Papilloedema Detection App: Initial Plan

Matheus Gomes

11/06/2018

Contents

Introduction

This document contains the description for the initial plan of the project. It is worth noting that only the Python programming part is considered in this document.

The first section will outline how the components were done previously in the Darwin project, and why certain parts will need to be refactored/removed in order to suit the task at hand. Furthermore, the later section will include the outline of the initial plan, with a flow diagram explaining inputs/outputs of each chosen components and a rough timeline of the tasks and deadlines.

1 Previous Implementation

The Darwin project focused on the detection of Papilloedema on given retinal images. Numerous diagnoses approaches were considered, but the chosen method was to detect the Optic Disc (OD) in the images, then use them to train a Convolutional Neural Network (CNN) which could provide a diagnosis for a given OD.

In terms of coding the components, the whole project was split into different parts: image pre-processing, OD detection and machine learning. The image pre-processing part dealt with variations in images by normalising images in the dataset with histogram manipulation, noise reduction and blur correction techniques. The OD detection component used PCA to extract the region of interest in every retinal image. Finally, CNN models were trained using TensorFlow and the obtained positive/negative images found in the dataset. Figure 1 shows a brief flow diagram of the Darwin project.

The image processing component was applied to the whole dataset. Firstly, the dataset was analysed, extracting information such as how blurred images were, contrast profiles, etc. Based on the extracted information, the images were then processed using the aforementioned normalisation methods. From the image pre-processing component, the OD extraction was then used on the whole notrmalised dataset and the corresponding ODs were passed onto the TensorFlow models.

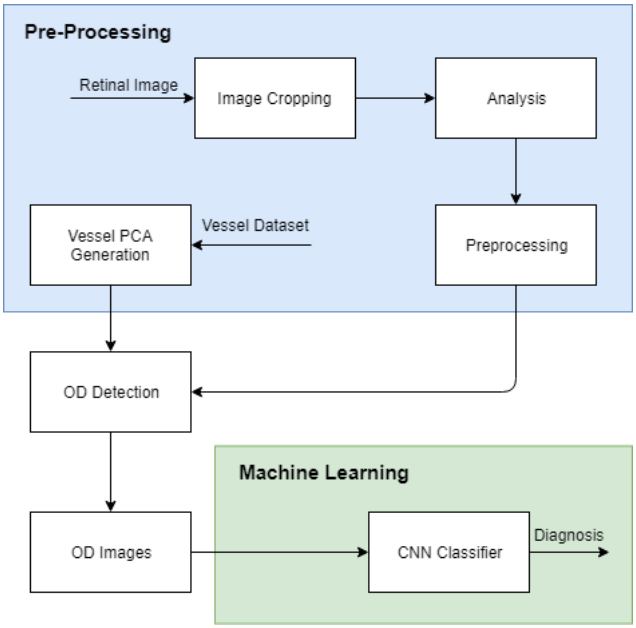

Figure 1: Darwin project flow diagram.

The Darwin project's split is exactly the same as this project's split, however, the order in which the components were applied will differ. Darwin's aims were made with the report & journal in mind, so the diagnosis process was designed to be used with datasets as opposed to web servers, where a more sequential approach is needed. For this reason, the design was changed with users in mind, where a single image will have to be processed and analysed at a time, and the diagnosis will be returned to the user.

2 Design Changes and the Plan

The functional aims of each component will remain the same as the Darwin components. Most of the changes being made are supposed to accommodate the server architecture, where a user would be sending the retinal image to the server, and the server would respond with some form of diagnosis. Due to the nature of the task, some components (such as the image processing), will have to be rewritten to work with single user images as opposed to a whole dataset. In addition, a more robust CNN will be designed for this project, since the CNN used for the Darwin project was a simple MNIST CNN obtained from the TensorFlow website, which is not very versatile in terms of tweaking the CNN architecture.

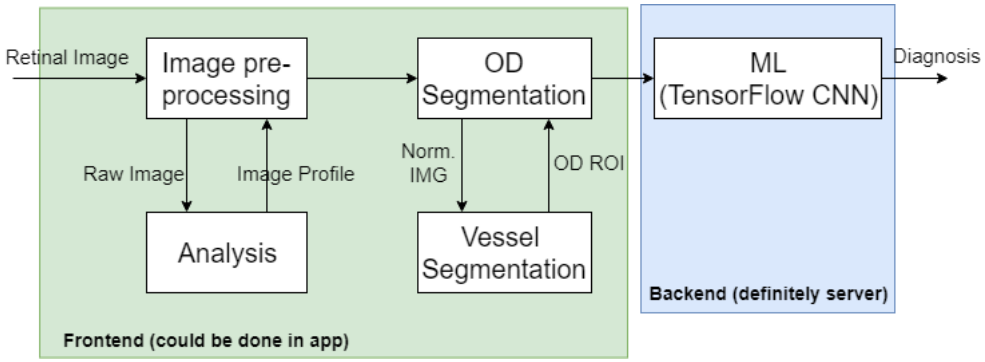

Figure 2: Rough flow diagram for Papilloedema detection app.

Figure 2 shows a simplified flow diagram of the components in the project. Unlike the flow diagram for the Darwin project, the Papilloedema detection app will require a more sequential flow, where the input is the image of the user's retina, and the output would be the diagnosis of the retinal image (from the CNN model). Although the components are nearly identical, the data flow is different: the communication between the preprocessing, OD segmentation and the CNN components will have to be automated, meaning that input/output of each component will be passed on automatically to the next component. This differs from the Darwin approach, as the output of each component on the dataset was being saved locally for manual analysis of the data, then manually input into other components.

3 Aims & Objectives Timeline

The purpose of this section is to provide a rough timeline for the development of the diagnosis system. Although parts of the project can simply be refactored from the Darwin project, there is still a lot of work to be done with respect to setting up the server environment, and redesigning certain components. The list below contains a rough estimate of will have to be done, and a timeline estimation for each task.

1. **Setting up the server**: The necessary dependencies will have to be installed on the server. ***11/06 - 14/06***
2. **Analysis of obtained data**: Analsing the current dataset of retinal images obtained with mobile phones, then modifying the analysis component of image pre-processing to suit the acquired data.***15/06 - 19/06***
3. **OD segmentation**: Change the current OD segmentation code, based on the analysis of the new dataset. Note this is only applicable if the mobile app will give the user options to automatically detect the OD. ***20/06 - 25/06***
4. **CNN design**: Create a CNN to make the diagnosis. Make use of TensorFlow's libraries to design a more robust CNN to diagnose the images sent by the users.***25/06 - 02/07***

It is worth noting that, since intermediate tasks will most likely have to be added, the timescales used above is quite generous, hence there will be time to complete additional stuff.

4 Current State

A Github repository was set for the project. In addition, boiler code and utility code which will be needed were added. This consists of OpenCV function wrappers, OD independent file management and analysis code that were previously written.

The plan for the rest of the week is to set up the server and to run some of the current code on the server so we can test the dependencies etc. For this reason, I do need the server details as soon as possible.