Retinal lesions segmentation using CNNs and adversarial training

A Degree Thesis Submitted to the Faculty of
Escola Tècnica d'Enginyeria de
Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Natàlia Gullón

In partial fulfilment of the requirements for the degree in
Telecommunications Technologies and Services Engineering

Advisor: Verónica Vilaplana

Barcelona, July 2018

Abstract

Convolutional Neural Networks (CNNs) are frequently used to tackle image classification and segmentation problems due to its recently proven successful results. In particular, in medical domain, it is more and more common to see automated techniques to help doctors in their diagnosis.

In this work, we study the retinal lesions segmentation problem using CNNs on the *Indian Diabetic Retinopathy Image Dataset* (IDRiD). Additionally, the idea of adversarial training used by Generative Adversarial Network (GAN) will be also added to the previous CNN to improve its results, making segmentation maps more accurate and realistic. A comparison between these two architectures will be made.

One of the main challenges we will be facing is the high-imbalance between lesions and healthy parts of the retina and the fact that some lesion classes are very scattered in small fractions. Thus, different loss functions, optimizers and training schemes will be studied and evaluated to see which one best addresses our problem.

Resum

Les Xarxes Neuronals Convolucionals (CNNs) són molt utilitzades per abordar problemes de classificació i segmentació d'imatges, gràcies a l'èxit dels resultats obtinguts. En el cas de l'àmbit de la medicina, és cada vegada més comú veure l'ús de tècniques automatitzades per tal d'ajudar als metges en els seus diagnòstics.

En aquest treball, s'estudia un problema de segmentació de lesions a la retina de l'ull utilitzant CNNs sobre la base de dades *Indian Diabetic Retinopathy Image Dataset* (IDRiD). A més a més, la idea d'entrenament adversari utilitzada en de les Xarxes Generatives Antagòniques (GANs) s'afegirà a l'anterior CNN per intentar millorar els seus resultats, generant uns mapes de segmentació més acurats i realistes. També es farà una comparació entre les dues arquitectures.

Un dels principals reptes al que ens enfrontarem serà l'alta desproporció entre les lesions i les parts sanes de la retina, així com que algunes lesions es troben molt disperses i en porcions molt petites. Conseqüentment, s'estudiaran i s'avaluaran diferents funcions de cost, optimitzadors i esquemes d'entrenament per veure quin d'ells aborda millor el nostre problema.

Resumen

Las Redes Neuronales Convolucionales (CNNs) son muy usadas usan para abordar problemas de clasificación y segmentación de imágenes, debido al éxito de los resultados obtenidos en numerosos casos. En el ámbito de la medicina, es cada vez más común ver el uso de técnicas automáticas para ayudar a los médicos en sus diagnósticos.

En este trabajo, se estudia un problema de segmentación de lesiones en la retina del ojo usando CNNs sobre la base de datos *Indian Diabetic Retinopathy Image Dataset* (IDRiD). Además, a esta CNN se le añadirá la idea de entrenamiento adversario usada en las Redes Generativas Antagónicas (GANs) para intentar mejorar sus resultados, generando unos mapas de segmentación más precisos y realistas. También se hará una comparación entre las dos arquitecturas.

Uno de los principales retos que tendremos que afrontar será la alta desproporción entre las lesiones y las zonas sanas de la retina, así como que algunas lesiones estén muy dispersas y en porciones muy pequeñas. Consecuentemente, diferentes funciones de coste, optimizadores y esquemas de entrenamiento se estudiarán y evaluarán para ver cuál de ellos se adapta mejor a nuestro problema.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my advisor Prof. Verónica Vilaplana for her support, patience, motivation and immense knowledge. Her guidance helped me during all the research and writing of this thesis.

Furthermore, I would also like to acknowledge Marc Combalia for his help and advices, which have been a great support to carry out this project. Thanks for sharing your knowledge and guide me in my first steps in the world of deep learning.

Last but not least, I would like to give thanks to my friends and family for their continuous encouragement and understanding.

Revision history and approval record

Revision	Date	Purpose
0	May 14, 2018	Document creation
1	June 28, 2018	Document revision
2	June 30, 2018	Document modification
3	July 2, 2018	Document final revision

DOCUMENT DISTRIBUTION LIST

Name	E-mail
Natàlia Gullón Altés	natalia.gullon@alu-etsetb.upc.edu
Verónica Vilaplana Besler	veronica.vilaplana@upc.edu

Written by:		Reviewed and approved by:	
Date	May 14, 2018	Date	June 28, 2018
Name	Natàlia Gullón Altés	Name	Verónica Vilaplana Besler
Position	Project author	Position	Project supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1 Problem statement

Most of the interpretations of medical data is done by experts, making the cost of these interpretations very high. Besides, the analysis of images made by human experts is quite limited due to its subjectivity, fatigue and complexity of the task. However, with the emergence of deep learning techniques, several algorithms and solutions with pretty good results have been proposed for medical imaging purposes [1].

One of the most important applications of deep learning techniques in health sector is the detection of diseases. To do that, classification and segmentation Neural Networks (NNs) are frequently used because they are able to detect certain abnormalities with high precision. Therefore, this kind of automated techniques might reduce costs and help experts in their diagnosis.

Therefore, this project will be focused on one of the examples of automation that would be of great assistance to doctors, which is the detection of Diabetic Retinopathy (DR), the leading cause of blindness among the working-age population.

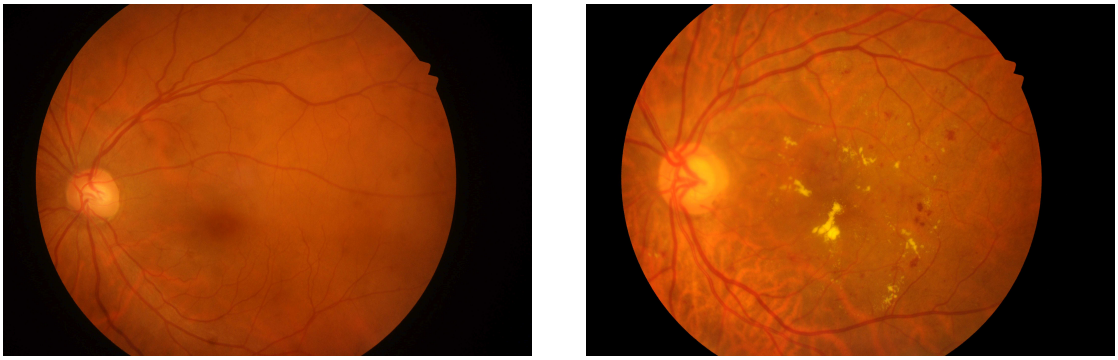
Diagnosing this disease is a time-consuming and manual process which requires a professional expert to examine and evaluate digital colour fundus images of the retina. Clinicians are able to identify DR by the presence of lesions associated with the vascular abnormalities caused by the disease. And although this approach is effective, it also demands a lot of resources and it is very difficult to achieve reliable early diagnosis at a reasonable cost. Therefore, the need for a comprehensive and automated method of DR screening is clearly recognised [2].

Diverse and representative retinal image sets are essential for developing and testing digital screening programs and automated algorithms. However, there are currently very few available annotated datasets, which make it very difficult to achieve good results.

1.1.1 Diabetic Retinopathy

Diabetic Retinopathy is an eye disease that affects people with diabetes and it is the most frequent cause of blindness among working-age adults. The damage occurs in the retina, and more precisely, it affects the retinal blood vessels which can bleed or leak fluid and lead to visual distortion [3, 4].

This disease usually has no symptoms until vision loss occurs. However, early detection and timely treatment can reduce or prevent its occurrence and, consequently, the vision loss. Therefore, diabetic patients need to be regularly screened with one of the methods of examinations available. One of the most commonly used is ophthalmology (funduscopy), which shows lesions of small vessels and functional abnormalities [5]. Figure 1.1 shows the obtained images of a funduscopy examination, where vessels and possible lesions can be observed.



(a) Healthy retina (b) Retina with lesions

Figure 1.1: Funduscopy images used to screen retinopathy

Due to its difficulty to achieve reliable early diagnosis at a reasonable cost, it is important to develop computer-aided diagnosis tools. An automatic retinal image analysis could ease mass screening of population with diabetes mellitus and help clinicians in utilizing their time more efficiently [6].

1.2 Project overview

This project is carried out at the Image Processing Group within the department of Signal Theory and Communications (TSC) of the Universitat Politècnica de Catalunya (UPC).

The objective of this project is to apply deep learning techniques to tackle a seg-

mentation problem with medical images in order to help clinicians in the diagnosis of a disease. For that purpose, Convolutional Neural Networks (CNNs) will be used. Also, the idea of adversarial training used by Generative Adversarial Networks (GANs) will be added to a conventional segmentation network in order to achieve more realistic and precise results. Afterwards, a comparison between both algorithms will be made to validate if the addition of adversarial training techniques are able to improve the results of a simple segmentation CNN.

In particular, fundus images are used to do a segmentation of the different type of retinal lesions that may appear in a patient with DR. The *Indian Diabetic Retinopathy Image Dataset* (IDRiD) from the *Diabetic Retinopathy: Segmentation and Grading Challenge* [6] is utilised and the obtained results will be compared to the first positions in the ranking of the Sub-Challenge 1.

The project is implemented in Python using Keras framework running on top of TensorFlow and, although the project was started from scratch, the part of using adversarial training in a segmentation problem was inspired by the paper presented in [7].

The tasks and milestones as well as the Gantt diagram of the project are detailed in Appendix B.

2. State of the art

2.1 Semantic segmentation

Deep Learning (DL) is becoming more and more popular in general data analysis and it has been termed one of the breakthrough technologies of last years.

In particular, CNNs have been proven to be powerful tools for computer vision tasks and they have been widely used to tackle image segmentation and classification problems, because they are able to automatically learn mid-level and high-level abstractions from raw data as images. Consequently, CNN architectures and other DL techniques have also been applied for medical image analysis and promising results are emerging [8].

In medical imaging, the accurate diagnosis and assessment of a disease not only depends on image acquisition but also on interpretation. While image acquisition has improved significantly over past years with the introduction of devices acquiring data at faster rates and increased resolution, image interpretation process has only recently begun to benefit from computer technology. Currently, most interpretations of medical images are still performed by experts, but these are limited due to its subjectivity and sometimes there are even large variations across interpreters. Therefore, DL techniques are key technologies to improve diagnosis by providing objective and accurate results and they have become the state-of-the-art foundation [1].

Although image classification is very useful and successful results have been obtained, in many visual tasks, especially in medical applications, the desired output should include localization instead of just classifying the inputs into a single class label. This is achieved by assigning a class label to each pixel of an image and it is the idea behind semantic segmentation using CNNs.

The use of CNNs for image segmentation has made significant progress in recent years. It has evolved from one of the first ideas proposed by Ciresan et al. [9], where the label of each pixel was predicted from raw pixel values around the target pixel, to the commonly

used U-Net architecture proposed by Ronneberger et al. [10]. This last architecture, U-Net, was a modification and extension of the Fully Convolutional Network proposed by Long et al. [11], the first segmentation network without fully connected layers and it follows a encoder-decoder scheme. It has proven to achieve good results in many different fields.

However, inspired by classical GANs proposed by Goodfellow et al. [12], adversarial neural networks for image segmentation have been proposed lately. The main idea of GANs is to simultaneously train two networks pitting one against each other. One of them captures the data distribution and generates fake images while the other has to classify whether the input images are real or generated. In image segmentation tasks, the generator should be replaced by a segmentation network, achieving more realistic outputs.

This approach has been recently proposed as an improvement of the U-Net, state-of-the-art to date, and several studies [7, 13–17] have proven a better performance in different medical image segmentation applications.

2.2 Automation of Diabetic Retinopathy diagnostics

There is an ever-increasing interest in the development of automatic medical diagnostic systems due to the advancement of computing technology. Especially, there is an particular interest in DR diagnosis because, as explained in 1.1.1, it is the leading cause of blindness among working-age adults and it can be prevented if detected and treated early.

DR has many symptoms and the most distinctive are Microaneurysms and Haemorrhages, which are dark lesions, and Hard Exudates and Cotton Wool Spots (also known as Soft Exudates), which are bright lesions [18].

However, most of the significant work that has been done related to DR are classification techniques to diagnose the disease. Initially, the methods were on two class classification for DR or no DR [19], but lately more accurate techniques appeared to automatically grade the state of DR into its different stages (No DR, Mild DR, Moderate DR, Severe DR and Proliferative DR) [20–22].

As far as we are aware, there are very few methods proposed to segment retinal lesions, maybe, due to the lack of data available, which is very difficult and hard to get and annotate.

The first work, and only to our knowledge, that segmented Exudates, Haemorrhages

and Microaneurysms automatically and simultaneously with the outputs evaluated on the basis of pixel was proposed by Tan et al. [23]. However, they achieved a low sensitivity.

Furthermore, apart from locating the lesions explained above, detecting the anatomical structures of blood vessels, fovea and optic disk play important role for early detection of DR and several methods have been proposed to segment these structures [24].

The proposed methods also use fundus images as inputs and the current state of the art techniques for semantic segmentation, that is adversarial training, has been studied to segment retinal vessels in [7]. Son et al. propose a method to improve the existing methods that tend to miss fine vessels or allow false positives at terminal branches with the use of adversarial training. The results look promising. Inspired by that work, we propose to apply the same technique to retinal lesions segmentation.

3. Methodology

Two different architectures have been tested with our segmentation problem to see which one performs better. The first one, is a CNN used to segment images and the second one is an extension of the first network with the addition of adversarial training, where two networks are pitting against each other in order to improve the results of the first network.

3.1 Convolutional Neural Networks

CNNs are deep artificial neural networks that are mainly used for two purposes: classification and segmentation. In the first one, the network needs to recognise and identify objects, shapes or parts of the image with certain characteristics. In the second one, the network not only needs to recognise and identify the objects but also to localise them. Therefore, it is similar to a classification problem, but at a pixel level.

They are very similar to ordinary NNs because they are also made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. In essence, the whole network expresses a differentiable score function: from the raw image pixels on one end to class scores at the other.

However, CNN architectures, unlike ordinary NN, make the explicit assumption that the inputs are images and this allows to fully exploit their characteristics and encode certain properties into the architecture. Therefore, it also makes the forward function more efficient to implement and vastly reduce the amount of parameters in the network [25].

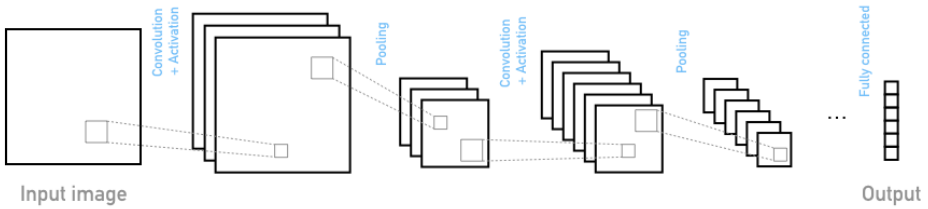


Figure 3.1: Typical structure of a classifying CNN

3.1.1 U-Net

3.1.1.1 Architecture

The U-Net is a CNN that is used to segment images, so every pixel has a class label assigned at the output. This is achieved with a contracting path that follows the typical architecture of a convolutional network, with convolutions and pooling operations, and a expansive path that consists of an upsampling of the feature map followed by a convolution ('up-convolution'). In order to precisely localize, high resolution features from the contracting path are combined with the upsampled output, so that the successive convolutional layer can use this information to assemble a more precise output. Finally, a 1x1 convolution is applied in the last layer to map each vector to the desired number of classes [10]. See Figure 3.2.

3.1.1.2 Loss functions

The loss function, also known as cost function, used to train a NN is the function that quantifies the agreement between the predicted scores and the ground truth labels. Therefore, training can be understood of an optimization problem in which weights are updated in order to minimize the loss function.

Different loss functions have been studied to see which one addresses better our segmentation problem. The cost functions that have been analysed with the U-Net are categorical cross-entropy, dice coefficient and generalised dice coefficient.

Categorical cross-entropy The cross-entropy loss function measures the average number of bits needed to identify an event from a set of possibilities, if a coding scheme is used based on a given probability distribution (p), rather than the real distribution (y).

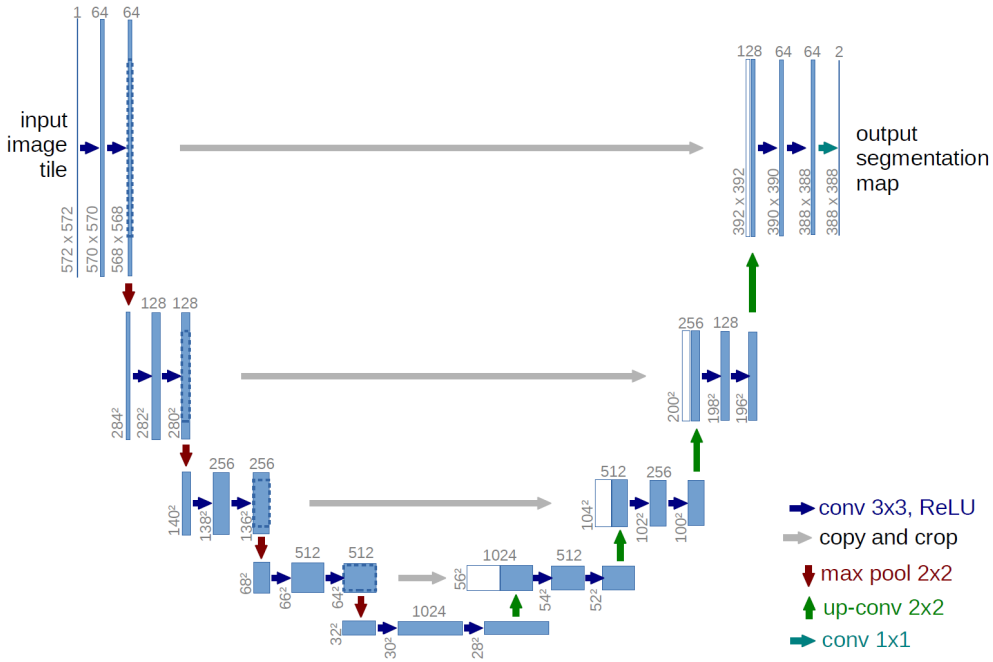


Figure 3.2: U-Net

It can be expressed as:

$$L = -\sum_{c=1}^{C}\sum_{n=1}^{N}(y_{c,n} \cdot \log p(y_{c,n})) \tag{3.1}$$

where y are the true labels, p are the predicted probability maps for each class and C are the number of classes. In case of image segmentation, the loss is calculated for all N pixel values.

Dice coefficient The dice coefficient is a measure of similarity that takes into account the overlap between the predicted values (P) and the ground truth (G). It is commonly used to assess segmentation performance. It is based in the following idea:

$$D = \frac{2|P \cap G|}{|P| + |G|} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.2}$$

where TP stands for the number True Positives, which are the pixels correctly classified as positive, FP for False Positives, which are the pixels wrongly classified as positives, and FN for False Negatives, that are the pixels wrongly classified as negatives [26, 27].

Therefore, the expression used to calculate the dice coefficient for a binary problem can be expressed as:

$$DC = 2 \cdot \frac{\sum_{n=1}^{N} p_n y_n + \epsilon}{\sum_{n=1}^{N} p_n + y_n + \epsilon} \tag{3.3}$$

where p are the predicted probability maps and y are the binary ground truth values. ϵ is used to ensure the loss function stability (avoid dividing by 0) [28]. It is calculated for all N pixel values.

In our multi-class segmentation problem, we used an extension of Equation 3.3 by doing an average of the dice coefficient for each class:

$$DL = 1 - \frac{1}{C} \sum_{c=1}^{C} DC_c \tag{3.4}$$

Generalised dice coefficient The generalised dice coefficient, unlike the above mentioned cost functions, takes into account the proportion between different classes. That is, weights are calculated and used to provide invariance to different label set properties. For a binary problem, it can be expressed as:

$$GDC = \frac{2 \cdot \sum_{c=1}^{2} w_c \sum_{n=1}^{N} y_{c,n} p_{c,n}}{\sum_{c=1}^{2} w_c \sum_{n=1}^{N} y_{c,n} + p_{c,n} + \epsilon} \tag{3.5}$$

where p are the predicted probability maps and y are the ground truth values. ϵ is used to ensure the loss function stability and w_c stands for the weights of the classes. The coefficient is calculated for all N pixel values [28].

The weights w are calculated as the square of the inverse of the volume of each class, so their contribution should correct the imbalance between classes:

$$w_c = \frac{1}{(\sum_{n=1}^{N} y_{c,n})^2} \tag{3.6}$$

In our problem, we used an extension of Equation 3.5 for C classes by doing an average of the GDC for each class:

$$GDL = 1 - \frac{1}{C}\sum_{c=1}^{C} GDC_c \tag{3.7}$$

3.1.1.3 Training scheme

The U-Net is trained with patches of the whole fundus images due to memory capacity. Furthermore, since there is a high-imbalance between lesion classes and healthy parts of the retina, dividing the image in small patches will also adjust the proportion between the number of pixels that belongs to lesion and healthy parts of the retina.

Then, the network is first trained with patches with a high percentage of lesion and for a few number of epochs. Thereupon, the network is trained with just the patches that has any type of lesion until the model converges. During this second stage of training, the network is evaluated with the validation dataset that includes patches with and without lesion.

3.2 Generative Adversarial Networks

GANs are deep neural network architectures comprised of two networks, pitting one against the other. In these architectures, one of the neural networks is called the generator that captures the data distribution and generates new data instances while the other, called discriminator, estimates the probability that a sample came from the training data rather than from the generator.

In other words, the discriminator decides whether each instance of data belongs to the actual training dataset or it is created by the generator. Therefore, the objective of the generator is to generate new fake images that are deemed authentic by the discriminator, whereas the goal of the discriminator is to recognize the authenticity of the instances [12]. The concept is illustrated in Figure 3.3.

3.2.1 Adversarial training

The idea used by GAN architectures can also be applied to train segmentation models. That is, a convolutional segmentation network is trained along with an adversarial network, which discriminates segmentation maps coming from the ground truth or from the segmentation network.

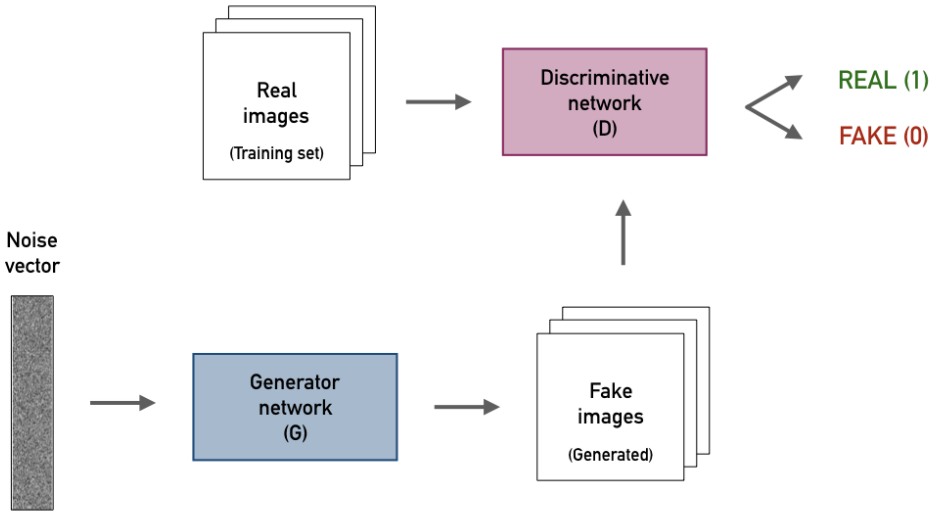


Figure 3.3: Concept of GANs

The objective is that the whole architecture can detect and correct higher-order inconsistencies between ground truth segmentation maps and the ones produced by the segmentation network. Therefore, the segmentation maps generated should be more realistic and accurate [13].

3.2.1.1 Architecture

A U-Net is used as segmentation network and it has the same structure as explained in 3.1.1.1. Another CNN is also used as discriminative network, but in this case, it is a classification network. That means that its output is a binary result which indicates whether the input is a real segmentation from the training set or if it is a generated segmentation map.

Therefore, the combination of these two networks will create another network whose inputs will be the images to segment with their ground truth. A segmentation will be performed by the U-Net and its outputs will be the inputs of the discriminative network that will classify them as real or fake. The output of the whole network will be the same as the output of the discriminator. This combined network will be trained

pitting it against the discriminator, so that the segmentation network gets to fool the discriminator, making the results realistic as if they were from the training set.

An outline of the entire adversarial model is illustrated in Figure 3.4.

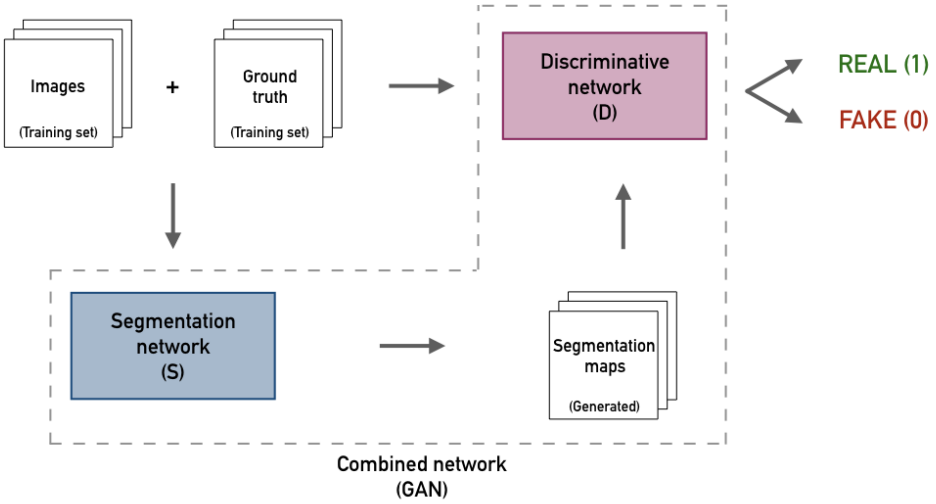


Figure 3.4: Outline of the architecture used with adversarial training

3.2.1.2 Loss functions

Different loss functions have also been studied to see which one best addresses our segmentation problem with the adversarial training scheme.

The adversarial training procedure is a two-player minimax game in which the combined network and discriminator are trained in an alternating way, so that the model can learn spatial pixel dependencies. For training the discriminator to make a good judgement, the loss function needs to be minimized, whereas the segmentation network should produce realistic outputs that are indiscernible to the real data. To do that, the combined model has to maximize the discriminator's loss function. At the same time, the combined network also penalizes the difference between ground truth and predicted segmentation maps.

Consequently, the loss has two terms with a trade-off coefficient: the first one encourages the segmentation model to predict the right class label at each pixel location (it minimizes the loss function used in the segmentation network) and the second one makes the outputs more similar to ground truth, so that the discriminator cannot distinguish whether the outputs are generated or real (it maximizes the loss function used in the discriminative network).

Therefore, we used a different loss function for the segmentation and discriminative networks. The cost functions that have been analysed for the discriminative network are binary cross-entropy and mean squared error. For the segmentation network's loss function, the cost functions described in 3.1.1.2 have been previously studied and the one with best performance is the only one used in adversarial training.

Binary cross-entropy The cross-entropy loss function penalizes at each position the deviation of the predicted labels from the true ones. More precisely, it measures the average number of bits needed to identify an event from a set of possibilities, if a coding scheme is used based on a given probability distribution (p), rather than the real distribution (y). For the binary case, it can be expressed as:

$$L = -(y \cdot \log p + (1 - y) \cdot \log (1 - p)) \tag{3.8}$$

where y is the real value and p is the predicted value.

Mean Squared Error The mean squared error measures the square of the difference between the predicted value, p, and the ground truth, y. It can be expressed as:

$$MSE = (y - p)^2 \tag{3.9}$$

where y is the real value and p is the predicted value.

3.2.1.3 Training scheme

When the model is trained using adversarial training, the previously trained U-Net is loaded as the segmentation network and the discriminative network is built. Then, the combination of these two networks is also built. The training scheme consists of several rounds in which the discriminative and the segmentation networks are alternately trained.

Firstly, the discriminator is trained for one step and its weights are updated, and

then, it is frozen and the combination model is also trained for one step. Training the combined model consequently trains the segmentation network and updates its weights. Afterwards, both models are evaluated with the validation set and the procedure is repeated for several rounds. This iterative process of training the discriminator and combined network alternately is where the minimax game takes place.

As training progresses, both networks should become more powerful and, eventually, the segmentation network should be able to produce predicted segmentation maps that are similar to ground truth.

3.3 Metrics

Two different metrics have been used to evaluate the performance of the model: Dice Score and the Area Under the Curve of Precision-Recall. All the metrics have been calculated for each class separately, so that we are able to evaluate the model performance for the different classes.

Furthermore, the sub-challenge 1 of the *Diabetic Retinopathy: Segmentation and Grading Challenge* [6] uses Area Under the Curve of Precision-Recall and also evaluates the performance of the networks for each class separately, so this way we are also able to compare our results with the ones from the challenge.

Dice Score The dice score is calculated according to Equation 3.2 for each class. This metric is the most frequently used in segmentation problems.

Area Under the Curve of Precision-Recall Area Under the Curve of Precision-Recall calculates, as its name suggests, the area under the Precision-Recall curve. Precision measures the fraction of samples classified as positive that are truly positive, while recall represents the fraction of positive samples that are correctly labelled [29]. They can be expressed as:

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.11}$$

where TP stands for the number true positives, FP for false positives and FN for false negatives.

Therefore, the precision-recall curve shows the trade-off between precision and recall for different thresholds. These thresholds indicate the value beyond which the predicted probability outputs can be considered as positives.

A system with high recall but low precision returns many results, but most of its predicted labels are incorrect when compared to the ground truth. A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the ground truth.

A high AUC consequently represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

An ideal network, whose probability of error is 0, would return all the results labeled correctly. Therefore, precision and recall would be 1, just as the area under the curve. It is illustrated in Figure 3.5.

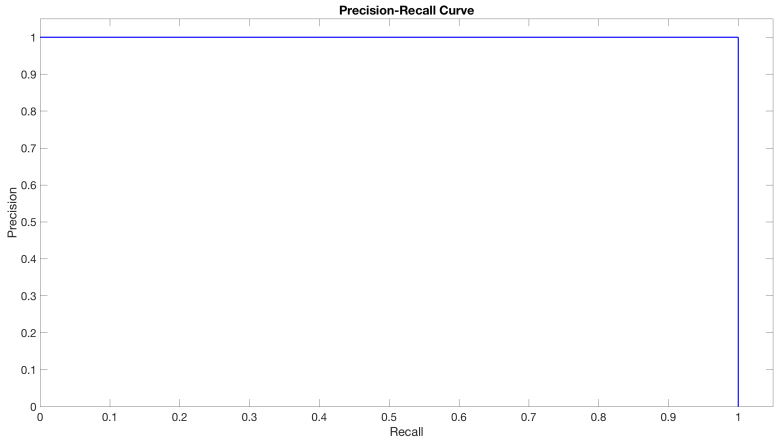


Figure 3.5: Ideal Precision-Recall curve

4. Experiments and results

4.1 Dataset

The dataset used in this project is the Indian Diabetic Retinopathy Image Dataset (IDRiD) from the *Diabetic Retinopathy: Segmentation and Grading Challenge* organised at IEEE International Symposium on Biomedical Imaging (ISBI-2018) [6].

It consists of 54 images with pixel level annotation for different retinal lesions such as Microaneurysms (MA), Soft Exudates (SE), Haemorrhages (HE) and Hard Exudates (EX).

However, not all the images present all these abnormalities. In particular, all 54 images have MA and EX, 53 have HE and only 26 of them have SE. This makes more difficult to do a good segmentation of this last type of lesion. On the other hand, MA are distributed over all the retina in very small portions, which also makes it difficult to segment properly. Figure 4.1 shows an example of the four types of lesions that may appear in dataset images.

To train the model, the whole dataset is divided in train, validation and test sets. From all the images, 44 of them (80%) are used for training, 5 (10%) for validation purposes and other 5 (10%) are used for testing. These partitions have been made randomly, but making sure that in test and validation sets, there are images with SE lesion, so that we are able to evaluate the model performance on this class.

Analysing fundus images, we can see that they show the retina in the centre of the image with a black edge around it, as shown in Figure 1.1. Therefore, a preprocessing should be applied to put the black background to level 0. To do that, a mask to filter the retina is made for every image. These masks are generated binarizing the original image with a previously studied threshold in the first place and, then, applying morphological operations, in particular, an opening followed by a closing are used.

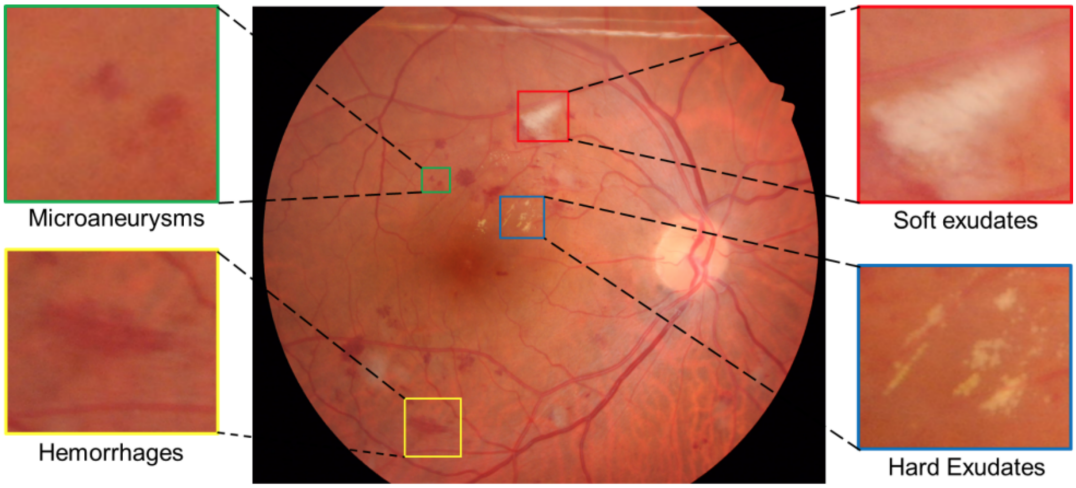


Figure 4.1: Examples of retinal lesions in dataset images

Another consideration to take into account is that the whole images have a size of 4288×2848, but they do not fit in memory. Consequently, the images are divided in patches of size 400×400 for training and validation and patches of 1600×1600 are used for testing. The size of the training patches is chosen to fit in memory, but they are also small enough, so that the proportion between lesion and healthy parts of the patches is balanced. If they were too big, the network would see almost the whole image as healthy and it would be more difficult to train the model.

4.2 U-Net

The U-Net follows the structure described in 3.1.1.1. The contracting path consists of the repeated application of 3×3 convolutions, each followed by a rectified linear unit (ReLU), and a 2×2 max pooling with a stride 2 for downsampling. The expansive path, otherwise, is composed of an upsampling of the feature map followed by a 2×2 deconvolution that halves the number of feature channels and a concatenation with the correspondingly feature map from the contracting path. Then, two 3×3 convolutions, each one followed by a ReLU, are added. Batch normalization is added after every convolutional layer as a regularization strategy. The final layer, a 1×1 convolution, is applied to map each feature vector to the desired number of classes, that is 5 in our particular case. The whole network has a total of 19 convolutional layers.

When training the network, the scheme explained in 3.1.1.3 is followed. More precisely, to train our U-Net, we first use patches with a minimum of 30% of lesion for 15 epochs. After that, we train the model with all the patches that have at least one pixel

of any type of lesion.

Despite that, obtaining best results when training a network relies on the correct selection of hyperparameters. As there are many hyperparameters to select and not all the possible combinations can be tried, after some experiments the ones with best outcomes were chosen.

Batch size is set to 8 and the learning rate to 0.0001. Different optimizers and loss functions have been tested to see which one obtains fits better to our problem. However, best results were all obtained with Adam optimizer.

Data Augmentation (DA) has also been tested and it led the network to underfit when many transformations were applied, so the number of flips and rotations were reduced. Then, the effect of invariance was also reduced, but in some cases the network performance improved.

A comparison between the three combinations with best results using the dice score as metric is presented in Table 4.1.

Optimizer	**DA**	**Loss function**	**EX**	**HE**	**MA**	**SE**
Adam	False	Categorical cross-entropy	0.579	0.365	0.374	0.281
Adam	False	Dice coefficient	0.727	0.429	0.423	0.0
Adam	True	Generalised dice coefficient	0.694	0.458	0.477	0.351

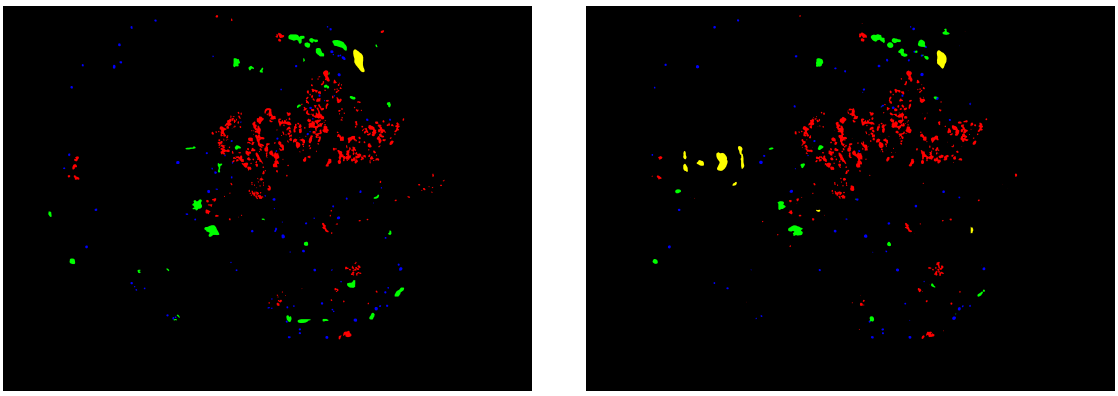
Table 4.1: Results using U-Net with Dice score metric

As we can see in Table 4.1, the best results for each class are not obtained with the same hyperparameters. The best performance for class EX is achieved with dice coefficient as loss function, whereas for the rest of the classes (HE, MA and SE), it is obtained with the cost function of generalised dice coefficient.

We can also observe that the class SE is not detected when training with dice coefficient loss. As explained in 4.1, this might happen due to the few samples of this lesion in the training set.

To qualitatively appreciate the performance of the network, an example of the predicted segmentation maps for the network trained with Adam optimizer and generalised dice coefficient loss compared to the original labels is shown in Figure 4.2.

We can note that the general localization and structure of the lesions are similar. However, with the naked eye, we can see that there are SE pixels misclassified in the left part of the image as well as some variations in the shape and localization of pixels,



(a) Ground truth (b) Predicted labels

Figure 4.2: Comparison between real and generated segmentation maps. In red, the segmentation of class EX; in green, class HE; in blue, class MA; and in yellow, class SE.

especially for the class HE.

The difference maps for the same test image are also presented in Figure 4.3. There is an image for each class, where we can see represented in different colours the True Positives (TP), that are the pixels classified as one particular lesion that are truly that lesion, the False Positives (FP), that are the pixels misclassified as the target lesion that do not truly belong to that class, and the False Negatives (FN), that are the pixels misclassified as another class but that truly belong to the lesion we are evaluating.

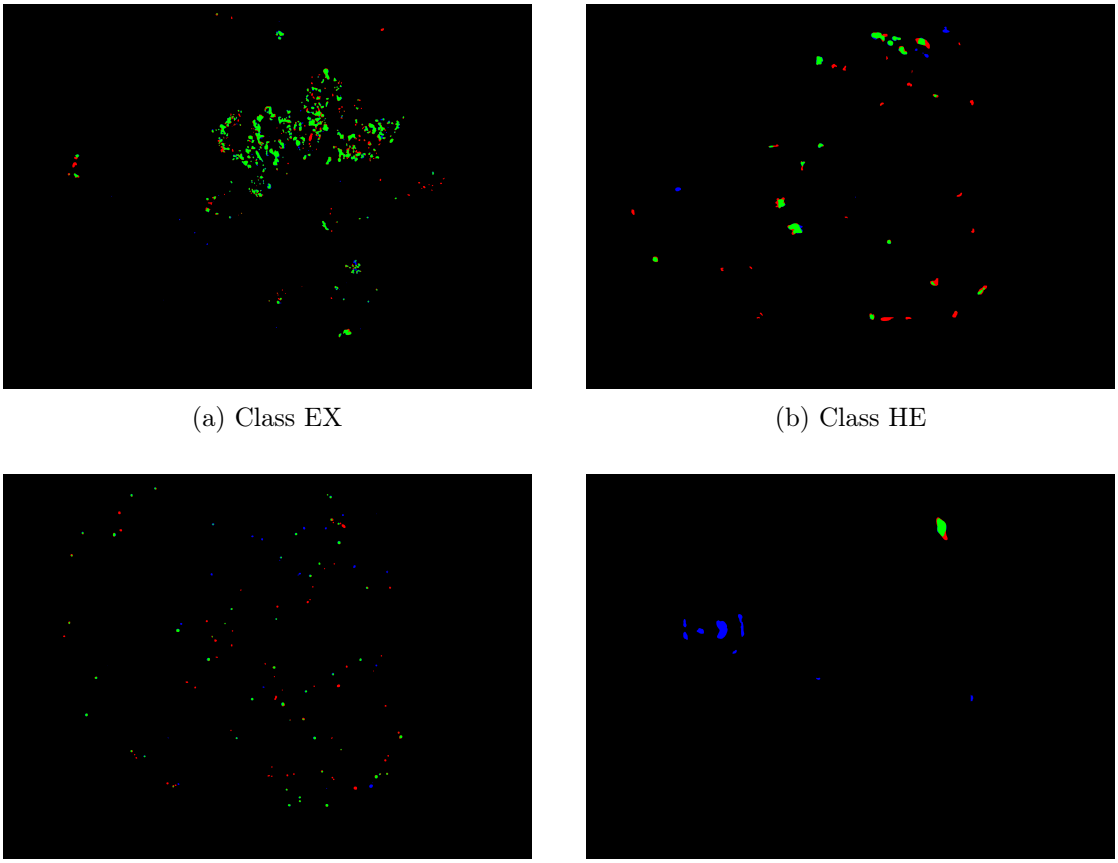
We can notice that, in general, FP and FN are proportioned. In particular, for class HE, the model tends to miss pixels, so there are more FN; but for class SE, it tends to wrongly classify pixels as lesion, so there are more FP.

The training and validation curves are illustrated in Figure 4.4, where the model's convergence can be observed. We can also see that validation curve starts decreasing, but after some epochs, it becomes quite stable and it does not improve, although it may seem that train loss could decrease even more.

Additionally, another comparison between the three combinations with best results is presented in Table 4.2, but now using the AUC precision-recall as metric. This way, the results can be compared to the ones uploaded to the challenge[1] *Diabetic retinopathy segmentation and grading challenge*. Table 4.3 summarizes the results of the first positions in the ranking.

Comparing the results obtained with AUC precision-recall metric and the ones ob-

[1]The leaderboard of Sub-Challenge 1 can be found in: https://idrid.grand-challenge.org/leaderboard/



(a) Class EX

(b) Class HE

(c) Class MA

(d) Class SE

Figure 4.3: Difference maps between ground truth and predictions. Green labels represent TP, red FN and blue FP.

Optimizer	**DA**	**Loss function**	**EX**	**HE**	**MA**	**SE**
Adam	False	Categorical cross-entropy	0.798	0.497	0.302	0.204
Adam	False	Dice coefficient	0.720	0.384	0.310	0.003
Adam	True	Generalised dice coefficient	0.687	0.420	0.354	0.385

Table 4.2: Results using U-Net with AUC PR metric

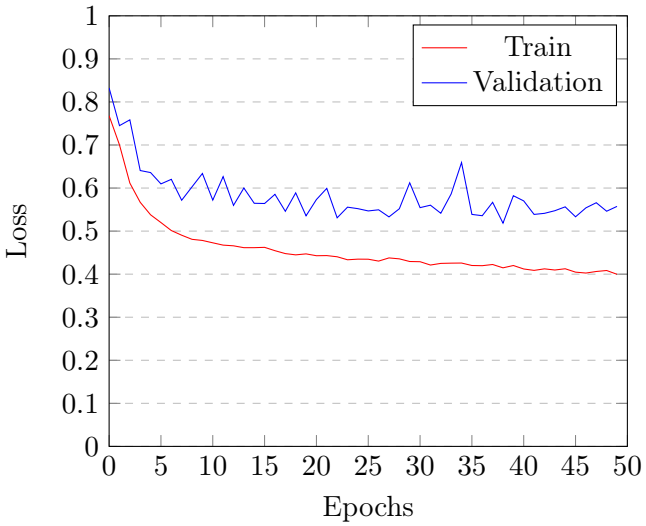


Figure 4.4: Training and validation curves of U-Net

Position	Team Name	EX	HE	MA	SE
1	VRT	0.7127	0.6804	0.4951	0.6995
2	PATech	0.8850	0.6490	0.4740	-
3	iFLYTEK-MIG	0.8741	0.5588	0.5017	0.6588
4	SOONER	0.7390	0.5395	0.4003	0.5369

Table 4.3: Summary of results of the first positions in Diabetic Retinopathy challenge

tained with dice score metric, we can notice a few distinctions. The one with best results, unlike with dice score metric, for classes EX and HE is achieved using categorical cross-entropy loss while the best results for classes MA and SE are obtained using generalised dice coefficient loss.

That is basically due to the fact that this metric is calculated with the probability maps outputs and dice score is calculated with the final predicted labels, whose values are integers. Therefore, it may happen that pixels are finally correctly predicted, even though their probability at the output is quite low.

Additionally, to have a high value of AUC, both precision and recall have to be high because the metric shows a trade-off between both metrics. As a consequence, if the model penalizes false positives or false negatives according to a particular criteria, it is possible that AUC is not very high because either precision or recall is low.

Consequently, we will use the dice score to compare the different experiments and the AUC precision-recall will only be used to compare the results with results from the Diabetic Retinopathy challenge. Having said that, best outcomes are a little worse than first positions in the ranking. However, depending on the class, the difference is bigger or smaller. In order to improve our results, we decided to add adversarial training.

4.3 Adversarial training

As explained in 3.2.1, two networks are needed to train our segmentation network using adversarial training. The actual segmentation network is the U-Net described in 4.2. In fact, the one with best performance is loaded and trained with this algorithm to further improve its performance. Therefore, this net is previously trained before adversarial training. In particular, for the results provided below, we used the network trained with Adam optimizer and generalised dice coefficient as loss function.

For the discriminative network, we used a CNN with a total of 10 convolutional layers and 1 fully connected layer to do the final classification between real or generated segmentation maps. The discriminator classifies the authenticity of data at image level, that is, the whole image is classified as real or fake.

It consists of the repeated application of 3×3 convolutions, each followed by a leaky rectified linear unit (Leaky ReLU), and a 2×2 max pooling with a stride 2 for downsampling. Batch normalization is also added after every convolutional layer as a regularization strategy. In the final layer, in order to do the binary classification, a global average pooling is applied followed by a fully connected layer, whose output is 1 for real or 0 for fake.

Adam optimizer and a learning rate of 0.0001 are used to train the discriminator. Additionally, two different losses, binary cross-entropy and mean squared error, have been tested and compared.

In order to train the generator, a combined model, which includes the segmentation and discriminative networks, is needed. In this model, the inputs are the images and ground truth labels, then, the segmentation maps using the segmentation network are computed and passed to the discriminator. The classification of the discriminator is the output of this combined model.

Adam optimizer is also used to train the combined model with a learning rate of 0.0001. The objective function, which plays a key role in adversarial training, is a combination between the loss of the discriminative model and the segmentation network loss. Therefore, the loss function has a component that penalizes that the discriminator classifies the generated images as generated and another component that penalizes the difference between the ground truth and generated segmentation maps. It can be expressed as:

$$L_{GAN} = \lambda \cdot L_{ADV} + L_{SEG} \tag{4.1}$$

where L_{ADV} (adversarial loss) is a binary loss that makes the generated segmentation maps to be more realistic so that the discriminator classify them as real, L_{SEG} (segmentation loss) is the loss used in the U-Net architecture that makes the generated segmentation maps similar to ground truth labels and λ is a trade-off coefficient.

Generalised dice coefficient is used as loss function for L_{SEG}, binary cross-entropy and mean squared error have been tested as loss functions for L_{ADV} and values of 0.05 and 0.07 have been tested as trade-off coefficients. The results, evaluated with dice score metric, are compared in Table 4.4.

L_{ADV}	λ	EX	HE	MA	SE
Binary cross-entropy	0.05	0.734	0.458	0.423	0.415
Binary cross-entropy	0.07	0.713	0.439	0.454	0.470
Mean Squared Error	0.05	0.707	0.476	0.462	0.405
Mean Squared Error	0.07	0.722	0.472	0.447	0.364

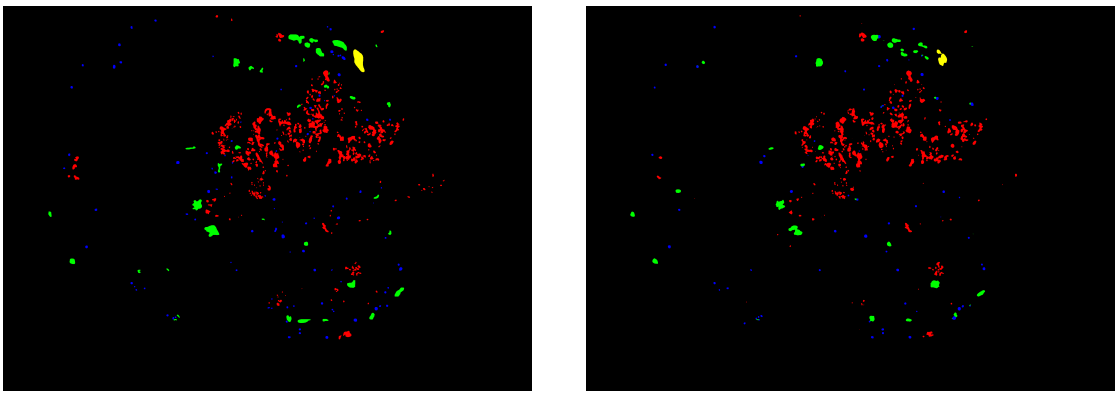
Table 4.4: Results using adversarial training with Dice score metric

As we can see, for some classes, the results improve the ones without adversarial training. However, it is difficult that the performance improve for all classes at the same

time, so we need to find a trade-off because when one class is segmented very precisely, another one worsens its accuracy.

We can also notice that the best results for each class are obtained with different functions losses and values of λ. Still, the one with best general results is obtained with mean squared error as adversarial loss function when λ is set to 0.05.

To qualitatively appreciate the performance of the model, a comparison between the predicted segmentation maps for the network with best performance and to the original labels is shown in Figure 4.5. It is the same test image illustrated in Figure 4.2 for the U-Net without adversarial training, so that they can also be compared.



(a) Ground truth (b) Predicted labels

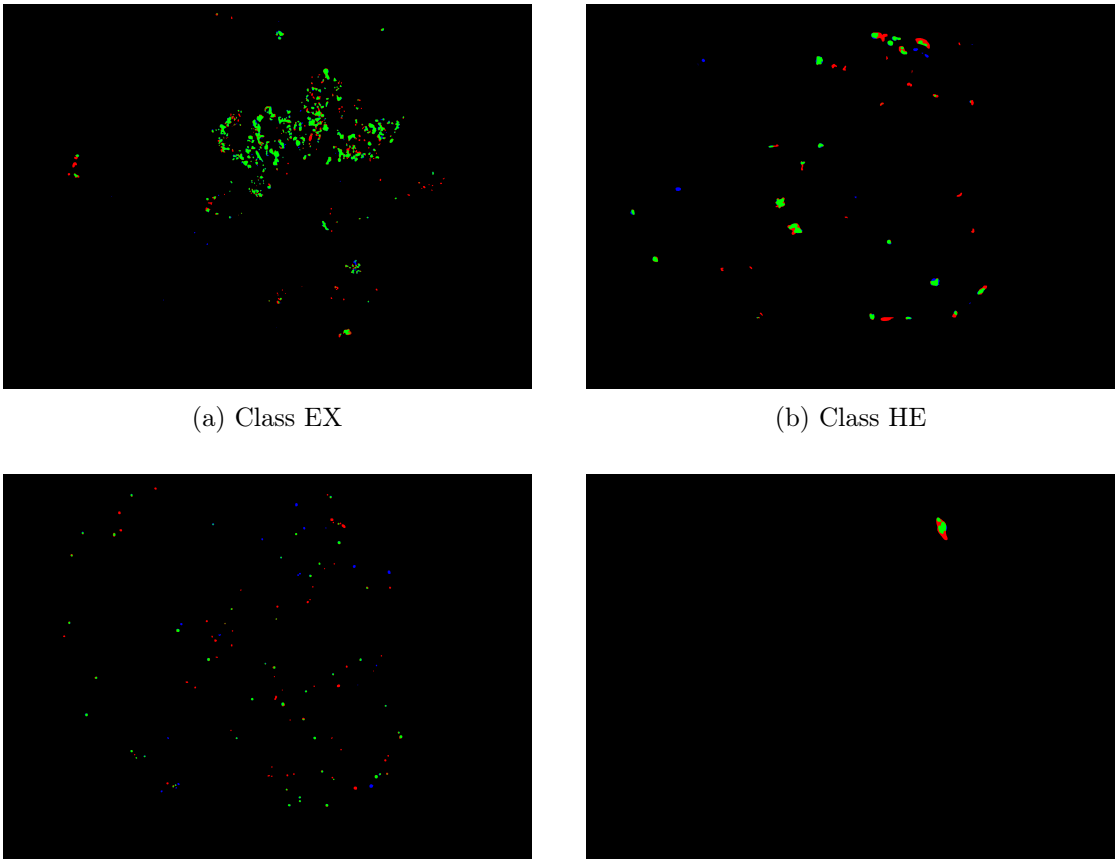
Figure 4.5: Comparison between real and generated segmentation maps using adversarial training. In red, the segmentation of class EX; in green, class HE; in blue, class MA; and in yellow, class SE.

In general, and for this particular image, the predicted labels are quite similar to the ground truth. The main visible difference is the large number of False Negatives in predictions of class MA.

Moreover, to better observe the difference between real and generated segmentation maps, the difference maps for the same example are presented in Figure 4.6.

We can observe that there are very few False Positives and main errors come from the presence of False Negatives in all the classes.

The training curves for the discriminative and segmentation networks are shown in Figure 4.7, where we can see that none of the network converge to zero. This is actually positive because it means that both sides are playing with each other and they are learning how to get better than the other, but none of them manage to trick the other too fast.



(a) Class EX

(b) Class HE

(c) Class MA

(d) Class SE

Figure 4.6: Difference maps between ground truth and predictions using adversarial training. Green labels represent TP, red FN and blue FP.

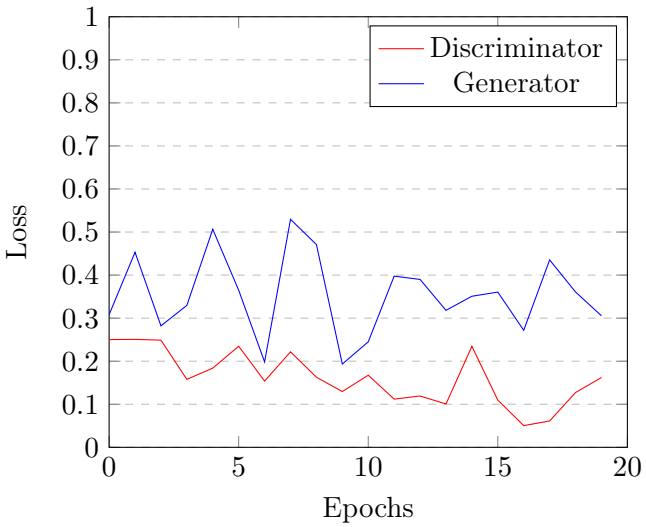


Figure 4.7: Training curves for discriminator and segmentation network in adversarial training

Additionally, the comparison between the best results obtained with different adversarial loss functions and λ values using AUC Precision-Recall metric is also presented in Table 4.5.

L_{ADV}	λ	EX	HE	MA	SE
Binary cross-entropy	0.05	0.718	0.430	0.315	0.415
Binary cross-entropy	0.07	0.703	0.417	0.358	0.456
Mean Squared Error	0.05	0.672	0.421	0.335	0.337
Mean Squared Error	0.07	0.707	0.438	0.329	0.294

Table 4.5: Results using adversarial training with AUC PR metric

As already mentioned before, the best results obtained with AUC precision-recall metric and the ones obtained with dice score metric are not the same due to the use of probability maps to calculate AUC PR. In this case, the best average performance would be achieved with binary cross-entropy as adversarial loss function and a λ value of 0.07. The results are still a little worse than first positions in the ranking for the sub-challenge 1 of *Diabetic retinopathy segmentation and grading challenge* (shown in Table 4.3), but they improved with respect to the ones obtained with the U-Net without adversarial training.

4.4 Discussion

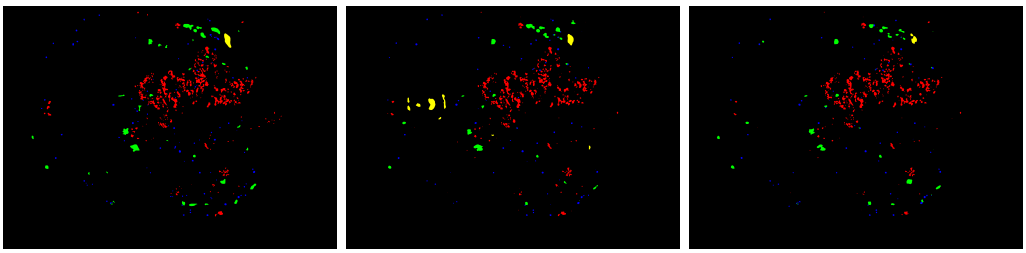
First of all, after doing several experiments and getting its results, we should point out that it is very difficult to achieve the best performance in all the classes. When the model fits very well to one particular class, then it automatically decreases its accuracy with another class. Therefore, a criteria must be established in order to correctly rank the results. This criteria should depend on the final purpose of the automatic segmentation.

If what we want is to detect Diabetic Retinopathy in its earliest stage, the most important class to correctly classify should be MA because it is the earliest visible sign of retinal damage. However, it is not very serious because it does not cause vision loss if controlled.

Consequently, we will not focus on MA, but on HE. The reason for that is the fact that when HE occurs, it means that DR is progressing and becoming a serious problem. First Haemorrhages appear in the last stage before developing Proliferative Diabetic Retinopathy, the most severe stage of the disease. Besides, the patients in that state need to follow a close medical monitoring because DR can progress at very fast pace.

Therefore, in case of doubt, when choosing best results we will focus on class HE as long as the other three classes are not significantly affected.

Having said that, comparing the results obtained with and without adversarial training, we can affirm that the addition of adversarial training improves the segmentation results. It is not only visible in the quantitative results, but also in the predicted segmentation maps. Figure 4.8 illustrates the difference between ground truth and predictions with and without adversarial training.



(a) Ground truth (b) Predicted labels without adversarial training (c) Predicted labels with adversarial training

Figure 4.8: Comparison between real and generated segmentation maps using adversarial training. In red, the segmentation of class EX; in green, class HE; in blue, class MA; and in yellow, class SE.

We can observe that the main improvement is in the reduction of False Positives. It

can be noted especially in class SE.

This improvement happens because the approach can capture and reduce the inconsistencies that a normal per-pixel loss average over the output does not capture. It learns high order regularities and transfers this information back to the segmentation model to achieve more realistic segmentation outcomes.

In particular, if we take a look at each class separately, we can observe that the one with biggest improvement is SE. It can be seen in the different maps, where all FP disappear. SE was also the class with lowest accuracy when using the U-Net without adversarial training.

Classes EX and HE also improve a little, but the change is not as big as the one with SE. However, class MA does not improve in any of the global best experiments with adversarial training. This might happen because the pixels belonging to that class are distributed in very small portions over all the retina. Therefore, it may be very difficult for the discriminator to find patterns to correctly classify the segmentation maps as real or fake and reduce inconsistencies.

Additionally, a table comparing best results with and without adversarial training is shown in Table 4.6. Dice scored metric is used to do the comparison.

Architecture	**EX**	**HE**	**MA**	**SE**
U-Net without adversarial training	0.694	0.458	0.477	0.351
U-Net with adversarial training	0.707	0.476	0.462	0.405

Table 4.6: Comparison of the addition of adversarial training

5. Budget

This project has been developed using the resources of the Image Processing Group at UPC. Therefore, the GPU power needed to implement the proposed solution did not have any cost.

Besides the expenses due to the computational power required, the main costs of this project come from the salary of the researchers and the time spent in it. The team who carried out the project is formed by me, the author of the work, and my advisor. I have considered myself as a junior engineer with a salary of 8€/hour and my advisor as a senior engineer with a salary of 20€/hour. The total duration of the project was 22 weeks, starting in mid-February and finishing in mid-July, as illustrated in the Gantt diagram in Figure B.1. Therefore, the budget can be calculated as shown in Table 5.1:

	Amount	Wage/hour	Dedication	Total
Junior engineer	1	8,00 €/ h	30 h /week	5,280.00 €
Senior engineer	1	20,00 €/ h	4 h /week	1,760.00 €
TOTAL				**7,040.00 €**

Table 5.1: Budget of the project

6. Conclusions and future development

The main goal of this project was to apply deep learning techniques to tackle a retinal lesions segmentation problem. We studied the use of a conventional U-Net and the addition of adversarial training as methods for automatic segmentation. We were able to prove that adversarial training applied to a segmentation network actually improve its performance.

This improvement can be observed quantitatively and qualitatively in the obtained results and they are achieved because the discriminative network is able to learn the higher order irregularities in the samples synthesized by the generator and transfer the information back to the segmentation model.

We also noticed that loss functions must include weights for the different classes when there is a high-imbalance between classes. However, the results could still be more accurate. That might be possible with a different choice of hyperparameters and doing much more experiments to correctly optimize the architecture's configuration.

Another important aspect to consider is the use of an appropriate metric to evaluate the models. I personally find more important that the final classification is properly done, without taking into account the probability maps at the output. Consequently, I think Dice Score gives a better insight of the performance of the model than Area Under the Curve Precision-Recall. That is why we optimized dice score when evaluating the networks, so it is quite difficult to correctly compare the results with the ones from the *Diabetic retinopathy segmentation and grading challenge*.

We should also mention that the dataset is very small, so the validation set has very few samples. Therefore, as a future development, a more robust estimate of the model's performance could be obtained by repeating the experiments with different partitions and training different networks to finally do an average of the results.

Additionally, the use of an adaptive sampling scheme would be very interesting to better tackle the imbalance between classes. There are several methodologies such as CASED [30] or the one proposed by Berger et al. in [31] that have proven to achieve good results in semantic segmentation problems with high data imbalance.

Bibliography

[1] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. In *Classification in BioApps*, pages 323–350. Springer, 2018.

[2] Kaggle competition. Diabetic retinopathy detection. https://www.kaggle.com/c/diabetic-retinopathy-detection. Online; Accessed: 2018-06-14.

[3] Donald S Fong, Lloyd Aiello, Thomas W Gardner, George L King, George Blankenship, Jerry D Cavallerano, Fredrick L Ferris, and Ronald Klein. Retinopathy in diabetes. *Diabetes care*, 27(suppl 1):s84–s87, 2004.

[4] National Eye Institute. Facts about diabetic eye disease. https://nei.nih.gov/health/diabetic/retinopathy. Online; Accessed: 2018-06-07.

[5] Joanna M Tarr, Kirti Kaul, Katarzyna Wolanska, Eva M Kohner, and Rakesh Chibber. Retinopathy in diabetes. In *Diabetes*, pages 88–106. Springer, 2013.

[6] Diabetic retinopathy segmentation and grading challenge. https://idrid.grand-challenge.org. Online; Accessed: 2018-06-06.

[7] Jaemin Son, Sang Jun Park, and Kyu-Hwan Jung. Retinal vessel segmentation in fundoscopic images with generative adversarial networks. *arXiv preprint arXiv:1706.09318*, 2017.

[8] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[9] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

[14] Pim Moeskops, Mitko Veta, Maxime W Lafarge, Koen AJ Eppenhof, and Josien PW Pluim. Adversarial training and dilated convolutions for brain mri segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 56–64. Springer, 2017.

[15] Wei Dai, Nanqing Dong, Zeya Wang, Xiaodan Liang, Hao Zhang, and Eric P Xing. Scan: Structure correcting adversarial network for organ segmentation in chest x-rays. 2018.

[16] Wentao Zhu, Xiang Xiang, Trac D Tran, Gregory D Hager, and Xiaohui Xie. Adversarial deep structured nets for mass segmentation from mammograms. In *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pages 847–850. IEEE, 2018.

[17] Yuan Xue, Tao Xu, Han Zhang, Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l_1 loss for medical image segmentation. *arXiv preprint arXiv:1706.01805*, 2017.

[18] Usman M Akram and Shoab A Khan. Automated detection of dark and bright lesions in retinal images for early detection of diabetic retinopathy. *Journal of medical systems*, 36(5):3151–3162, 2012.

[19] GG Gardner, D Keating, Tom H Williamson, and Alex T Elliott. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. *British journal of Ophthalmology*, 80(11):940–944, 1996.

[20] Udyavara R Acharya, Choo M Lim, E Yin Kwee Ng, Caroline Chee, and Toshiyo Tamura. Computer-based detection of diabetes retinopathy stages using digital fundus images. *Proceedings of the institution of mechanical engineers, part H: journal of engineering in medicine*, 223(5):545–553, 2009.

[21] Harry Pratt, Frans Coenen, Deborah M Broadbent, Simon P Harding, and Yalin Zheng. Convolutional neural networks for diabetic retinopathy. *Procedia Computer Science*, 90:200–205, 2016.

[22] Yehui Yang, Tao Li, Wensi Li, Haishan Wu, Wei Fan, and Wensheng Zhang. Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 533–540. Springer, 2017.

[23] Jen Hong Tan, Hamido Fujita, Sobha Sivaprasad, Sulatha V Bhandary, A Krishna Rao, Kuang Chua Chua, and U Rajendra Acharya. Automated segmentation of exudates, haemorrhages, microaneurysms using single convolutional neural network. *Information Sciences*, 420:66–76, 2017.

[24] Muthu Rama Krishnan Mookiah, U Rajendra Acharya, Chua Kuang Chua, Choo Min Lim, EYK Ng, and Augustinus Laude. Computer-aided diagnosis of diabetic retinopathy: A review. *Computers in biology and medicine*, 43(12):2136–2155, 2013.

[25] Stanford University. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.stanford.edu. Online; Accessed: 2018-06-17.

[26] Lucas Fidon, Wenqi Li, Luis C Garcia-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sébastien Ourselin, and Tom Vercauteren. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. *arXiv preprint arXiv:1707.00478*, 2017.

[27] William R Crum, Oscar Camara, and Derek LG Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE transactions on medical imaging*, 25(11):1451–1461, 2006.

[28] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer, 2017.

[29] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[30] Andrew Jesson, Nicolas Guizard, Sina Hamidi Ghalehjegh, Damien Goblot, Florian Soudan, and Nicolas Chapados. Cased: Curriculum adaptive sampling for extreme data imbalance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 639–646. Springer, 2017.

[31] Lorenz Berger, Eoin Hyde, Jorge Cardoso, and Sebastien Ourselin. An adaptive sampling scheme to efficiently train fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1709.02764*, 2017.

[32] Natàlia Gullón. Code of the project. https://github.com/nataliagullon/segmentation-retinal-lesions.

A. Code of the project

The code of the project can be found in a GitHub repository called *Segmentation-Retinal-Lesions* [32]. It is developed in Python using Keras framework running on top of TensorFlow.

B. Work Plan

B.1 Tasks and milestones

- WP1: Project proposal and work plan
  - T1: Project description
  - T2: Project development plan
  - T3: Document review and approval
  - **Milestone:** Documentation (NGullon-PP-and-WP.pdf)
- WP2: Information research
  - T1: CS231n Stanford course about deep learning techniques
  - T2: Study of GAN architectures and its medical applications
  - T3: Familiarization with Keras framework
  - **Milestone:** Software
- WP3: Software development
  - T1: Learn how to handle GAN models
  - T2: Study different kinds of implementations used in GANs
  - T3: Software adaptation to a particular medical application
  - T4: Improvement of the software
  - T5: Results assessment until date
  - **Milestone:** Software and documentation
- WP4: Critical review
  - T1: Analyse the current development and compare it to the initial work plan
  - T2: Review the work plan and make changes if necessary
  - T3: Document review and approval

Milestone: Documentation (NGullon-CR.pdf)

- WP5: Test and results assessment
  - T1: Test the performance of the model
  - T2: Last improvements of the software
  - T3: Get final results and compare them to the state-of-the-art results
  - **Milestone:** Software
- WP6: Final report
  - T1: Write the document
  - T2: Document review and approval
  - **Milestone:** Documentation (NGullon-FR.pdf)
- WP7: Oral presentation
  - T1: Slides presentation
  - T2: Presentation review and approval
  - T3: Presentation rehearsal
  - **Milestone:** Documentation (NGullon-pres.pdf)

B.2 Gantt diagram

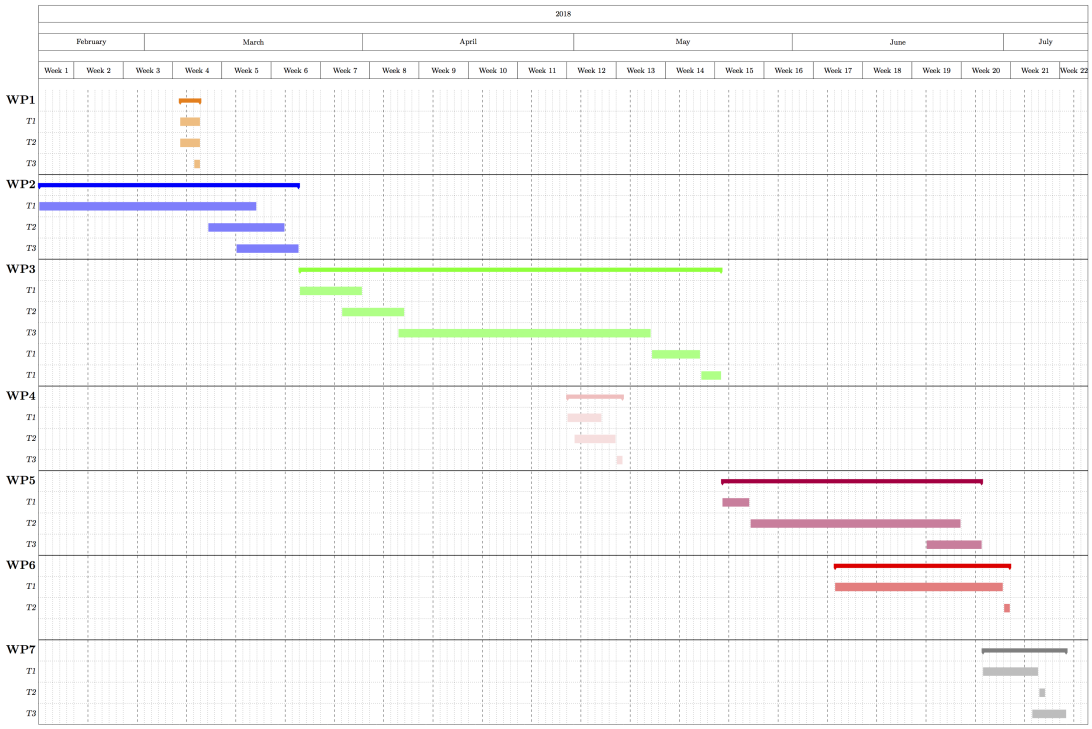


Figure B.1: Gantt diagram

C. Additional results

In this chapter, results from all the experiments presented in 4 are shown, so that they can be compared qualitatively. Setups are indicated and a test image is used to do the comparison in Figure C.1.

Experiment A setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** False
- **Batch size:** 8
- **Loss function:** Categorical cross-entropy
- **Adversarial training:** No

Experiment B setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** False
- **Batch size:** 8
- **Loss function:** Dice coefficient
- **Adversarial training:** No

Experiment C setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** True
- **Batch size:** 8
- **Loss function:** Generalised dice coefficient
- **Adversarial training:** No

Experiment D setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** True
- **Batch size:** 8
- **Loss function:** Generalised dice coefficient
- **Adversarial training:** Yes
  Adversarial loss function: Binary cross-entropy
  Trade-off coefficient (λ): 0.05

Experiment E setup

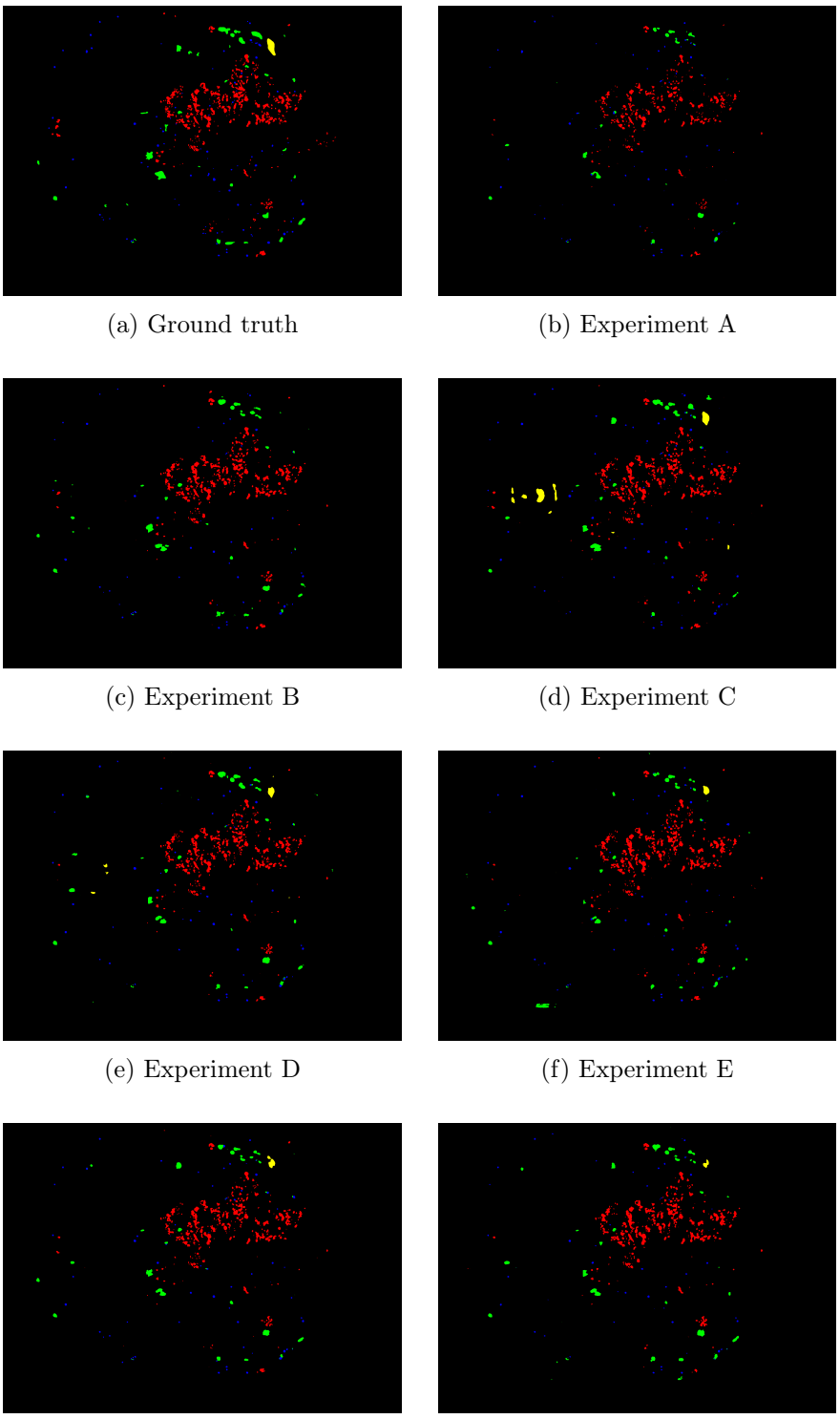
- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** True
- **Batch size:** 8
- **Loss function:** Generalised dice coefficient
- **Adversarial training:** Yes
  Adversarial loss function: Binary cross-entropy
  Trade-off coefficient (λ): 0.07

Experiment F setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** True
- **Batch size:** 8
- **Loss function:** Generalised dice coefficient
- **Adversarial training:** Yes
  Adversarial loss function: Mean Squared Error
  Trade-off coefficient (λ): 0.05

Experiment G setup

- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Data augmentation:** True
- **Batch size:** 8
- **Loss function:** Generalised dice coefficient
- **Adversarial training:** Yes
  Adversarial loss function: Mean Squared Error
  Trade-off coefficient (λ): 0.07



(a) Ground truth (b) Experiment A

(c) Experiment B (d) Experiment C

(e) Experiment D (f) Experiment E

(g) Experiment F (h) Experiment G

Figure C.1: Comparison between real and generated segmentation maps. In red, the segmentation of class EX; in green, class HE; in blue, class MA; and in yellow, class SE.

Acronyms

CNN Convolutional Neural Network.

DL Deep Learning.

DR Diabetic Retinopathy.

EX Hard Exudates.

FN False Negatives.

FP False Positives.

GAN Generative Adversarial Network.

HE Haemorrhages.

MA Microaneurysms.

NN Neural Network.

SE Soft Exudates.

TP True Positives.