Raquel de Frutos Arranz

A TOOL FOR FACILITATING THE GENERATION OF GOLD TRUTH IN MEDICAL IMAGES FOR POSTERIOR DEEP LEARNING ANALYSIS

FINAL MASTER'S THESIS

Directed by Dr. Aïda Valls & Mohammed Yousef Salem Ali

University Master's Degree in Computer Security Engineering and Artificial Intelligence

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2020-2021

Contents

Abstract

The objective of this Master Thesis is to develop an annotation desktop software to assist medical specialists when manual labelling medical images to mark some targets they detect. This task is essential to train automatic Machine Learning methods for segmentation or classification of images. The tool will then facilitate the generation of gold truth images, which will be used as a dataset for some Deep Learning analysis, which is a recent technique for image classification that is achieving very good performance. The software will allow the user to annotate using different shapes and adding the desired number of labels. After annotating an image, its mask and annotation will be saved to then feed a deep learning method that is also implemented in the software.

Furthermore, as manual labelling requires much time and the knowledge of an expert, it is not easy to collect a huge dataset of annotated medical images. For this purpose, the proposed software will also include augmentation processes, to automatically generate more images.

The software proposed is designed for classification tasks related for Diabetic Retinopathy disease, but it can be also used for any other image analysis tasks, in medical domain or not.

Keywords

Medical image annotation, Machine Learning, Deep Learning, Gold Truth, Diabetic Retinopathy

Chapter 1. Introduction.

1.1.Problem setting.

Computers have and will continue to play an important role in the everyday life at present. Computer vision systems try to understand the real world as a human eye does.

Image analysis refers to these techniques which are applied to find information from an image or an image dataset. Deep learning methods are based on Neural Networks, a well-known technique of Machine Learning, developed in the field of Artificial Intelligence. Deep learning techniques using computers are mainly applied to image analysis such as image recognition, classification or segmentation, tasks which were usually performed by the human eye, so there are still several challenges and difficulties. They have become very popular in the last years due to its high performance in many different fields of application [1].

We will focus on the case of objects detection in medical images (e.g., normal, or abnormal structures in a part of the body). In contrast to the other image processing tasks, such as the enhance the aesthetic of an image or the extraction of some features of the image.

The use of deep learning methods for image analysis requires of a large dataset of labelled images (called Ground Truth). The labels are marks that indicate where the elements of the image are that the computer must find. To generate the ground truth, a person must first visualize the image and mark on it the targets that the system must detect. For example, we must mark cancer tumours in a tomography, or exudates in an eye fundus image. Manual labelling is time consuming as it needs to be precise in the marking of the different elements in the image.

Consequently, it is difficult to collect large sets of labelled images. To overcome this scarcity, we can use data augmentation methods, which generate several versions from a single labelled image.

Given that problematic, this master thesis has de goal of developing a desktop software tool for assisting the medical personnel in labelling the images and collecting a proper dataset to be used as gold truth, especially for the Diabetic Retinopathy disease.

ITAKA-RIVI group, a research group founded in January 2007, is working on a research project related to the Diabetic Retinopathy disease. Diabetic Retinopathy is an ocular pathology that appears in about 12% of diabetic people and which can cause the blindness of the person if not detected and treated in early stages. In this research project, they have developed a clinical support application to estimate the risk of developing this disease [2]. The annotation tool proposed on this Master Thesis aims support the work done in this project.

1.2.Brief overview of Deep Learning in image analysis.

Deep learning algorithms have been used to analyse medical images, especially in recent years. Algorithms are used to different classification, detection, or segmentation in medical images. When using these systems to diagnose a disease or medical problem it is important to become results with high sensitivity and specificity, which should be at least higher than 80% [3].

Usually some pre-trained networks (transfer learning) are used with good results [4] on classification task, medical images are taken to determine a diagnostic. These techniques are also used to classify objects or lesions within images. Deep learning algorithms have been already proved with good results in medical images classifications [5] [6].

Dealing with medical images is not always as simple as working on other areas. One of the main problems of the deep medical analysis is the high variety of formats of the medical image. For example, most of the input data is 3D. Furthermore, the use of deep learning in medical images, in contrast to common use in 2D colour image, requires assessment and validation to get annotated or labelled medical images and it requires the work of a specialist and time [4]. Furthermore, annotated medical images usually have label noise or uncertainty about tumours in images which could be seen as label subclasses. One possible solution to this problem could be the application of intelligence already on the training part with the risk of finding noise again.

Then, when using convolutional neural networks on medical image tasks, the output class should usually also include localization of each pixel [7], and not just a class label. Other problem related to the application of deep learning models on medical images is the lack of huge databases [8] of images and annotations, which are needed to train the deep learning algorithms, because usually this kind of medical data was not digital in the past years. Other problem is the variability among annotations or opinions from different specialists.

On top of this, the analysis of medical images finds huge class imbalance or difficulties to get images containing the abnormal class (for example images with a malicious tumour). To face this issue, several data augmentation algorithms can be performed [9].

1.3.Project objectives.

This master thesis has de goal of developing a desktop software tool for assisting the medical personnel (Retinal Specialists, Glaucoma Specialists and Ophthalmologists) in labelling the images and collecting a proper dataset to be used as gold truth. The main sub goals in this software are:

1. Enable the upload of different types of images, in different formats and sizes (as we want a general tool to be used in files obtained from different types of cameras or machines).
2. Enable the marking of multiple types of lesions/objects with assistance from the software. The set of labels must be configurable as well as its marking type.
3. Include different data augmentation methods to generate new images and masks.
4. Enable the possibility to train and test a deep learning model from the system.

5. Test the system with the Diabetic Retinopathy images. The system can be further used in different domains or medical problems, but we will focus mainly on different classification tasks related to the Diabetic Retinopathy disease, in the frame of a Spanish research project, to continue with the Diabetic retinopathy project [2].

1.4. Document structure.

The rest of this Master Thesis is structured as follows. In Chapter 2 we introduce the main deep learning techniques, focusing on the use on medical images. Chapter 3 discuses about data augmentation. Chapter 4 describes image annotation and shows some image annotation tools. In Chapter 5 the annotation tool developed on this Master Thesis is presented and tested. Chapter 6 analyses the implemented work and comment possible improvements.

Chapter 2. Deep Learning Theory.

Deep Learning is a technique which uses a huge datasheet of labelled images to allow computers to learn and predict features needing numerous levels or modules but without any predefined rule [1].

There are two categories inside machine learning: supervised and unsupervised learning. Supervised learning tries to find model parameters to predict data with loss functions using a datasheet based on features and labels and a process of training. On the other hand, unsupervised learning tries to find features on data without knowing anything of these labels.

Deep Learning methods are a recent extension of a supervised model of classification named Neural Network, that appeared in 1943 by McCulloch and Pitts. The increase of power of the computers has enabled the creation of a next generation of quite complex Neural Networks. The basic types of these models are the following:

-Neural networks or Artificial Neural Networks (ANNs) are models which try to simulate the human brain [10].

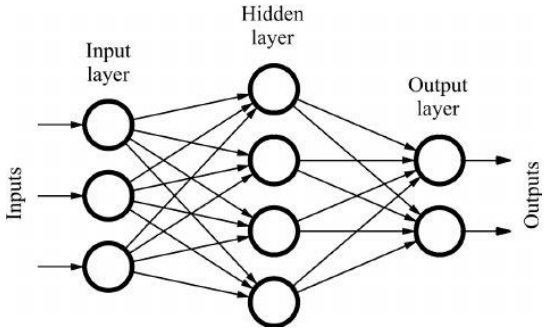


Figure 1. Example of a feed-forward neural neuron [67]

Each Neural Network can be seen as a combination of some input layers, hidden layers, and output layers. Each hidden layer follows a combination of the input layers, where $\omega_{ij}^{(\alpha)}$ are the weights and $b_i^{(\alpha)}$ the biases, as follows:

$$h_{i|}^{(\alpha)} = \sum_{j=1}^{N_{\alpha-1}} \omega_{ij}^{(\alpha)} x_j + b_i^{(\alpha)} \qquad (1)$$

The output of each neuron, represented as i, is calculated after applying the activation function:

$$V_i^{(\alpha)} = g\,(h_i^{(\alpha)}) \qquad (2)$$

The activation function can adopt different approaches, such the sigmoid, the Heaviside function or the Rectifier Linear Unit (ReLU), which will be used on the U-Net architecture, explained later:

$$g(x) = \max(0, x) \qquad (3)$$

Merging both equations, the final output is:

$$\vartheta_i = g\left(\sum_j \omega_{ij}^{(2)} V_j^{(1)} + b_i^2\right) = g\left(\sum_j \omega_{ij}^{(2)} g\left(\sum_{k=1} \omega_{jk}^{(1)} x_k + b_j^{(1)}\right) + b_i^2\right) \quad (4)$$

The difference between the desired output by the adjust of the parameters (Back-Propagation) and the outcome with the real parameters can be obtained with a Loss function, such the cross-entropy:

$$L_y = -\log\left(\frac{e^{fy}}{\sum_j e^{fj}}\right) \quad (5)$$

In this case, the detection of objects on an image, such as the abnormalities present on a fundus image, is a segmentation or a classification problem where the output is a scored class. To find the smallest distance it is needed to apply some method, as the gradient descent:

$$\nabla L_y = \left(\frac{\delta L_y}{\delta \omega_{11}^{(1)}}, \dots, \frac{\delta L_y}{\delta \omega_{mn}^{(\alpha)}}, \frac{\delta L_y}{\delta b_1^{(1)}}, \dots, \frac{\delta L_y}{\delta b_m^{(\alpha)}}\right) \quad (6)$$

The parameters will be updated at each iteration or epoch over the training data until the accuracy is 1, by selecting an appropriate learning rate η:

$$\omega'^{(\alpha)}_{ij} = \omega_{ij}^{(\alpha)} + \eta \nabla L_y \quad (7)$$

-Deep Convolutional neural networks (CNN) are a special type of Neural Networks with an excellent performance in pattern recognition tasks due to the multiple feature extraction steps [11], especially in computer vision applications. CNN are also made up of neurons with learnable wights and biases.

Usually, a CNN is composed by alternate convolutional and pooling layers followed by fully connected layer. A convolutional layer is formed by some convolutional kernels to divide the image into small pieces and extract features. Pooling layers helps to determine the relative position of the features. The activation function is applied as a decision to help learning patterns. Dropout adds some randomness to improve the network and the fully connected layer is usually used at the end of the network for classification.

-Recurrent neural networks (RNNs) are a type of artificial neural networks based on a timing sequence of nodes [12], with good results on handwriting or speech recognition as they can process inputs with diverse lengths.

On the other hand, unsupervised learning refers to algorithms which learns patterns from data which have not been tagged, which will not be significant for this Master Thesis. Some unsupervised learning techniques which are usually used on medical images tasks are Auto-encoders (AEs) and stacked auto-encoders (SAEs), Restricted Boltzmann machines (RBMs) and deep belief networks (DBNs), Variational Auto-Encoders (VAE) and generative adversarial networks (GAN) [4].

2.1.Deep learning in medical images.

As summarized on the introduction, deep learning is widely used on medical images analysis. The detection of organ, region and landmark in medical images is another goal to detect small lesions, which also deals with the treatment of complex 3D images, which are usually treated as a mix of 2D orthogonal plane. Usually, CNNs are used to solve these kinds of problems.

Segmentation is widely extended as a deep learning technique in medical images, specially to detect parameters about volume and shape. Image segmentation refers to the process of splitting an image into several regions and it is the first and a principal task in image analysis and pattern recognition [13]. CNNs and RNNs are used for this purpose.

Deep Learning algorithms are used also for further task on medical images, such as registration, Content-based image retrieval (CBIR) or image generation. Registration consists on performing some coordinate transformation between two images. CBIR id used to identify similarities on patients' clinical histories. For creating new images, CNNs are used.

The most used DL methods in medical images are supervised algorithms such as Artificial Neural Networks (ANNs) or Deep Convolutional Neural Networks (CNNs). Usually, for medical analysis simple CNN architectures such LeNet or AlexNet are used, but also others like GoogLeNet Inception v3 [4].

Deep learning techniques are used on several medical application areas [4]. For example, to segment and classify brain diseases, especially for Alzheimer. Apart of this, other diseases such as tumours or white matter lesions and methods such mapping learning from patches are also studied using DNNs. Furthermore, DL algorithms based on CNNs and RNNs are used to detect nodules or intestinal lungs to detect chest diseases. Another area of interest is breast, where experts use DNN to deal with 2D images, especially for the detection of breast cancer. DL algorithms are also used on cardiac images, where some techniques such MRI are used to get results, especially on left ventricle segmentation applications. On musculoskeletal applications, some techniques are implemented to segment and detect parts and abnormalities in bones or joints. On abdomen images, some techniques such as MRI and CT are used to segment and detect organs (liver, bladder, pancreas, bladder) to determine tumours. Also, DL techniques are used on digital pathology and microscopy, to detect, segment, or classify organs or diseases in large scale images and to normalize images. A huge area of interest of DL in medical images is eye, which will also concern us on this project. Most efforts are done to study the colour fundus imaging or CFI to detect eye or retinal diseases or abnormalities, such diabetic retinopathy.

2.2.Deep learning on Diabetic Retinopathy disease.

As introduced before, to continue with the research performed by the ITAKA-RIVI group, this Master Thesis will focus on Diabetes Retinopathy detection through medical images.

Diabetes mellitus (DM) is a chronic disease which affects to 387 million people in the world [14] and it is expected than the cipher grows in the future.

A consequence of diabetes is the ocular problem diabetic retinopathy (DR) which occurs when fluid from the blood vessels damage the retina [15] and is one of the leading causes of blindness among people worldwide [2] [16]. One of each tree persons diagnosed with DM, will be also affected with DR [14]. In Europe, 3%–4.1% people suffer DR and in Spain, 26.1% inhabitants with Diabetes Type II also are affected by DR.

DR can be classified according to the severity of the disease [17]. For example, the ICDR scale [18] presents 5 categories: no DR, mild non-proliferative DR, moderate non-proliferative DR, severe non-proliferative DR and proliferative DR.

-Mild retinopathy [19]: some small microaneurysms in the tiny blood vessels are found.

-Moderate retinopathy: on this stage, some blood vessels are blocked.

-Severe retinopathy: more blood vessels are blocked, leading to retina abnormalities or haemorrhages.

-Proliferative retinopathy: on this severe stage the blood vessels are severe affected.

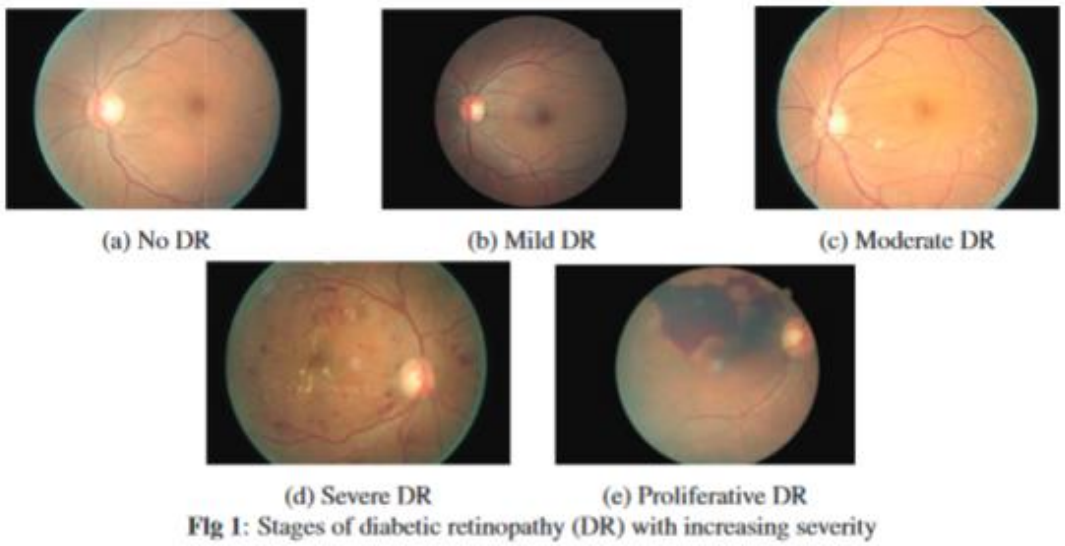


Figure 2. Five stage DR representation [20].

There is huge research with retina fundus image to detect Diabetic Retinopathy (DR), using different DR techniques and algorithms. As other DR techniques, the process consist on several

steps: data acquisition, pre-processing of images, feature extraction and classifier or algorithm used. A good pre-processing and augmentation can improve the results, better than changing the network architecture.

A prompt retina analysis to detect different ocular lesions such microaneurysms, neovascularization, haemorrhages and exudates would be useful to detect this disease [2], which leads to a huge variability and dependence of an expert analysis to detect DR [16]. When there are four or more microaneurysms are found on a retina fundus image, DR can be usually diagnosed [21]. Exudates are typically showed as whitish/yellowish patches of different sizes and shapes [17]. An automatic grading of diabetic retinopathy would help to an early detection and improve the treatments [6]. Deep learning can be used to train an algorithm to detect it.

The best method for DR screening is retinal photography by nonmydriatic fundus camera [21], but a periodic screening is an expensive preventive method difficult to be applied to all patients. A better way to early diagnose and control DR could improve and cheapen DR therapy while reduce the risk of blindness [3].

A fundus image shows the back of the eyeball which mainly consist on the retina whose analysis allows the expert to detect abnormalities in the eye and the blood vessels [22].

In the literature [23] we found that some pre-processing steps produce good results when applied to retina fundus images, such as contrast and brightness corrections. They also indicate that a green channel image helps to detect retina abnormalities.

There have been published some competitions to identify DR in eye fundus images [24].

Some of the researchers developed an algorithm to detect diabetic retinopathy in retinal images and scale it into severity categories using some dataset as the EyePACS or the Messidor-2, obtaining a good machine learning with high sensitivity and specificity, higher than a 90% [6] [5], but these algorithms still need to be further proved before being implemented in medical institutions or substituted for ophthalmologic test. A new version on this deep learning algorithm but using different datasheets will be presented on [25] but obtaining lower results. Most of them uses the Inception-v3 architecture and some normalizing and pre-processing steps such as resizing each image to 299x299 pixels and locating the fundus centre in the middle of the image.

Further research [16] improve these algorithms, for example taking a bigger image resolution and changing the architecture to Inception-v4 architecture, obtaining a better AUC.

On the other hand, the risk of being diagnosed with DR has been also studied with other methods such as decision support system in [2] or fuzzy rules [3] [26]. Other study [27] developed an automatic decision tool based on supervised pixel classification to detect microaneurysms and

haemorrhages, which are signs of DR visible as dark signs on retina images, to detect or classify the severity of the disease. This program first deleted the background or the main components of the retina to avoid false positives and used pre-processed images. MAs and HAs are then segmented by h-maxima transformation, thresholding and feature extraction.

Deep learning methods based on Convolutional Neural Networks (CNNs) [20] or using support vector machines (SVM) [15] to detect exudates and in [28] to identify or classify DR into the five classes on fundus images have been also studied. A mixed system of fuzzy logic and SVM is proposed to classify DR [29]. Bayesian statistical classifier [17], back propagation neural network [19], some algorithms based on the edges [30] or Gabor and Gaussian filters [31] are also used to classify and detect DR. A computer tool was developed in [22] to detect abnormalities related to DR in blood vessels. There is also commercial software [32] used to analyse fundus retina image and detect microaneurysm or haemorrhage and diagnose DR.

2.3.Existing datasets of DR.

There are several databases and datasets which contains retina fundus images and are used on diabetic retinopathy medical images tasks. Some of them are: EyePACS, which is license-free Web-based DRS system which allows the access to images, information and collaboration among experts about diabetic retinopathy [33], Messidor-2 data set [34] which contains 1748 images but it does not contain official annotations of the images, Kaggle [35], DIARETDB0 [36] contains 130 color fundus images, 20 of them showing DR and the rest no, the DIARETDB1 [37] [38] which is a public database containing 89 colour fundus images and ground truth images marked by experts using an annotation tool, DRIVE [39] database contains 40 colour fundus images, but only 7 are affected by DR, VARPA Retina Images for Authentication (VARIA) is composed by 233 grayscale images [40], VICAVR database [41] contains 58 colour retina images, APTOS [42], ODIR [43], Drishti-GS [44], STARE [45], CMIF (collection of multispectral images of the fundus) [46], ROC (retinopathy online challenge) [47], REVIEW (retinal vessel image set for estimation of widths) [48].

Despite the existence of these databases, in many of them the images are only labelled indicating the level of DR or the presence/absence of DR, but the images does not contain annotations with the localization of the different types of lesions, or the annotation of the normal eye structures (blood vessels, optic disk and macula), which are relevant for improving the analysis of these kind of images for this pathology. This lack of annotations is the motivation of this Master Thesis work.

2.4.U-Net architecture.

There are several architectures valid for deep learning. On this project we will employ the U-Net architecture, which is now explained.

The U-Net architecture consist of 23 convolutional layers divided into a contracting and an expansive path [7]. The contracting one, on the left part, is composed by sets of two 3x3 convolutions, a rectified linear unit (ReLU) and 2x2 max pooling operation with stride 2 for down sampling. These sets are repeated through the contracting path and the number of feature channels are doubled at each step.

On the other hand, the expansive path is composed by an up sampling of the feature map followed by a 2x2 convolution, a concatenation with the cropped feature map and two 3x3 convolutions, each followed by a ReLU. The last layer is a 1x1 convolution to map each 64-component feature vector to the number of classes. This network composed by successive layers is used to increase the output resolution and to localize features. This model also applies data augmentation techniques with elastic deformations, which also allows the network to learn invariance.

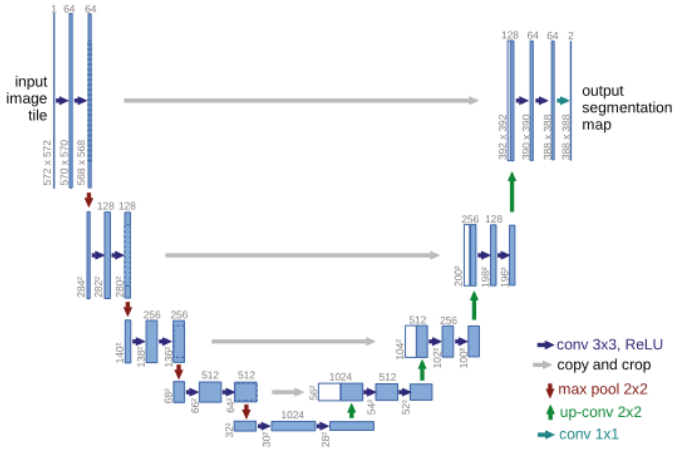


Figure 3. U-net architecture [7].

2.5.Evaluation.

To measure the accuracy of a classification or prediction algorithm, the ROC curve or receiver operating characteristic is used, especially in medical research [49]. It is based, as we discussed before on the measures of sensitivity (SN) and specificity (SP) which can be calculated as, being true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Sensitivity measures the proportion of positive results that are correctly considered positive with respect to all positive results. Specificity measures the proportion of negative results that are correctly considered negative, with respect to all negative points.

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \qquad (8)$$

$$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \qquad (9)$$

Other measures are also used to determine the performance of an algorithm, such as precision, recall or accuracy. Precision measures the number of correct positive results divided by the number of positive results predicted by the classifier. Recall is the number of correct positive results divided by all the results that should have been identified as positive.

$$Precision = \frac{TP}{TP+FP} \qquad (10)$$

$$Recall = \frac{TP}{TP+FN} \qquad (11)$$

$$Accuracy = \frac{TP+TM}{TP+FP+FN+TN} \qquad (12)$$

Furthermore, Accuracy is the ratio of correct predictions to the total number of input samples. Logarithmic Loss, a good metric for multi-class classifications, penalise the false classifications.

On the other hand, IoU or Jaccard index, compares the similarity or intersection between two shapes or boxes to determine the true positive and false positives in a set of predictions [50].

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} \qquad (13)$$

Chapter 3. Data Augmentation.

Image augmentation is a technique frequently used to expand the datasheet in computer vision or deep learning. To face the problem of the limited data in medical analysis explained before, data augmentation methods can help to transform and increase the number of data and thus improve the algorithms, adding more information to the training dataset [8]. One single image can be augmented several times with different methods or styles, getting a final image with a different appearance but same image content [51].

Huge medical images datasheets are often difficult to achieve because image taking, and labelling is expensive and need to be done by an expert. Patient privacy and unusual diseases are also factors that play against. There are some packages specially designed for medical image augmentations, which fit to biomedical image formats and time-series or z-stacked/layered images.

Oversampling is a method to generate more images of the less dominant class.

3.1. Data augmentation techniques.

3.1.1. Geometric transformations.

When dealing with geometric transformations it is important to keep the image save or keep the image labels, which add an extra work to the user to validate if the new images could be accepted or not.

- Flipping: it is an easy transformation, and it exists horizontal and vertical axis version.
- Cropping operations result on image size reduction or translation.
- Rotation augmentations consist of rotate the image left or right an angle between 1 and 359º.
- Translation or shifting means moving the image direction up, down, right, or left and filling the empty space with noise, usually Gaussian noise.
- Noise addition augmentations add a matrix of noise, usually Gaussian.

3.1.2. Colour or photometric transformations.

Colour images are formed by 3 matrices to represent the R, G and B values. Each matrix has a size height x width. On the other hand, grayscale images are formed by a single matrix height x width. By dealing with matrix some operations related to colour can be performed such as one colour channel (R, G or B) conversion, brightness, or intensity modification. Also colour transformations not always keep the image labels information if some representative colour is deleted.

3.1.3. Augmentation based on deep learning.

- Feature space augmentation. Neural networks can be used to achieve images with less dimensions.
- GAN or Generative Adversarial Network refers to an artificial creation of data using min-max strategy based on unsupervised deep learning [51]. Adversarial training means the implementation of some networks with contrasting objectives. This recent method uses two adversarial networks, one of them, the generator or G(z) to generate the new image, and the other one, the discriminator or D(z) to help to distinguish fake and real images.
- Neural Style Transfer or methods based on CNNs. Some functionalities of CNNs are for example the random erasing
- Meta leaning are methods used to optimize neural networks by using neural networks.

3.1.4. Further augmentation techniques.

- Kernel filters: These filters are used to sharpen and blur images.
- Mixing images: this technique consists of averaging both image pixels.
- Texture transfer methods [51] get the texture from an image and apply it on a new image.

Data augmentation is also done on DR classification task such as [20], where they did a rotation, horizontal and vertical flip and shifts to generate more images.

Chapter 4. Image Annotation.

Ground truth labelled images that include marks for the boundaries of relevant objects are needed by deep learning algorithms for object detection and recognition [52], especially for supervised learning. But still a lot of work has to been done to achieve large annotated datasheets.

There are different kinds of image annotation. The test-based annotation consists of manual annotation by humans, the content-based image retrieval (CBIR) refers to an automatic low level annotation by chape, texture or colour, and the automatic image annotation (AIA) where images are annotated with semantic labels using a learnt model using machine learning [53]. We will focus on the first type of manual annotation tools.

4.1. Annotation tools.

There are several tools or software for annotation. Some of them are described below.

Visual Object Tagging Tool (VoTT) is an open-source annotation and labelling tool for image and video, developed in TypeScript by Microsoft. Being available for Windows, Linux and OSX, it can be installed or also, run on the command window and used from a Web Browser. Once a new project is created, the user can open several images or videos and start labelling them using tags. It allows to save the annotations on different formats: Azure Custom Vision Service, JSON, csv, Pascal VOC, CNTK (Microsoft Cognitive Toolkit) [54].

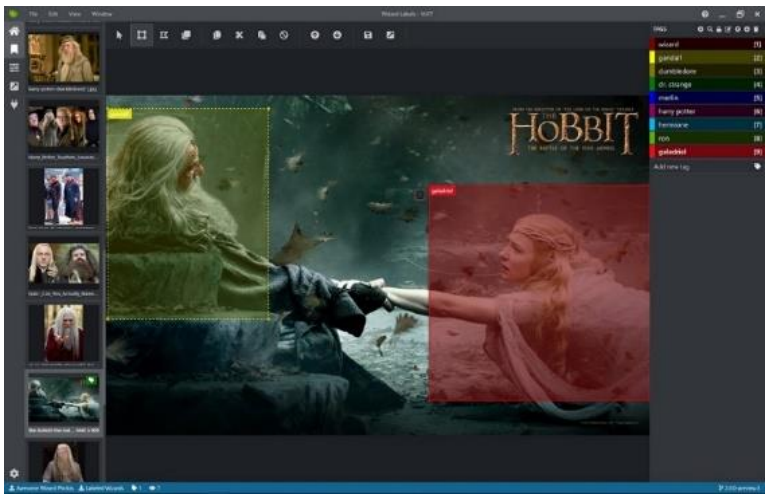


Figure 4. VoTT screen appearance [54].

LabelImg [55] is other image annotation tool which allow the user to draw rectangle labels of different categories and save it into different formats.

Figure 5. LabelImg screen appearance [55].

A further annotation tool, LabelMe, is a Javascript online tool which allows the user to draw diverse shapes of diverse objects. It is possible to share images and annotations, which are saved on XML format [52]. Rectlabel [56] is an image annotation tool to label images or videos to detect objects and segment the images by using boundary boxes. It allows the user to draw rectangles, points, or different polygons. Then it is possible to generate masks. A further automatic labelling is possible using Core models. Further annotation tools are Labelbox [57], Supervise.ly [58], Diffgram [59], Prodigy [60], DataTurks [61], ImageTagger [62], Fast Annotation Tool [63] or PolygonRNN+ [64].

4.2. Annotation tools for diabetic retinopathy.

There are also some tools which have been developed to annotate and find abnormalities on fundus image to detect DR, such as [65]. This tool allows the user to draw different shapes (Polygon, ellipse, point, free hand) and select among different eye abnormalities as labels. It uses a K nearest neighbour's classifier model to determine if a pixel is part of the region of interest (ROI) or not. As pre-processing step, they extract the green image channel to better see the eye abnormalities.

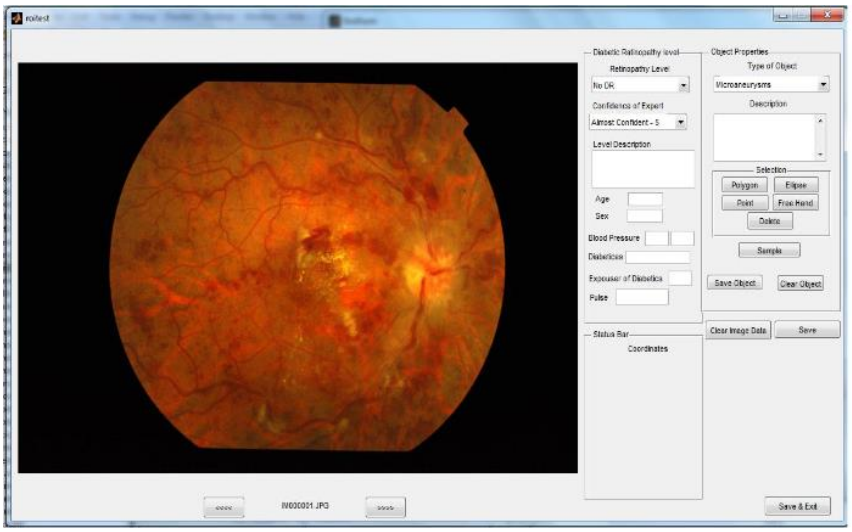


Figure 6. DR annotation tool screen appearance [65].

OPTAN web interface [66] is an annotation tool where ophthalmologist students can do manual annotations over a DR database which already contains annotations performed by specialists and shows a comparative similarity index between them. circles, ellipses, dots, polygons, and lines are allowed.

Figure 7. OPTAN DR annotation tool screen appearance [66].

A recent smartphone annotation tool to detect DR was also developed in [67] which allows mobility, working online and exporting images. They implement further options such as zoom, brightness and contrast changing or green level image.

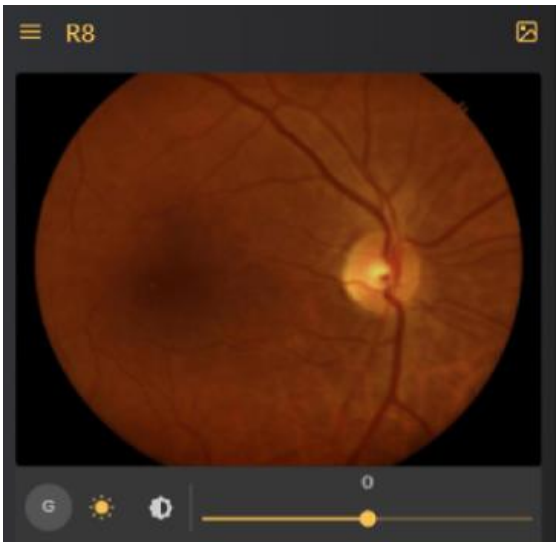


Figure 8. Smartphone DR annotation tool screen appearance [67].

SCREEN-DR [68] is a collaborative platform which allows textual and visual annotations, discard no DR images, grade DR severity, or evaluate the quality of the images.

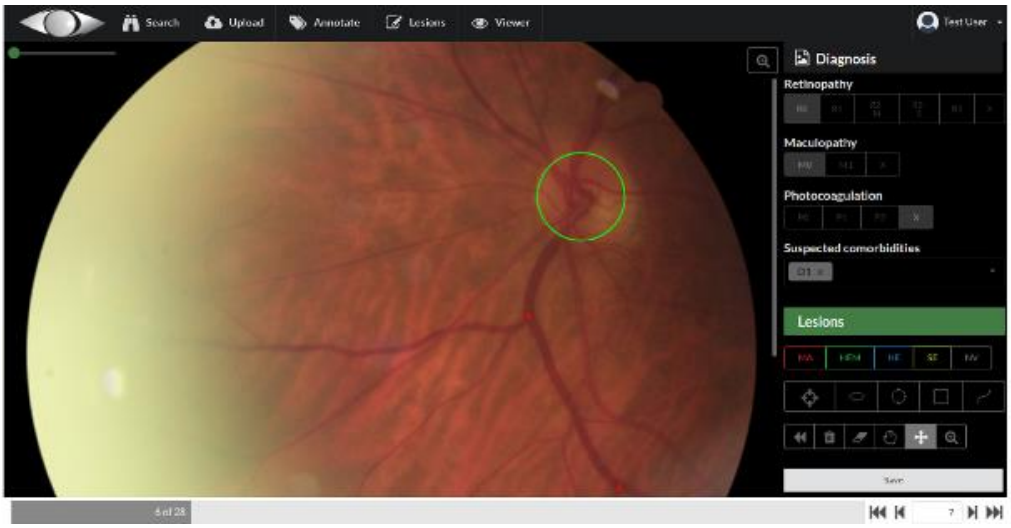


Figure 9. SCREEN- DR annotation tool screen appearance [68].

In [69] they developed a desktop tool for marking, showing also patient information such age and gender. Other authors developed a tool for grading the severity of DR [70] to help the specialists to grade retina images and reach an agreement.

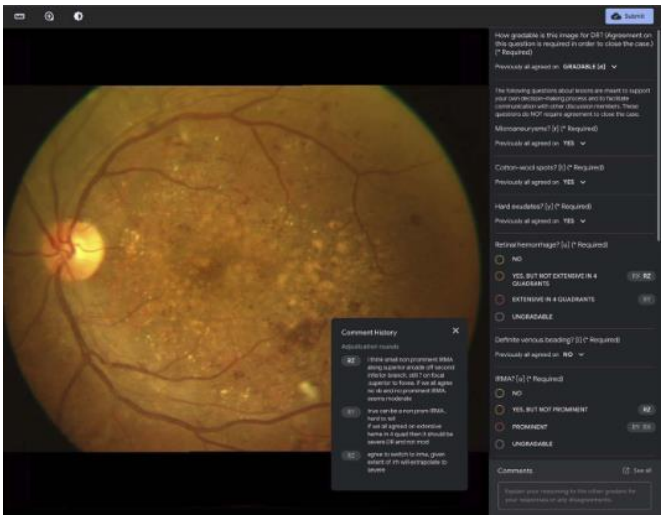


Figure 10. DR annotation tool screen appearance [70].

Another tool described in [71] is based on a 5-step reading system allows to read retina images and detect any kind of abnormalities. This software integrates algo an automated detection system.

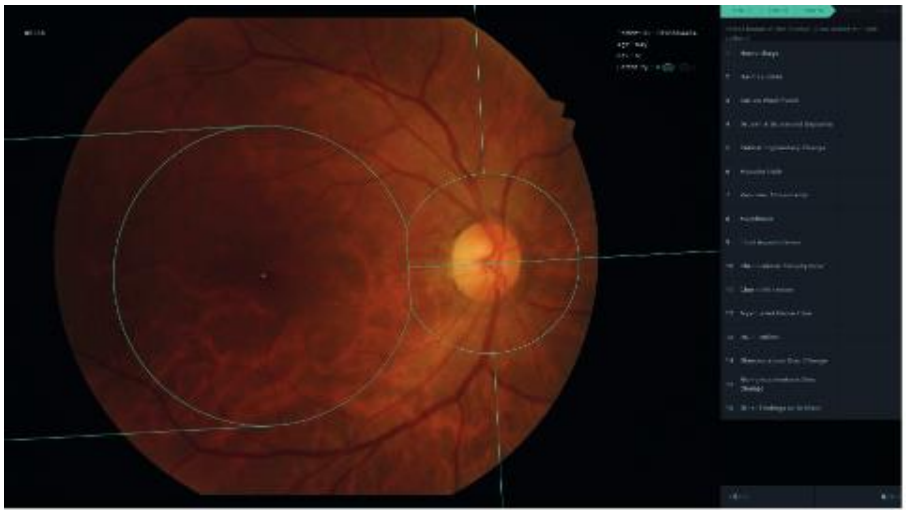


Figure 11. DR annotation tool screen appearance [71].

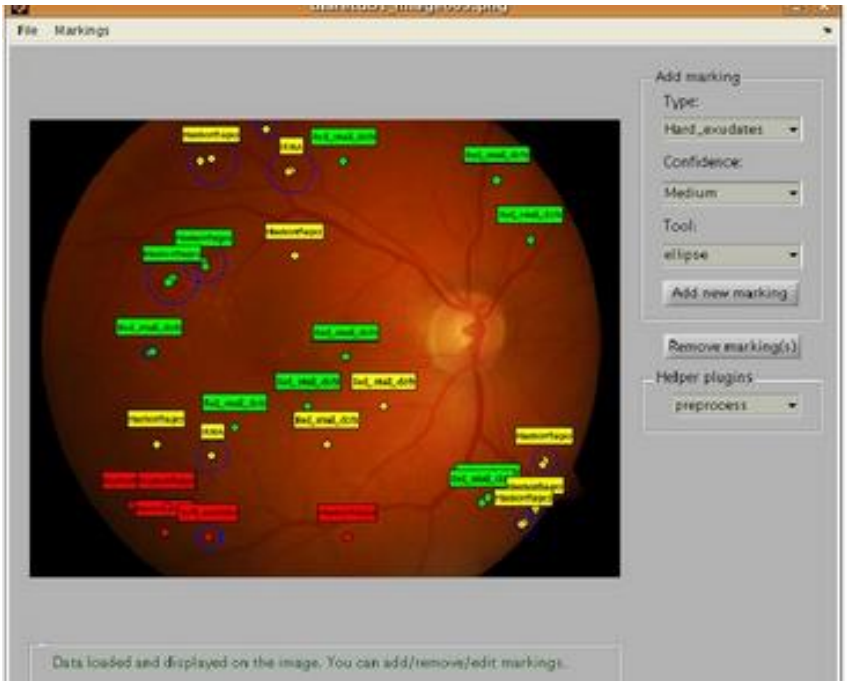


Figure 12. DR annotation tool screen appearance [72].

A comparation list among different annotation tools and their main characteristics is presented below.

Software name	author/ company	main features	Limitations	Possible shapes	Video	Diabetic retinopathy	Data augmentation	Exportation formats	Masks
VoTT [54]	Microsoft	Online	Only bounding boxes	Rectangle	Yes	No	No	Azur, JSON, csv, Pascal VOC, CNTK	Yes
LabelImg [55]		No project manage-ment		Rectangle	No	No	No	PASCAL VOC, YOLO, CreateML	No
LabelMe [52]	MIT Computer Science and Artificial Intelligence Laboratory	GUI customization (flags, predefined labels)		polygon, rectangle, circle, line and point	Yes	No	No	VOC, COCO	Yes
Labelbox annotation tool [57]	Labelbox	Pixel control, flexibility.	Limited free version	Several	Yes	No	No	JSON, CSV	Yes
Supervise.ly [58]		Workflow, project manage-ment	Web App	Rectangle, line, polygon, box	Yes	No	Yes	JSON, PNG, COCO	Yes
Rectlabel [56]		Bounding box		Rectangle, points, polygons	No	No	No	YOLO, CreateML, COCO JSON, CSV	Yes
VGG Image Annotator [73]	Visual Geometry Group, Oxford University	No project managemen t	Online/ offline as HTML.	Rectangle, Circle, polygon, point, line.	No	No	No	CSV, JSON	Yes
[65]	Alam, R., Poon, B., Mohammed, M. K et al.	Pre-processing: extract grey channel		Polygon, ellipse, point, free hand	No	Yes	No	XML	Yes
OPTAN [66]	Fitri, N. A., Pratama, A., Handayani, A. et al.	Comparatio n		circles, dots, polygons, and lines	No	Yes	No		No
[67]	Morya, A. K., Gowdar, J., Kaushal, A.. et al.	Smartphone annotation tool, DR grading			No	Yes	No	PNG	No

SCREEN-DR [68]	Pedrosa, M., Silva, J. M.. et al.	Textual and visual annotations, grade DR severity or quality of the images.	In production	Several	No	Yes	No	JSON, DICOM	No
[70]	Schaekermann, M., Hammel, et al.	Grading DR tool	No annotations possible	-	No	Yes	No	-	No
[71]	Sang Jun Park, Joo Young Shin et al.	multi-annotation per image.		Geometric regions	No	Yes	No	-	No
[72]	Tomi Kauppie et al.	Confidence selection. Compare opinions.		Circle, ellipse, polygon	No	Yes	No	XML	No

Table 1. Annotation tools comparative.

There already exist several annotation tools and specific annotation tools for Diabetic Retinopathy labelling. However, this Master Thesis aims to develop a new annotation software to specially achieve data anonymization. Most of the proposed tools are web based and, when using sensitive data from patients, it is needed to have a desktop tool and not sharing any information. Furthermore, it is difficult to find a software which combines annotation and augmentation tasks. The proposed tool includes both, which is interesting, due to the difficulty of generating new medical images to get a proper dataset to feed of the algorithm. The developed tool will also include several shapes for annotation, several exportation formats and mask generation. Finally, this desktop tool will enable to user to run a Deep Learning model with the images previously annotated. These characteristics cannot be found in the existing software's.

4.3. Annotation formats.

Annotations can be saved into different formats, but there are some ones which are commonly used on classification or segmentation deep learning tasks: Pascal VOC, COCO format and YOLO Format.

4.3.1. COCO Format.

This file format describes annotation properties and image metadata, which ae stored into a json file. Usually there is created only one json file with the information and annotations from the whole dataset. This file is split into the following parts:

-Info. It contains information about the datasheet.

-Licenses of the datasheet.

-Categories with their ID.

-List of images in the datasheet with some information.

-List of annotations containing their ID, boundary box and label information. The parameter "iscrowd" is used to indicate if there are several objects of the same category together.

4.3.2. Pascal VOC.

Pascal VOC stores annotation in an XML file for each annotated image with the name of each image. The data stored contains information about the path and folder where the image is located, the image name itself, image characteristics such as size and depth of image (RGB images, where the depth will be 3 or grayscale, depth is equal to 1). It also contains the following tags related to the annotation or objects: name, pose, truncated (if the bounding box of the object does not correspond to the full extent of the object), difficult means that the object is difficult to recognize and bounding box containing the object.

4.3.3. YOLO format.

This labelling format generates a .txt file for each image with the same name containing the annotations of each object in one line. It specifies the object class, origin coordinates, height, and width.

	Pascal VOC	COCO	YOLO
File	XML	JSON	txt
File number	One for each image in dataset	One for the whole dataset	One for each image in dataset
Bounding Box	(xmin-top left, ymin-top left,xmax-bottom right, ymax-bottom right)	(x-top left, y-top left, width, height)	<object-class> <x> <y> <width> <height>

Table 2. Annotations format comparative.

Chapter 5. The developed Annotation Tool.

An image annotation software or user interface has been implemented to help a specialist to annotate medical images, especially to examine fundus retina images and do a fast accurate DR diagnosis. On this section we will describe the main features of the software implemented. A complete software guide is attached on *Appendix I.*

Although this master thesis aims at developing an annotation tool for diagnosing and classifying diabetic retinopathy fundus image, but same tool could be also used for fundus image of other ocular diseases as well as other medical areas such as radiology.

After manually annotating images, Deep learning consists of training a neural network using a large dataset of images. First, the parameters of the neural network are set to random values and for each image they are updated to decrease the error, comparing the severity grade of the function and the known grade of the training set. Once the training part is done, the algorithm will be able to detect diabetic retinopathy on new images thanks to local features [6].

The main characteristic of the software is that it provides patient data confidentiality, as it should be run or installed on a computer. The tool background is dark Gray with minor tint changes to distinguish the different sections but not such as to catch attention, the same way as presented in [67].

5.1. Annotation Tool appearance.

The annotation main window is divided into six panels: Project Management Panel, Label Management Panel, Deep Leaning Model Panel, Image Augmentation Panel, Image Displayer and Save Annotations Panel.

On the left side, the Project Management Panel allows to create or load a project, open an image or an image directory and navigate through images. On the right side, the Label Management Panel allows to create or delete label categories and draw different shapes. Drawn shapes over the image are also displayed on this panel. On the window centre, images are displayed, indicating the mouse coordinates. The Image Augmentation Panel located on the centre downside of the window, enables the possibility to perform an augmentation process to generate more images. The Deep Leaning Model Panel on the left downside part, allows to train and test a DL model. The tool also allows to save the annotations into different formats, on the Save Annotations Panel, on the right down side.

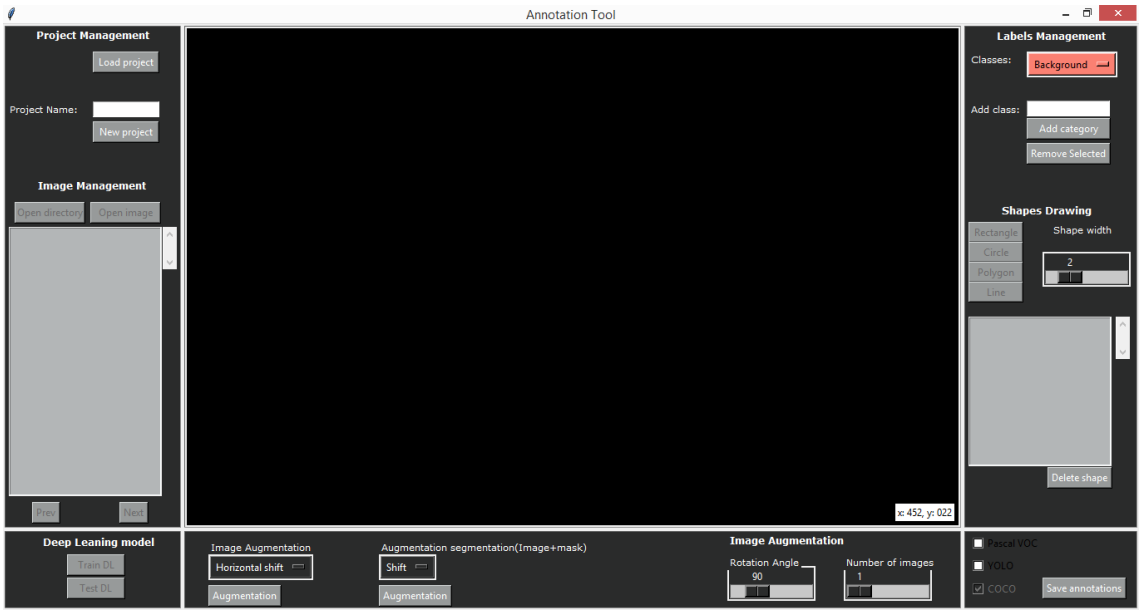


Figure 13. Annotation tool main *screen appearance.*

5.2.Software code.

The software is developed in Python version 3.5. and it uses Tkinter as graphic interface. The complete code is split into 6 scripts. The main script, AT.py contains the annotation tool design and the main features and functions code. Then, there is a script for the DL train model, train.py and another one for the test, test.py. The file model.py contains the DL model. Furthermore, the script data.py also used for the DL process, contains information about how images and mask are read. The last script, logger.py, is an interface which connects the Tool main window and the DL process. The augmentation process is based on Keras and Albumentations libraries.

5.2.1. DL model code.

The code is structured on this way, allowing an easy change of the DL architecture or model if desired. The DL architecture implemented for this Thesis is U-Net, but this algorithm can be easily changed, by just coding a new model on the script model.py. Other information to feed the model is included also on the train.py script, such as the project folder containing images and mask, number of classes and model hyperparameters. On the other hand, data.py script specifies and loads the train and test images. Test.py scrip just loads the saved model, writes the predicted masks and calculates some evaluation parameters. The DL model is based on Tensorflow library [74].

5.3.Annotation tool features.

- Desktop software.
- Project management. The software allows to create projects or to work on an already saved project. The user can specify the location and name of the project.
- Images or directories loading. The user can load single images or a whole directory containing images to work with.
- Diverse image formats. The software allows to load images with different formats (DICOM, PNG, JPG, TIFF) and sizes.
- Image visualization. When loading an image or image folder, it is printed on the main screen window. The user can navigate through images thanks to a Previous and Next button and an Image Browser scroll.
- Image marking. The user can annotate and mark lesions or objects in images using different shapes: rectangle, circle or ellipse, polygon and line. The label and its category will be drawn on the image.
- Label deleting. Once a label is drawn, the user can also delete it.
- Labels management. Some predefined labels related to DR are automatically loaded when the application runs: Background, Aneurysms, Retinal Haemorrhages, Exudates, Cotton wool spots, Vessels, IRMA, Vascular Abnormalities and Venous Beading. The user can easily delete one or all of them and create new category labels. When loading a saved project, the categories of that project are also loaded. Each category is represented by a random colour. The shapes of each category are also drawn on the colour category label.
- Annotation saving. Once the manual annotations have been performed, they can be saved into Pascal VOC, COCO and YOLO format. COCO saving format is mandatory because the system is especially prepared to read JSON files.
- Mask generation. RGB and greyscale mask will be automatically generated after the annotations.
- Image augmentation. Two augmentation processes are implemented, one for image augmentation and other for image and mask augmentation. For image augmentation the user can select among: Horizontal shift, Vertical shift, Horizontal Flip, Vertical Flip, Rotation, Brightness modification or zoom. For image and mask augmentation the user has different options: Shift, Brightness Modification, Horizontal Flip, Vertical Flip, Rotation, Blur or Crop. The system also allows to select the desired number of generated images in both cases and, in case of Rotation augmentation, also select the rotation degrees.
- Loading annotations project. When the user loads a saved project, images and their annotations will be loaded and printed on the main window display.

- Deep Learning model training and test. With a saved project with images and annotations, the software can train and test a DL algorithm. The predefined algorithm is U-Net, but it can be easily changed by other architecture. The system automatically split data into training (80%) and test (20%).

5.4.Testing the software.

5.4.1. Data annotation.

To test the annotation tool, we will simulate a complete annotation process. The first step is to annotate the images. This task is time consuming and complicated and will also require the assistance of an eye specialist. For that reason, only two images will be manually annotated using the tool, which are enough to prove the effectiveness of the software.

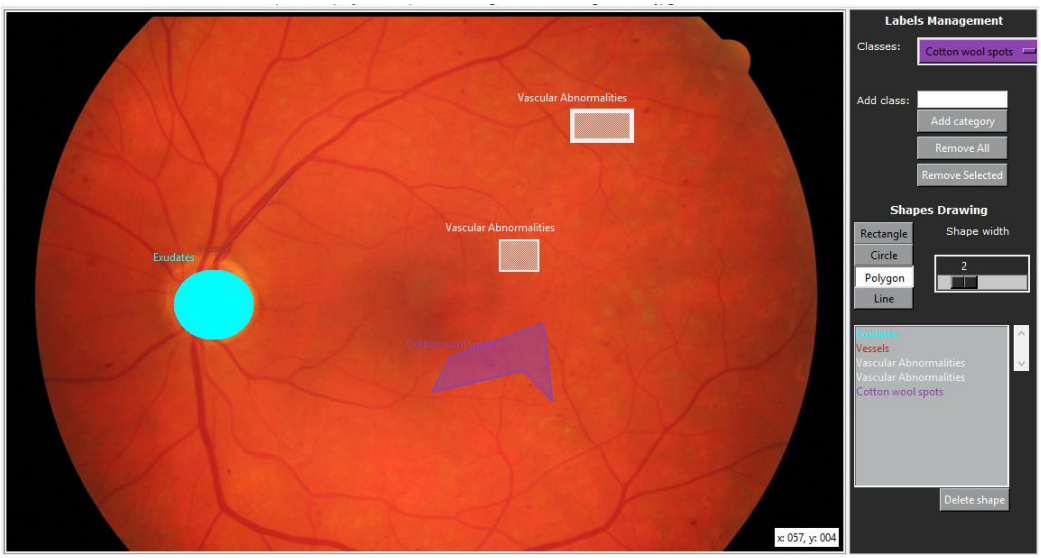


Figure 14. DR fundus image example with annotations (rectangle, ellipse, and polygon).

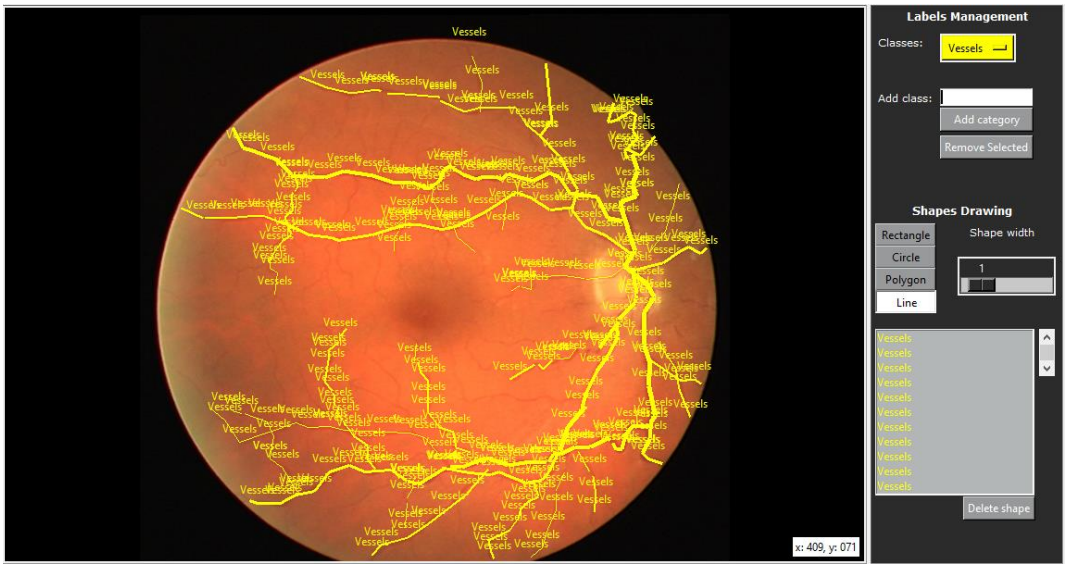


Figure 15. DR fundus image with annotations used to train the DL model.

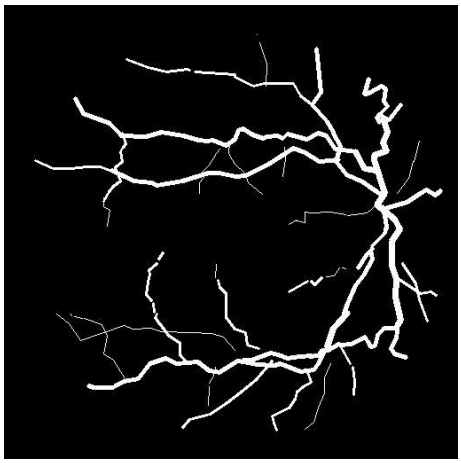
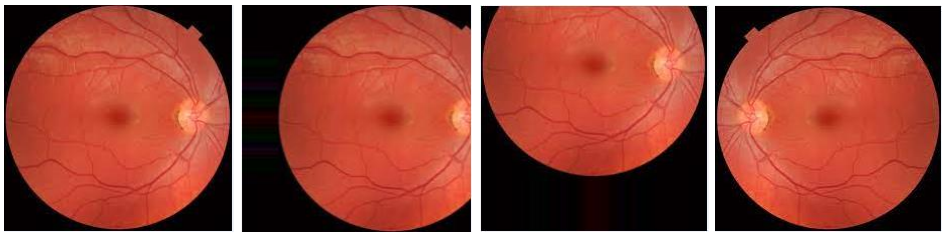


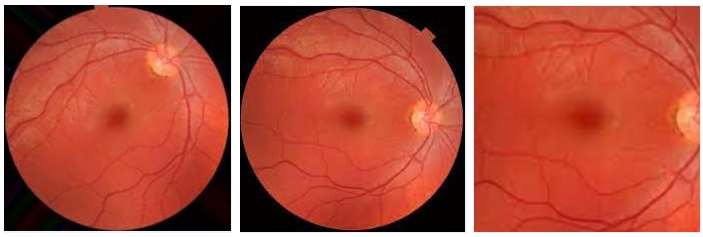
Figure 16. DR fundus mask generated to train the DL model.

5.4.2. Data augmentation.

To get a proper dataset, it is needed to perform a data augmentation process using the annotation software. A first single image performed with the annotation software is displayed below.



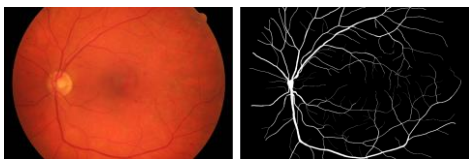
(a) (b) (c) (d)



(e) (f) (g)

Figure 17. DR fundus image augmented. (a) Original image, (b) Horizontal shift augmented image, (c) Vertical shift augmented image, (d) Vertical flip augmented image, (e) Rotation of 58º augmented image, (f) Brightness augmented image, (g) Zoom augmented image

A further augmentation process of image and mask is performed.



(a)

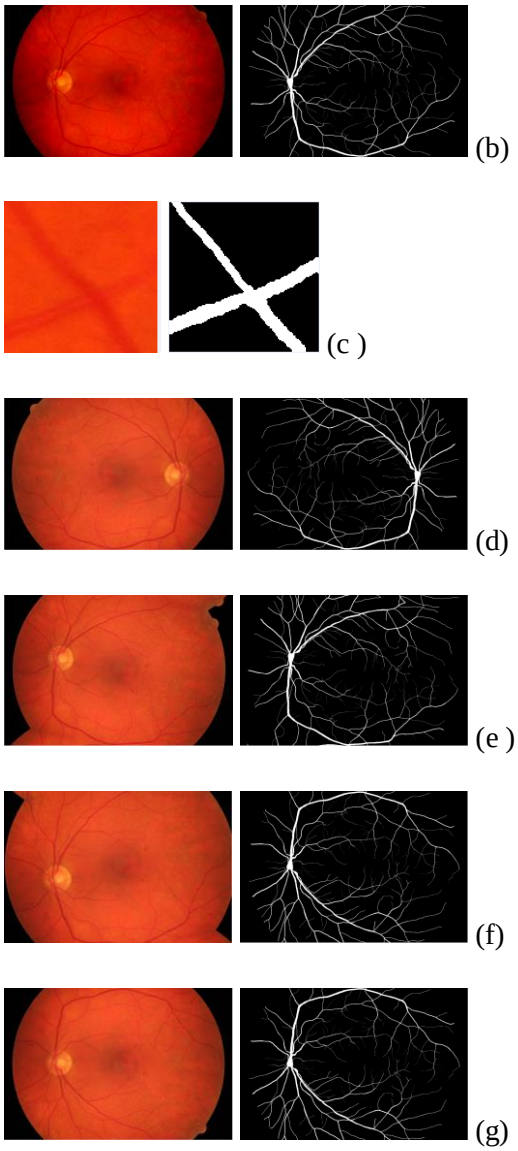


Figure 18. DR fundus image and mask augmented. (a) Original image and mask, (b) Brightness augmented image and mask, (c) Crop 128x128 augmented image and mask., (d) Horizontal Flip augmented image and mask, (e) Rotation augmented image and mask, (f) Shift augmented image and mask, (g) Vertical Flip augmented image and mask.

5.4.3. DL model training.

To test the Annotation Tool and train and test the DL model, two experiments will be performed. The first prove will be done with images already annotated by experts, from the DRIVE dataset [39]. The 40 images from this dataset and their corresponding masks will be divided into 20 images to the train part and 20 images to the test. On this case, train and test images will not be divided into 80% and 20%, but on 50% and 50% because the total number of available images is small. It is a binary problem with two classes, vessels and background, so the activation function used is sigmoid.

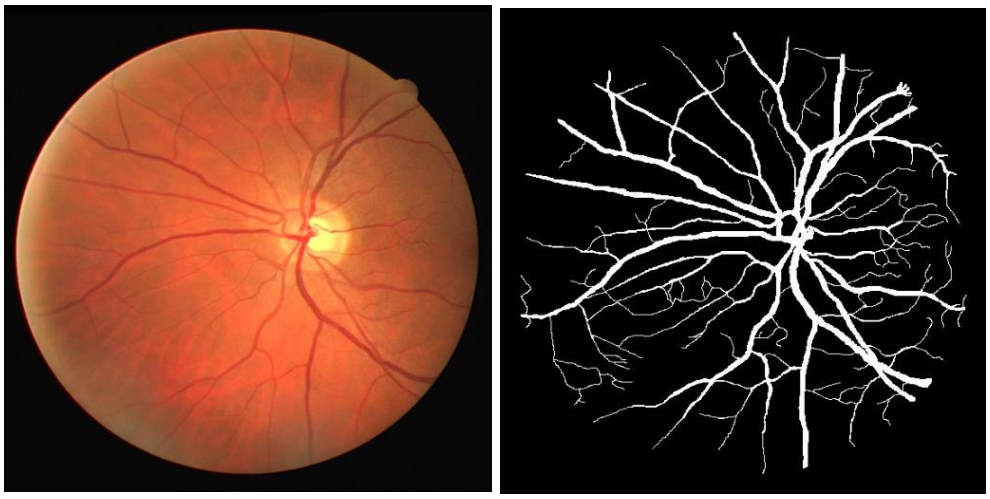


Figure 19. DR fundus image and its mask from the DRIVE dataset.

They will be used to train the U-Net architecture proposed on the Annotation Tool in 10 epochs, obtaining the following results.

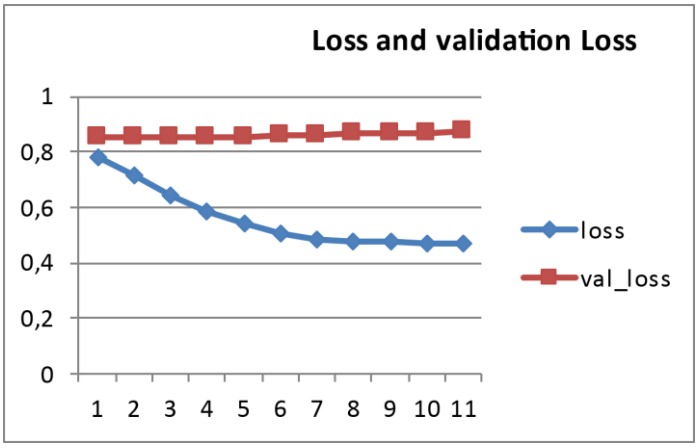


Figure 20. Loss and validation Loss with DRIVE dataset images.

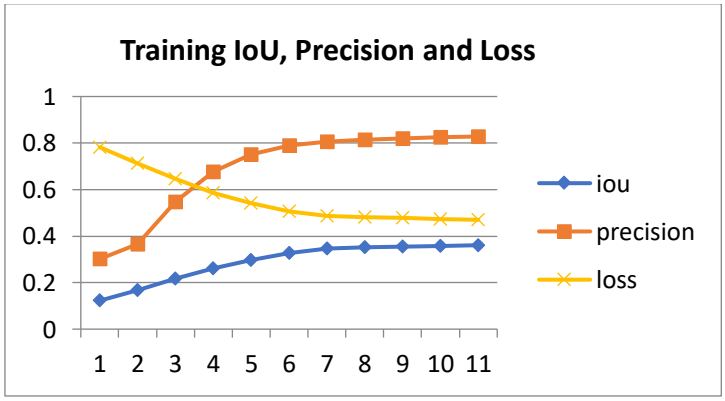
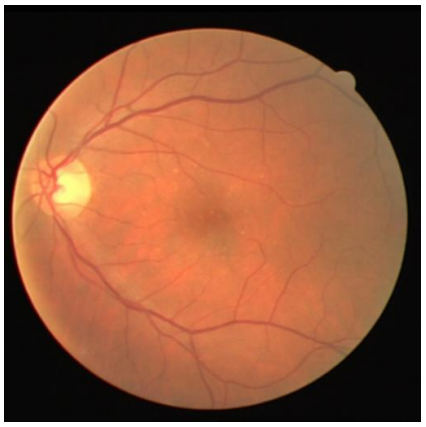
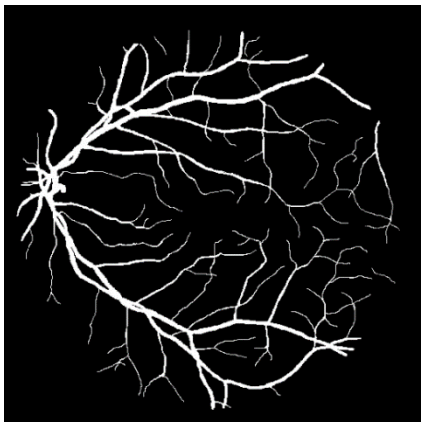


Figure 21. Training IoU, Precision and Loss with DRIVE dataset images.

Then, the test will be achieved with an Accuracy of 0.4371 and a Recall of 0.7268.

The second prove will be done will manually annotated images with the Annotation Tool. This process is time consuming and should be performed by and expert, so the expected results of the annotation and mask are with less quality than the ones from the DRIVE datasheet. Two fundus images taken also from the DRIVE datasheet are manually annotated. One of the images which was manual annotated is shown in *Figure 22*. It can be seen a proper annotation by an expert on the corresponding mask on *22b*, while a less quality annotation is performed by testing the Annotation Tool, on *22c*.


(a)


(b)

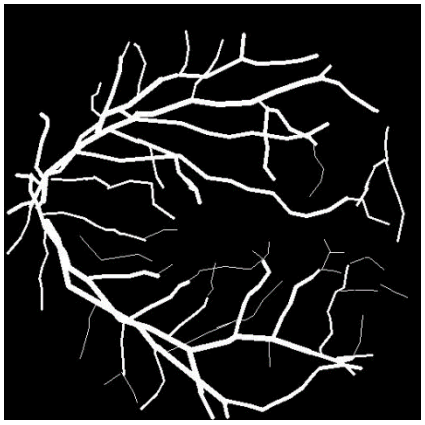

(c)

Figure 22. DR fundus image from the DRIVE dataset used on the second experiment. (a) Real fundus image. (b)Comparison with the Image mask from an expert annotation of the DRIVE dataset. (c) Manual annotated image with the Annotation Tool.

To get the same number of images as the previous experiment, some augmentation process is done. Finally, 20 images are generated for the train step and 20 for the test. Some graphics with training results are displayed below.

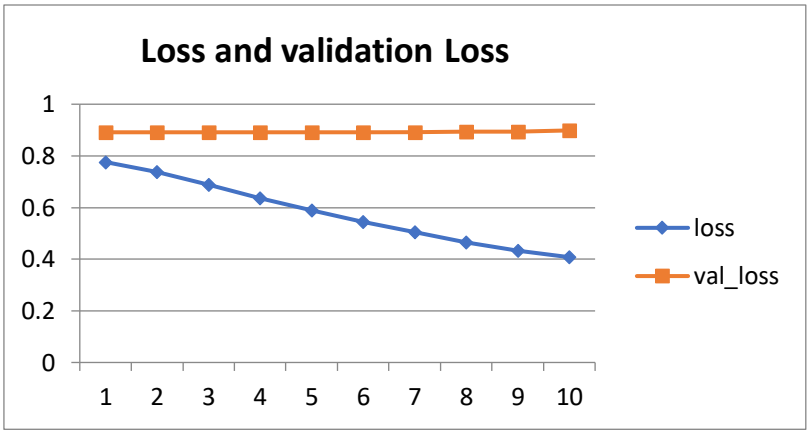


Figure 23. Loss and validation Loss with manual annotated images.

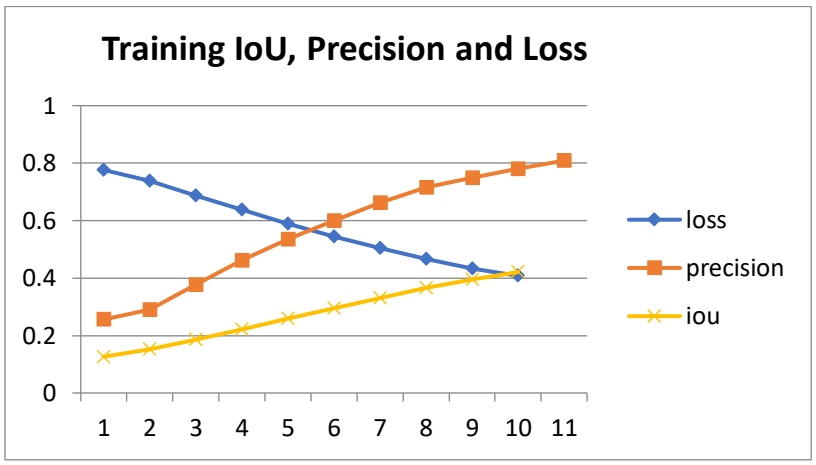


Figure 24. Training IoU, Precision and Loss with with manual annotated images.

Then, the test will be achieved with an Accuracy of 0.3288 and a Recall of 0.64342.

If both experiments, the one with the images annotated by experts and the second one with the manual annotated and augmented images, are compared, the first one gives better results on the training step and with the test images. Both cases are increasing Precision and IoU at each epoch, so both models better detect the true positives at each training iteration; and Loss decreases.

However, the difference of the measures is not big, considering that on the second experiment only two annotated images were taken as input and the rest were augmented images. For better results, it is important to count with a proper dataset of data. Furthermore, on the second case, images were manual labelled with possible mistakes, as they were not annotated by an eye specialist. On top of this, having a larger dataset or increasing the number of iterations or epoch, results could also improve. With this analysis, we could conclude that the Annotation Tool allows to annotate and properly mark images to then test a DL model as expected but need the assistance of an expert while doing the annotations.

Chapter 6. Conclusions and future work.

We proposed on this work an Annotation Tool to assist medical specialists to annotate medical images and generate a proper dataset of Ground Truth images being able to feed a Deep Learning algorithm.

The software allows the user to upload different types of images in different formats and sizes and to mark lesions or objects using different shapes, as required. Furthermore, an augmentation procedure is also implemented. On top of this, it is possible to train and test a Deep Leaning model directly from the Annotation Tool, which can be easily changed. The software has been also proved and validated while annotating some images and feeding the DL model and comparing the results with the same procedure but with annotated images from a database.

Despite the good results, there are some improvements that could be applied on the Annotation Tool, such as Zoom possibility to better visualize the medical images or a colour selection for each category, instead of being a random colour. An intelligent wizard could also be implemented to assist the user to select the most appropriate augmentation procedure.

References.

[1] LeCun, Y., Bengio, Y., and Hinton, G. *Deep learning nature.* 436-444 (2015).

[2] Romero-Aroca, P., Valls, A., Moreno, A., Sagarra-Alamo, R., Basora-Gallisa, J., Saleh, E. and Puig, D. *A clinical decision support system for diabetic retinopathy screening: creating a clinical support application.* Telemedicine and e-Health, 25(1), 31-40 (2019).

[3] Saleh, E., Valls, A., Moreno, A., Romero-Aroca, P., de la Riva-Fernandez, S., and Sagarra-Alamo, R. *Diabetic retinopathy risk estimation using fuzzy rules on electronic health record data.* In International Conference on Modeling Decisions for Artificial Intelligence. Springer: 263-274 (2016, September).

[4] Litjens and Geert. *A survey on deep learning in medical image analysis*. Medical image analysis, vol. 42: 60-88 (2017)

[5] Gulshan, V., Rajan, R. P., Widner, K., Wu, D., Wubbels, P., Rhodes, T. and Webster, D. R. *Performance of a deep-learning algorithm vs manual grading for detecting diabetic retinopathy in India*. JAMA ophthalmology, 137(9), 987-993 (2019).

[6] Gulshan V, Peng L and Coram M. *Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs*. JAMA (2016).

[7] Ronneberger, O., Fischer, P., and Brox, T. *U-net: Convolutional networks for biomedical image segmentation*. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham: 234-241 (2015).

[8] Shorten, Connor, and Taghi M. Khoshgoftaar, *A survey on image data augmentation for deep learning.* Journal of Big Data 6.1: 1-48 (2019)

[9] Hussain, Z., Gimenez, F., Yi, D. and Rubin, D. *Differential data augmentation techniques for medical imaging classification tasks.* AMIA annual symposium proceedings. American Medical Informatics Association: Vol. 2017: 979 (2017).

[10] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. and Arshad, H. *State-of-the-art in artificial neural network applications: A survey* (2018).

[11] Khan, A., Sohail, A., Zahoora, U. and Qureshi, A. S. *A survey of the recent architectures of deep convolutional neural networks.* Artificial Intelligence Review (2020).

[12] Pascanu, R., Mikolov, T., and Bengio, Y. *On the difficulty of training recurrent neural networks*. International conference on machine learning: 1310-1318 (2013).

[13] Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J. *Color image segmentation: advances and prospects*. Pattern recognition (2001).

[14] R. V. Paul Chan. *Diabetic Retinopathy - Europe*. Weill Cornell Medical College, New York, New York (2016).

[15] Acharya, R., Chua, C. K., Ng, E. Y. K., Yu, W. and Chee, C. *Application of higher order spectra for the identification of diabetes retinopathy stages.* Journal of medical systems: 481-488 (2008).

[16] Krause, J., Gulshan, V., Rahimy, E., Karth, P., Widner, K., Corrado, G. S. and Webster, D. R. *Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy*. Ophthalmology: 1264-1272 (2018).

[17] Wang, H., Hsu, W., Goh, K., and Lee, M. *An effective approach to detect lesions in colour retinal images*. Proc. IEEE Conf. Comp.Vis. Pattern Recognit. 2:181–187 (2000)

[18] American Academy of Ophthalmology. *International clinical diabetic retinopathy disease severity scale, detailed table.* Available at: http://www.icoph.org/dynamic/attachments/resources/diabetic-retinopathy-detail.pdf. (Accessed 14 July 2021)

[19] Wong, L. Y., Acharya, U. R., Venkatesh, Y. V., Chee, C., Lim, C. M., and Ng, E. Y. K., *Identification of different stages of diabetic retinopathy using retinal optical images* (2008).

[20] Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P. and Zheng, Y. *Convolutional neural networks for diabetic retinopathy*. Procedia computer science, 90, 200-205 (2016).

[21] Romero, P., Sagarra, R., Ferrer, J., Fernández-Ballart, J. and Baget, M. *The incorporation of family physicians in the assessment of diabetic retinopathy by non-mydriatic fundus camera*. Diabetes research and clinical practice: 184-188 (2010).

[22] Hayashi, Junichiro. *A development of computer-aided diagnosis system using fundus images.* Proceedings Seventh International Conference on Virtual Systems and Multimedia. IEEE (2001).

[23] Rasta, S. H., Partovi, M. E., Seyedarabi, H. and Javadzadeh, A. *A comparative study on preprocessing techniques in diabetic retinopathy retinal images: illumination correction and contrast enhancement*. Journal of Medical signals and sensors (2015).

[24] *Kaggle diabetic retinopathy detection competition.* Available from: https://www.kaggle.com/c/diabetic-retinopathy-detection. (Accessed 12 July 2021)

[25] Voets, M., Møllersen, K. and Bongo, L. A. *Reproduction study using public data of: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs* (2019).

[26] I.E. Grisan, A. Pesce, A. Giani, M. Foracchia, and A. Ruggeri, *A new tracking system for the robust extraction of retinal vessel structure*, in: 26th Annual International Conference of the IEEE EMBS San Francisco, USA: 1620–1623 (2004).

[27] Saleh, Marwan D., and Eswaran, C. *An automated decision-support system for non-proliferative diabetic retinopathy disease based on MAs and HAs detection.* Computer methods and programs in biomedicine 108.1: 186-196 (2012).

[28] Adarsh, P. and Jeyakumari, D. *Multiclass SVM-based automated diagnosis of diabetic retinopathy*. International Conference on Communication and Signal Processing: 206-210 (2013).

[29] Xiaohui, Zhang, and A. Chutatape. *Detection and classification of bright lesions in color fundus images*. International Conference on Image Processing, Vol. 1. (2004).

[30] N. Kandiraju, S. Dua and H.W. Thompson, *Design and implementation of a unique blood vessel detection algorithm towards early diagnosis of diabetic retinopathy*. Proceedings of the International Conference on Information Technology: Coding and Computing, IEEE Computer society: 26–31 (2005).

[31] Vallabha, D., Dorairaj, R., Namuduri, K. and Thompson, H. *Automated detection and classification of vascular abnormalities in diabetic retinopathy*. Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers: 1625-1629 (2004).

[32] *RetinaLyze System*; Retinalyze A/S, Hørsholm, Denmark. Available from: https://www.retinalyze.com/ (Accessed 22 July 2021)

[33] Cuadros, J. and Bresnick, G. *EyePACS: an adaptable telemedicine system for diabetic retinopathy screening*. Journal of diabetes science and technology: 509–516 (2009).

[34] Available from: https://www.adcis.net/en/third-party/messidor2/ (Accessed 22 July 2021)

[35] Available from: https://www.kaggle.com/ (Accessed 23 July 2021)

[36] Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Uusitalo, H., Kälviäinen, H. and Pietilä, J., *DIARETDB0: Evaluation Database and Methodology for Diabetic Retinopathy Algorithms*, Technical report.

[37] DIARETDB1 Available from: http://www2.it.lut.fi/project/imageret/diaretdb1/ (Accessed 15 July 2021)

[38] Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Raninen A., Voutilainen R., Uusitalo, H., Kälviäinen, H. and Pietilä, J. *DIARETDB1 diabetic retinopathy database and evaluation protocol*, Technical report.

[39] *Digital Retinal Image for Vessel Extraction (DRIVE).* Available from: http://www.isi.uu.nl/Research/Databases/DRIVE/ (Accessed 14 July 2021)

[40] *VARPA Retina Images Authentication (VARIA) database.* Available from: http://www.varpa.es/varia.html (Accessed 15 July 2021)

[41] *VICAVR Database*, Available from: http://www.varpa.es/vicavr.html (Accessed 15 July 2021)

[42] *APTOS. Blindness detection.* Available from: https://www.kaggle.com/c/aptos2019-blindness-detection/. (Accessed 14 July 2021)

[43] *Peking university international competition on ocular disease intelligent recognition.* Available from: https://odir2019.grand-challenge.org/dataset/. (Accessed 15 July 2021)

[44] Sivaswamy J, Krishnadas SR, Chakravarty A, Joshi GD and Ujjwal SA. *Comprehensive retinal image dataset for the assessment of glaucoma from the optic nerve head analysis.* JSM Biomed Imaging Data Papers (2015).

[45] *Structured analysis of the retina (STARE).* Available from: http://www.clemson.edu/ces/ (Accessed 22 July 2021)

[46] *Collection of multispectral images of the fundus (CMIF)*. Available from: http://www.cs.bham.ac.uk/research/projects/fundus-multispectral/ (Accessed 22 July 2021)

[47] *Retinopathy online challenge (ROC).* Available from: http://roc.healthcare.uiowa.edu/ (Accessed 22 July 2021)

[48] *Review: retinal vessel image set for estimation of widths (REVIEW).* Available from: http://reviewdb.lincoln.ac.uk/ (Accessed 22 July 2021)

[49] Lasko TA, Bhagwat JG, Zou KH and Ohno-Machado L. *The use of receiver operating characteristic curves in biomedical informatics*. Journal of Biomedical Informatics: 404–415 (2005).

[50] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. and Savarese, S. *Generalized intersection over union: A metric and a loss for bounding box regression*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 658-666 (2019).

[51] Mikołajczyk, Agnieszka, and Michał Grochowski. *Data augmentation for improving deep learning in image classification problem*. 2018 international interdisciplinary PhD workshop (IIPhDW). IEEE (2018).

[52] Russell, B. C., Torralba, A., Murphy, K. P. and Freeman, W. T. *LabelMe: a database and web-based tool for image annotation*. International journal of computer vision: 157-173. (2008).

[53] Zhang, D., Islam, M. M., and Lu, G. *A review on automatic image annotation techniques. Pattern Recognition*, 45(1): 346-362 (2012).

[54] *Visual Object Tagging Tool: An electron app for building end to end Object Detection Models from Images and Videos*. Available from: https://github.com/microsoft/VoTT (Accessed 25 July 2021)

[55] *LabelImg.* Available from: https://github.com/tzutalin/labelImg (Accessed 25 July 2021)

[56] *Rectlabel* Available from: https://rectlabel.com/ (Accessed 25 July 2021)

[57] Labelbox. Available from: https://labelbox.com/ (Accessed 25 July 2021)

[58] *Supervise.ly*. Available from: https://supervise.ly/ (Accessed 25 July 2021)

[59] *Diffgram.* Available from: https://diffgram.com/ (Accessed 25 July 2021)

[60] *Prodigy*. Available from: https://prodi.gy (Accessed 25 July 2021)

[61] *DataTurks.* Available from: https://dataturks.com (Accessed 25 July 2021)

[62] *Image Tagger*. Available from: https://github.com/bit-bots/imagetagger (Accessed 25 July 2021)

[63] Fast Annotation Tool Available from: https://github.com/christopher5106/FastAnnotationTool (Accessed 25 July 2021)

[64] PolygonRNN+. Available from: http://www.cs.toronto.edu/~amlan/demo/ (Accessed 25 July 2021)

[65] Alam, R., Poon, B., Mohammed, M. K., Amin, M. A. and Yan, H. *Annotating Retina Fundus Images for Teaching and Learning Diabetic Retinopathy Software*. INFORMATION TECHNOLOGY IN INDUSTRY (2014).

[66] Fitri, N. A., Pratama, A., Handayani, A. and Rajab, T. L. E. *Web-based Training Application to Support Manual Annotation of Diabetic Retinopathy Features on Retinal Fundus Images*. 2019 International Biomedical Instrumentation and Technology Conference (IBITeC) Vol. 1: 98-102 (2019).

[67] Morya, A. K., Gowdar, J., Kaushal, A., Makwana, N., Biswas, S., Raj, P. and Raghav, P. *Evaluating the Viability of a Smartphone-Based Annotation Tool for Faster and Accurate Image Labelling for Artificial Intelligence in Diabetic Retinopathy*. Clinical Ophthalmology (2021).

[68] Pedrosa, M., Silva, J. M., Silva, J. F., Matos, S. and Costa, C. *SCREEN-DR*: *Collaborative platform for diabetic retinopathy*. International journal of medical informatics, 120: 137-146 (2018).

[69] Park SJ, Shin JY and Kim S. *Image reading tool for efficient generation of a multi-dimensional categorical image database for machine learning algorithm training*. J Korean Med Sci. (2018).

[70] Schaekermann, M., Hammel, N., Terry, M., Ali, T. K., Liu, Y., Basham, B. and Sayres, R. *Remote tool-based adjudication for grading diabetic retinopathy. Translational vision science & technology* (2019).

[71] Park, S. J., Shin, J. Y., Kim, S., Son, J., Jung, K. H. and Park, K. H. *A novel fundus image reading tool for efficient generation of a multi-dimensional categorical image database for machine learning algorithm training*. Journal of Korean medical science (2018).

[72] Kauppi T, Kämäräinen JK, Lensu L, Kalesnykiene V, Sorri I, Uusitalo H and Kälviäinen H. *Constructing benchmark databases and protocols for medical image analysis: diabetic retinopathy*. Comput Math Methods Med. (2013).

[73] *VGG Image Annotator*. Available from: https://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html (Accessed 25 July 2021)

[74] *Multiclass-Segmentation in U-Net.* Available from: https://github.com/nikhilroxtomar/Multiclass-Segmentation-in-Unet (Accessed 25 July 2021)

Appendix I. Annotation Tool User Guide.

Once the application is launched, the annotation tool main window will be printed. This window is divided into six panels: Project Management Panel, Label Management Panel, Deep Leaning Model Panel, Image Augmentation Panel, Image Displayer and Save Annotations Panel.

Figure 1. Annotation tool main *screen appearance*.

1. Project and image management.

The panel of the left side or "Project Management Panel" allows to navigate over the project and images. It allows the user to perform different actions. When the tool is started, there are only two possible options, load a saved project or create a new project. The rest of buttons are blocked.

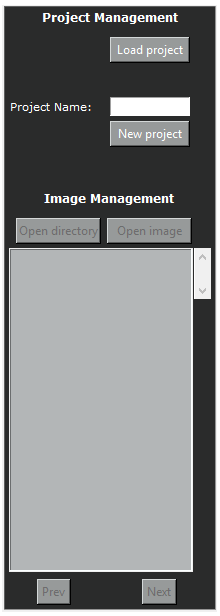


Figure 2. Project Management panel.

- Create a new project. There is a text entry where the user can write a desired name for the new project. Then, there is a "New Project" button which, when pressed, will open a folder searching window to select the path to create the project. After this, there will be created an empty folder with this name.

- Load a saved project: there exists also the possibility to open an already saved project with a button called "Load project". The only requirement to load a project is that the user should have already annotated and saved any image. Otherwise, the project folder will be empty.

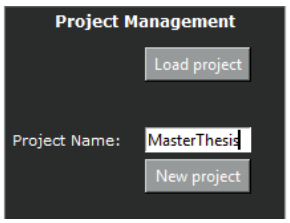


Figure 3. Load or create a new project.

- Open an image or a directory. Once the project is created or opened, the user can start annotating images. For this purpose, there are two more buttons, one to open a single image and other to open a directory which contains several images. When loading images, they will be displayed on a Listbox. When there are several images loaded, a scrollbar helps the user to scroll up and down.

  Images will be automatically saved in our project folder once they have been annotated.

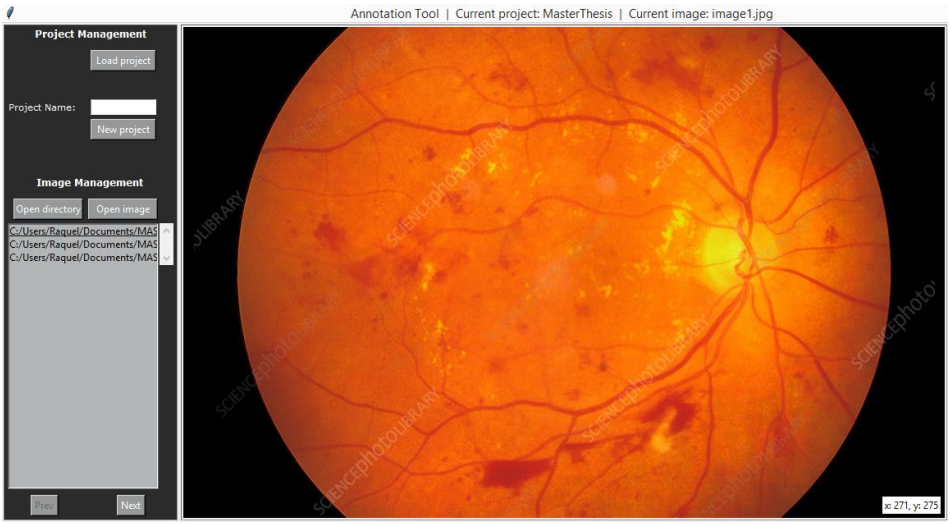


Figure 4. Creation of a new project, loading several images and displaying one of them.

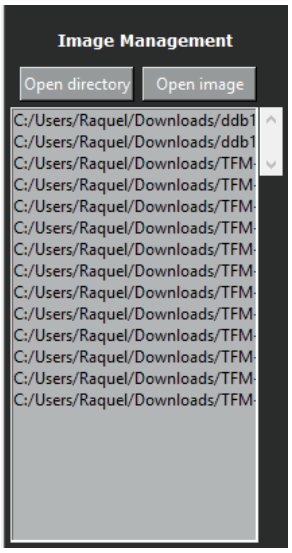


Figure 4. Image browser.

- Images navigation. The user can easily navigate through images. There is a "Next Image" and a "Previous Image" button and there also exist the possibility to select an image on the Listbox by clicking it. When navigating in images, they will be displayed on the Image Displayer main window.

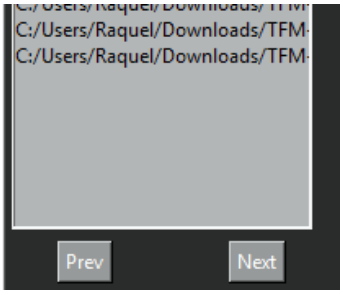


Figure 5. Images navigator.

- Current project information. On the top title, there is information to guide the user over the current project and current image where he/she is working at that moment.
- Image displayer. The selected image is displayed on the main window, being adapted to the window size. Mouse coordinates are displayed also on this window on the right downside.

2. Labels Management Panel.

This panel on the right side of the window allows the user to manage the labels or classes. There are some predefined labels which will be load when launching the application.

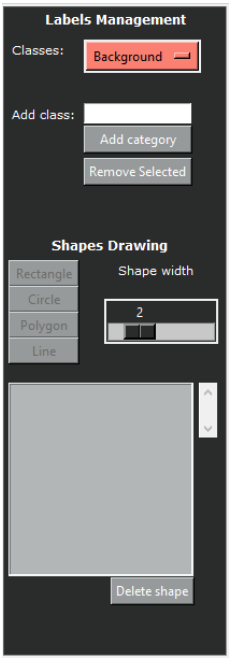


Figure 6. Labels management and drawing panel.

- Add a new category label. The user should write the new desired category name and then click on the "Add category" button. This new category will be added to the list and assigned a new colour.

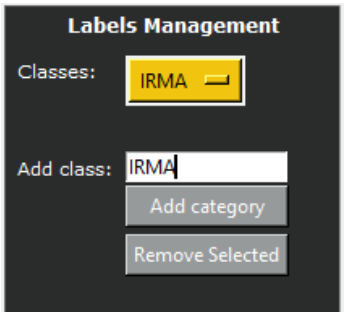


Figure 7. Categories management.

- Delete a selected class. To delete one class label, the user must first select it on the list and then click on the button "Remove Selected". This category will be deleted from the list. If there are some shapes of this category drawn on one or several images from the project, the user will be alerted.

For example, if the user wants to delete the category Vessels, which have been drawn, this message will be shown:

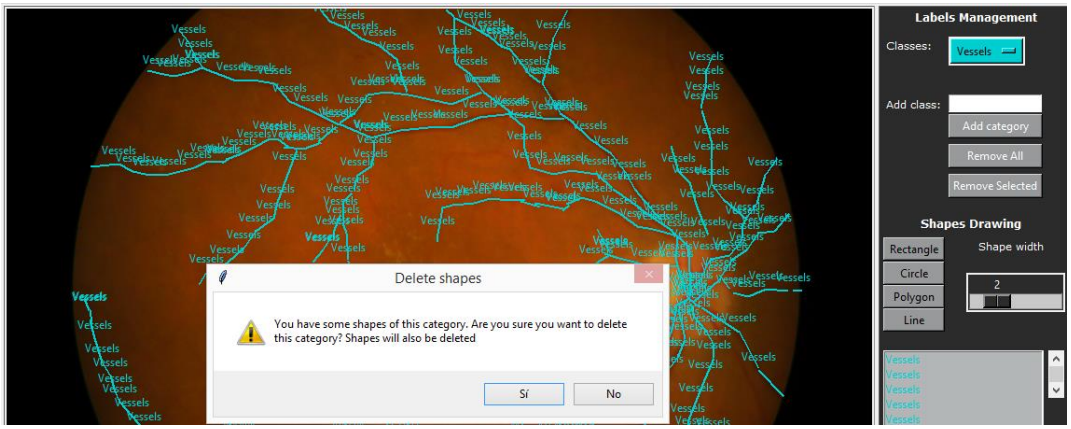


Figure 8. Deleting a category.

If the user clicks "Yes", the category "Vessels", on this case, will be deleted and also all the annotations of this category.

Figure 9. Image without shapes after deleting a category.

3. Drawing shapes.

Once a project is created or loaded and some images are opened and displayed, the user can start drawing shapes. There are some drawing shapes possibilities: rectangle, circle, polygon, and line.

- Drawing rectangle or circle: the user should first select the button "rectangle" or "circle" and then perform a click on the left mouse button on the desired location. Without releasing the mouse, expand the cursor until the end location. The shape will be created. On this moment, the polygon will be filled with the corresponding class colour.
- Drawing a polygon: first select the button named "polygon". The user then should click the left mouse button to draw the first polygon point. After that, the user should release the mouse and move the cursor until the desired location for the next point or polygon vertex. This action can be repeated as vertex desired. While drawing, the polygon lines have a white colour. When the user wants to close the polygon shape, it is needed to close

the mouse cursor near the first vertex. On this moment, the polygon will be filled with the corresponding class colour. If the user wants to delete the started polygon, he can do it by crossing some lines.

- Drawing a line: The user should click on the left mouse button and, without releasing it, move the mouse cursor until the desired location of the line.

The user can also select the shape width while drawing by moving the shape width scroll.

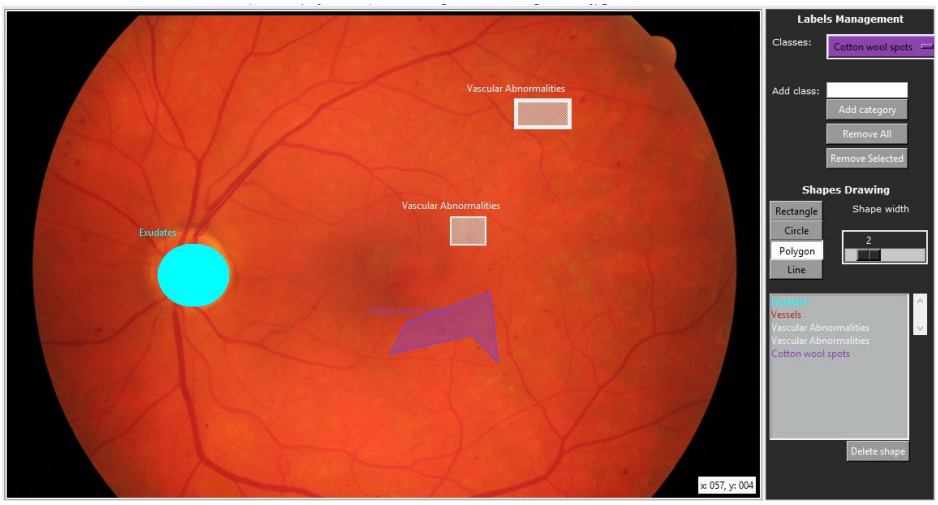


Figure 10. Image and shapes drawn.

- Deleting shapes. Once shapes are drawing, the user can see them on the listox. A desired shape can be deleted by selecting it on the listbox and clicking on the "Delete shape" button.

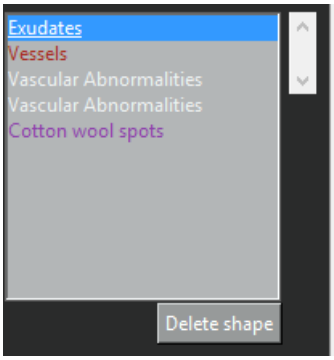


Figure 11. List of drawn shapes.

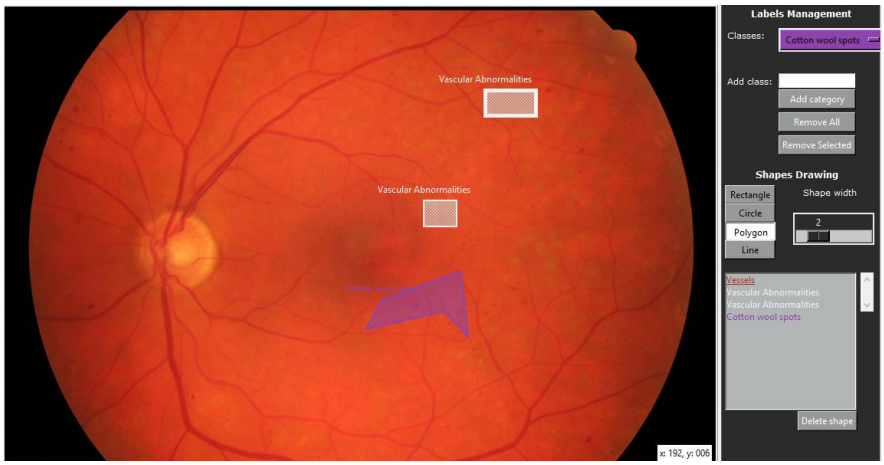


Figure 12. Previous annotated image after deleting circular shape of category "Exudates".

4. Data saving and folder organization.

The user can save the annotations of the current project when desired by clicking on the right down button "Save annotations".

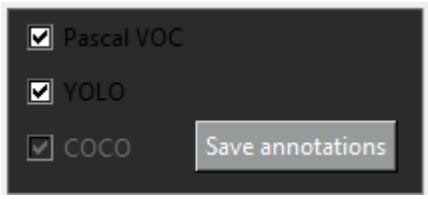


Figure 13. Annotation saving formats.

By default, all annotations will be saved into COCO Format, and it corresponding JSON file will be always generated. The user can also save the annotations into Pascal VOC and YOLO format if checking on their Checkbuttons.

When a project is created and saved, the folders structure is the following:

```
MasterThesis                                    #Name of the project
|
|--annotations/                                 #Annotations folder
|--------dl_masks/                              #Greyscale masks
|--------real_masks/                            #RGB masks
|--------labels.json                            #Annotations JSON file
|--------trainval.txt    #Images for training the DL model
|--------test.txt        #Images for testing the DL model
|--images/               #Images folder
```

|--results #Results from the DL test
|

It contains a global folder with the name of the folder which is composed by 3 sub folders: annotations, images, and results. The folder Annotations contains 2 folders, one for saving the generated RGB masks and the other the greyscale masks which will be used for the DL algorithm, after the annotations. This Annotations folder also contains the JSON file with the annotations of all images. If the user had also selected YOLO and Pascal VOC as annotation formats, one .txt or .XML will be also generated and saved on the annotations folder, for each annotated image, with the same name as the image.

Not only annotations will be saved, but also a copy of all the annotated images will be created on the images folder. The results folder is still empty but will later contain the results if a DL test process is performed.

When annotations are saved, images annotated, also classes information and shapes drawn will be saved into the JSON file and can be recovered and visualized when loading the project.

Mask will be also generated, a RGB mask and a greyscale mask on their corresponding folders, both on .png format. Mask will assume the same name as the input image.

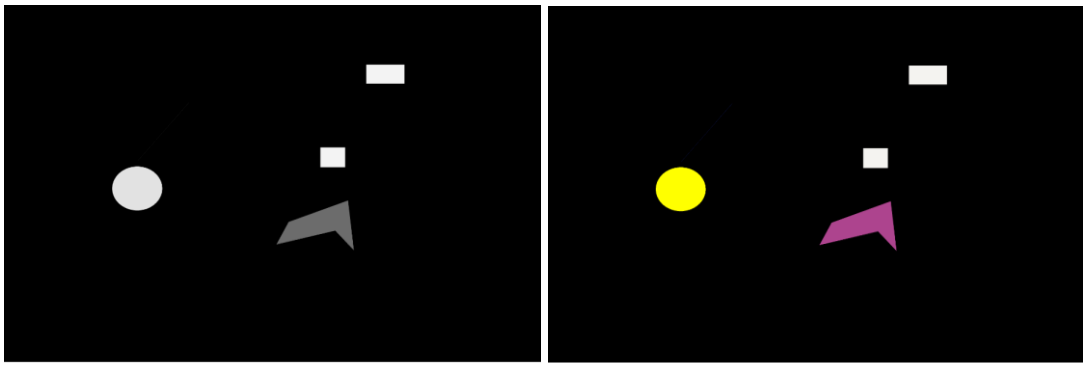


Figure 14. Grayscale and RGB mask of an annotated image.

Inside the JSON file there are several information. A JSON extract file of the example showed on this chapter is shown. It contains information related to the dataset:


```
{
  "info": {
    "year": "",
    "version": "",
    "description": "",
    "contributor": "",
    "url": "",
    "date_created": ""
  },
  "licenses": {
    "url": [],
    "id": "",
    "name": ""
  },
```

Figure 15. JSON File. Database information.

It also contains information about the label categories:


```
"categories": [
  {
    "id": "1630257725.0983214",
    "name": "Background",
    "color": "#FA8072",
    "supercategory": ""
  },
  {
    "id": "1630257725.0983214",
    "name": "Aneurysms",
    "color": "#FF0000",
    "supercategory": ""
```


Figure 16. JSON File. Categories information.

For each of the annotated images:


```
"images": [
  {
    "id": "01_dr_blur_augmented0.jpg+1630257738.9357858",
    "license": "",
    "file_name": "C:/Users/Raquel/Downloads/TFM--llanes1(1)/TFM--llanes1
      /Thesis_DL/images/01_dr_blur_augmented0.jpg",
    "height": 3504,
    "width": 2336,
    "date_captured": ""
  }
```


Figure 17. JSON File. Images information.

For each of the shapes drawn:


```
"id": "1630257769.8282444",
"image_id": "01_dr_blur_augmented0.jpg+1630257738.9357858",
"shape_type": "Rectangle",
"category": "Vascular Abnormalities",
"category_id": "1630257725.0983214",
"bbox": [
  549,
  255,
  43,
  34
],
"segmentation": [
  [
    549,
    255
  ],
  [
    592,
    255
  ],
  [
    592,
    289
  ],
  [
    549,
```


Figure 18. JSON File. Annotations information.

5. Image augmentation panel.

On the bottom annotation tool there is a panel to perform image augmentation. Two processes can be performed: Single image augmentation (on the left) and augmentation of image and its

mask (on the right). Both works on the same way; the user should select one augmentation option on the listbox and then click on the "Augmentation" button. Then a new window will be opened to ask the user for the selected image for data augmentation, or, in the case of augmentation for image and mask, the user should also select the mask for augmentation. This mask should correspond to the same image. The augmentation files will be saved on the same location as the input image. User can also select the number of images generated and, in case a rotation process is selected, also the rotation angle. These options are on the right side of the panel.

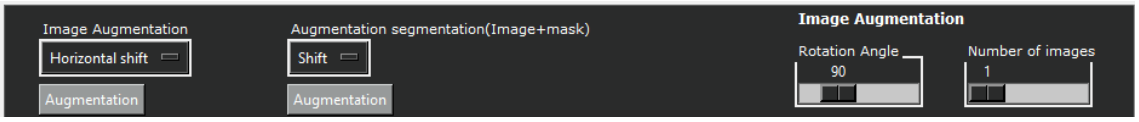


Figure 19. Augmentation panel.

6. DL Model.

The annotation tool also gives the possibility to train and test a DL model with the annotated images. The DL model will be fed with the images located on the project folder, so it is important to not make changes or delete any item. First, when selecting on the "Train DL" button, two .txt files will be generated on the annotations folder: trainval.txt and test.txt, containing the image names for the training and the testing. These images are all the annotated images from the project, split by 80% to the training and 20% to the test. These buttons are only available when there is an open project with annotations.

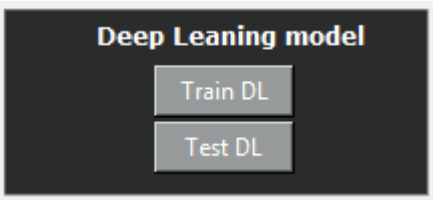


Figure 20. DL panel.

Annotated images should be enough to be able to train/test the model. If not, an alert will be displayed, and the model will not be trained/tested.

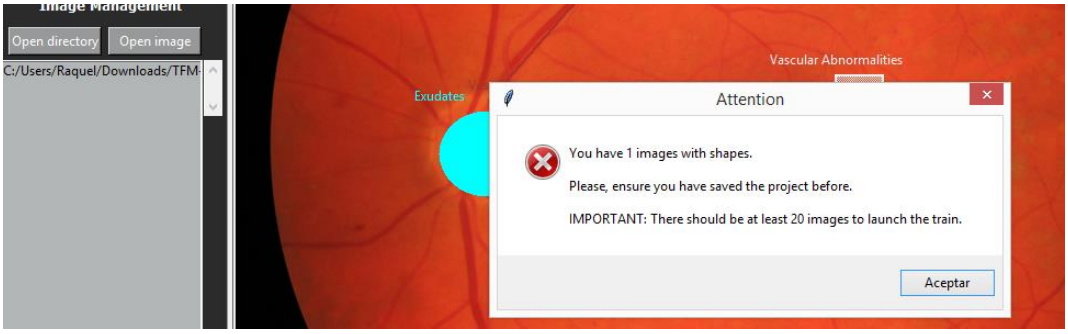


Figure 21. DL alert when there are not enough images.

When clicking on the Train/Test button, a new console will be opened.

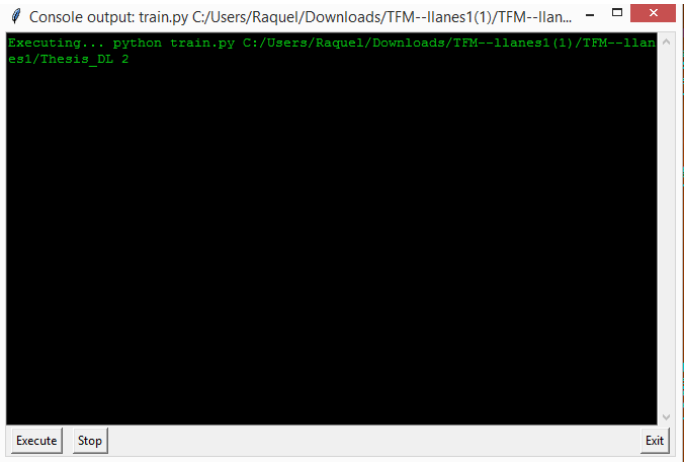


Figure 22. DL console.

The process will start and maybe take some time to finish.

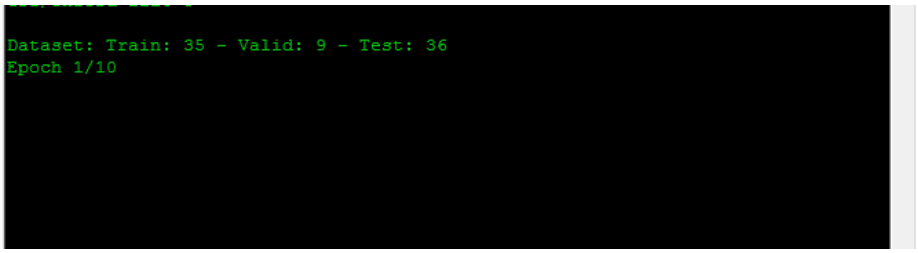


Figure 23. DL console executing the model.

To start the process, the user should click on the “Execute” button.

To stop the process, click on “Stop” button and to exit the process on the “Exit” button.