OCCULAR DISEASE RECOGNITION

-ADITYA MUKHERJEE
RA1911003010210

Acknowledgement

- I would like to thank Grzegorz Meller for the inspiration to the project

Abstract

- **Approximately 12 million people** every year suffer from eye diseases like **macular degeneration, glaucoma, and cataracts**
- Some of these are not even in detected and if left alone it can be fatal sometimes
- The equipment required for the detection can also be costly also weather the doctor remained equipped to decipher is a question too

- Thus my answer to this problem statement was to create a Image classifier using pytorch, that is trained on over 2000 images of eye problems and can then help you identify if you have any problem in your eye or not from just a click

Intorduction

- Early ocular disease detection is an economic and effective way to prevent blindness caused by diabetes, glaucoma, cataract, age-related macular degeneration (AMD), and many other diseases.
- According to World Health Organization (WHO) at present, at least 2.2 billion people around the world have vision impairments, of whom at least 1 billion have a vision impairment that could have been prevented.
- Rapid and automatic detection of diseases is critical and urgent in reducing the ophthalmologist's workload and prevents vision damage of patients.
- Computer vision and deep learning can automatically detect ocular diseases after providing high-quality medical eye fundus images.

- Thus, this projects works on creating an accurate and well to do Project in the eye funds recognition

Dataset

- Ocular Disease Intelligent Recognition (ODIR) is a structured ophthalmic database of 5,000 patients with age, color fundus photographs from left and right eyes, and doctors' diagnostic keywords from doctors.
- This dataset is meant to represent the "real-life" set of patient information collected by Shanggong Medical Technology Co., Ltd. from different hospitals/medical centers in China.
- In these institutions, fundus images are captured by various cameras in the market, such as Canon, Zeiss, and Kowa, resulting in varied image resolutions.
- Annotations were labeled by trained human readers with quality control management. They classify patients into eight labels including normal (N), diabetes (D), glaucoma (G), cataract (C), AMD (A), hypertension (H), myopia (M), and other diseases/abnormalities (O).

- After preliminary data exploration I found the following main challenges of the ODIR dataset:
- Highly unbalanced data. Most images are classified as normal (1140 examples), while specific diseases like for example hypertension have only 100 occurrences in the dataset.
- The dataset contains multi-label diseases because each eye can have not only one single disease but also a combination of many.
- Images labeled as “other diseases/abnormalities” (O) contain images associated to more than 10 different diseases stretching the variability to a greater extent.
- ·Very big and different image resolutions. Most images have sizes of around 2976x2976 or 2592x1728 pixels.

- Certain data pre processing was done on the images to help overcome this problems
- This was done using opencv module
- As the number of images was less , Image augmentation techniques was used to help increase the test metrics
- The were done in the following file
- https://colab.research.google.com/github/adityamukherjee42/Ocular-Disease-Recognition-/blob/main/Occular_disease_preprocessing.ipynb#scrollTo=qK0DU--zmv6u

Building Convolutional Neural Network

- In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery.
- Input layer takes 250x250 RGB images. The first 2D convolution layer shifts over the input image using a window of the size of 3x3 pixels to extract features and save them on a multi-dimensional array, in my example number of filters for the first layer equals 64,
- And to reduce overfitting and computation time , a maxpool layer was used so the cube or the image size will be (50, 50, 32)

- Similary more layers were created with similar parameters
- After each convolution layer, a rectified linear activation function (ReLU) is applied. Activation has the authority to decide if neuron needs to be activated or not measuring the weighted sum. ReLU returns the value provided as input directly, or the value 0.0 if the input is 0.0 or less. Because rectified linear units are nearly linear, they preserve many of the properties that make linear models easy to optimize with gradient-based methods. They also preserve many of the properties that make the linear model generalize well

- And A Dropout layer with 60% percent drop out rate was created to create more varations and avoid overfitting

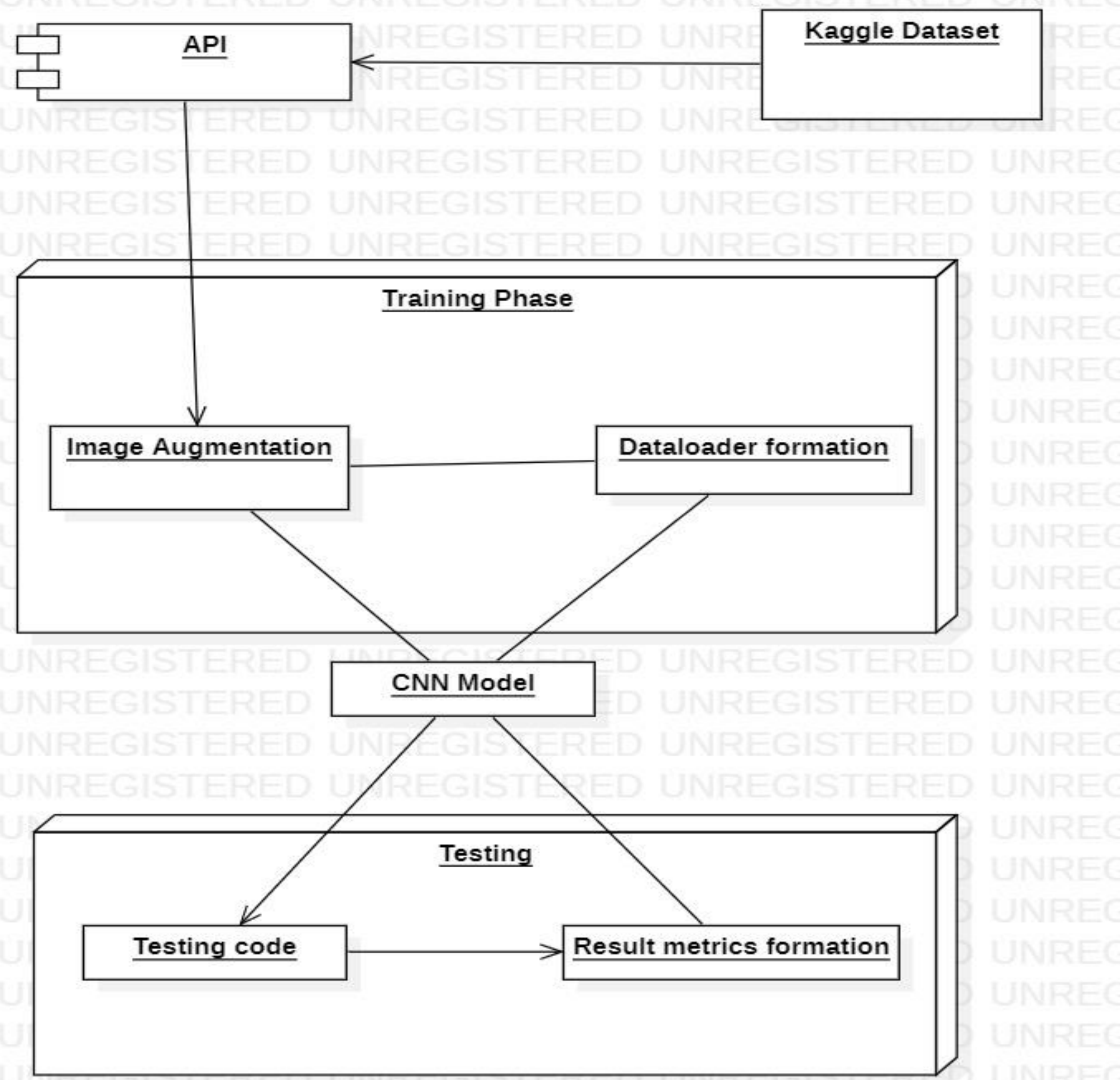
Training

- For Training , K-Fold cross validation technique was used
- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
- The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation

Result metrics

```
Accuracy: 0.810000
Precision: 0.606965
Recall: 0.884058
F1 score: 0.719764
```

PIPELINE ARCHITECTURE DIAGRAM



Conclusion

- In this new age of Big data, my aim here was to create a universal pipeline using pytorch so that anyone starting off can see this and take it as an inspiration
- Also while this pipeline was created , It was kept in mind that that it will be trained again with more points that will result in better accuracy
- The usage of this software ,can be widespread.
- Medical with machine learning is need of the hour as with insufficient doctors available especially in the village places

- The Pandemic has also emphasized the same, The huge loss of life could have been reduced if we had an infrastructure where inaccessible people could have been diagnosed using a mobile app or the net
- While Internet is still inaccessible in some areas, the advancement is a necessary in this age
- Also with cases like fake doctors or doctors making human errors can be eliminated with the use of Software and technologies like this

- Thus,keeping all of this in mind this project was created
- Further up gradation like training the model with more images and giving it an frontend and hosting it on the net will be done

Important links

- https://colab.research.google.com/drive/17fn_JbMUsWom9Ry74vT8UDa8m301uxyF#scrollTo=WywrSAdKGzWw

- https://colab.research.google.com/github/adityamukherjee42/Ocular-Disease-Recognition-/blob/main/Occular_disease_preprocessing.ipynb#scrollTo=k5dJoabfhyn4

- https://github.com/adityamukherjee42/Ocular-Disease-Recognition-

Stack used

- Pytorch-For creating CNN
- Opencv-For image Augmentation
- Streamlit-For frontend

Dataset used

- https://www.kaggle.com/kondwani/eye-disease-dataset

Refrence

- https://towardsdatascience.com/ocular-disease-recognition-using-convolutional-neural-networks-c04d63a7a2da

THANK YOU