

# PhD. Work Plan

Date: 2015-12-17

Author: Jordi de la Torre

## 1. Objective

Design a system capable of classify retine images according to the standard classification of diabetic retinopathy level.

## 2. Dataset

The dataset consist of a set of high resolution retine images taken under a variety of imaging conditions.

Every image has been rated by a clinician for the presence of retinopathy according to the following scale:

- 0 - No diabetic retinopathy
- 1 - Mild
- 2 - Moderate
- 3 - Severe
- 4 - Proliferative

The dataset consist of 88704 labeled high resolution retine images. The training set is made of 35127 images and the test set of 53577.

## 3. Proposed work pipeline

### 3.1 Data preprocessing

The original images are taken in very different conditions, with different cameras and different resolutions. The first task is to standarize as much as possible the images. The proposed data preprocessing pipeline is the next:

- Remove image borders
- Resize all images to the same size
- Equalize luminance, brightness conditions
- Globally normalize

#### Remove image borders

Nowadays, one of the bottle necks of deep learning is memory limitation. In order to not waste resources is important to remove from the images all the non important parts. In this problem, the black background doesn't give us any information and should be removed. As all the raw images come with an important part of background, the first thing that has to be done is the trimming of the images in order to remove it as much as possible. The retine has a spherical form, so as the image is rectangular is inevitable to have some background. The purpose is to make one of the dimensions equal to the diameter of the retine.

## Resizing

The objective is to find a standardized input size for feeding the neural network. Some different input sizes will be tested. The proposed sizes are the next ones:

- 256x256
- 384x384
- 512x512

Different models will be constructed with this sizes in order to find the optimal size that improves the classification accuracy.

## Equalize luminance, brightness conditions

In order to reduce the training time, some preprocessing will be done to equalize the luminance and brightness conditions of the images. As has previously stated, the images have been taken on different conditions, some of them are overexposed, others underexposed, etc. Some pretreatment would make easier the training. The proposed strategies are the next ones:

- Histogram equalization
- Contrastive normalization of luminance channel
- Histogram equalization: All the images are similar in the sense that all are retines. The proposal is to look for a well exposed image and try to match his histogram as much as possible with all the images of the dataset.
- Contrastive normalization of the luminance channel: Map the RGB channels into the YUV space to separate the luminance from the color information. Normalize locally the luminance channel using a contrastive normalization operator for each defined neighborhood, define by a gaussian kernel.

## Globally normalize

Learning time is reduced when the input is normalized. Mean and standard deviation of the three channels will be calculated over the full training set. This values will be posteriorusly used for the normalization of the test set. After the global normalization all the three channels will have 0 - mean, 1 - standard deviation. All the models will be tested using as an input either a RGB or a YUV input.

## 3.2 Data augmentation

Deep learning requires a lot of data to get the most of their prediction power. Some data augmentation techniques has been used historically to increase the data availability and to improve the generalization of the models. Due to the spherical nature of the retine, we will use 0..360 random rotations of the images as a augmentation technique as well as x and y axis mirrors.

## 3.3 ML Model proposals

The idea is to use convolutional neural networks (CNNs) to solve the problem. It

was proved that a 2-layer neural network can approximate any function, but in the last years of reserch have been proved that deeper architectures can make the same job with less computing units. With some problems has been proved that just adding one layer can reduce exponentially the required number of computation units to achieve a previously fixed maximum error level.

In order to check this results with our dataset we will use different architectures with different number of layers in order to check if the expressivity of the model grows with the deepness. The activation function to be used is the ReLU. This activation function has stated in the last years as the best one in order of expressivity and fast convergence.

The proposed model creation pipeline is the next:

- 1- CNN with gradient descent and random initialization
- 2- CNN with unsupervised pretraining
- 3- Using a general purpose DCNN like IMAGENET for making the prediction
- 4- Using similarity models
- 5- Image key feature detection

#### 3.2.1 CNN with gradient descent and random initialization

Some models will be constructed using the gradient descent with random initialization. The idea is to test some models using a direct final classification scheme (multiclass classification), others with some binary classification schemes (0 vs 1,2,3,4), (1 vs 2,3,4), etc. The weights will be randomly initialized.

#### 3.2.2 CNN with unsupervised learning

The dataset has a lot more of 0 class images than 1,2,3,4. Due to this when the training is made, has to make an equilibrated sampling of images. One strategy to use all the capability of the training set is to make a unsupervised learning pretraining phase and after that, make the final classification training using gradient descent. This is what will be done in this phase. Autoencoders will be used to train every layer. A stacking strategy will be used afterwards in order to construct the deep architecture for posterior classification. Different kinds of autoencoders will be used (denoising, etc.). There's space for exploring new types of them and create new ones.

#### 3.2.3 Using a general purpose DCNN like IMAGENET for making the prediction

There's some papers where the deep features of a general purpose net has been used to solve totally different problems. The idea is to try to explore if we are able to use the knowledge of a already trained on a completely different purpouse deep net to solve our problem.

#### 3.2.4 Using similarity models

The idea here is to check for similarity of images. Feeding to images to the system see if they are of the same class or not.

#### 3.2.5 Image key feature detection

The last and most challenging task is to try to extract from a trained model which parts of the image are the main responsables of the final conclusion, that is to say, been able to find the reasons of the final conclusion found by the model.