## **Milestone 2**

|  |
| --- |
| **Model Pipeline Design Title Page** |

**Special Instructions:**

Here are some pre-processing steps needed before running code written. There are some hyperparameters tuned during training including which parameters were chosen, ranges explored over cross validation and why they were selected.

**Objective:**

To train a model that is capable of diagnosing fundus images so that low end medical staff/volunteers can provide quicker and easier service within 3rd world countries.

**Deliverable:**

A toolkit or library containing an implementation of all methods used to build and test the solution as well as sample input/output pairs so you can reproduce our results easily. This should be accompanied by clear documentation explaining how everything works with examples where possible.

|  |
| --- |
| **Design Planning Summary** |

With ODIR fundus image data a Convolutional Neural Network (CNN) model will be trained. The image data must have black space trimmed to increase runtime and allows more valuable information to be spread out within in segmented cell. There must also be a vetting of each image for proper testing and image resizing for standardization within the preprocessing stage. Once the data is processed into the CNN model there are three different analysis parameters to look at after prediction. One is just giving us an accuracy score which gives no indication about how confident the output is; one classifies each image based on whether certain eye diseases exist or not like Glaucoma or Diabetic Retinopathy. The last is simply looking at the confidence level itself. In both cases the top layer shows that it has been successful. But in order for the model to work effectively, it needs to be trained on sufficient number of pictures showing signs of disease.

|  |
| --- |
| **Overview of Model Pipeline Design (OSEMN Model)** |

The design concepts must address the following:

1. **How will the data be obtained?**

The Data comes from Ocular Disease Intelligent Recognition ODIR research center. It contains 8 different classifications of diagnosis, including observations with multiple classifications. This data has been involved in data science competitions for developing artificial intelligence models for diagnosing ocular disease.

1. **How will the data be scrubbed or cleaned?**

The fundus images have black spaced trimmed and resized to a standardized dimension. Also the data must be filtered and vetted through to ensure proper diagnosis within the images before actual and predictive comparison can be done.

1. **How the data will be explored and visualized (e.g., to detect patterns and trends)?**

This CNN model has been simplified to do multi-level classification instead of binary. Each image will be connected to a patient ID and the output will made a prediction on classification, which includes Normal, Cataracts, Glaucoma, Myopia, Diabetic, Age-related Macular Degeneration, Hypertension, and finally Abnormal.

1. **What data model will be used (e.g., how will you set up a predictive model)?**

As already previously mentioned, a Convolutional Neural Network capable of multi-level classification will be used. It will hold multiple levels of layering and different sample sizing designed to optimized specific parameters within the model.

1. **What methods will be used to interpret the results of analysis?**

Several other models will be run along-side the CNN for comparison and a measurement of actual vs predicted result testing will be performed.

|  |
| --- |
| **Detailed Model Pipeline Design** |

The fundus images will go through a Convolutional Neural Network (CNN) a model that is known for its ability to transfer pixelated images into arrays of numbers that allows the computer to understand what the image contains. This means, we can use this CNN models and train it on retinal data from the ODIR archive in order to make predictions on diagnosis for each patient in the absence of medical professionals.

**Convolutional Neural Network Model:**

Diagram

Description automatically generated

The design overview should include:

1. **The data source:**

<https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k>

1. **The dataset types and formatting:**

The dataset is composed with images that will be formatted into numeric color and binary (black and white) arrays for analysis. For example, the image above consists of four channels of colors red, green, blue, and gray. Each channel will be broken down into segments, and depending upon the situation may or may not include black spaces within the image.

1. **The data cleaning procedure:**

The fundus images are trimmed of useless black-space and resized into a standardized format. They also need to be cleaned before feeding into the deep learning system, removing all unnecessary noise and ensuring that the entire image is contained within the boundaries of the frame. Afterward the data is converted into a series of vectors whose dimensions range between zero and one hundred fifty-one, representing the pixels inside of every image.

1. **Method of initial data exploration and visualization:**

The data is processed to reveal reverse contrast and black and white formats for analysis. An array is generated for the new image while taking care to maintain aspect ratio. The final step is to process the image through a CNN to get predicted values for each pixel.

1. **The data model used and its nature (e.g., predictive):**

A CNN that will process each fundus image to create a prediction of disease diagnosis. A type of artificial neural network that can recognize features in images, and are often used for tasks involving visual recognition, particularly in image classification problems. CNNs are designed to solve these image recognition issues, specifically working with 2D image inputs and producing outputs via probability estimates. CNNs take advantage of the spatial relationship of local image features, meaning they learn to identify objects in relation to other neighboring parts of the same object, rather than treating each part independently.

1. **The methodology for interpreting the analysis results:**

Interpreting the data from each predicted result will be done through a statistical analysis of its confidence levels and accuracy. A Convolutional layer performs a local operation on neurons across a small region of the previous hidden layers, sliding across the feature maps at every pixel position. The filter weights are shared between different positions within the image. These weight matrices are learned during backpropagation, where errors from the output are propagated backwards towards the input. A Max Pooling layer takes the largest response activation among several neurons in a given window, reducing the size of the output representation. This reduces the complexity of subsequent layers while maintaining the most salient features. A Fully Connected Layer connects every neuron to every other in the next layer. It flattens the output of the pooling layer so that connections can span the full width and height of the image.

1. **Any configuration changes that will be required to develop and implement the proposed solution:**

The initial design has been scrapped in face of limitations and restrictions of the nature of the assigned project. However, it serves as a good base to expand upon in later projects.

1. **Describe the approach and resources required to assure system security, if applicable; otherwise, explain why security is not relevant:**

To add security to this project, it must run only within a web browser without needing installation of anything. Thus the user must have access to their internet connection and camera for capturing the image(s), then uploading them directly into the site. The website is supposed to display an image once uploaded along with two buttons asking if said image shows any signs of either Diabetic Retinopathy or Glaucoma. If these are true positives, there should also appear a third button telling the viewer how confident the software was in making such a prediction - ranging from 0-1 in increments of .01. This value indicates the certainty rate the system had in making such a judgment.

1. **Use the template to list the hardware and software technologies:**

* Python Packages
  + Tensorflow
  + streamlit
  + PIL
  + Image
  + ImageOps
  + os
* 16-Core Processor
* GPU with Tensor Cores

|  |
| --- |
| **Projects Requirements Review** |

The initial plan was to use an ensemble of Convolutional Neural Networks (CNN) to do multiple binary classifications in the fashion of a multi-classifying decision tree, but due to the size of the model and the mandatory milestone 3 web-based application, this path would much too large for a single person to handle, or even transfer to a web-based application through .h5py files. While my computer is capable of Tensorflow, my GPU wasn’t being picked up, which resulted in my model being purely trained on my Processor, which is quite slow. So throughout this project I’ve faced a plethora of limitations and restrictions that I wasn’t aware of before starting this project, yet I am determined to continue since I believe that this topic is of importance and highly interesting.