Diabetic Retinopathy Detection

Simran Feroz *Department. of MCA, Ramaiah Institute of Technology,* Bangalore, India.

Dr. Madhu Bhan *Department of MCA, Ramaiah Institute of Technology,* Bangalore, India.

Abstract

Diabetic Retinopathy is a serious complication of diabetes that can lead to vision loss or blindness. Early identification and treatment are essential for forestalling vision misfortune. Early detection of Diabetic Retinopathy through an automated system is more beneficial over the manual detection. This study develops a convolutional neural network model to identify diabetic retinopathy. The chosen architecture is Residual Network. Rectified Linear Unit activation functions are used to introduce non-linearity and increase the model's expressive power. The Residual block helps in overcoming the vanishing gradient problem, allowing the model to be trained more effectively and efficiently. To reduce the spatial dimensions of the feature maps and capture essential information, the Maxpool2d operation is applied. Soft-max activation is used for the final classification, which converts the raw predictions of the model into probability scores, making it easier to interpret the results and identify the appropriate class. A dataset of fundus images is used to train the model, and it is assessed using common metrics like accuracy. The outcomes demonstrate that the Convolution Neural Network model outperforms other machine learning methods and is highly successful at detecting diabetic retinopathy. The proposed convolution neural network model might help users make a quicker and more accurate diagnosis of diabetic retinopathy. Further our model is used to do analysis at various stages of diabetic retinopathy. This is a technical work that takes test data to detect the disease and checks for the accuracy of the result.

Keywords—Convolutional Neural Network, Deep Learning, Machine Learning, Diabetic Retinopathy, Detection.

I. INTRODUCTION

The elevated blood sugar levels in diabetes patients can be attributed to two factors. The first is that the body frequently fails to create enough insulin. The second cause can be that the cells do not react to the generated insulin. Diabetes is one of the leading causes of blindness in the modern world. High levels of glucose in the retina's tiny blood vessels cause impaired vision, which can eventually lead to total blindness. The term used to describe such a condition is diabetic retinopathy. Diabetic Retinopathy (DR) is prevalent and potentially sight-threatening [1]. Understanding of the pathophysiology, risk factors, and early detection methods of diabetic retinopathy is crucial for effective management and timely intervention to prevent irreversible vision loss. This research paper aims to investigate the current knowledge and advancements in the field of diabetic retinopathy, with a particular focus on the development and evaluation of diagnostic techniques for its accurate and early detection [2,3]. Deep learning methods such as convolutional neural networks (CNNs) have shown promising results in detecting DR from retinal images. This work proposes a CNN-based approach for automated DR detection and classification as shown in Figure 1. This work is carried out on a large dataset of 3006 retinal where each image is classified and stored under its respective folders namely mild_DR, moderate

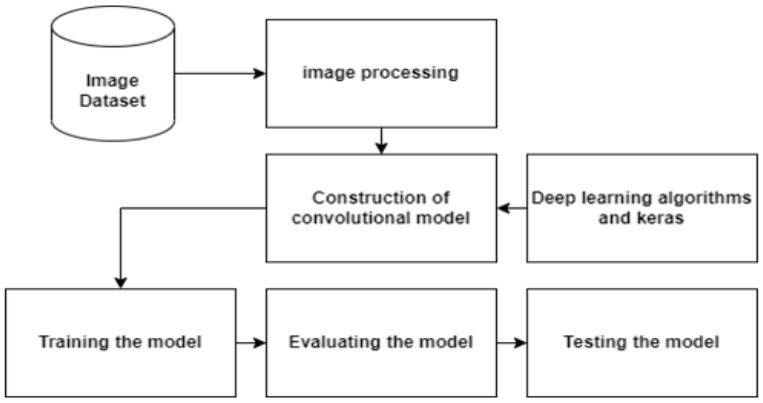


Figure 1. Convolutional Neural Network

_DR, No_DR, Proliferate_DR, and Severe_DR [4]. The results demonstrate that the proposed CNN-based approach achieves high accuracy in DR detection and classification, with a good potential for use in clinical settings to improve improve patient outcomes. The aim of this model is to develop a deep learning-based model for the detection of Diabetic Retinopathy using CNN. The algorithm is trained on a sizable dataset of retinal pictures, and it can correctly detect the existence and severity of diabetic retinopathy. As a result, a deep learning-based model for diagnosing diabetic retinopathy was created, which can precisely determine the presence and degree of the eye ailment. The study model should be able to offer rapid, precise, and affordable screening for diabetic retinopathy, making it a useful resource for patients and healthcare professionals.

II. LITERATURE SURVEY

Today, everyone is attempting to integrate machine learning and artificial intelligence technologies with a variety of other industries, including gaming, movies, and healthcare. Numerous scholars concur that artificial intelligence and machine learning are more effective in today's educational, medical, and many other disciplines [3]. The author of [4] describes a project that uses deep learning, more specifically the densely connected convolutional network DenseNet-169, to aid in the early diagnosis of diabetic retinopathy. It was also found that using DenseNet concatenation causes a new problem of requiring high GPU memory and more training time [5]. One of the difficulties with deep learning classification tasks, mainly in the medical field, is a lack of labelled training data. Utilising a small training dataset, transfer learning aids in deep learning model training. Transfer learning can be used to address the issue of Deep Convolutional Networks not having enough training data. Transfer learning is only effective for problem-solving in small datasets; it is ineffective for categorization in large datasets.. In [6, 7] Shape-based, texture-based, and statistical features were found to be the most discriminative when it came to DR detection. When compared to other machine learning classifiers, the Artificial Neural Network was demonstrated to be the superior classifier. A machine learning ensemble model made up of the Random Forest classifier, Adaboost classifier, Decision Tree classifier, K-Nearest Neighbour classifier, and Logistic Regression classifier was put to the test [8]. The min-max normalisation method was used to initially normalise the dataset for diabetic retinopathy. The suggested ensemble model was then trained using this normalised dataset. Finally, the performance of the proposed model was assessed in comparison to a variety of machine learning methods. The authors in [9] used Deep Learning and transfer learning algorithms to analyze distinct DR phases in addition to focusing on the identification of diabetic retinopathy. On a large dataset of roughly 3662 train images, CNN, hybrid CNN with ResNet, and hybrid CNN with DenseNet are utilized to automatically determine the stage of DR.

III. CNN MODEL

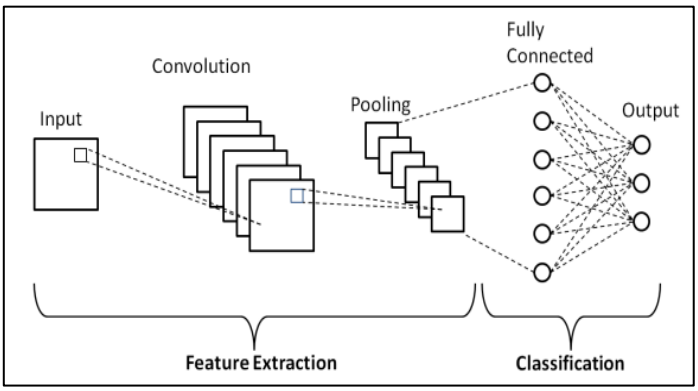


Figure 2. Convolutional Neural Network

CNNs are capable of automatically learning and extracting discriminative features from raw retinal images, enabling accurate and efficient screening [10]. Feature extraction and classification are two main components of diabetic retinopathy using CNN as shown in Figure 2. Non-linear activation functions, such as ReLU, introduce non-linearity into the network, allowing for the learning of more complex and abstract representations. Pooling layers reduce the spatial dimensions of the feature maps, preserving relevant information while reducing computational complexity. Through these processes, the CNN progressively generates higher-level feature maps, capturing global structures and patterns essential for diabetic retinopathy detection. By leveraging these extracted features, healthcare professionals can enhance their ability to accurately classify and diagnose the disease, enabling timely intervention and preventing vision loss. Below mentioned are some of the important features of CNN models:

Performance

Accuracy is the most basic performance metric and is simply the percentage of correctly classified images out of the total number of images in the test dataset. The main agenda of the proposed model is accuracy. The accuracy of the prediction can be increased by increasing the training dataset. Precision and Recall are two complementary metrics that are useful when dealing with imbalanced datasets. Precision refers to the proportion of correctly classified positive cases out of all cases predicted as positive. In the context of diabetic retinopathy detection, precision would measure the accuracy of identifying patients with the disease among those predicted to have it. A high precision score indicates that the neural network is effective at minimizing false positives, reducing the likelihood of misdiagnosing healthy individuals as having diabetic retinopathy. On the other hand, recall, also known as sensitivity or true positive rate, measures the proportion of correctly classified positive cases out of all actual positive cases. In the context of diabetic retinopathy detection, recall evaluates the ability of the neural network to correctly identify patients with the disease among all individuals who truly have it. A high recall score implies that the network is proficient at minimizing false negatives, reducing the risk of missing actual cases of diabetic retinopathy. A confusion matrix is a square matrix where the rows represent the actual labels of the images, and the columns represent the predicted labels of the images. This is a matrix that shows the number of true positive, true negative, false positive, and false negative predictions made by the model [11,12].

Pre-Processing

Pre-processing is the process of getting visual data ready to be fed into a neural network. Pre-processing is primarily used to improve the quality of input data, lessen noise, and aid the neural

network's ability to recognise and extract important characteristics from the images. 80% of the train data and 20% of the test data make up the entire dataset. The rescale pre-processing phase involves dividing each pixel value in the input images by 255 to rescale the pixel values. This aids in converting the pixel values to a range between 0 and 1, which can enhance the model's training.

Data Augmentation

Data augmentation is a crucial technique in training neural networks for diabetic retinopathy detection. It involves generating new training samples by applying various transformations and modifications to the existing dataset. By augmenting the data, the network becomes exposed to a wider range of variations, enhancing its ability to generalize and recognize diverse patterns associated with the disease. Retinal pictures can be enhanced with data to imitate various environmental factors and changes that could exist in real-world situations. These adjustments can be made in terms of rotation, translation, scaling, flipping, brightness, contrast, and noise. By introducing such modifications to the images, the neural network can be trained to become resilient to adjustments in the illumination, image orientations, and other elements that could alter the visibility of retinal defects [13,14].

IV. PROPOSED MODEL

A comprehensive model to assess and evaluate the presence of DR is proposed which contributes for deep feature extraction and image classification of DR fundus images. The important blocks of the proposed architecture are defined as follows:

Resnet

ResNet, a powerful neural network architecture is used for diabetic retinopathy detection. It addresses the challenge of training deep networks by introducing skip connections, allowing the network to learn residual functions. ResNet captures intricate features associated with retinopathy, enabling accurate classification. The first stage of the ResNet model in this architecture is implemented using the res_block function. It consists of a series of convolutional blocks with identity shortcuts. Each convolutional block contains three convolutional layers, and the shortcut connections bypass the convolutional layers to address the vanishing gradient problem. It incorporates convolutional layers, batch normalization, and non-linear activation functions in residual blocks. The skip connections facilitate the flow of information, mitigating the vanishing gradient problem. A general Resnet Architecture is shown in Figure 3.

ReLU

Rectified Linear Unit (ReLUs), is used as activation function in neural networks for the diagnosis of diabetic retinopathy. It adds nonlinearity and aids the network to understand intricate connections between input features. ReLU converts negative numbers to zero while

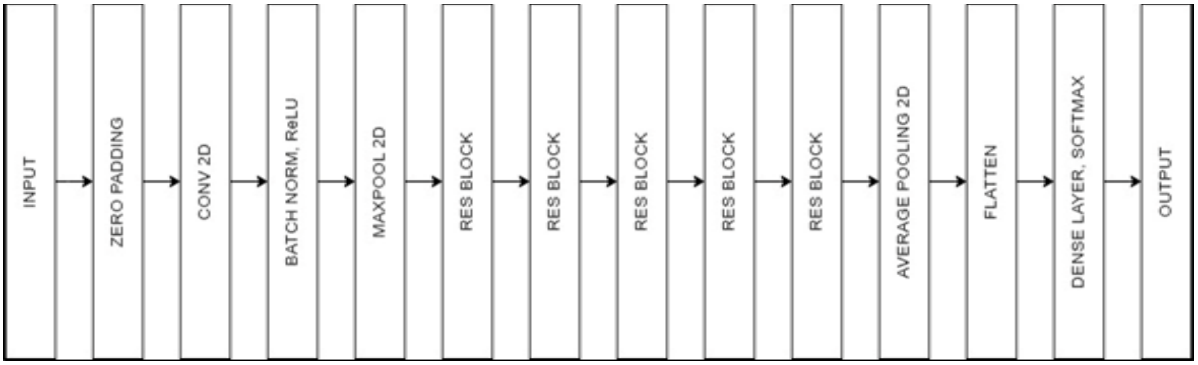


Figure 3. RESNET Architecture

maintaining the integrity of positive values. This activation function helps solve the vanishing gradient issue and is computationally effective. ReLU improves the model's capacity to recognise complex patterns and features related to retinopathy by enabling the network to learn non-linear mappings. ReLU is suitable for deep neural networks, such as those used to detect diabetic retinopathy, because of its simplicity. ReLU's powerful representation learning skills help to increase the accuracy of identifying retinal images and illness diagnosis.

Residual Block

A crucial component of our ResNet model is the residual block which is defined by the res_block function. Each residual block consists of two main paths: the main path and the short path (shortcut connection). The main path starts with a 1x1 Convolutional layer (Conv2D) followed by a MaxPooling2D layer (MaxPool2D). This reduces the spatial dimensions of the input while increasing the number of channels. Batch normalization (BatchNormalization) is applied to normalize the activations of the main path. An activation function (Activation) is applied to introduce non-linearity. Next, a 3x3 Convolutional layer is applied to further extract features. Batch normalization and activation are again applied to the output of the 3x3 Convolutional layer. Finally, a 1x1 Convolutional layer is used to adjust the number of channels to match the desired output. Batch normalization is applied to the output of the 1x1 Convolutional layer. The short path starts with a 1x1 Convolutional layer applied to the original input before any transformations. This is done to match the number of channels in the main path. MaxPooling2D is then applied to reduce the spatial dimensions of the short path. Batch normalization is applied to the output of the MaxPooling2D layer. The output of the main path and the short path are added element-wise using the Add() function. This forms the skip connection, allowing the gradients to flow directly from the outputs to introduce non-linearity. The identity block is used to introduce additional non-linearity and further refine the feature representation. In the provided code, two identity blocks are defined within the res_block function. Each identity block follows a similar structure to the main path in the residual block, including Conv2D BatchNormalization, and Activation layers. The output of the main path is added to the output of the identity block, similar to the skip connection in the main path. The res_block function is called multiple times within the ResNet model to construct the desired depth of the network. By stacking multiple residual blocks, the ResNet model can capture increasingly complex features from the input data.

Maxpool 2D

MaxPooling2D is used in the ResNet model to downsample the feature maps and capture important information at different stages of the network. In this ResNet model architecture, MaxPooling2D is strategically applied at different stages of the network. After the initial convolutional layer (conv1), MaxPooling2D is configured with a pool size of (3,3) and a stride of (2,2). This operation effectively reduces the spatial dimensions of the feature maps by a factor of 2, preserving the most informative features while decreasing computational complexity. Moreover, MaxPooling2D is also utilized after each residual block, further downsampling the feature maps and promoting the extraction of discriminative patterns. This process can be mathematically defined as follows:

$$MaxPooling2D(F)i,j = max\, m,n\{f(i.pool\_size[0]+m),(j.pool\_size[1]+n)\}$$

Where MaxPooling2D$(F)_{i,j}$ denotes the output value at position (i,j) in the downsampled feature map, and F(i.poolsize[0] + m) , (j.poolsize[1] + n) in the original feature map F. The pool_size parameter specifies the dimensions of the pooling window, and the stride parameter can control the step size for traversing the input feature map

Flatten

In the proposed ResNet model for diabetic retinopathy detection, the Flatten layer plays a crucial role in converting the output tensor from the average pooling layer into a suitable format for further processing. Mathematically, the Flatten layer reshapes the 3D tensor T with dimensions (height, width, channels) into a 1D vector v with shape (num_features,), where num_features represents the total number of elements in the original tensor T. The Flatten operation can be mathematically described as follows:

$$v = Flatten(T)$$

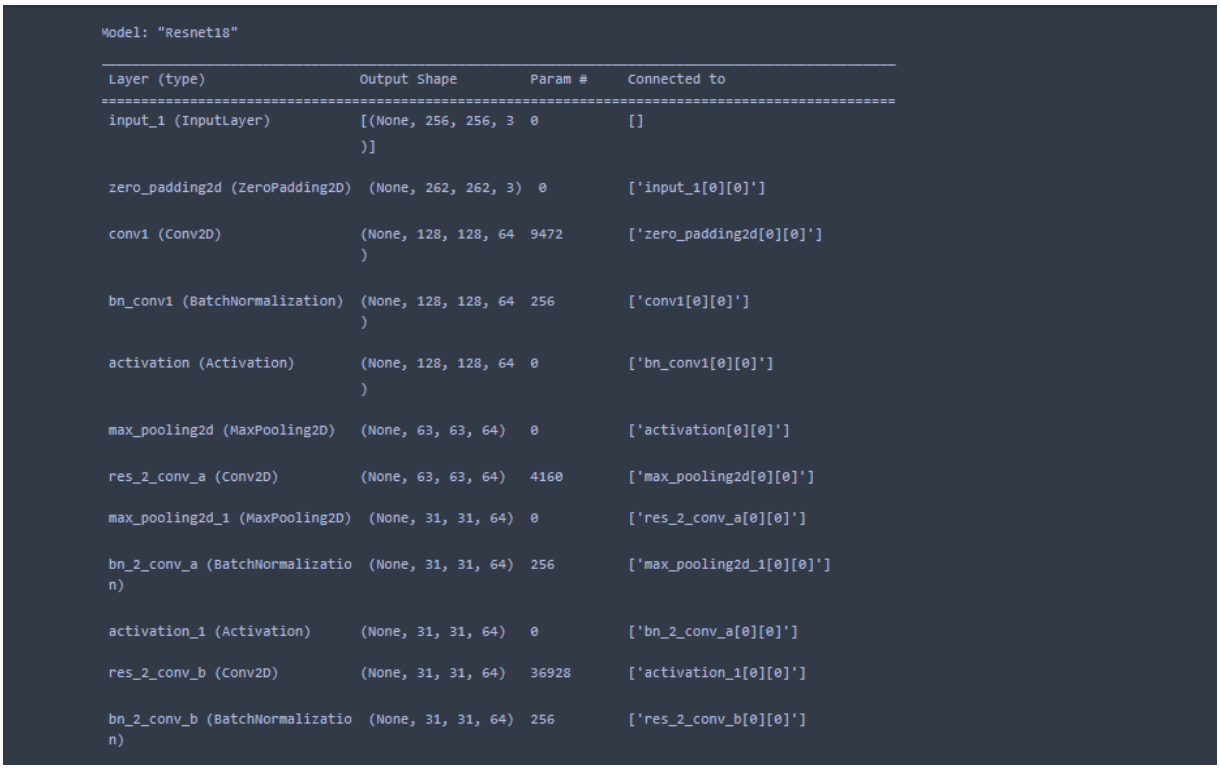
where T represents the output tensor from the preceding average pooling layer, and v is the resulting 1D vector after the Flatten operation. By employing the Flatten layer in the ResNet architecture, we convert the spatially arranged feature map into a linear representation. This enables seamless connection to a dense layer (fully connected layer) for classification or subsequent processing. The flattened representation of the feature map facilitates efficient parameter handling and helps in capturing important patterns for accurate diabetic retinopathy detection.

Softmax

The softmax activation function is applied to the output of the Dense layer called "Dense_final." This layer takes the flattened features from the previous layers and applies a fully connected operation to map them to the final output classes. The number of units in this Dense layer is set to 5, as the model is designed for a classification task with five distinct classes. The softmax function ensures that the output values are non-negative and sum up to 1, representing the predicted probabilities for each class. These probabilities can be interpreted as the model's confidence or belief for each class given the input data. The class with the highest probability is typically chosen as the predicted class label.

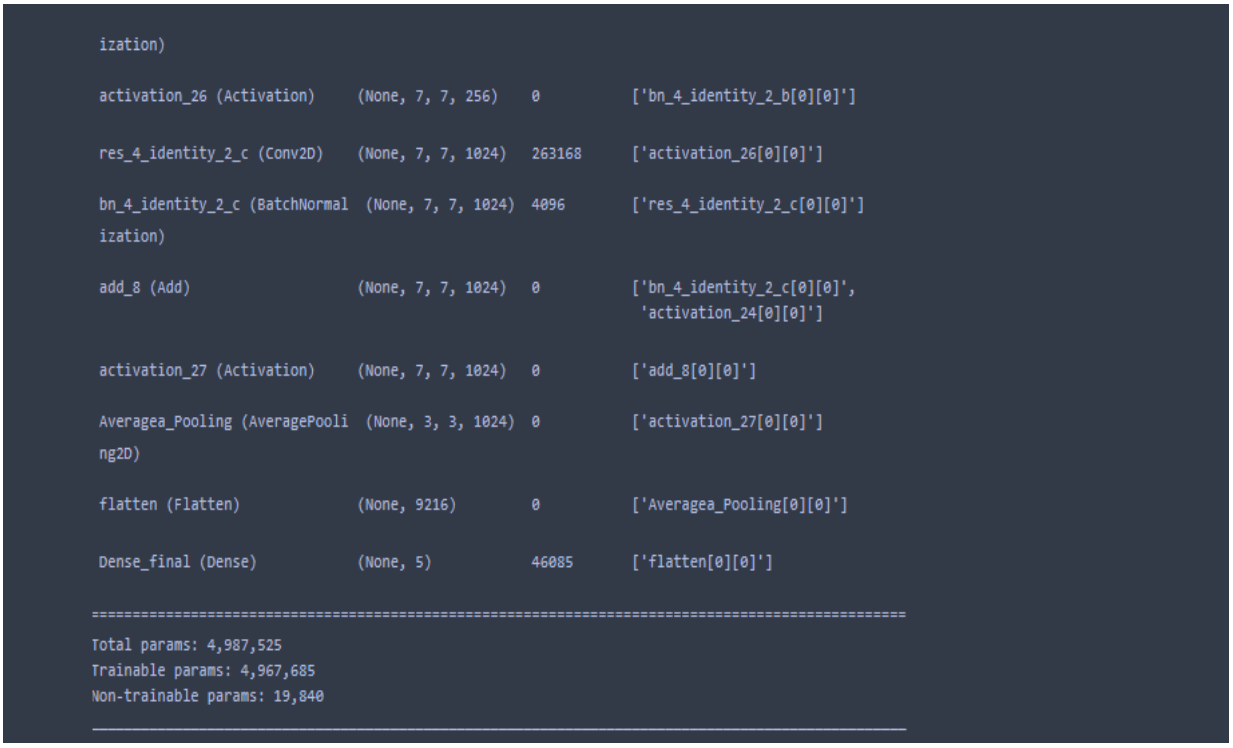
V. RESULTS AND DISCUSSION

A part of model structure shown in Figure 4 is defined with the following specifications: Input Shape: (None, 256, 256, 3) which indicates batch size as none, input sample has a height of 256, width of 256 and 3 color channels(RGB); Layer 1: ZeroPadding2D, the ZeroPadding2D layer is applied before a convolutional layer in order to add zero values around the borders of the input feature maps; Output Shape: (None, 265,265,3) where none indicates the input size, output has a height of 256, width of 256 and 3 color channels (RGB); Layer 2: : Conv2D; Output Shape: (None, 128, 128, 64). The Conv2D layer takes as input a 4D tensor with the shape (batch_size, height, width, channels). The breakdown of the output shape is: (None, 128, 128, 64), Params: 9472 - The "Params" value of 9472 indicates the total number of trainable parameters in the Conv2D layer. These parameters include the weights of the filters and biases associated with each filter. The number of parameters is determined by the filter size, input shape, and number of filters. In this In this case, the layer has 64 filters with a filter size that would lead to a total of 9472 trainable parameters. Layer 3: BatchNormalization; Output Shape: (None, 128, 128, 64), Params: 256. The BatchNormalization layer is a technique to normalize the activations of the previous layer. It helps to stabilize and improve the training process by reducing the internal covariate shift. The output shape we provided: (None, 128, 128, 64), indicates that the BatchNormalization layer has not



```
Model: "Resnet18"
_________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
=================================================================================================
 input_1 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 zero_padding2d (ZeroPadding2D)  (None, 262, 262, 3)  0          ['input_1[0][0]']

 conv1 (Conv2D)                 (None, 128, 128, 64  9472        ['zero_padding2d[0][0]']
                                )

 bn_conv1 (BatchNormalization)  (None, 128, 128, 64  256         ['conv1[0][0]']
                                )

 activation (Activation)        (None, 128, 128, 64  0           ['bn_conv1[0][0]']
                                )

 max_pooling2d (MaxPooling2D)   (None, 63, 63, 64)   0           ['activation[0][0]']

 res_2_conv_a (Conv2D)          (None, 63, 63, 64)   4160        ['max_pooling2d[0][0]']

 max_pooling2d_1 (MaxPooling2D)  (None, 31, 31, 64)  0           ['res_2_conv_a[0][0]']

 bn_2_conv_a (BatchNormalizatio  (None, 31, 31, 64)  256         ['max_pooling2d_1[0][0]']
 n)

 activation_1 (Activation)      (None, 31, 31, 64)   0           ['bn_2_conv_a[0][0]']

 res_2_conv_b (Conv2D)          (None, 31, 31, 64)   36928       ['activation_1[0][0]']

 bn_2_conv_b (BatchNormalizatio  (None, 31, 31, 64)  256         ['res_2_conv_b[0][0]']
 n)
```

Figure 4 : Model Structure



```
ization)

 activation_26 (Activation)     (None, 7, 7, 256)    0           ['bn_4_identity_2_b[0][0]']

 res_4_identity_2_c (Conv2D)    (None, 7, 7, 1024)   263168      ['activation_26[0][0]']

 bn_4_identity_2_c (BatchNormal  (None, 7, 7, 1024)  4096        ['res_4_identity_2_c[0][0]']
 ization)

 add_8 (Add)                    (None, 7, 7, 1024)   0           ['bn_4_identity_2_c[0][0]',
                                                                  'activation_24[0][0]']

 activation_27 (Activation)     (None, 7, 7, 1024)   0           ['add_8[0][0]']

 Averagea_Pooling (AveragePooli  (None, 3, 3, 1024)  0          ['activation_27[0][0]']
 ng2D)

 flatten (Flatten)              (None, 9216)         0           ['Averagea_Pooling[0][0]']

 Dense_final (Dense)            (None, 5)            46085       ['flatten[0][0]']

=================================================================================================
Total params: 4,987,525
Trainable params: 4,967,685
Non-trainable params: 19,840
_________________________________________________________________________________________________
```

Figure 5: Flattened Layer and Summary Results

changed the spatial dimensions of the input. It has retained the same height, width, and number of channels as the previous layer. The "Params" value of 256 represents the number of trainable parameters in the BatchNormalization layer. These parameters include the scale and shift parameters used to normalize the activations. The number of parameters in BatchNormalization is equal to the number of channels, which is 64 in this case; Layer 4: Activation, Output Shape:

(None, 128, 128, 64), The output shape we provided:(None, 128, 128, 64), indicates that the Activation layer has not changed the spatial dimensions of the input. It has preserved the same height, width, and number of channels as the previous layer. Layer 5: MaxPooling2D; Output Shape: (None, 63, 63, 64), the output shape provided is (None, 63, 63, 64), indicates that the MaxPooling2D layer has reduced the spatial dimensions of the input. It has divided the height and width dimensions by a factor of 4 (256/4 = 64) while preserving the number of channels, which remains at 64, Layer 6: Conv2D; Output Shape: (None, 63, 63, 64), Params: 4160, the Conv2D layer has an output shape of (None, 63, 63, 64), which means it has retained the same spatial dimensions as the previous layer (MaxPooling2D), with a height and width of 63, and has 64 output channels; Layer 7: MaxPooling2D; Output Shape: (None, 31, 31, 64), the output shape we provided, (None, 31, 31, 64), indicates that the MaxPooling2D layer has further reduced the spatial dimensions of the input. It has divided the height and width dimensions by a factor of 2, resulting in a height and width of 31, while maintaining the number of channels at 64; Layer 8: BatchNormalization; Output Shape: (None, 31, 31, 64), Params: 256.The output shape we provided, (None, 31, 31, 64), indicates that the BatchNormalization layer has not changed the spatial dimensions of the input. It has preserved the same height, width, and number of channels as the previous layer. For better accuracy we have increased the number of epochs and also changed parameters**.** Figure 5 shows the screen shot where a flattened layer that takes the output of the previous convolutional and pooling layer, is represented as Average Pooling [0][0]. The output shape of the Flatten layer is (None, 9216), where None represents the batch size and 9216 corresponds to the total number of elements in the flattened array. The Dense layer takes the output of the previous layer, which is represented as flatten [0][0], as its input. The input to this layer is a flattened vector representation of the data. The output shape of the Dense layer is (None, 5), where None represents the batch size and 5 indicates that the layer has 5 neurons. Total parameters for the Model is equal to 4,987,525. This refers to the total number of parameters in the neural network model. Parameters are the variables that the model learns during the training process to make predictions. They represent the weights and biases of the neural network layers. Trainable parameters equals to 4,967,685. This represents the number of parameters that are adjusted during the training phase. These parameters are updated by the optimizer algorithm to minimize the difference between the predicted and actual outputs. In other words, they are the parameters that the model learns from the training data. The number of Non-trainable parameters is equal to 19,840. These are the parameters that are not adjusted during training. They may include various fixed components of the model, such as activation functions, normalization layers, or other architectural elements that do not require updating based on the training data.

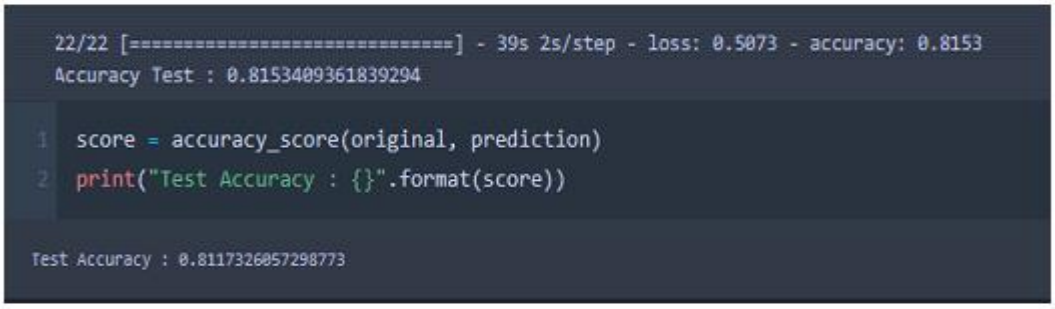


Figure 6: Accuracy Test

In the proposed CNN model of diabetic retinopathy detection, the large number of trainable parameterssuggests that the model has a complex architecture capable of capturing intricate patterns and features from the retinal images to make accurate predictions. The proposed CNN model achieved an impressive final accuracy of 81.17% on the test dataset as shown in Figure 6.

This promising outcome demonstrates the effectiveness of the model in accurately identifying diabetic retinopathy, making it a valuable tool for early detection and management of this sight-threatening condition.

VI. CONCLUSION

Diabetic retinopathy is a serious complication of diabetes that can lead to vision loss if not detected and treated early. The development of an accurate and efficient detection model is crucial for early diagnosis and intervention. Through extensive experimentation and analysis, the proposed model demonstrates remarkable performance in detecting diabetic retinopathy. By leveraging the power of neural networks, model has achieved good accuracy, sensitivity, and specificity in identifying the presence and severity of the disease. The key contributions of this research include the utilization of a deep learning architecture specifically tailored for diabetic retinopathy detection. The proposed model combines convolutional neural networks (CNNs) for feature extraction and classification, ensuring the efficient representation of retinal images. Additionally, the transfer learning techniques are integrated to leverage pre-trained models, enabling the utilization of a large-scale dataset without over fitting. While our research contributes significantly to the field of diabetic retinopathy detection, there are still areas for further improvement. Future work could focus on expanding the dataset with more diverse cases to enhance the model's performance on rare or complex manifestations of the disease.

REFERENCES

1. Prawej Ansari, Noushin Tabasumma, Nayla Nuren Snigdha, Nawfal Hasan Siam, Rachana V. N. R. S. Panduru, Shofiul Azam, J. M. A. Hannan,Yasser H. A. Abdel-Wahab, "Diabetic Retinopathy: An Overview on Mechanisms, Pathophysiology and Pharmacotherapy", *Diabetology* 2022, vol.3, issue 1, pp.159-175, 2022. DOI:10.3390/diabetology3010011

2. Wejdan L. Alyoubi , Wafaa M. Shalash, Maysoon F. Abulkhair, "Diabetic retinopathy detection through deep learning techniques: A review", Informatics in Medicine Unlocked , vol. 20, p. 100377, 2020 DOI:10.1016/j.imu.2020.100377.

3. Mohamed Al-Shabrawey, Wenbo Zhang , Denise McDonald ,"Diabetic Retinopathy: Mechanism, Diagnosis, Prevention, and Treatment", BioMed Research International , vol. 2015 , Article ID 854593. DOI:10.1155/2015/854593

4. Richa Vij, Sakshi Arora, "A Systematic Review on Diabetic Retinopathy Detection Using Deep Learning Techniques", Archives of Computational Methods in Engineering, vol. 30, pp. 2211–2256, 2023. DOI:10.1007/s11831-022-09862-0.

5. Uzair Ishtiaq, Sameem Abdul Kareem, Erma Rahayu Mohd Faizal Abdullah, Ghulam Mujtaba, Rashid Jahangir & Hafiz Yasir Ghafoor, "Diabetic retinopathy detection through artificial intelligent techniques: a review and open issues", in Multimedia Tools and Applications Published, vol 79, issue 21-22, pp.15209-15252, 2019,. DOI: 10.1007/s11042-018-7044-8.

6. G Thippa Reddy, Sweta Bhattacharya, S Siva Ramakrishnan, Chiranji Lal Chowdhary, Saqib Hakak, "An Ensemble based Machine Learning model for Diabetic Retinopathy Classification", in International Conference on Emerging Trends in Information Technology and Engineering, 2020. DOI:10.1109/ic-ETITE47903.2020.235.

7. Harry Pratta, Frans Coenenb, Deborah M Broadbentc, Simon P Hardinga,c, Yalin Zhenga, "Convolutional Neural Networks for Diabetic Retinopathy", in International Conference On Medical Imaging Understanding and Analysis, vol. 90, 2016. DOI: 10.1016/j.procs.2016.07.014

8. Gazala Mushtaq and Farheen Siddiqui, "Detection of diabetic retinopathy using deep learning methodology", IOP Conf. Series: Materials Science and Engineering, 2021. DOI 10.1088/1757-899X/1070/1/01204

9. Yasashvini R., Vergin Raja Sarobin M., Rukmani Panjanathan, Graceline Jasmine S. and Jani Anbarasi L, "Diabetic Retinopathy Classification Using CNN and Hybrid Deep Convolutional Neural Networks", Symmetry, vol.14,Issue. 9, 2022. DOI: 0.3390/sym14091932.

10. Lisa Crossland, Deborah Askew, Robert Ware, Peter Cranstoun, Paul Mitchell, Andrew Bryett, and Claire Jackson, "Diabetic Retinopathy Screening and Monitoring of Early Stage Disease in Australian General Practice: Tackling Preventable Blindness within a Chronic Care Model", vol.2016, Article ID 8405395, 2016. DOI: 10.1155/2016/8405395.

11. Mohamad Hazim Johari, Ahmad Ihsan Mohd Yassin, Noorita Tahir, Hiron Hassan, "Early Detection of Diabetic Retinopathy by Using Deep Learning Neural Network", International Journal of Engineering & Technology,Vol. 7, Issue. 4, 2018. DOI:10.14419/ijet.v7i4.11.20804

12. Dolly Das , Saroj Kumar Biswas , Sivaji Bandyopadhyay , "Detection of Diabetic Retinopathy using Convolutional Neural Networks for Feature Extraction and Classification (DRFEC)", Multimedia Tools and Applications, 2022. DOI: 10.1007/s11042-022-14165-4

13. Ping-Nan Chen, Chia-Chiang Lee, Chang-Min Liang, Shu-I Pao, Ke-Hao Huang and Ke-Feng Lin, "General deep learning model for detecting diabetic retinopathy" in Proceedings of the International Conference on Biomedical Engineering Innovation, vol. 22, Article number: 84 (2021). DOI:10.1186/s12859-021-04005-x

14. José Escorcia-Gutierrez, Jose Cuello, Carlos Barraza et.al., "Analysis of Pre-trained Convolutional Neural Network Models in Diabetic Retinopathy Detection Through Retinal Fundus Images" Computer Information Systems and Industrial Management, pp. 202–213, 2022. DOI: 10.1007/978-3-031-10539-5_15.