segment these newly taken images of the retina. In this study, a deep learning algorithm consisting of a Convolutional Neural Network (CNN) and an image preprocessing step has been developed to automate the image analysis of these images. This could help speed up the screening of the morphology of the vessels in the human retina of diabetic patients. The proposed program will be trained by using photographs from the DRIVE (Digital Retinal Images for Vessel Extraction) database that have been obtained during a diabetic retinopathy screening program in the Netherlands.[2]

For image recognition and finding regions of interest in medical images, one of the most widely used methods are CNNs. Instead of using predefined kernels (sets of connection weights used by the units in feature maps), CNNs learn data-specific kernels, which are used to extract local information from the images. Because of weight sharing property of CNNs, they are not so demanding to computer as traditional neural networks.[4] During the years of extensive study on CNNs, a lot of different architectures of training nets have been proposed. Each of these have their pros and cons and are used to tackle different kinds of medical image processing problems. In general, it is probably not wise to invent a new architecture for a problem, but one should take a look at the existing architectures, which might indeed be more efficient, accurate and robust.

For most nets it's common that the input image is divisible by number two several times (e.g. input image size 32x32, 64x64 or 92x92 pixels). Additionally, it's preferable to stack several small kernel convolutional layers rather than using one larger kernel on one layer. This is because several layers make features (*e.g.* edges of vessels) more expressive and because it is computationally more effective.[6]

One of the simplest and easiest of CNNs, also used in this study, to implement is the LeNet-5 which consists of 7 layers. At first there are alternating convolutional layers (kernel is slided, *i.e.* convolved, with the image and dot products are computed) and subsampling layers (reduce spatial size of the representation to reduce the amount of parameters and hence control overfitting), two of each. These layers are followed by two full connection layers (full connections to all activations in the previous layer) and finally the pixel classification is done on output layer with Gaussian connection.[4] Other net architectures are for example AlexNet, GoogLeNet and

ResNet.

Another CNN used during this study is the ConvNet. It has much resemblance with the LeNet-5, but has more convolutional layers and more feature maps, which is more likely to give better results than the former CNN. The complete architecture will be discussed in the Methods.

In the following sections the used materials, *i.e.* the used retinal images, and methods for implementing the vessel segmentation with CNN are thoroughly explained. Subsequently we present the results of segmentation our algorithm yields, and ultimately we sum up the conclusions of this study.

II. MATERIALS

A. Image data

In this study the ophthalmoscopy images from human fundus were obtained from the DRIVE database.[2] The data is freely available for research purposes and for people to test their vessel segmentation algorithms. The dataset consists of a training set and a test set, both containing the 20 original pictures of the retina with the masks corresponding to the field of view (FOV) of the ophthalmoscope. From the 20 images in training set, 19 were used for training and 1 for validation. For the training images, one manual segmentation of the vasculature is available and can be used to train the CNN: supervised learning can be done. Two manual segmentations of the vasculature are available for the test images. One of these manual segmentations is considered as the ground truth, which can be used to evaluate the accuracy of the algorithm. The other set of manual segmentations can be used to compare with the output of this algorithm.

B. Programming environment

The Deep Learning algorithm implementation for these images is done with Python. Anaconda is used as a platform for the Python implementation due to useful packages being directly available with Anaconda. For the creation of the algorithm, the two packages Lasagne and Theano were used, as well as basic packages, such as Numpy, Matplotlib and Image for matrix usage, image visualisation and image processing, respectively. The code was written using Jupyter notebook, a straightforward coding platform it is easy to quickly see the output and test different parts of the programs with independently of the rest of other parts of the script. For the actual running of the algorithm a GPU server is used, as it is much quicker and has greater memory for training the network compared to CPU-driven training.

III. METHODS

A. Preprocessing

Before training of the network some preprocessing has to be done on the image data. Firstly, the necessary packages are imported and the used images - the original images of the retina, the masks as well as the manual segmentation of the vasculature - are loaded. The images are patched and then converted into matrices and gray-scale images, which gives the possibility to perform calculations with.

The patch size is determined to be 32x32 pixels for the LeNet-5 and 31x31 pixels for the ConvNet, and all possible patches of this size are extracted from all images. Patches on the edges of the image, so patches with pixels outside the FOV of the image, are removed from the dataset. This is based on the kernel size of the convolutional layers. Also, the patches outside of the mask, and therefore outside of the region of interest of the retina, are removed. Since there are also patches with pixels on the edge of mask which have skewed information, the possibility is made to choose to remove those edge patches or keep them in the data set.

At last, after selecting specific patches for training the algorithm, the output is prepared as a hot encoding, which is needed for the deep learning algorithm. That is, for each patch information about whether the center pixel of that patch is vessel (value 1) or non-vessel (value 0) is arranged into a matrix with two columns and with the amount of pixels rows consisting zeros and ones.

B. Convolutional Neural Network Architectures

Two different deep learning algorithms, LeNet-5 and a differently assembled ConvNet, are set up with the Lasagne and Theano packages. Theano is an open source package and is developed to handle computations for neural network algorithms. Theano is a compiler in Python for mathematical expressions and designed to run as fast as possible on CPUs or GPUs, by turning the structures into very efficient code. On top of Theano, the library Lasagne is built. Lasagne shows the variables of Theano, making it possible to modify the model very easily.[7] So, Lasagne is used to construct a deep learning network. Next, Theano is used to train, validate and test the network. To actually train the algorithm, a function is constructed that continuously puts random batches of input (images to be trained) and output data (manual segmentations) in the algorithm and while training, the accuracy of the algorithm is measured with validation batches.

There were several variables that accounted for the accuracy and training speed of the network, including the learning rate (controls the size of update steps when updating network weights), batch size (how many image patches are in each random batch), number of batches and chosen activation function (which brings nonlinearity to connections between neurons). We hypothesize that when using smaller batches and smaller amount of them, the training would not be so extensive and resulting segmented images would have poorer quality. Just training the algorithm with no change in variables would not enhance the performance. Therefore the former variables of the training of the algorithm are variated to test if it would perform better in the end.

LeNet-5 Architecture

For the training a LeNet-5 (see Fig. 1) was chosen, because of its simplicity, relative effectiveness and previous good results when dealing with pattern recognition problems.

The first convolutional layer (C1) had 6 filters and the second one (C3) had 16 filters, with both a convolution

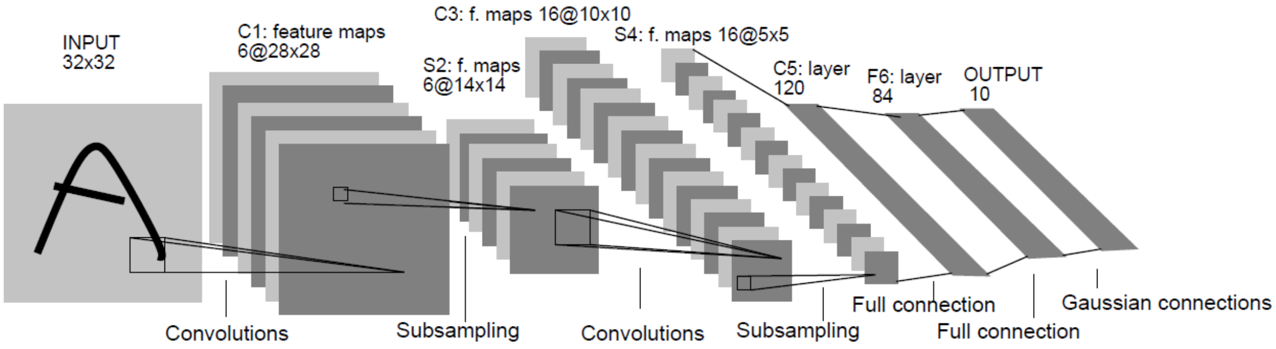

Fig. 1. LeNet-5 architecture. [4]

kernel of 5x5. The convolutional layers were alternating with subsampling layers (S2, S4), where a maxpooling operation (*i.e.* maximum pixel value is chosen from each kernel) was executed with a kernel size of 2x2. Subsequently to these layers, *i.e.* C1-S2-C3-S4, there were two full connection layers (F5 or C5, F6) which had 120 and 84 classes, respectively. Finally there was the output layer with 2 classes; vessel and non-vessel for each pixel. The sizes of the kernels and number of classes were taken from literature. [4]

ConvNet [8]

Another Convolutional Neural Network is ConvNet, just like CNN it has a gridlike structure and is designed for preprocessing data. ConvNet is implemented with the same preprocessing step, except this CNN takes patches of 31x31 as input size. It is a very efficient and effective net for deep learning. Using ConvNet too, makes it possible to compare the different networks. ConvNet consists of 5 types of layers: convolutional, pooling, fully-connected, dropout (promotes better generalization by forcing a fraction of the neurons to be inactive during each episode of learning[10] and rectifying linear unit. In a deep ConvNet, a larger area of the input data is used in the deeper layers, therefore a higher level of abstraction is formed. Multiple models to solve the problem are used, this is called ensemble learning. The output of all the models are combined in the final result, reducing the risk of overfitting of the training data. Generally, this provides a higher accuracy. All convolutional layers had 64 filters with a kernel size of 4x4. The maxpooling layers have a kernel of 2x2. The output layer consisted of 2 classes; vessel and non-vessel (see Fig. 2). The sizes of the kernels, number of filters and layers were taken from literature.[8]

C. Quality Measurement / Assessing the algorithm

The performance of the algorithm will be checked with different processes. First, the generated image of the algorithm will be compared with the manually segmented image from the dataset. Theoretically, these images should be identical. Practically, this is difficult to achieve. Secondly, there are statistical methods to assess the algorithm. Accuracy α is defined as sum of true positives TP and true negatives TN divided by number of pixels in image, *i.e.*

$$\alpha = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}, \tag{1}$$

where FP stands for false positive and FN for false negative. Sensitivity β, which measures correctly identified positives, is defined as

$$\beta = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{2}$$

and specificity γ, which measures correctly identified negatives, as

$$\gamma = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}. \tag{3}$$

The Cohen's kappa statistic can be used to determine the inter-observer acknowledgement with the following equation[9]:

$$\kappa = \frac{p_o - p_p}{1 - p_p}, \tag{4}$$

where observed accuracy p_o is

$$p_0 = \alpha$$

and potential accuracy p_p is

$$p_p = \frac{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN}) + (\mathrm{FN}+\mathrm{TN})(\mathrm{FP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})^2}.$$

The closer α and κ are to 1, the better the agreement between segmentations. Besides assessing inter- or even intra-observer accuracy these equations can be used to define the accuracy of CNN. In this study the statistics were calculated as a mean ± standard deviation over of all 20 test images. There were two manual segmentations, so in the following subindex 1 refers to segmentation made by first observer and subindex 2 by second observer.

Loss tells how well the model is doing for training and validation sets. It is a summation of the errors made for each example (*i.e.* batch) in training and validation sets, respectively. In case of classification problem, the training loss and validation loss are negative log-likelihood. If training loss is much lower than validation loss then the CNN might be over fitting, *i.e.* the CNN ”memorizes” the training examples and doesn't effectively delineate the vessels from the test set images.

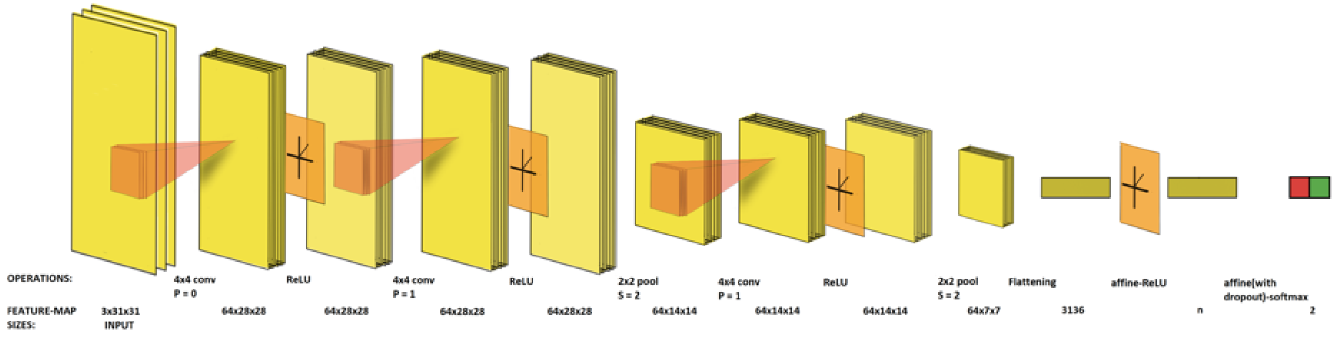

Fig. 2. ConvNet ensemble [8]

IV. RESULTS

In this section the output images of our CNN are presented and compared to manual segmentations. The comparison is done with the help of previously introduced means of quality measurement. The inter-observer accuracy over the 20 test images were $\alpha = 0.136 \pm 0.013$, $\beta = 0.136 \pm 0.013$, $\gamma = 0.000 \pm 0.000$, and $\kappa = 0.000 \pm 0.000$. All the statistics regarding the CNN are collected into table I.

Firstly, the effect of number of batches was investigated. In Fig. 3 there is shown the original image (test set image number 2), manual segmentation by first observer and segmentation when training was done with 1000 and 400 batches, respectively. Here, the learning rate was set to 0.00001 and each batch consisted of 500 patches (*i.e.* batch size was 500). Similarly, the effect of batch size was assessed (see Fig. 4). In Fig. 4 test image number 17 is shown with manual segmentation by 2nd observer; now learning rate was 0.00001 again and 1000 batches were used each having size of 500 and 80 patches. From these images it can be clearly seen, as was hypothesized, that the lower the number of batches and smaller the batches were, the poorer the segmentation results would turn out to be. Actually what matters is the product of these two parameters; the product must be large enough to get decent segmentations. In the case of smaller number of batches the resulting images seem to have thinner vessels and a lot of noise-like clusters between the vessels. Only the main branches of vessels are distinguishable in Fig. 3D. When it comes to the batch size, lowering it leads to blurring and thickening of vessels. On top of that, some vessels even disappeared in Fig. 4D.

Subsequently the learning rate was tested. Keeping the number of batches to 1000 and batch size as 400, three rates were studied, 0.00001, 0.0001 and 0.001. These are presented in Fig. 5 with the manual segmentation by observer 2 (test image number 2). There were 1000 batches with each the size of 500 patches. It can be seen that when learning rate is decreased (but not too much), the CNN outputs better quality segmentations. With a low learning rate the network would need much more batches to achieve the same quality as with higher learning rate. Higher learning rate network accomplishes to delineate even the smaller vessels, whereas the slow learning rate network blurs or even loses them. Additionally, the vessels are thinner when higher learning rate CNN is used. If the learning rate is increased too much, the resulting image consists of only vessel or non-vessel pixels.

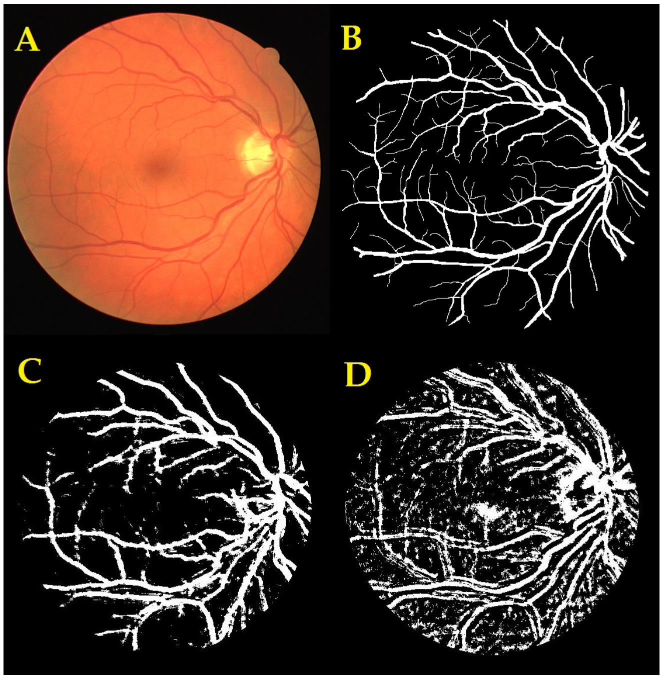

Fig. 3. Effect of number of batches on test image number 2. A) Original image. B) Manual segmentation by 1st observer. C) Segmentation by CNN when 1000 batches were used for training. D) Segmentation by CNN when 400 batches were used for training.

Previously only the rectified linear unit (ReLU) was used as activation function. Next, the activation function was changed to sigmoid and results alongside original image and manual segmentation by observer 1 (from test image) are presented in Fig. 6. In both CNN outputs the learning rate was 0.001 with 1000 batches size 500. Both activation functions produce almost similar results, except sigmoid has more noise between the vessels. That's why the overall performance is better when ReLU was used as activation function.

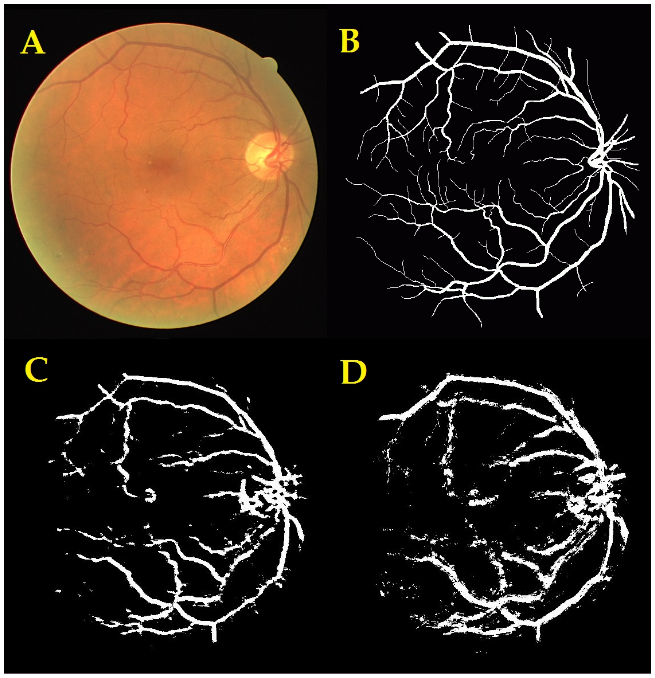

Fig. 4. Effect of batch size (number of patches in one batch) on test image number 17. A) Original image. B) Manual segmentation by 2nd observer. C) Segmentation by CNN when batch size 500 was used for training. D) Segmentation by CNN when batch size 80 was used for training.

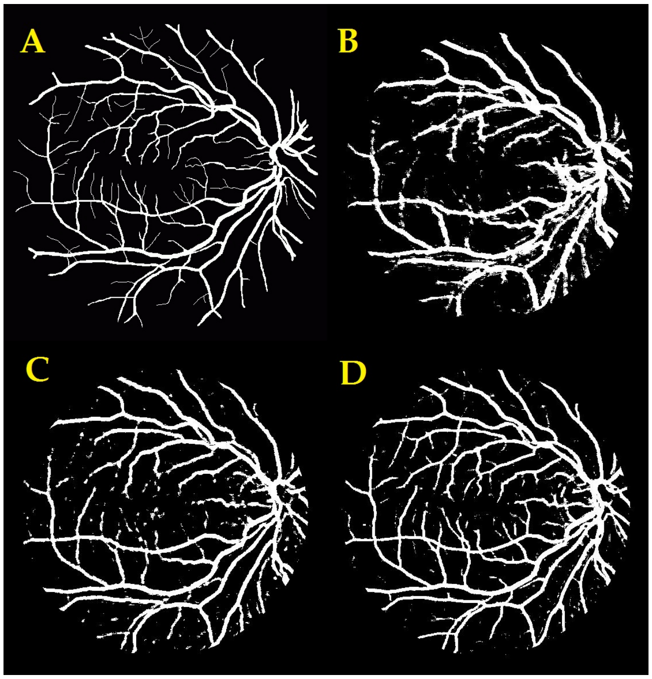

Fig. 5. Effect of learning on test image number 2. A) Manual segmentation by 2nd observer. B) Segmentation by CNN when learning rate was set to 0.00001. C) Segmentation by CNN when learning rate was set to 0.0001. D) Segmentation by CNN when learning rate was set to 0.001.

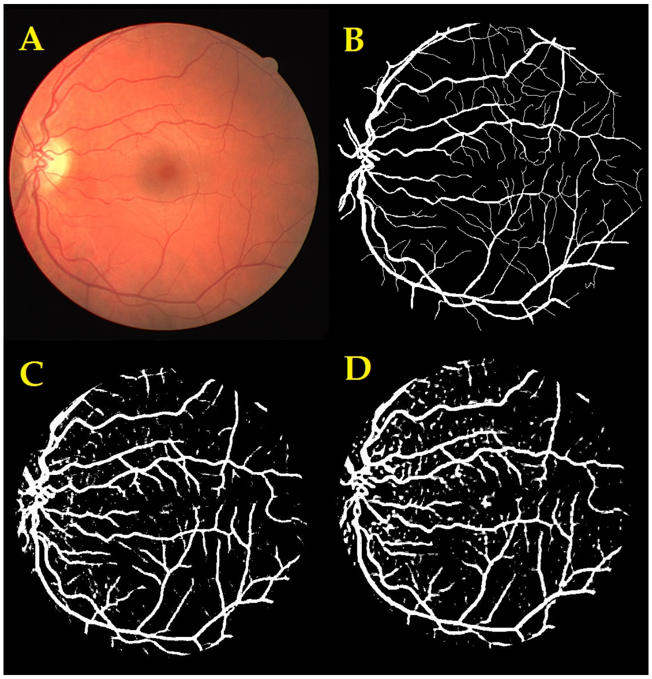

Fig. 6. Effect of chosen activation function on test image number 11. A) Original image. B) Manual segmentation by 1st observer. C) Segmentation by CNN with ReLU. D) Segmentation by CNN with sigmoid.

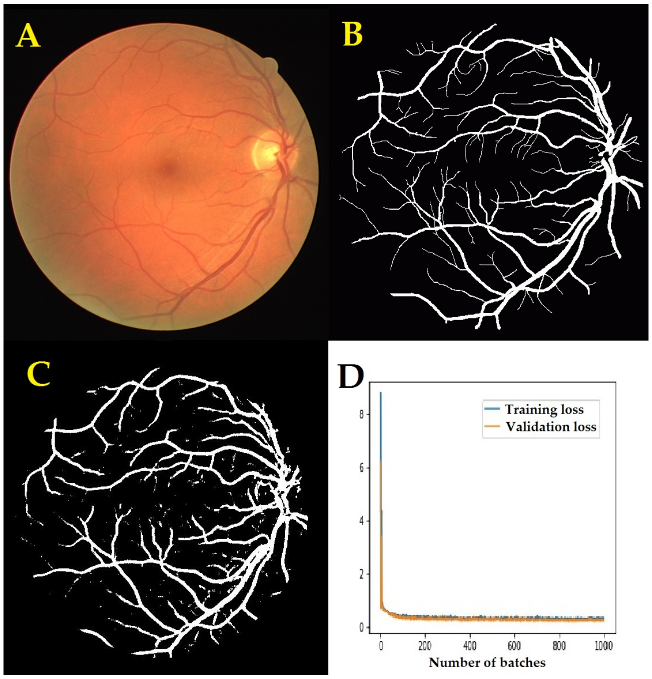

Fig. 7. The best quality segmentation obtained with the algortihm on image number 18. A) Original image. B) Manual segmentation by 2nd observer. C) Segmentation by CNN. D) Loss curve.

Ultimately for the best matching of output pictures to the manual segmentations the following parameters were used: learning rate = 0.001, number of batches = 1000, batch size = 500, activation function as ReLU. The results are shown in Fig. 7, where there is original retinal image number 18, manual segmentation by observer 2nd observer and the loss curve containing training loss and validation loss.

TABLE I

Different statistics used to measure the performance of the CNN. α is accuracy (1), β is sensitivity (2), γ is specificity (3) and κ is the Cohen's kappa (4), and the presented value is mean of all 20 test images $\pm$ standard deviation. On each row the first value corresponds to the comparison to manual segmentation made by 1st observer. LR = learning rate, #B = number of batches, BS = batch size and AF = activation function (ReLU = rectified linear unit, S = sigmoid).

Experiment	α	β	γ	κ
LR = 0.00001, #B = 1000, BS = 500, AF = ReLU	0.856 ± 0.054	0.509 ± 0.095	0.962 ± 0.012	0.524 ± 0.078
LR = 0.0001, #B = 1000, BS = 500, AF = ReLU	0.902 ± 0.030	0.625 ± 0.092	0.966 ± 0.011	0.635 ± 0.064
LR = 0.001, #B = 1000, BS = 500, AF = ReLU	0.916 ± 0.014	0.650 ± 0.065	0.976 ± 0.008	0.685 ± 0.042
LR = 0.00001, #B = 400, BS = 500, AF = ReLU	0.726 ± 0.055	0.279 ± 0.043	0.923 ± 0.009	0.234 ± 0.044
LR = 0.00001, #B = 1000, BS = 80, AF = ReLU	0.807 ± 0.067	0.411 ± 0.075	0.952 ± 0.012	0.413 ± 0.071
LR = 0.00001, #B = 1000 BS = 500, AF = S	0.904 ± 0.018	0.612 ± 0.066	0.972 ± 0.008	0.646 ± 0.048

V. Discussion

First of all the inter-observer accuracy was very low, which means that even though the manual segmentations look similar in the first glance, there are great differences how they have defined the vessel ridges. Basically this results in poor accuracies when our algorithm was tested on second observer's segmentations, because training was done on first observer's delineations. This leads to inconsistencies in the sense that for example lower number of batches yielded better accuracy when comparing to the second observer's results. Different parameters have been adjusted to improve the quality of the algorithm. If the amount of batches increase, then the quality of the segmentation will increase. The same goes for the decrease of the learning rate, but the learning rate must be high enough to reach the good solution in given time. This is why in this study a higher learning rate gave better results; the slower ones didn't have enough iterations to reach the better solution. If learning rate was too high, classification of pixels fails. However, the adjustment of these parameters have a direct influence on the computation time. The higher the amount of batches and learning rate, the longer the computation time. The number of units indicates the amount of parameters that could be learned by the neural network. The higher this number, the more it can learn, but this will increase the computation time as well and will be very demanding of the computer.

It was meant to compare the output of the LeNet-5 with the Ensemble ConvNet. However, it was not managed to attain an output from the Ensemble ConvNet, due to technical drawbacks. The Ensemble ConvNet works on the CPU, but due to memory issues no results were attained with this network. The same network was also runned on the GPU and did not give any results due to technical problems. This is a major drawback for the quality assessment of the network.

VI. Conclusions

During this study, two different CNNs are made to facilitate automated vessel segmentation of images from the retina, which could be much more efficient than segmenting the images manually. This method is evaluated experimented on the images from the DRIVE database. Using deep learning to analyse images is a promising tool to enhance the image analysis and prevent subjective induced segmentation errors.

References

[1] Sharon D. Solomon, Emily Chew, Elia J. Duh, Lucia Sobrin, Jennifer K. Sun, Brian L. VanderBeek, Charles C. Wykoff and Thomas W. Gardner, "Diabetic Retinopathy: A Position Statement by the America Diabeters Association", Diabetes Care, 2017, vol. 40, pp. 412-418, DOI: 10.2337/dc16-2641.

[2] Joes Staal, Michael D. Abramoff, Max A. Viergever and Bram van Ginneken, "Ridge-Based Vessel Segmentation in Color Images of the Retina", IEEE Transactions on medical imaging, April 2004, vol. 23, no. 4, pp. 501-509.

[3] D. C. Klonoff and D.M. Schwartz, "An economic analysis of interventions for diabetes", Diabetes Care, 2000, vol. 23, no. 3, pp. 390-404.

[4] LeCun Y., Bottou L., Bengio Y. and Haffner P. "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, 86:11, 2278 - 2324, 1998.

[5] Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1990). "Handwritten Digit Recognition with a Back-Propagation Network. NIPS, 1989

[6] Simonyan, K. & Zisserman, A., "Very deep Convolutional Networks for Large-Scale Image Recognition", ICLR, 2015

[7] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In International conference on artificial intelligence and statistics (pp. 249-256).

[8] D. Maji, A. Santara, P. Mitra and D. Sheet, "Ensemble of Deep Convolutional Neural Networks for Learning to Detect Retinal Vessels in Fundus Images", CoRR, 2016

[9] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data", Biometrics, 1977, Vol. 33, No. 1, pp. 159-174

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, vol. 15, pp. 1929−1958, 2014.