University of Regina

Neural Network for Automated Detection of Diabetic Retinopathy in the Eye

CS476

Instructor: Dr. Samira Sadaoui

Application Domain: Healthcare / Medical Science

Group Members:

Vaibhav Sharma and Will Wallace

Problem Statement and Motivation:

Our motivation is to create an application that can be of value in helping to screen patients for diabetic retinopathy, a disease of the eye which is preventable but is also unfortunately one of the most predominant causes of blindness worldwide. By adapting pre existing tools such as neural networks, which can be trained to differentiate between healthy and diseased eyes, we hope to provide an easily accessible and usable system that will give meaningful and quick results to specialists who may have a backlog of retinal images they need to check for disease. More and more patients are needing care every day, and specialists cannot be expected to do everything on their own. By providing faster screening and diagnosis methods in a healthcare setting, patients can receive appropriate treatment in a more timely manner. Being able to quickly scan images for disease is an important step towards this.

User Roles:

End User: Typically, the end user will be a physician specializing in eye care (an ophthalmologist) or someone else working in the field, e.g. a technician employed at an eyecare clinic. It is assumed that the end user working at such a clinic will already have access to the cameras and equipment needed to provide high-resolution images of patients' retinas.
Functional Requirements for End User:
• Upload an image: The user should be able to upload an image of the retina to be processed by the neural network.
• View results: After the processing has completed, the system should provide output consisting of the image's classification in terms of retinopathy severity (ranging from none, i.e. healthy, to more advanced stages such as proliferative retinopathy).

Software Functional Requirements:

- User must be able to upload an image upto 100mb in size.
- User must be able to use "Image Picker" to upload a picture.
- User must get a result when evaluating an image

Software Quality Requirements:

Robustness:

The system should be fault-tolerant to the extent that unexpected keystrokes or “misclicks” do not jeopardize the integrity of the system in any way. For example, accidentally hitting the Enter key during image processing should not affect the output or cause the system to be interrupted or delayed. If the end user uploads an invalid file, such as an MP3 or any other undesired file type, this should be detected and no image processing should be attempted. An appropriate error message should be displayed, explaining in plain language why the input was unacceptable.

Correctness:

Neither humans nor machines can offer one hundred percent accuracy when it comes to determining what an image shows; after all, two experts may disagree about what they are looking at when presented with the same picture. However, the system should offer a degree of accuracy that correlates well to what a panel of experts would determine if they looked at the same image. For example, if 90% of specialists agree that a particular image shows a healthy eye, we should expect the system to give a similar report.
When an uploaded image is processed and the results are provided to the end user, these results should correlate well with what the ophthalmologist sees; the specialist and the machine should be in agreement in the majority of cases.
Note: The exact probability of coming up with the correct result is impossible to predict until the network is trained; obviously, anything less than fifty percent would be useless.

Time Efficiency:

It is expected that the system should be able to be quickly and efficiently used in a variety of settings, wherever eye care and diabetes screening is done. This may include a small private clinic where only a few patients are seen each day, or a busy hospital environment that is aimed at screening as many patients as it can reasonably see in a day.
In either case, but especially in a busy hospital setting, the end user should expect to receive the results of their diagnostic imaging in only seconds. The doctor or technician using the system should not be left wondering how long it will be before they can see the output.
From the viewpoint of the programmer responsible for software maintenance, time efficiency is less critical. In fact, allowing the system more time while training (or retraining with new data) is preferable so that accuracy is better assured.

User Friendliness:

The vast majority of the system’s usage will be undertaken by the end users and not computer experts (except when maintenance is being done).

For this reason, the system should be clear and simple to understand. Proper UI design heuristics should be adhered to; e.g. a minimalist design is preferable so that the screen remains uncluttered and the user's attention can be directed to where it needs to be.
The options available to the end user should be displayed in an uncluttered environment so that there is no confusion as to what the user can do. For example, icons or buttons representing access to the two main functional requirements of the end user should be clearly displayed whenever available, e.g. Upload Image or View Results.
From the viewpoint of the maintenance specialist, any visible code should be well-documented and easily understood with the aid of comments.