Application of Deep Learning Techniques to the Diagnosis of Medical Images

Pedro Miguel Carreto Vaz
pedromcvaz@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2018

Abstract

Diabetic Retinopathy (DR) is the leading cause of visual disability worldwide. Although it is highly treatable when diagnosed in its earlier stages, there is currently a need of cheaper and more accurate ways to do so. Medical images have been used in diagnosis for a long time. Recent advancements in the computer vision field have shown remarkable results through the use of Convolutional Neural Networks, that have been able to reach state-of-the-art results in image segmentation. In this master's thesis, we implemented a V-Net like architecture in Python and study how image preprocessing techniques to highlight lesions associated with DR, and different optimization metrics have an impact on its results. The results show that the impact of this variables changes according to the lesion that we try to segment and that the V-Net is capable of obtaining good results for some of the segmentation problems.
Keywords: Diabetic Retinopathy, Computer Vision, Convolutional Neural Networks, Segmentation, V-Net.

1. Introduction

Diabetic Retinopathy (DR) is a medical condition that affects the eyes of diabetic people. This disease is caused by the high blood sugar levels that cause long-term damage to the blood vessels in the retina, by making them swell and leak. This condition is the most prevalent cause of vision impairment worldwide, affecting mainly the working population.

The longer a person has diabetes, the higher the chances of developing DR, which may cause blindness if left untreated. However, with proper treatment and monitoring of the eye, the number of cases of this disease can be significantly reduced. Nowadays, the most common method used to detect DR is for experts to manually analyze the images of the retina and detect the abnormalities associated with it. This method is effective, but it is also slow and resource-heavy.

Computer-aided diagnosis has gained popularity in the recent years and has managed to obtain increasingly better results. A key factor in its success has been deep learning, that has particularly good results in image analysis.

In 2018, the Indian Diabetic Retinopathy Image Dataset (IDRiD) organized the Diabetic Retinopathy Segmentation and Grading Challenge [1], to try to tackle this problem.

In this study, we propose a convolutional neural network that aims to capture the structures associated with DR.

1.1. Problem Statement

The sub-challenge 1 from the IDRiD challenge, aims for the creation of an algorithm capable of performing segmentation of the lesions associated with DR. There are four lesions associated with DR: Micro Aneurysms (MA), Hemorrhages (HE), Hard Exudates (EX), and Soft Exudates (SE) highlighted in fig. 1.

In order to properly diagnose DR we must be able to detect the presence of these lesions in the images. As it can be seen in fig. 1, the shape, color, and size of each lesion is different from one another, with micro aneurysms being very small and hard to see to someone that isn't an expert, and hard exudates being larger structures that can easily be identified. Therefore, one of the main problems that we must face is how to reduce the noise in the images in order to better extract features of DR from retina images.

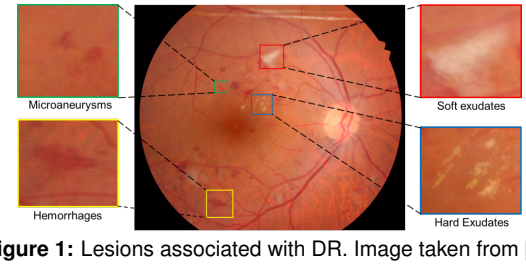

Figure 1: Lesions associated with DR. Image taken from [1].

1.2. Document Organization

The rest of the document follows the following structure. In Section 2 we introduce the concepts of Machine Learning, Deep Learning, Computer Vision and Convolutional Neural Networks. In Section 3, we describe the dataset that was provided to us by IDRiD. Section 4 contains a description of the process of our experiments, describing the environment, our network architecture, and the operations done. In Section 5 we show the results of our work and compare it to the results of a benchmark architecture and the leaderboard participants of the IDRiD challenge. Finally, in Section 6 we give our conclusions and discuss future work.

2. Problem Context

In this section, we provide the theoretical background necessary for understanding the methods discussed in the next chapters. We start by discussing relevant details of machine learning, neural networks, and computer vision. Finally, we explain how these disciplines are combined in convolutional neural networks, exposing some architectures.

2.1. Machine Learning

Machine Learning (ML) is a subset of artificial intelligence where computer algorithms are used to autonomously learn from data and information. Classical computer programs are explicitly programmed by hand to perform a task. However, ML programs change and improve via a learning algorithm [7].

There are four main types of ML: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning. In our work we focused on the first.

In Supervised Learning, the ML algorithm performs a mapping from an input to an output based on input-output pairs examples that are provided to it and that usually were labeled by humans. After the learning process, this algorithm can perform the same kind of mapping in previously unseen inputs.

The tasks associated with machine learning are many, however the one that is relevant to our work is segmentation, which is the process of dividing an image into multiple non-overlapping regions, according to a certain criterion, making it a particular case of classification. By dividing the original image, we can reduce significantly the search area, making the resulting images more meaningful and easier to analyze. The result of this process is a set of image segments that, when put together, cover the entire image. Each set of pixels shares similar properties that were defined by the criterion that was given when doing the segmentation, being easy to distinguish different sets based on these properties.

2.2. Neural Networks

Neural Networks (NNs) are a type of ML algorithm inspired by the biology of the human brain.

The most basic unit of a NN is called perceptron [20], an algorithm that given one or more inputs x_N multiplies them by their weights w_N and adds all the multiplied values in a weighted sum $\sum$. Finally the result from the weighted sum serves as input to the activation function $f(x)$, that produces the output y. During training, the weights w_N are updated in order to produce the desired output.

By combining perceptrons and organizing them into layers, we can form Multi-Layer Neural Networks (MLNNs). The typical structure of MLNNs includes three different types of layers: the input layer, the hidden layers, and the output layer.

2.3. Deep Learning

Deep Learning (DL) is a class of ML algorithms that make use of NNs [12]. With the continuous increase in the size of the datasets necessary to solve more complex problems and the cheapening and increased computational power of modern machines, in particular of GPUs, DL gained popularity over the recent years, being the focal point of many different works [6], [22].

2.4. Computer Vision

Computer Vision (CV) is a big field of research that seeks to make computers able to analyze and understand images and videos the same way humans do, being the focal point of many recent works [10], [9].

Medical image diagnosis has been a very commonly used method ever since the discovery of x-rays in 1895. With the recent advancements of deep learning, medical image analysis has been a focus of numerous studies that try to solve problems like detection, classification, segmentation, and computer-aided diagnosis [17].

2.5. Convolutional Neural Networks

With the advancements in computer vision, it quickly became obvious that regular fully connected NNs wouldn't provide a solution since even low-resolution images required a lot of time to perform the needed computations and there was a lot of overfitting. And so CNNs where created to specifically tackle the problem of computer vision, through the use of the convolution operation. When presented with a new image, the CNN doesn't know where the features that it is looking for are going to be in the image, so it runs through the whole image and tries every possible location, resulting in a filter for that image. To this process, we call convolution. By combining a set of con-

volution filters we form a convolutional layer of the network [5]. In CNNs, the convolution operations replace the regular multiplication operations done in fully connected NNs, and since the filters created from the convolution operations are used for all parts of the image, the number of free parameters needed is drastically reduced when comparing to a fully connected network [13].

2.6. Architectures

The architecture of a CNN has been proven to have a significant impact in its performance, and for different tasks, such as classification and segmentation, the same architecture may have a great performance in doing one, but not so great when it comes to doing the other one.

The most commonly used architecture in the computer vision literature, when it comes to CNNs, is the classical approach of a simple stack of multiple convolutional layers, usually ending in a fully-connected layer. The idea is that each layer extracts features from the previous layer into the hierarchy to which it is connected.

2.6.1 Residual Connections

As explained in the study "Deep residual learning for image recognition" [8], one can view residual learning as the following: we have a function $\mathcal{H}(x)$ as a mapping that we want to fit using a stack of layers, x being the input of the first layer. The residual function, in this case, would be $\mathcal{H}(x) - x$. Since Deep Learning is based on the premise that multiple non-linear layers can asymptotically approximate complicated functions, one could hypothesize the same for its residual function. Therefore rather than trying to approximate $\mathcal{H}(x)$, we approximate its residual function, $\mathcal{F}(x) := \mathcal{H}(x) - x$, which means that $\mathcal{F}(x) + x := \mathcal{H}(x)$. An example of the application of residual connections in a Neural Network can be seen in Fig.2.

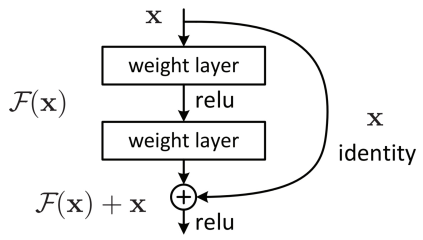

Figure 2: Example of a Residual Connection in a Neural Network. Image taken from the original paper [8].

2.6.2 U-Net

The U-Net architecture [19] provided a big step-up in the image segmentation field. Usually, to train a CNN it is necessary thousands of images, something that is not available in the medical field. However, this is not the case for a CNN with a U-Net architecture [19]. By applying elastic deformations to their dataset, they were able to augment it, which also allows the network to learn invariance without needing to analyze an image with this kind of transformation. This is very important in the biomedical field since this kind of deformations are common.

When it comes to the architecture itself, it was based on the so-called "fully convolutional network" [14], modifying and extending it, so that it could work with very few training images. The main idea of this network is to supplement the contracting network with another one, where the pooling operators are replaced by upsampling ones, that makes the resolution of the output increase. In order to capture high-resolution features, there are connections between the contracting and upsampling paths. Finally, a convolutional layer can learn to correctly assemble a more precise output based on this. One particularly important modification made when building the U-Net was the introduction of a large number of feature channels in the upsampling path, which made it identical to the contraction path and thus acquiring the characteristic U-shape. Another particularity of this architecture is the absence of fully connected layers. The segmentation map only contains pixels that have their full context available, and for that, an overlap-tile strategy was used, were the borders of the image where seen as mirrors of each patch.

2.6.3 V-Net

The V-Net architecture has made its first appearance in the study "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation" [16], with the purpose of doing the segmentation of prostate volumes present in MRI images. They proposed the use of a Fully Convolutional Network trained end-to-end, however, to the date of that work, this kind of Neural Networks had only been used in 2D images, such as in the case of U-Net [19]. One option would be to do the segmentation of the volumes slice wise, however, some of the contexts might be lost in the process, so the team responsible for the V-Net decided to use volumetric convolutions instead. They also proposed a novel objective function based on Dice coefficient maximization. Similarly to the U-Net [19] architecture, V-Net [16] also has both a convolution and de-convolution paths, where the input image is compressed and then decompressed to its original form respectively.

As explained in their work [16], the left side of the network is composed of several stages that work at

different resolutions, starting from the original image and then compressing it down, with each stage being composed of one to three convolutional layers. Based on the work done in residual connections which we previously discussed in this section [8], each stage was made to learn a residual function, where the input image is used first in the convolutional layers and processed through the non-linearities and later added to the output of the last convolutional layer of that stage. According to the authors, this use of residual connections improved the time performance of the network. In this network, the commonly used max-pooling layers were replaced by convolutional layers that double the number of feature maps as they reduce the resolution, so at the end of each stage the number of feature channels doubles as well.

The right side of the network is responsible for the decompression of the image. Each stage extracts features and doubles the spatial support of the lower resolution feature maps and is also applied a de-convolution operation followed by one to three convolutional layers. Similarly to the left side of the network, residual functions are used in the convolution stages. In a way similar to the U-Net [19], the features that were extracted from the left side of the network are forwarded to the right side, which allows to gather details from the image that would have been lost otherwise.

3. Dataset

In this section, we will describe the original datasets used to train and test our CNN, both of them provided by the IDRiD Challenge [1].

Both training and test datasets have the original images of the retina (.jpg files), that are the same for all the lesions, and the ground-truth images (.tif files) that have a set for each one of the lesions and for the optic disk. The images of the retina were captured by the specialist at an Eye Clinic located in Nanded, Maharashtra, India, using a Kowa VX-10 alpha digital fundus camera. According to the IDRiD Challenge webpage [1]. The images in the datasets were marked by experts, that signaled typical DR lesions as well as the OD. All images are sized 4288x2848.

The training set is composed of 54 images of the retina, 54 ground-truth images of MA, EX, and OD, 53 ground-truth images of HE, and 26 ground-truth images of SE. As it can be seen, some of the lesions are not as well represented in the dataset, namely SE, which is present in less than half of the original images of the retina.

The test set is half the size of the training set, with only 27 images of the retina, 27 ground-truth images of MA, HE, EX, and OD, and 14 ground-truth images of SE. Again, SE present about half of the original images of the retina.

4. Methodology

In this section, we will describe the practical implementation of our CNN, starting by discussing the process that the data is put through, from the raw data to the beginning of the training process. We will also describe the specifics of the architecture of our network, and lastly, we will explain the training process.

4.1. Process Description

We start by preprocessing both the train and test data (both original images and groundtruth images), using various methods. The next step is to resize and augment the dataset through a series of transformations done to the images. After that, the augmented train data is used by our model to train and the test data is used to obtain our results.

4.2. Data Operations

Sometimes, the images from the dataset can contain noise, which makes it harder for the network to extract and understand the important features in the image that it must learn to recognize. On one hand, it is possible to fine-tune the network in order to make it robust to noise [3]. Another alternative is through image preprocessing techniques [24]. In our study, we focused on the latter.

4.2.1 Preprocessing

We started by changing all images (both original images and groundtruth images) to .png files since .jpg format compresses the images in a way that sometimes the exact pixel rgb value changes, which can affect the results when comparing the predicted value to the real one. After that, we resize the images to 560x384, a size that allowed our network to train without having memory issues, and that kept the properties of the image.

After resizing the images, we create three different datasets, one where retina images were not altered any further, another one where we enhance the contrast of the image, and a final one where we set the images to green channel only.

By enhancing the contrast of an image we make the structures in it more distinguishable from one another and the background. As discussed in the study “A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina” [23], enhancing the contrast of the images can improve the detection of certain structures within the retina such as hemorrhages and micro aneurysms.

In the study “Segmentation of blood vessels from red-free and fluorescein retinal images” [15], an algorithm for performing segmentation of blood vessels in retinal images used red-free images (green channel only) to successfully improve the results of

their algorithm. Also, according to the study done by C. Sinthanayothin, J. F. Boyce, T. H. Williamson, H. L. Cook, E. Mensah, S. Lal, and D. Usher [21], green channel only images contained more information and greater contrast for red features characteristic of hemorrhages and micro aneurysms, which led us to decide to test this technique in our solution.

We performed closing operations on the groundtruth images of the train data with hopes that it would make it easier to detect small structures that are close to each other, leading to two different sets of groundtruth images. The closing operation consists of the erosion of the image followed by dilation. The basic idea is that by learning to identify the area of the lesions instead of the precise position of the image we can obtain results that although to a computer that can easily see pixel level differences in an image may seem different, but can guide a human expert to where in the image he should look at.

Since the original groundtruth images where .tif files, we verified that the rgb pixel values where not binary black([0,0,0]) or red([255,0,0]), and there were some outliers like [1,3,10], probably created from the conversion from .tif to .png, we applied a filter that set every pixel with a value for red lower than one hundred, to black and to red otherwise. We applied this filter before and after the closing operation.

Finally, before the training process begins, we perform standardization of the retina images by subtracting the mean and dividing by the standard deviation:

$$x_{stand} = \frac{x - \mu}{\sigma} \tag{1}$$

The idea is to reduce the range of the rgb values of the pixels in the images in order to center the data. In the process of training our network the initial inputs will be multiplied by the weights and added to the bias in order to trigger the activations, that are then backpropagated with the gradients to train the model. The reason why standardization of the images is important is that we make the training less sensitive to the scale of the features and reduce the dimensional space, making it easier for the gradient to converge.

4.3. Dataset Augmentation

After preprocessing the images, we end up with three different datasets, each one with 54 images. In order to augment the dataset we apply augmentation methods to the training dataset, making it more robust and not as susceptible to overfit.

So in order to augment our dataset we start by mirroring vertically all of the images of the retina as well as the groundtruth images. We then proceed to rotate both the original images and the mirrored ones by 90, 180, and 270 degrees, transforming the original 54 images into 432 different images.

The augmentation of the dataset makes our model more robust and more tolerant to variations in the images. This is very important given the work that we are trying to do since the images provided to us have differences in the brightness level, as well as the fact that structures in the retina have slight changes in shape and size form eye to eye. By rotating and mirroring the images, we make our model tolerant to variations in the orientation of the image and to the shape of the structures of the retina such as the optic disc or the exudates.

4.4. Architectures

In this section, we will elaborate on the architectures used for the two CNNs that we created in this study. The main architecture and the one that was our goal to use to try to solve the problem of segmenting lesions in the retina associated with DR was the V-Net [16]. The second one was the U-Net [19]. The IDRiD Challenge [1] leader-board for the sub-challenge one has the results from the algorithms sent by its participants, however, we don't know the specifics about their architectures. The reason we decided to create a second CNN with the U-Net architecture was to have a benchmark algorithm whose architecture was known to us to compare the results from our original CNN.

Our implementation off the U-Net was basically the one described in the original paper [19], and because we already described the general architectures of the U-Net and V-Net, we will focus on more specific things of our implementation of the V-Net.

4.4.1 V-Net

Our implementation of the V-Net starts of with a 384x560x3 input, meaning that our network will receive as input 560x384 rgb images (the x and y values are switched in our representation). We have two paths, the contracting path, and the expanding path.

The contracting path is composed of 6 layers, including the input layer, which we called "Down Layer 0". From the first layer to the last one, one, one, two, three, three, and three convolution operations are done respectively. In between layers one to four, a down convolution operation is performed, where the number of channels is doubled, meaning that we start with 16 channels and end up with 256 channels on layer five. After each convolution operation, we performed batch normalization and then applied the activation function Leaky ReLU (LReLU) to the output of the convolution. Batch

normalization consists in normalizing the activations of the previous layer at each batch, i.e. applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1 [2]. In each of the layers, we implemented a residual connection that sums the input of the layer with its output.

The expanding path is composed of five layers including the output layer, which we called "output", where the layers are made to mirror the ones from the contracting path in terms of number of channels and convolution operations. This means that "Up Layer 1" from the expanding path matches with "Down Layer 4" from the contracting path, "Up Layer 2" from the expanding path matches with "Down Layer 3" from the contracting path, and so on until the last layer of the expanding path. Once again residual connections where made in each layer, and the same operations of batch normalization and LReLU where applied to the output of each convolution in order to match the operations from the contracting path. The key difference in the expanding path is the replacement of the down convolution operations by de-convolution operations, where the number of channels is halved instead of doubled. In the output layer, we used the sigmoid activation function.

4.5. Training

The training of our model was performed during eighty epochs, in batches of five. We chose batches of five mainly because of memory issues when trying to used larger size batches.

When compiling the model, we use the Adam optimizer [11], [18], an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. As loss function, we used binary cross entropy. Finally, we used three different metrics that we tried to optimize, one at the time, the Dice coefficient, the f1 value, and the value of the loss function. The f1 value can be calculated as follows:

$$f1 = 2 * \frac{precision * recall}{precision + recall} \tag{2}$$

Before we understand what precision and recall are, we need to understand what are True Positives (TP), False Positives (FP), and False Negatives (FN). A TP means that the pixel in the original groundtruth is red and the predicted pixel value from our model is red as well. An FP means that the pixel in the original groundtruth is black and the predicted pixel value from our model is red. An FN means that the pixel in the original groundtruth is red and the predicted pixel value from our model is black. This way precision is the number of pixels that our model classified as red and that are indeed red in the groundtruth image, dividing by all of the pixels that our model classified as red:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall is the number of pixels that our model classified as red and that are indeed red in the groundtruth image, dividing by the is the amount of pixels that our model classified as red and that are indeed red in the groundtruth image, plus the number of pixels that our model classified as black, but that are red in the groundtruth image:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

So in the end, we trained our model using the three different metrics for each of the three different datasets and for each of the two groundtruth datasets, resulting in 18 different results for each lesion and optic disc segmentation.

5. Results & Discussion

In this section, we will start by describing the evaluation metrics. Then we will discuss the results of our implementation of the V-Net architecture and how the changes in the experimental variables affected the results. Finally, we will evaluate our results by comparing them to the results from the benchmark U-Net architecture and to the ones in the leaderboard of the IDRiD challenge.

5.1. Evaluation Metrics

In order to evaluate the results from our model, we used four different metrics: Precision, Recall, F1, and Area Under Precision-Recall Curve.

The Precision, Recall, and F1 metrics are the same that we just described at the end of the previous section, so we will focus on the AUC metric.

5.1.1 Area Under Precision-Recall Curve

This metric is the one used by the IDRiD Challenge [1] to evaluate the results from the algorithms submitted for sub-challenge one, so ideally we want to maximize this metric.

To compute the AUC we start by computing the Precision and Recall Values at eleven equally spaced threshold instances, i.e. [0, 0.1, 0.2, ... , 1]. However, in this case, the True Positives (TP), False Positives (FP), and False Negatives (FN) aren't computed in the same way as we previously described, but by computing the IoU. This evaluation metric that ranges from zero to one, and is used to detect the accuracy of an object detector on a particular dataset, and is the same as the one

used in *The PASCAL Visual Object Classes (VOC) Challenge* [4]. Let's say that we are trying to predict bounding boxes for objects in an image just like in the VOC Challenge. The IoU is the area of overlap divided by the area of the union of those bounding boxes:

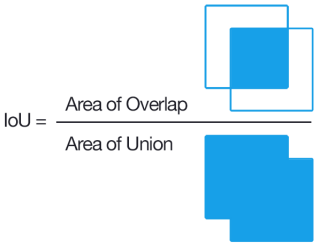

Figure 3: IoU Formula.

So, let's say that we are trying to compute the Precision and Recall at the threshold value of 0.5. In this case, we start by going through all of the predicted bounding boxes for the test dataset. An image would be considered a TP if the IoU value for that image is equal or greater than 0.5. On the other hand, if the value is less than 0.5, the image would be considered both an FP and an FN. After that, we use the formulas 3 and 4 from the previous section to compute Precision and Recall.

We now can make the parallel between the bounding box problem and our segmentation problem, since in its most basic they are essentially the same, as they are both areas, however, instead of squares, we have more irregular shapes.

We repeat the process of computing Precision and Recall for all the threshold values and the we plot both curves. Finally, we compute the integral of precision as a function of recall. This process is essentially the same as computing the Average Precision value for this set of images.

5.2. Discussing Results

We conducted two experiments, one with our version of the V-Net and the other with the U-Net. As discussed before, we tested our two models with three different datasets for the original images, two different datasets for the groundtruth images, and with three different metrics to optimize for all of the four lesions and the optic disc segmentations. In the end, we gathered ninety different results for each of the networks.

In this subsection we will analyze our implementation of the V-Net, comparing the results obtained when changing the experimental variables, particularly how changing the optimization metric, the image dataset and the ground truth dataset impacts the evaluation metrics. Finally, we will also describe the best experimental result. For the sake of simplicity, we will exclude talking about the variations in the Precision and Recall metrics, since the F1 metric already relates them and favors balanced results.

5.2.1 V-Net Results

When looking at the experimental results for **Micro Aneurysms**, we can conclude:

- Out of the three optimization metrics, F1 was the one that obtained better results for both the evaluation metrics F1 and AUC. Since that by using this optimization metric, the model tries to maximize the F1 value between the predicted segmentation and the groundtruth when training, it is no surprise that the result for it was better than the other two metrics. Also, since micro aneurysms are very small structures, any wrongly placed pixel adds a considerable area to the union between the prediction and the groundtruth, and therefore reducing the AUC value. This way, an optimization metric that specifically focuses on maximizing the number of true positives and reduce the number of both false positives and false negatives, should be the most suited for the task at hand.
- When looking at the average results between the three image datasets, we unexpectedly observed that the Original dataset had better F1 and AUC results, although the difference in AUC value between it and the Green Channel Only dataset is negligible. We expected that, as discussed in subsection 4.2, the green channel only images would highlight the micro aneurysms in the image and thus provide better results.
- Between the Original and Closed groundtruth image datasets, the Original managed to obtain better results by a very small margin. These results are exactly as it was to be expected since the morphological closing operation doesn't really change much in the groundtruth images for micro aneurysms, since the structures in it are very small and dispersed through the image.
- The combination of experimental variables that managed both the best F1 and AUC scores was *F1, Green, Original*. Here we see that, although that in average, the Green Channel Only dataset managed very similar results to the Original dataset, it still managed to get the best result overall, being part of the only combination of variables capable of reaching 0.2 AUC value.

When looking at the experimental results for **Hemorrhages**, we can conclude:

- All three optimization metrics obtained very similar results for both F1 and AUC. Hemorrhages can appear in many shapes and sizes, so there isn't a particular metric that would seem to improve the results considerably.
- The Original image dataset achieved better results for the F1 measure and slightly better results for the AUC measure. The difference between the average F1 results for the Original image dataset and the other two was unexpected, especially when comparing to the Green Channel Only image dataset, since the hemorrhages seem more highlighted.
- The Original groundtruth image and the Closed groundtruth image datasets have very similar results for both measures, with the Closed one having slightly better average results.
- The combination of experimental variables that managed both the best F1 and AUC scores was *F1, Original, Original*. This time around, the results were pretty similar across all of the combinations, so there isn't much to discuss.

When looking at the experimental results for **Hard Exudates**,we can conclude:

- There is no significant impact in the evaluation metrics when we change the optimization metric. Hard exudates are larger structures in the image, so the average of the results are very similar across the three evaluation metrics.
- When comparing the results between the datasets, the Green channel only image data set had the best results, followed by the Contrast Enhanced image dataset and then the Original image dataset. This can be explained by the fact that the exudates are more distinguishable from the background in the first two datasets.
- The results from the Original and the Closed groundtruth image datasets are interesting. Although the Original had better results for F1 and AUC, the Closed was better at giving an overall idea to the location and shape of the lesions. This will be further discussed when we evaluate our results later in this chapter.
- The combination of experimental variables that got the best F1 score was *F1, Green, Original*, and the combination with the best AUC score was *F1, Enhanced, Original*, although both combinations were almost exactly identical in results. This once again highlights the that the lesions where more distinct in these image datasets.

When looking at the experimental results for **Soft Exudates**,we can conclude:

- The F1 metric had worst results in both the F1 and AUC scores than the other two optimization metrics, that were basically equal in AUC score, but the Loss metric had slightly better results for the F1 metric. We don't have a concrete explanation for the reason why the F1 optimization metric performed worse in this case.
- The results from the three image datasets were very similar for both of the evaluation metrics, with the original dataset achieving slightly better AUC score. Although soft exudates have a similar color to the hard exudates they are generally smaller.
- The Original groundtruth images dataset achieved slightly better F1 and AUC scores than the Closed dataset.
- The smaller sample of soft exudates cases in the dataset might explain why these results were more inconclusive than to the other problems.
- The combination of experimental variables that managed both the best F1 and AUC scores was *Loss, Original, Closed*, being the only combination able to reach 0.4 AUC score.

When looking at the experimental results for **Optic Disc**, we can conclude:

- The three optimization metrics obtained similar scores for F1 and AUC. The Optic Discs are large and distinguishable structures, so all three optimization metrics seem to converge to 0.92 F1 score and 0.78 AUC score.
- The Original image dataset obtained the best score for both metrics, averaging at 0.95 and 0.85 for F1 and AUC respectively. This results can be explained by the fact that the blood vessels in the retina that overlap the optic disc being more visible in both the Contrast Enhanced and Green Channel Only datasets, making it harder to correctly identify the optic disc.
- The Original groundtruth dataset got better results for both metrics. This is unexpected since the groundtruth images didn't change much from the morphological closing operation.

- The combination of experimental variables that managed both the best F1 and AUC scores was *Loss, Original, Closed*, achieving 0.95 mean F1 score and 0.87 AUC score.
- These were our most successful results in terms of evaluation metrics score.

5.3. Closed Groundtruth Dataset Results

Before we evaluate our results, we would like to briefly talk about the results of the Closed groundtruth images dataset. When we thought about applying the morphological closing operation to our groundtruth images, the main idea was not that it would help increase the scores of the evaluation metrics, but that maybe instead of making our segmentations more accurate, we could be better at identifying the area that was affected by the lesion and that way it would help the experts to know where to look for them. For some of the lesions, in particular, hard exudates, we managed to get some results that went according to our expectations.

6. Evaluating Results

In this section, we will evaluate our results, first by comparing them with the benchmark U-Net network that we created and then with the results from the leaderboard algorithms from the IDRiD challenge.

6.1. Comparing with U-Net

Overall we can conclude that the V-Net has a better performance in every segmentation problem, especially when segmenting soft exudates and optic discs. The residual connections, the batch normalization and Leaky ReLU activation function after each convolution, and the replacement of the Max-Poll 2x2 operations in between layers by a down convolution operation make it so that the V-Net can better capture the relevant features to the data and achieve better results.

6.2. The IDRiD Leaderboard

The metric used for ranking was the AUC score. There were a total of 22 participants, with some of them submitting results for some of the segmentation problems.

If we had participated in the challenge:

- We would place 14th in micro aneurysms segmentation with a score of 0.2021 from the *F1, Green, Original* experiment, 0.1962 higher than the 15th place and 0.058 lower than the 13th place;
- We would have placed 12th in hemorrhages segmentation, with a score of 0.2853 from the *F1, Original, Original* experiment, 0.1445 higher than the 13th place and 0.0062 lower than the 11th place;
- We would have placed 9th in soft exudates segmentation, with a score of 0.4130 from the *Loss, Original, Closed* experiment, 0.1397 higher than the 10th place and 0.0894 lower than the 8th place;
- We would have placed 18th in hard exudates segmentation, with a score of 0.4335 from the *F1, Enhanced, Original* experiment, 0.0617 higher than the 19th place and 0.0683 lower than the 17th place;

Although the groundtruth images for the optic discs were provided to the participants, the results from its segmentation weren't part of the competition so we don't have results to compare to our own.

Overall our results are good, but not as good as the top results, with our highest ranking being 9th place in soft exudates segmentation.

7. Conclusions

In this study, we propose a V-Net architecture implementation for performing segmentation of lesions in the retina associated with diabetic retinopathy. We make use of image preprocessing for both the retina images and the groundtruth images, as well as different optimization metrics and discuss how they impact the results of the V-Net.

The preprocessing of the retina images improved the results for some of the segmentation problems, but for the other didn't show a significant impact. The optimization metric used in the model showed to have more impact when segmenting smaller lesions, namely the F1 metric that showed better results when performing segmentation of micro aneurysms. The morphological closing operation for the groundtruth images showed worst results for the evaluation metrics but was better at identifying the area in which the lesions were located, more evidently when performing hard exudates segmentation.

Overall we believe we achieved reasonably good results that can help experts to have a general idea of the location of the lesions, although these results aren't good enough for automatic lesion detection on their own.

References

[1] *IDRiD - Home*, 2018 (accessed September 21, 2018).

[2] *Keras, Normalization Layers*, 2018 (accessed September 21, 2018).

[3] S. Diamond, V. Sitzmann, S. Boyd, G. Wetzstein, and F. Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487*, 2017.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[5] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[6] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[15] M. E. Martinez-Perez, A. D. Hughes, S. A. Thom, A. A. Bharath, and K. H. Parker. Segmentation of blood vessels from red-free and fluorescein retinal images. *Medical image analysis*, 11(1):47–61, 2007.

[16] F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *arXiv:1606.04797v1*, 2016.

[17] A. Qayyum, S. M. Anwar, M. Majid, M. Awais, and M. Alnowami. Medical Image Analysis using Convolutional Neural Networks: A Review. 0.

[18] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. 2018.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 234–241, 2015.

[20] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[21] C. Sinthanayothin, J. F. Boyce, T. H. Williamson, H. L. Cook, E. Mensah, S. Lal, and D. Usher. Automated detection of diabetic retinopathy on digital fundus images. *Diabetic medicine*, 19(2):105–112, 2002.

[22] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[23] T. Walter, J.-C. Klein, P. Massin, and A. Erginay. A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina. *IEEE transactions on medical imaging*, 21(10):1236–1243, 2002.

[24] J. Yim and K.-A. Sohn. Enhancing the performance of convolutional neural networks on quality degraded datasets. *arXiv preprint arXiv:1710.06805*, 2017.