Classification and Detection of Eye Diseases Using Feature Engineering and Image Processing

Shikhar Srivastava
Computer Science Department
Western University
London, Ontario
ssriva25@uwo.ca

Abstract— **The classification of eye diseases, including Normal, Diabetic Retinopathy, Cataract, and Glaucoma, plays a crucial role in early diagnosis and treatment planning. This study presents a comprehensive approach to automate the classification process using image processing, machine learning, and deep learning techniques. Various features are extracted from the images, including entropy, moments, and region properties, to characterize different aspects of the eye conditions. Machine learning algorithms such as Random Forest classifier, Logistic Regression, and multilayer perceptron models are employed to predict the class labels based on the extracted features. Furthermore, deep learning models, particularly two Convolutional Neural Network (ResNet – 18 and a customized CNN), are applied directly to the image data for classification. The effectiveness of different methods is assessed in terms of accuracy, sensitivity, and specificity, providing valuable insights into the performance of each approach. The proposed system not only facilitates accurate classification of eye diseases but also holds potential for aiding healthcare professionals in identifying infected cells promptly, which can significantly impact disease management and patient outcomes. This research contributes to the advancement of machine learning and deep learning techniques in medical image analysis, offering a promising avenue for early disease detection and intervention.**

Keywords— Feature Extraction, Machine Learning, RBC, Image Processing

I. INTRODUCTION

This study addresses the crucial task of classifying various eye conditions using a multifaceted approach that integrates image processing, machine learning, and deep neural networks. The dataset under consideration consists of a diverse collection of retinal images, encompassing classes such as Normal, Diabetic Retinopathy, Cataract, and Glaucoma, each comprising approximately 1000 images.

The proposed system begins by converting the RGB retinal images into 2-dimensional grayscale representations to facilitate the subsequent feature extraction process. Utilizing a range of filters, including but not limited to those for Entropy, Shannon-Entropy, moments, and region properties, the system extracts rich and discriminative features from the retinal images. This feature extraction stage is pivotal in capturing relevant information pertinent to the different eye conditions present in the dataset. Subsequently, these extracted features are harnessed within the framework of various machine learning algorithms, including Random Forest, Logistic Regression, and multi-layer perceptron models, for binary classification. Leveraging the distinctive characteristics of each algorithm, the system endeavors to discern patterns within the feature space that delineate the different eye conditions with a high degree of accuracy.

Notably, the system attains commendable performance metrics, achieving accuracy rates of 82.86%, 79.68%, and 83.26% for the Random Forest, Logistic Regression, and multi-layer perceptron model, respectively. Furthermore, to enhance the classification prowess, a Convolutional Neural Network Res-Net 18 is incorporated, yielding an accuracy of 84.20% and a custom-CNN yielding an accuracy of 94.31% in classifying the retinal images. The system's utility extends to providing valuable insights into the presence of eye conditions within the retinal images, thereby facilitating early disease detection and diagnosis. This automated approach, which seamlessly integrates image processing techniques with sophisticated machine learning algorithms, represents a significant advancement in the realm of ophthalmological diagnostics.

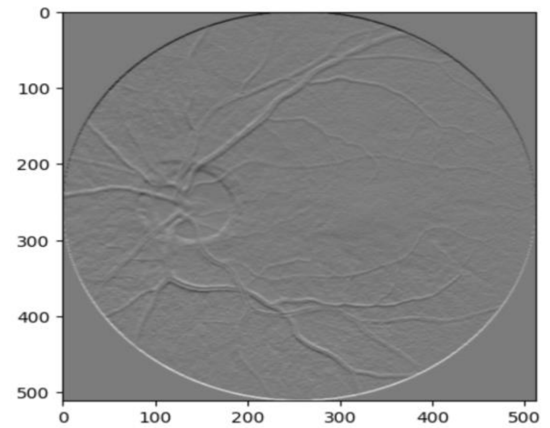


Fig. 1. Image of an eye after applying 2D-Convulution in Greyscale

Moreover, owing to its portability and user-friendly design, the proposed system holds promise for widespread deployment in healthcare settings, offering clinicians and medical practitioners a reliable and efficient tool for expeditious and accurate diagnosis of various eye diseases. Thus, the amalgamation of image processing, artificial intelligence, and machine learning techniques presents a compelling solution to the intricate task of eye condition classification, with profound implications for improving patient care and treatment outcomes in the field of ophthalmology..

II. Related Work

In the realm of image processing and analysis, a multitude of researchers have made noteworthy contributions, each bringing forth innovative methodologies and insights to advance the field of computer vision. Among these pioneers, Robert Haralick's seminal work [1] stands out, focusing on the extraction and utilization of textual features for image classification. Through a meticulous exploration of texture and tone, Haralick elucidates the intricate relationship between these features and their significance in delineating shapes, objects, and regions within images. By leveraging metrics such as entropy, inverse difference moments, and sum average, Haralick's framework empowers image classification systems with the ability to discern subtle nuances and patterns, thereby enhancing interpretive capabilities.

Building upon this foundation, Huang Z & Leng J [2] delve into the domain of image scaling, employing the analytical prowess of Hu Moments Invariants to unravel the complexities inherent in this process. With a keen focus on capturing the intrinsic characteristics of patterns within images, their research sheds light on the robustness and adaptability of Hu Moments Invariants, which remain steadfast in the face of variations in size, orientation, and projection. Furthermore, their findings underscore the dynamic interplay between Hu moments invariants and spatial resolution, offering valuable insights into the mechanisms governing image scaling dynamics. As such, these invariant moments emerge as indispensable tools for conducting comprehensive analyses of image scaling phenomena, with implications spanning diverse applications.

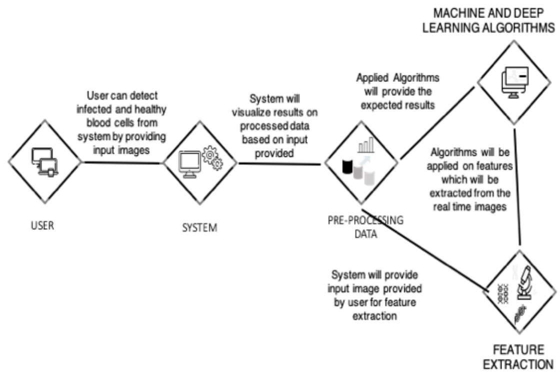
In a complementary endeavor, Chander Bhuvan [5] delves into the realm of image preprocessing and segmentation, recognizing the critical role of noise reduction in enhancing segmentation accuracy. Leveraging the adaptive capabilities of median filtering techniques, Bhuvan's research seeks to mitigate the deleterious effects of noise interference, thereby laying the groundwork for more precise segmentation outcomes. By integrating advanced thresholding methodologies such as Otsu and Zach thresholds, Bhuvan's framework achieves segmentation accuracy comparable to ground truth data, albeit with a nuanced understanding of the scalability challenges inherent in datasets of varying sizes. Thus, Bhuvan's research underscores the importance of methodological refinements in optimizing segmentation efficacy, with potential ramifications for a myriad of real-world applications.

In a parallel trajectory, Kaiming, Xiangyu [3], spearheading research efforts under the aegis of the Microsoft research team, chart new frontiers in deep learning architecture with the introduction of a deeper residual network. By meticulously reformulating network layers to address the pervasive issue of degradation, their research heralds a paradigm shift in the realm of deep neural networks. Through the optimization of residual layers, the proposed Res-Net model not only achieves heightened accuracy but also demonstrates remarkable robustness in handling increasingly complex classification tasks. As such, the Res-Net model emerges as a cornerstone in the field of image classification, serving as a benchmark against which other classification algorithms are evaluated. In essence, Kaiming and Xiangyu's research epitomizes the relentless pursuit of excellence in machine learning, with far-reaching implications for advancing the frontiers of computer vision and beyond.

III. Proposed Framework

The proposed framework is designed to facilitate the detection and classification of images containing healthy or infected red blood cells based on user-provided input images. Users interact with the system via a web application interface, through which they submit image data for classification. Upon receiving input data, the system initiates a series of processes aimed at preprocessing the images and extracting relevant features for classification. Initially, the system preprocesses the images by removing unwanted areas and converting them from RGB to 2-channel grayscale images. This preprocessing step is crucial for enhancing the quality of input data and facilitating more accurate classification outcomes. The preprocessed images are then forwarded to the classification models for analysis.

Two distinct classification approaches are employed within the proposed framework. First, the preprocessed images are fed into a CNN models, a deep learning convolutional algorithm capable of identifying the type of image (infected or healthy cell) with high accuracy. Additionally, the images undergo feature extraction, after which they are subjected to classification using random



forest classifier and multi-layer perceptron (MLP) models.

Fig2. The Proposed System Architecture

The internal workings of the system leverage image processing, machine learning, and deep neural networks to achieve accurate classification results. By integrating these technologies, the system can effectively predict whether an image depicts an infected or healthy red blood cell. This predictive capability is essential for early detection and diagnosis of blood cell infections, enabling timely intervention and treatment.

The proposed system architecture, as depicted in Fig. 2, illustrates the various components and processes involved in image classification. The system's effectiveness is evaluated based on metrics such as accuracy, sensitivity, and specificity, ensuring robust performance across diverse datasets and scenarios. Through this comprehensive approach, the proposed framework aims to enhance the efficiency and accuracy of red blood cell classification, thereby contributing to improved healthcare outcomes and patient care.

A. Dataset

The dataset, sourced from various repositories including IDRiD, HRF, Oculur Recognition, retinal_dataset, and DRIVE, is available on Kaggle. It comprises approximately 1000 images for each of the four classes: Normal, Diabetic Retinopathy, Cataract, and Glaucoma. This dataset serves as a valuable resource for research and analysis in ophthalmology. Researchers and practitioners can access the dataset on Kaggle via the following link:

https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification

Through access to diverse retinal images, this dataset enables the development and evaluation of machine learning and deep learning methods for automated classification and diagnosis of retinal diseases, ultimately enhancing patient care and clinical decision-making in ophthalmology.

B. Feature Extraction

Various features have been extracted from the dataset, each contributing uniquely to the classification process. These features encompass a range of characteristics derived from the images, including Moments, Hu Moments, Entropy, Shannon-Entropy, and Region Properties. Each of these features provides valuable information that aids in the classification of retinal images into their respective categories. Moments and Hu Moments capture the shape and texture properties of the images, while Entropy and Shannon-Entropy quantify the randomness and complexity within the images. Additionally, Region Properties describe various attributes of distinct regions within the images, further enhancing the discriminative power of the classification model. By leveraging this diverse set of features, the classification algorithm can effectively differentiate between different retinal conditions, enabling accurate diagnosis and treatment planning in the field of ophthalmology.

a) Entropy

Entropy quantifies information weight in data distributions, increasing with data complexity, pivotal for discerning patterns in retinal image classification.

Entropy = $\sum_i p(k)\ log_2(p(k))$

b) Eccentricity

Eccentricity indicates the ratio of major axis to minor axis in plots, ranging from 0 to 1.

Eccentricity = a / b

c) Moments

Moments, denoted by μ., are employed for interpretation and calculating specific weighted averages.

$$\mu_{m,n} = \iint (a\text{-}\ c_x)^m\ (b\text{-}c_y)^n\ f\ (x,y)\ dy\,dx$$

d) Hu Moments

Hu Moments, stemming from normalized central moments, yield invariant descriptors comprising seven moment invariants essential for pattern recognition.

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20}\text{-}\ \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30}\text{-}\ 3\eta_{12})^2 + (3\eta_{21}\text{-}\ \mu_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2$$
$$\phi_5 = (\eta_{30}\text{-}\ 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ -3(\eta_{21} + \eta_{03})^2] + (3\eta_{21}\text{-}\ \eta_{03})(\eta_{21} + \eta_{03}) \\ [3(\eta_{30} + \eta_{12})^2\text{-}(\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20}\text{-}\ \eta_{02})[(\eta_{30} + \eta_{12})^2\text{-}(\eta_{21} + \eta_{03})^2] \\ +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21}\text{-}\ \eta_{03})(\eta_{30} + \eta_{12}) \\ [(\eta_{30} + \eta_{12})^2\text{-}3(\eta_{21} + \eta_{03})^2] \\ \text{-}(\eta_{30}\text{-}\ 3\eta_{12})(\eta_{21} + \eta_{03}) \\ [3(\eta_{30} + \eta_{12})^2\text{-}(\eta_{21} + \eta_{03})^2]$$

e) Energy

Energy derives from pixel intensity within an image.

Energy = $\sum_i [p(i)^2]$

f) Mean Area

The mean area is computed as the average of all pixels within the image..

Mean = $\sum_i i.p(i)$

g) Standard deviation Area

The standard deviation of area in an image quantifies variations in pixel intensity.

Standard deviation area = $\sqrt{\sum_i (i\text{-}\mu)^2 p(i)}$

C. Classification Algorithms on Feature Extraction

a) Random Forest Classifier

Random Forest classifier is a powerful machine learning algorithm that extends the capabilities of decision trees by leveraging ensemble learning. By constructing an ensemble of decision trees, Random Forest can handle complex classification tasks more effectively. Each decision tree in the ensemble is trained on a random subset of the features and data instances, reducing the risk of overfitting and improving generalization performance. During training, each decision tree independently learns to classify the data based on a subset of features and instances. This diversity among decision trees allows the Random Forest to capture different aspects of the underlying data distribution. When making predictions, each decision tree in the ensemble contributes its classification decision through a voting mechanism. The final prediction is determined by aggregating the individual decisions of all decision trees in the forest. One of the key advantages of Random Forest is its ability to handle high-dimensional data with complex relationships between features. Additionally, Random Forest is robust to noisy data and outliers, making it suitable for a wide range of real-world applications. Moreover, Random Forest requires minimal hyperparameter tuning and is less prone to overfitting compared to individual decision trees.

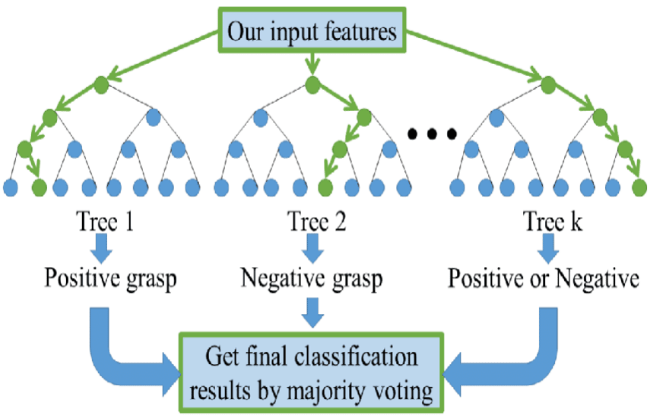


Fig. 2. Working Structure of Random Forest Classifier Algorithm

b) Logistic Regression

Logistic Regression is utilized for predicting discrete dependent variables, often in binary format. It employs the Sigmoid Function, also known as the logistic function, which follows an S-shaped curve, facilitating classification problem-solving. The output of the logistic function yields probabilities ranging from 0 to 1, corresponding to the input data provided. The formula for the Sigmoid Function is::

$$y = \frac{1}{1 + e^{-(wx+b)}}$$

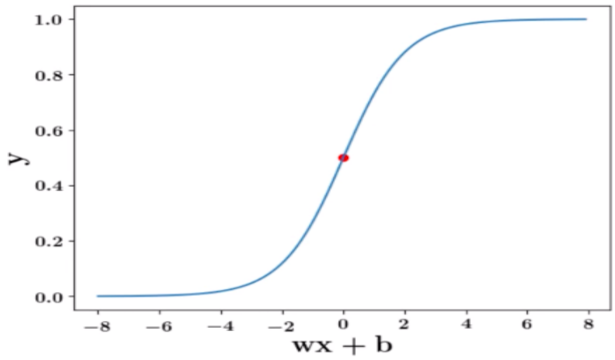


Fig. 3. Plot of Sigmoid Function

c) Multilayer Perceptron Model

A multi-layer perceptron model is a type of neural network. The backpropagation training algorithm is pressed. In this model there are 3 layers, in the order of 100, 50 and 10. The final layer or output layer will perform the classification using softmax. The algorithm is performed by taking input values as x, and each x value is multiplied by its weight w and added to a bias value b, and the overall neuron has an activation function. The activation function value is sent to the sigmoid function to get some output for each neuron. Similarly, many neurons are connected in the network, and each neuron performs in the same way. The multi-layer perceptron model can be used with the backpropagation algorithm in which the value of each weight can be changed on each epoch by differentiation concerning each term. Adam, a stochastic gradient-based optimizer is used for the model to minimize loss and sigmoid activation is used throughout layers.

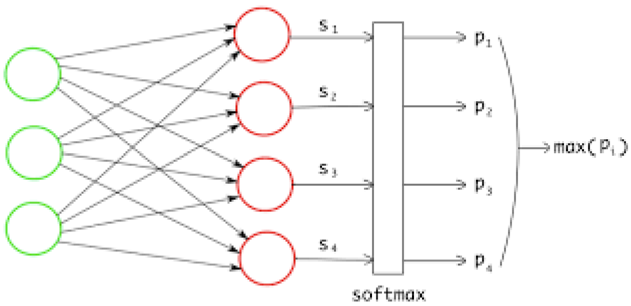


Fig. 4. Final layer of softmax function in our Multi-Layer Perceptron Model Architecture, the maximum of 4 probabilities is chosen as the predicted label

D. Classification Algorithm on Image Data

a) Residual Neural Network (ResNet – 18)

Residual Residual Neural Networks (ResNets) are advanced deep learning architectures designed to address the challenges of training very deep neural networks. They achieve this by introducing skip connections, or shortcuts, that allow the gradient to flow directly through the network without being diminished by passing through every layer. In essence, ResNets enable the network to learn residual functions, which are the differences between the original input and the desired output. By learning these residual functions, ResNets can effectively train deep networks with hundreds or even thousands of layers without encountering the vanishing gradient problem, allowing ResNets to learn complex features at different levels of abstraction, leading to superior performance on various tasks such as image classification, object detection, and semantic segmentation. To make use of this, we trained ResNet on 3000 transformed training images representing all classes of eye disorders equally.

b) Customized Convolutional Neural Network (Custom-CNN)

The Custom CNN architecture is a convolutional neural network (CNN) designed for image classification tasks. It consists of three convolutional layers followed by max-pooling layers for feature extraction and spatial down-sampling. The output from the convolutional layers is passed through fully connected layers for classification, with ReLU activation functions applied throughout the network to introduce non-linearity. A dropout layer with a dropout probability of 0.5 is included after the first fully connected layer to prevent overfitting. This architecture follows a standard design pattern for image classification, leveraging convolutional and pooling layers for feature extraction and fully connected layers for classification. The combination of convolutional and fully connected layers, along with activation functions and dropout regularization, allows the network to learn complex patterns in the data and generalize well to unseen examples.

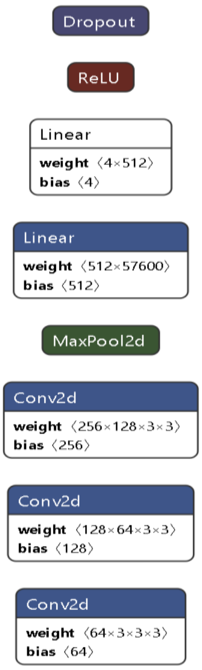


Fig. 5 Architecture of Custom-CNN.

IV. Observations and Results

The Random Forest classifier, Logistic Regression and Multi-layer Perceptron model has been implemented on the extracted features from both the classes infected and healthy to perform binary classification. The sample images have been taken to extract the features. The classes have been labelled as normal=1, diabetic retinopathy=2, cataract=3and glaucoma=4 during the feature extraction and creation of dataframe.

A. *Preprocessing*

For Observations, A sample of thousand images has been taken from each of the four classes for the feature extraction to perform the multi-class classification. Then conversion has been made from RGB to 2 channel grayscale images to extract features in-depth.

a) Normal Eye in RGB *b) Normal Eye in Grayscale*

Fig, 6 Cell images before and after preprocessing.

B. *Classification Algorithms result*

The performance is measured for the model in terms of accuracy, and can be measured further with metrics such as sensitivity, and specificity as an evaluation for each class separately using the terms TP: true positive, TN: true negative, FP: false positive and FN: false negative. Specific formulas are as follows:

Accuracy=TP+TN/TP+FP+TN+FN

Sensitivity, also called the true positive rate, determines the percentage of accurately identified disease subjects by the equation:

Sensitivity=TP/TP+FN

Specificity, also called the true negative rate, determines the percentage of accurately identified healthy subjects by the equation: Specificity=TN/TN+FP

a) Random Forest Classifier

The random forest classification has been applied to the features to predict the type. The classifier has been trained and provides the prediction with an accuracy of 82.86% on the testing data(20% split of original data).

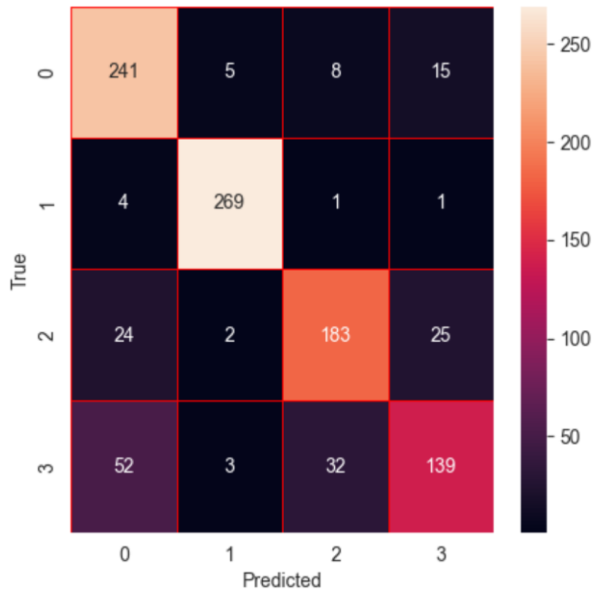


Fig 7. Confusion Matrix resulted from Random Forest Classifier

b) Logistic Regression

Logistic Regression is executed on the extracted features of the four classes. The algorithm shows an accuracy of 79.68%. In Logistic Regression, the extracted features have been considered as input to predict the class using the sigmoid function. The sigmoid function will give the output in the range of 0 to 1 to perform the binary classification.

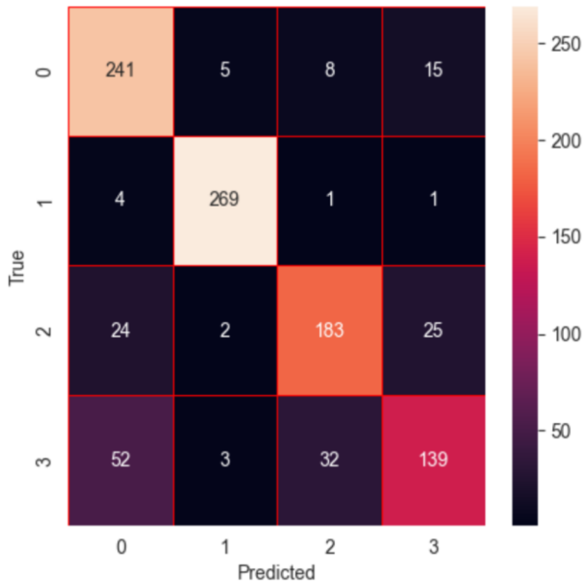


Fig. 8 Confusion Matrix Resulted from Logistic Regression

c) Multi-Layer Perceptron Model

Multi-layer Perceptron Model has been implemented on the extracted features. The model converged after 459 iterations. At tolerance of 0.0001. The model shows 83.2% accuracy, and generated a confusion matrix on testing set shown in Fig 9. The input layer in the model will accept the features as x value, and the final output will be the provided sample belongs to one of the four classes. The loss plot for the training is shown in the Figure 10, training is stopped as soon as loss stops decreasing for 10 epochs.

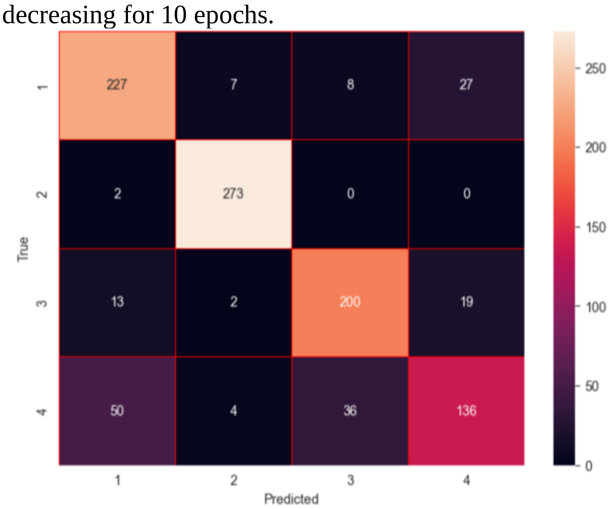


Fig. 9 Confusion Matrix resulted from Multi-Layer Perceptron Model

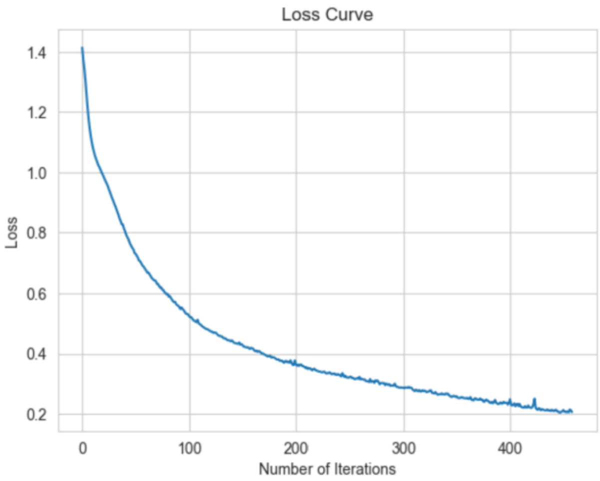


Fig. 10 Loss Plot of Multi Layer Perceptron Model

d) Residual Neural Network (ResNet 18)

The deep Residual learning model ResNet 18 has been trained on images from the training split of 75% of total data. The network can predict the class with an accuracy of 83.16 percent on the leftover testing split. Training for random forest classifier and multi-layer perceptron model is done on extracted features. Preparation for the deep residual network(Resnet) has been done on image dataset directly to predict the class type, the loss plot for training is in Figure 12. Upon testing Res-Net algorithm, output layer is able to predict which class the image belongs to, and the following confusion matrix resulted from testing is shown below in Figure 11.

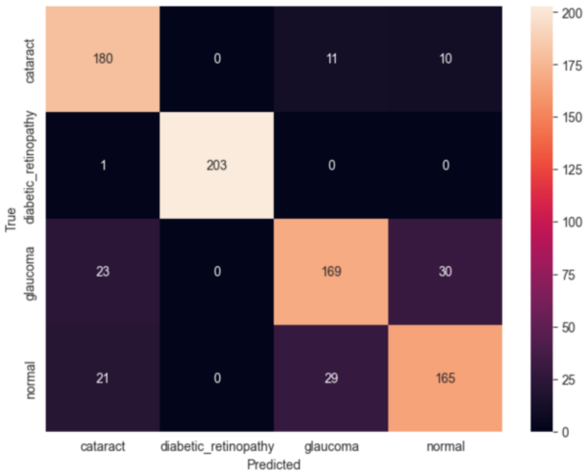


Fig 11. Result of Resnet-18 Confusion Matrix

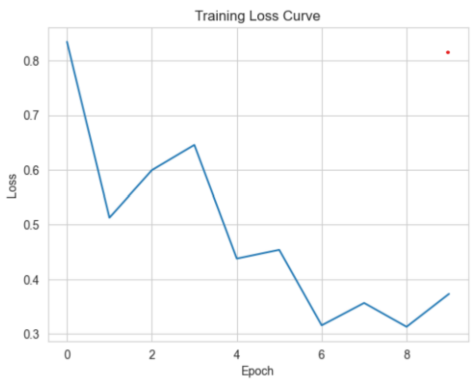


Fig 12. Confusion Matrix resulted from Resnet-18 model

e) Customized Covulutional Neural Network

Custom-CNN model has been trained on images from the training split of 75% of total data. The network can predict the class with an accuracy of 94.31 percent on the leftover testing split. Training for the Custom-CNN has been done on image dataset directly to predict the class type, the loss plot for training is in Figure 14. Upon testing Custom-CNN algorithm, output layer is able to predict which class the image belongs to, and the following confusion matrix resulted from testing is shown below in Figure 13.

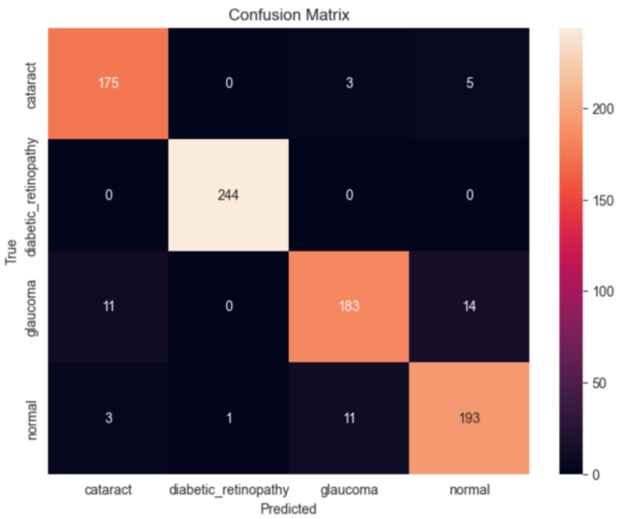


Fig 13. Confusion Matrix resulted from Custom-CNN model

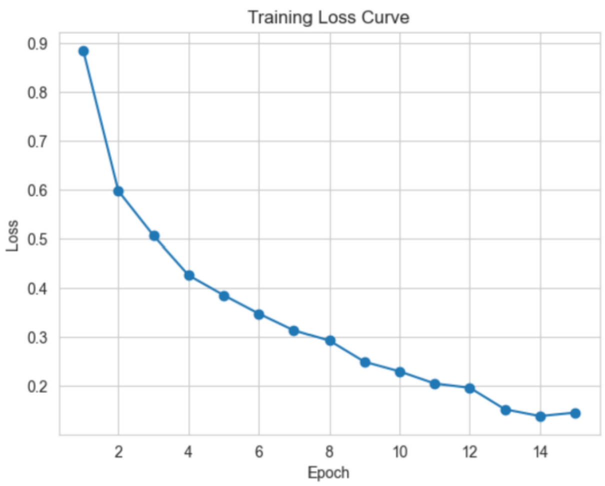


Fig 14. Loss Plot of Custom-CNN model

C. *Comparision of Implemented Algorithms*

The comparison has been made concerning the accuracy, in which the Custom-CNN shows the best results with a much higher accuracy as compared to the other four algorithms. Res-Net algorithm is efficient and can be considered for the classification of images with raw data however, it doesn't show much or any improvement over the classification methods such as random forest and Multi-Layer Perceptron model performed on extracted features(all very close to 83%). Logistic Regression is showing least accuracy of all at 79.6%. The application of different algorithms on image data directly shows high accuracy as compared to implementing algorithm on extracted data however it does require more computational power and time. The classification algoritms trained on extracted feature take close to 2 minutes to train while the convolutional neural networks took close to 15 minutes to train. The proposed feature set can be considered to implement the algorithms for the classification of images where saving computational energy and time is a priority. Random Forest Classifier is fast and has very low overfitting of data that makes the algorithm best and should be considered. The multi-layer perceptron model shows very less errors with the better results as it has been trained on extracted features and can perform better on extensive data. Fig 15. shows the performance of implemented algorithms with their respective accuracies. Further comparison can we drawn between performance of algorithms by calculating metrics such as specificity, sensitivity and F1-score for each of the classes or specific classes of interest. One specific example of subset of interest in these classes could be infected eyes and non-infected eyes.

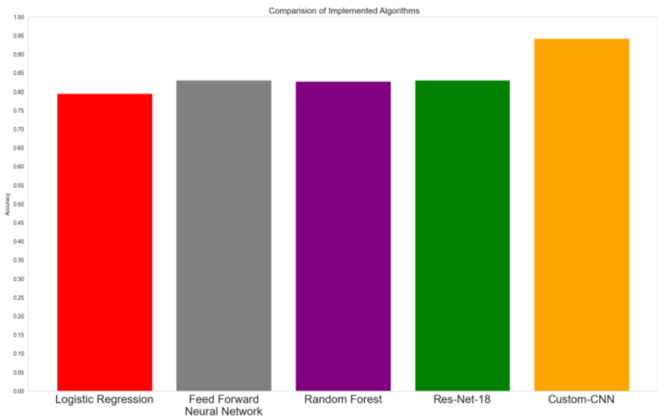


Fig 15. Comparision Plot of implemented Algorithms based on Accuracy

D. *Deployment of model as a web Application*

The Custom-CNN model has been deployed as a web application using Flask. The Custom-CNN algorithm has been trained on the complete image dataset and loaded as a model into a file to deploy the web application. The application works by taking input data as an image and classify the data using the loaded model into the app. The preprocessing gets performed on the input image, and then the model predicts the class of the input image. The result can be seen on the web application after user uploads the image. A message prints notifying user of the class type. The model predicts the class with an accuracy of 94.31%. The working and results of application are also provided in Figure 16 and Figure 17.

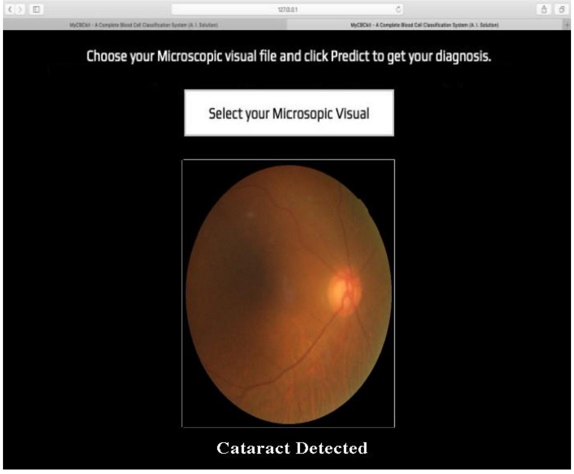


Fig 16. Deployment of Model on a Web Applicalication which correctly predicts the input image as an eye infected with Cataract.

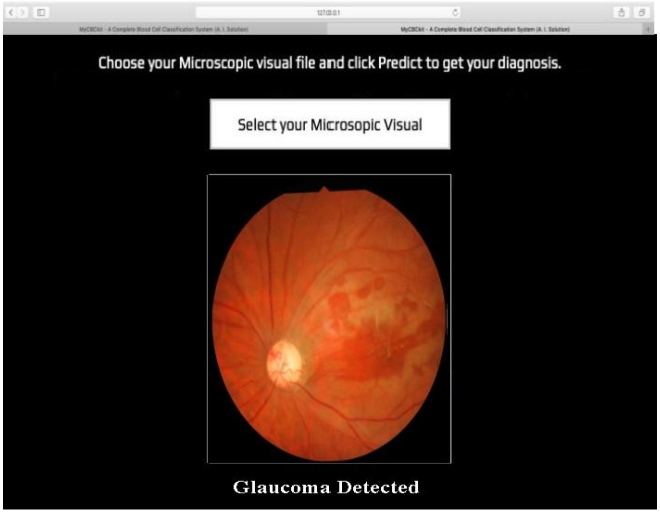


Fig 17. Deployment of Model on a Web Applicalication which correctly predicts the input image as an eye infected with Glaucoma.

V. CONCLUSION

This study has presented the classification of infected eyes using a set of different features and applications of random forest classifiers, Logistic Regression and multi-layer perceptron models to predict which type of infection the eye has. Our study can be summarised as follows: 1. preprocessing of images by removing unwanted areas, focusing on the cells, converting RGB to grayscale images, 2.1 extracting features from the preprocessed images, and applied random forest classifier, Logistic Regression and multi-layer perceptron model. 2.2 Applying Res-Net, a convolutional neural network algorithm and a customized CNN to predict the type of images by training the algorithm on preprocessed image data directly. 3 The comparison has been made concerning the accuracy in which the Customized CNN model shows the best results with high accuracy and can be applied to large data as compared to the other algorithms. 4 Deployment of the model as a web application to perform the classification. The efficiency of other algorithms is also considerable, however training of deep learning models on image directly can take high time compared to the other algorithms as it takes image data as input. The proposed feature set used in our feature extraction process can be considered to implement the algorithms for the classification of images where saving computational energy and time is a priority in future studies.

This proposed model can help the healthcare system to automate the process of detecting which type of infection is damaging the eye. This model can also prove to be very beneficial in regions with shortage of eye specialists and good doctors. Implementing the model can also streamline processes in hospitals with long patient queues, potentially reducing waiting times and enhancing overall efficiency and reducing human errors.

VI. REFERENCES

[1] Robert M. Haralick, K. Shanmugam, and Itshak Dinsten, “Texture Features for Image Classification” Institute of Electrical and Electronics Engineers, Inc, vol. SMC-3, pp. 610–621, Nov 1973.

[2] Zhihu Huang, Jinsong Leng, “Analysis of Hu’s Moments Invariants on Image Scaling and Rotation” Cowan University, pp. 476–480, 2010.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun ”Deep Residual Learning for Image Recoginition” Microsoft Research, arXiv:1512.03385v1, 2015

[4] Rajivkumar Mente, B V Dhandra, Gururaj Mukarambi “Image Recognition Using Shape Descritpor: Eccentricity and color” Solapur University, pp. 2277–7830, 2014

[5] Chander Bhuvan, Shivansh Bansal, Rinki Gupta, AnupamaBhan “Computer based diagnosis of malaria in thin blood smears using thresholding based approach” Amity University, 2020

[6] Simon Xinmeng Liao, “Image Analysis by moments” The Department of Electrical and Computer Engineering, The University of Manitoba, Canada, 1993.

[7] Luis B Almedia ”Multilayer perceptron”, Handbook of Neural Computation, Oxford University Press, 1997

[8] Gautam, Anjali, Bhadauria, Harvindra “Classification of White Blood Cells Based on Morphological Features” IEEE, pp. 2363–2368, 2014

[9] M. Sajjad, S. Khan, Z. Jan, K. Muhammad “Leukocytes Classification and Segmentation in Microscopic Blood Smear: A Resource-Aware Healthcare Service in Smart Cities”, IEEE, pp. 3475–3489, 2017

[10] R. Ahasan, A.U. Ratul, A.S.M. Bakibillah, “White blood cells nucleus segmentation from microscopic images of strained peripheral blood film during leukemia and normal condition” ICIEV,pp. 361-366,2016

[11] Vitalflux. (n.d.). Keras Multi-class Classification using Iris Dataset. Retrieved from https://vitalflux.com/keras-multi-class-classification-using-iris-dataset/

[12] Netron. (n.d.). Retrieved from https://netron.app/

[13] Grammarly. (n.d.). Retrieved from https://www.grammarly.com/