Multi-labeled Disease Classification of Retinal Fundus Images

1st Rishabh Sharma
College of Computing and Informatics
Drexel Unversity
Philadelphia, USA
rs3738@drexel.edu

2nd Jasjiv Singh
College of Computing and Informatics
Drexel Unversity
Philadelphia, USA
js5324@drexel.edu

3rdUjjwal Malik
College of Computing and Informatics
Drexel Unversity
Philadelphia, USA
um43@drexel.edu

***Abstract*—This research paper presents an investigation of deep learning models for multi-class classification of retinal fundus images using convolutional neural networks (CNNs). The primary objective is to compare the performance of various CNN architectures for accurately and efficiently classifying retinal images into multiple disease categories. The study employs a publicly available dataset of retinal images and implements VGG16, InceptionV3, and ResNet50 models for comparative analysis of their classification accuracy and computational efficiency. The results demonstrate that deep learning models can effectively classify retinal images into multiple categories with high accuracy. The study provides insights into the efficacy of CNN models in retinal disease classification and contributes to the development of automated diagnosis systems for retinal diseases.**

***Index Terms*—Deep Learning, CNN, Neural Networks, ResNet, InceptionV3, machine learning.**

I. Introduction

Retinal problems affect millions of people each year and are a major public health concern worldwide. For the purpose of preventing vision loss and blindness, retinal disorders must be identified and treated early. Deep learning techniques have made recent strides, and the categorization and analysis of medical images has shown promising results. Due to its capacity to extract intricate information from medical images, convolutional neural networks (CNNs) have become a prominent option for automated diagnosis systems. This study examines how well different CNN architectures perform when classifying retinal fundus images into many categories. Comparing the precision and effectiveness of the VGG16, InceptionV3, and ResNet50 models in classifying retinal pictures into various illness categories is the main goal. The other objective of this study is to implement the neural network from scratch using libraries like pandas and numpy. For the following study our end goal was to classify the images under the following 1 or more categories: Diabetic Retinopathy, Media Haze, Tessellation, Optic Disk Coloboma.

II. Previous Work

Recent advancements in deep learning techniques have shown promising results in medical image analysis and classification, particularly in ophthalmology. In the context of retinal image analysis, several studies have investigated the use of deep learning models for the detection and classification of retinal diseases.

A new method for detecting Diabetic Retinopathy (DR) in fundus images[1] has been presented, which combines transfer learning with hybrid feature extraction and uses various classifiers. The proposed model is capable of accurately classifying fundus images into different stages of DR, and its performance has been compared to recent research articles, demonstrating significant improvement in average accuracy. Using a hybrid feature vector and SVM classifier, the model achieved a maximum average accuracy of 97.80 percent for binary classification and 89.29 percent for multi-class classification.

Retinal disorders can be detected by analyzing fundus images, which are commonly used by medical professionals such as ophthalmologists. These images can also be used to predict the severity of diseases and provide early warnings. In recent years, machine learning algorithms have been increasingly used in medical science, including ophthalmology. In this other study[2] ,they aimed to use deep neural networks to automatically classify healthy and diseased retinal fundus images. Deep learning is a highly accurate machine learning algorithm that has shown promising results in computer vision problems. They employed convolutional neural networks (CNN) to classify retinal images as either healthy or diseased.

Most of the work in the field of diabetic retinopathy has been based on disease detection or manual extraction of features, but in another one such research[3], it aims at automatic diagnosis of the disease into its different stages using deep learning. This paper presents the design and implementation of GPU accelerated deep convolutional neural networks to automatically diagnose and thereby classify high-resolution retinal images into 5 stages of the disease based on severity. The single model accuracy of the convolutional neural networks presented in this paper is 0.386 on a quadratic weighted kappa metric and ensembling of three such similar models resulted in a score of 0.3996.

In another research paper[4], the CNN (Convolutional Neural Networks) algorithm labeled the dataset of OCT retinal images into four types: CNV, DME, DRUSEN, and Natural Retina. We have also done several preprocessing on the images before passing these to the neural network. They implemented different models for our algorithm where individual models have different hidden layers. At the end of our following research, we have found that our algorithm CNN generates 93 percent accuracy.

III. Data Collection and Image Preprocessing

A. Data Collection

The Data for this study was collected from Retinal Image Analysis for multi-Disease Detection Challenge. The initial Data set consisted of 46 Disease Class labels. For the sake of this study we narrowed down the classes to 4. These 4 classes consists of maximum number of image data and are also the most common diseases in retinal fundus. These diseases are: Diabetic Retinopathy (376 images), Media Haze (317 Images), Tessellation (186 Images) and Optic Disk Coloboma (282 Images). These images were used as training data. The other sample of 600 images combined were split into testing and validation data sets.

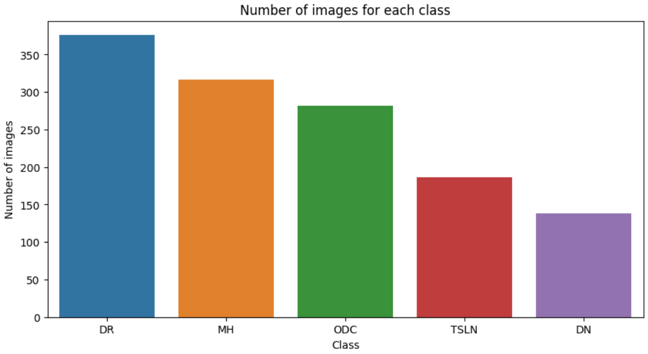

Fig. 1. Sample size for the top 5 diseases

B. Image Preprocessing

Retinal fundus images are commonly used in computer-aided diagnosis systems for detecting various eye diseases. Prior to feeding the images into a convolutional neural network (CNN) model, some preprocessing steps are required to enhance the quality of the images and convert them into numpy arrays.

Firstly, the retinal fundus image is loaded into the computer memory using a suitable image loading library such as OpenCV. The image is then converted into grayscale to reduce the dimensionality of the image and facilitate further processing. First, the image is resized to a suitable input size for the CNN model. This typically involves cropping or padding the image to ensure that it matches the desired input dimensions of the CNN model. The image is then normalized to have a mean of 0 and a standard deviation of 1 to improve the convergence of the CNN model during training. We experimented with various sizes of images to ensure the best and most optimum size for the image data(Computation wise). We then applied circular crop on the image data to ensure no additional unnecessary data is given to the model.

Gaussian blur is applied to reduce noise and smooth out the image. This is important for improving the robustness of the CNN model and preventing overfitting. After applying Gaussian blur we were able to see through low contrast and uneven illumination due to the natural variation in the retina.

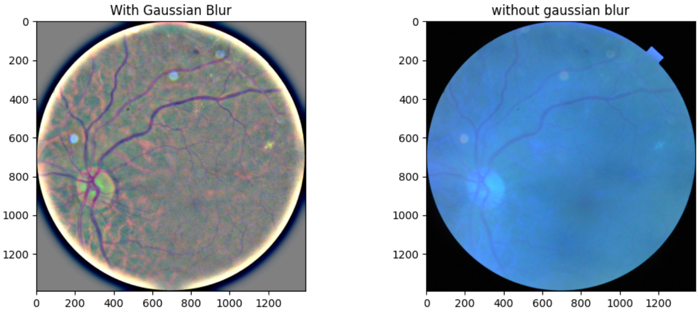

Fig. 2. Image after and before applying Gaussian Blur

The processed images were then converted into numpy arrays to be given to the model as an input.

IV. Architecture

For the following study we created various unique networks and architectures to design our Convolutional Neural Network. We also Drew inference from various other architectures including a Multi layer perceptron model for multi-class classification problem.

A. Multi Layer Perceptron Model

Multi-labeled image classification is a challenging task in computer vision, where an image can have multiple labels associated with it. An MLP model is a type of artificial neural network that can be used for multi-labeled image classification. This model consists of multiple layers of perceptrons, where each perceptron is a simple mathematical function that takes input values and produces an output value.

The MLP model for multi-labeled image classification consists of an input layer, one or more hidden layers, and an output layer. The input layer takes the raw pixel values of an image as input which after being z-scored are then fed forward to the hidden layers. Each hidden layer consists of multiple perceptrons, and the output of each perceptron is determined by a weighted sum of the inputs, followed by an activation function. The output layer of the MLP model consists of multiple neurons, where each neuron represents a label. The final output of the model is a vector of probabilities that represents the likelihood of each label being present in the image. For the following problem we followed a classic architecture which consisted of an input layer followed by 4 hidden layers which consisted of Dense layer, an activation layer(ReLU Layer) and a drop out layer. The output layer consisted of a Dense Layer with 4 perceptrons. And the Output activation function was sigmoid followed by binary-cross entropy loss layer. For the inputs, the data which was initially in the shape of RGB image$(128, 128, 3)$, was converted to a single dimension$(1, 128 * 128 * 3)$ to be given to the input layer. For various architectures the model gave out varied results.Thus it was important to select the best features for the model.

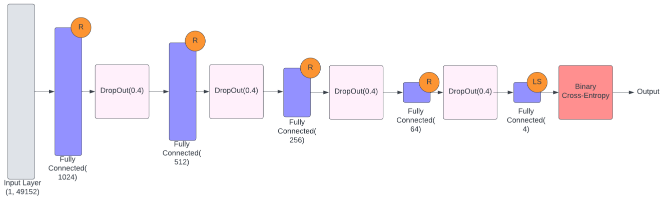

Fig. 3. Multi Layered Perceptron Architecture

1) Hyper-parameter Choices: For the MLP model, After experimenting through various combinations of parameter choices, We analysed the most optimum learning rate for the model to be 0.001. Learning rates of 0.01 resulted in fast convergence for the training data but performed poorly on validation data. Number of hidden layers for the problem were taken to be 4. Adding more hidden layers can increase the capacity of the model, allowing it to capture more complex relationships in the data, but can also increase the risk of overfitting, where the model fits the training data too closely and fails to generalize to new data. After testing the model on various numbers of hidden layers, We observed that 4 hidden layers helped in well fitted results for the training data and also the validation data.

ReLU activation was taken as inner activation functions as while dealing with the image data which is normalized, using any other activation function can lead to vanishing gradient problem. To counter this ReLU or Leaky ReLU seemed plausible choices.

To prevent overfiiting, we deployed regularizing measures like adding dopout layers to the model after internal activation layers. While not using these layers the model did show signs of overfitting.

ADAM can be a good choice for image classification problems in MLP models due to its adaptive learning rates, momentum, efficiency, and ease of implementation. ADAM is computationally efficient and requires relatively low memory compared to other optimization algorithms like BFGS or L-BFGS. ADAM is relatively invariant to the scale of the gradients, which means it can be a good choice for neural networks with widely varying gradients, such as those encountered in deep networks for image classification. Due to the following reasons ADAM optimization was used. Number of epochs were put to 10 with a batch size of 50.

B. Convolutional Neural Network

For the CNN, after trying out various pre existing architectures such as ResNet, VGG16 and InceptionV3, We discovered that due to excessive requirement of computation, it is better to move to smaller number of hidden layers along with decreased image shape. To keep the model simplified, a variant of LeNet architecture is used with the input RGB retina image size of (64, 64, 3). As it is a multi labeled classification, our output layer used will be the same as that of our multi-layer perceptron model. Using sigmoid output function with binary cross entropy/ log loss to calculate the probability of each disease and setting the argmax threshold of 0.5. The architecture used is displayed in the below figure

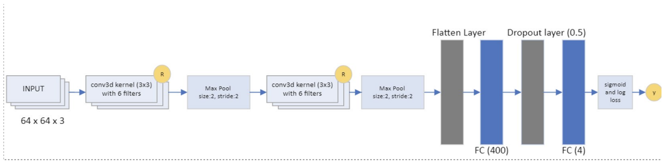

Fig. 4. Convolutional Neural Network Architecture

The core idea of convolution is to apply a filter (also called a kernel or a feature detector) to an input signal, in order to extract relevant features. In the case of a convolutional layer in a CNN, the input signal is a 3D tensor representing an image (or a batch of images), and the filter is a small 3D tensor (e.g., a 3x3x3 cube) that slides over the input tensor, computing dot products at each location.

The result of applying a filter to an image is a feature map, which captures some specific aspect of the image, such as edges, corners, or blobs. By applying multiple filters, a convolutional layer can extract multiple feature maps, each representing a different aspect of the input.

The mathematical equation for a 2D convolution operation (i.e., sliding a 2D filter over a 2D input) can be expressed as follows:

$$y_{i,j} = \sum_{k,l} w_{k,l} x_{i-k,j-l} + b \tag{1}$$

where $y_{i,j}$ is the output at position (i, j) in the feature map, $w_{k,l}$ is the weight of the filter at position (k, l), $x_{i-k,j-l}$ is the input at position $(i-k, j-l)$, and b is the bias term. This equation performs a dot product between the filter and a local patch of the input, and adds a bias term.

For a 3D convolutional layer, the equation is similar, but with an additional dimension:

$$y_{i,j,k} = \sum_{l,m,n} w_{l,m,n,k} x_{i-l,j-m,n-n} + b_k \tag{2}$$

where $y_{i,j,k}$ is the output at position (i, j, k) in the feature map, $w_{l,m,n,k}$ is the weight of the filter at position (l, m, n, k), $x_{i-l,j-m,n-n}$ is the input at position $(i-l, j-m, n-n)$, and b_k is the bias term for the k-th feature map.

To reduce the dimensionality of the feature maps, a convolutional layer often applies pooling after the convolution operation. Pooling computes a summary statistic (e.g., max, mean, or sum) over a small region of the feature map, and outputs a lower-resolution version of the map. This helps to reduce the number of parameters and computation in subsequent layers, and also provides some degree of translation invariance.

1) Hyperparameter Decisions: The model was trained using Mini-Batch Stochastic gradient descent, with batch size of 4 for 5 epochs. The learning rate for the model was taken as 0.0001 Through experimenting through various learning rates, we found out that faster learning rates shooted the optimum parameters where as the learning rate of $1e^{-}5$ was the reason for algorithm to converge slowly or get stuck in a suboptimal solution.

Weights for the model was initialized using Xavier initialization. With myriad experimentation on the LeNet architecture we observed that even for small and equally distributed data set, the model with convolution layers with $3 * 3$ kernels and 8 filters, As smaller size of filters, help in capturing fine edges of the image, it was relevant to use a 2x2 or a 3x3 filter for the purpose. We took 6 filters as it makes it optimum computationally as well as helps in capturing complex features of the image with each layer.

A drop out of 0.5 was used After the internal activation to reduce the overfitting. For the 2 fully connected layers or Dense layers, the only inner activation was taken to be ReLU as it will neglect all the values less than 0 and avoid vaishing gradient because of linear properties.

V. Evaluation

A. MLP

The multi-layer perceptron (MLP) is a popular neural network architecture for multi-labeled classification problems, and achieving an accuracy of 85 $\%$ on the training data and 67$\%$ on the validation data is a promising result. This indicates that the model is able to capture complex relationships in the data and generalize well to new, unseen examples.

However, it's important to note that the validation accuracy is lower than the training accuracy, which suggests that the model may be overfitting to the training data. Overfitting occurs when a model becomes too complex and learns to fit the noise in the training data rather than the underlying patterns. This can lead to poor generalization performance on new examples.

To address overfitting, it may be helpful to incorporate regularization techniques such as L2 regularization into the model. Additionally, it's important to carefully evaluate the model's performance on a held-out test set to get an unbiased estimate of its true performance on new data.

Overall, an 85$\%$ training accuracy and 67$\%$ validation accuracy are promising results for an MLP model in multi-labeled classification, but further analysis and optimization may be necessary to improve the model's generalization performance and ensure its suitability for real-world applications.

B. CNN

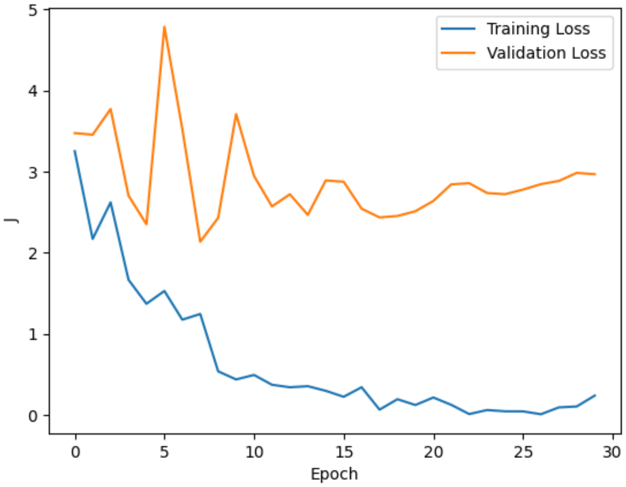

Fig. 5. training V/S validation Loss for Multi layered Perceptron

Using the convolutional Neural Network gave significantly good results and was able to classify the images to their associated multiple labels properly both for the training and validation data sets. The training accuracy for the model was $\%$ and the validation accuracy was $\%$. After trying multiple models with change of shape of the input data, hyperparameters and Architecture, The best architecture was the one shown in fig 4. with shape of $(64, 64, 3)$.

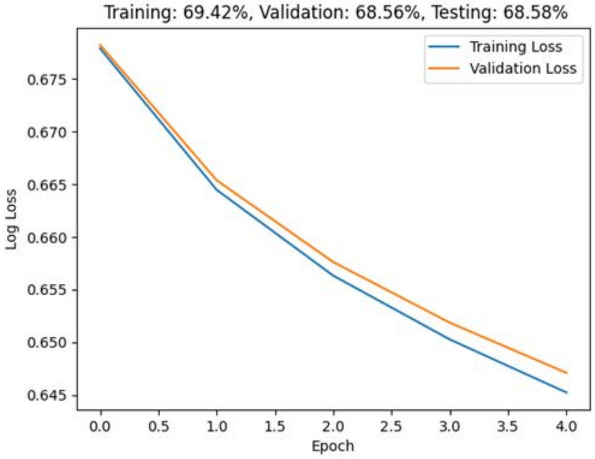

Fig. 6. training V/S validation Loss for Convolution Neural Network for 200 images in training data.

VI. Conclusions

With the following study we discovered several approaches to implement multi-label classification of retinal fundus images into 4 diseases. With this study we observed drawing intuition from the retinal images with the help of deep MLP Model and Convolutional Neural Networks. Multi Layered Perceptron model displayed signs of overfitting in the model. Where as the CNN model fit perfectly over the data for both training and validation.

CNN Model 1 and 2 carry same architecture except for one factor that is batch size. CNN Model 1 had batch size =4 and CNN Model 2 had a batch size of 8.

TABLE I
TRAINING AND VALIDATION ACCURACY TABLE

No.	***Model***	***Training Acc.***	***Validation Acc.***
1.	MLP Model	85%	67%
2.	CNN Model 1	84%	69%
3.	CNN Model 2	83%	68.5%

VII. FUTURE WORK

With the study of implementation of CNN for multi labeled classification along with the challenges faced, The future scope of the study will be the following.

a) : For the image preprocessing, We did not use CAHE(Contrast Adaptive Histogram Equalization). Retinal images can have uneven illumination and contrast, making it challenging for the classification model to accurately identify and classify features of interest. By applying CAHE, we can enhance the contrast and visibility of the retinal images, making it easier for the classification model to detect features and classify them. Additionally, CAHE can be applied locally to different parts of the image, depending on the local contrast characteristics. This can help to improve the contrast and visibility of smaller and more subtle features, which may otherwise be overlooked or difficult to identify. Overall, by applying CAHE as a pre-processing step in the classification of retinal images, we can improve the quality of the images and make it easier for the classification model to accurately identify and classify features of interest, which can lead to more accurate diagnosis and treatment of retinal diseases.

b) : For the ease of the project, The Models had to be made compact, As numpy operations for tensors can be very lacking and require heavy computation. This lead to development of smaller networks instead of dense networks. In the future scope of the project we can implement a VGG16 or Xception model, using GPU and cupy library for faster computation.

c) : For the following problem, the dataset was multi labeled, with minimum number of images for each category. For the future work, We plan to find a better data source with more images for each individual Category.

REFERENCES

[1] Butt MM, Iskandar DNFA, Abdelhamid SE, Latif G, Alghazo R. Diabetic Retinopathy Detection from Fundus Images of the Eye Using Hybrid Deep Learning Features. Diagnostics (Basel). 2022 Jul 1;12(7):1607. doi: 10.3390/diagnostics12071607. PMID: 35885512; PMCID: PMC9324358.29;2022:7968200. doi: 10.1155/2022/7968200. PMID: 35676956; PMCID: PMC9168160.

[2] S. A. Toki, S. Rahman, S. M. Billah Fahim, A. Al Mostakim and M. K. Rhaman, ”RetinalNet-500: A newly developed CNN Model for Eye Disease Detection,” 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2022, pp. 459-463, doi: 10.1109/MIUCC55081.2022.9781785.

[3] D. Doshi, A. Shenoy, D. Sidhpura and P. Gharpure, ”Diabetic retinopathy detection using deep convolutional neural networks,” 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, India, 2016, pp. 261-266, doi: 10.1109/CAST.2016.7914977.

[4] Mohammad, A., Utso, M. Z., Habib, S. B., Das, A. K. (2021). Predicting Retinal Diseases using Efficient Image Processing and Convolutional Neural Network (CNN). Journal of Engineering Advancements, 2(04), 221–227.

[5] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway et al., “Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally,” in Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 4761–4772, 2017.

[6] D. Xiao, S. Yu, J. Vignarajan, D. An, M. Tay-Kearney and Y. Kanagasingam, “Retinal hemorrhage detection by rule-based and machine learning approach,” in Proc. of 39th Annual Int. Conf. of the IEEE Engineering inMedicine and Biology Society, Seogwipo, South Korea, pp. 660–663, 2017

[7] Pachade S, Porwal P, Thulkar D, et al. Retinal Fundus Multi-Disease Image Dataset (RFMiD): A Dataset for Multi-Disease Detection Research. Data. 2021;

[8] Edward Ho, Edward Wang, Saerom Youn, Asaanth Sivajohan, Kevin Lane, Jin Chun, Cindy M. L. Hutnik; Deep Ensemble Learning for Retinal Image Classification. Trans. Vis. Sci. Tech. 2022;11(10):39. doi: https://doi.org/10.1167/tvst.11.10.39.

[9] Mohamed Akil, Yaroub Elloumi, Rostom Kachouri. Detection of Retinal Abnormalities in Fundus Image Using CNN Deep Learning Networks. Elsevier. State of the Art in Neural Networks, 1, Ayman S. El-Baz; Jasjit S. Suri, inPress. ffhal-02428351f