A

Project Report

on

A Convolutional Neural Network with Attention For Multi-Stage Diabetic Retinopathy Detection

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2021-25

in

Computer Science and Engineering

By

Gaurav Payal 2100290100066

Geetika Verma 2100290100067

Jigyasha Bhushan 2100290100075

Kanishk Rawat 2100290100080

Under the supervision of

Prof. Bharti Chugh

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow
(Formerly UPTU)

May 2025

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material that to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name(s): Gaurav Payal
Geetika Verma
Jigyasha Bhushan
Kanishk Rawat

Roll No.: 2100290100066
2100290100067
2100290100075
2100290100080

Date: 15-05-2025

CERTIFICATE

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Supervisor Name

(Designation)

Date: 15-05-2025

Bharti Chugh

(Assistant Professor)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Prof. Bharti Chugh, Assistant Professor, Department of Computer Science and Engineering, KIET Group of Institutions, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness, and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have seen the light of day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Dean, Computer Science and Engineering Department, KIET Group of Institutions, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution to the completion of the project.

Date: 15-05-2025

Signature:

Name : Gaurav Payal
Geetika Verma
Jigyasha Bhushan
kanishk Rawat

Roll No.: 2100290100066
2100290100067
2100290100075
2100290100080

ABSTRACT

Diabetic Retinopathy (DR) continues to be among the major causes of vision loss and blindness in the world, particularly among adults of working age. With the growing prevalence of diabetes, so too does DR, necessitating early and precise diagnosis to avoid irreversible vision loss. Conventional approaches to diagnosis that depend on human inspection of retinal images by ophthalmologists are time- and resource-consuming and prone to human error. Therefore, there exists an urgent need for strong automated systems that will help clinicians efficiently and accurately detect and classify DR.

This work investigates the ability of the ResNet-50 convolutional neural network (CNN) architecture for multi-stage classification of diabetic retinopathy from retinal fundus images. ResNet-50, which is a deep residual learning model with 50 layers, has been recognized to possess strong feature extraction abilities, particularly in visual tasks that are complex. Utilizing this architecture, our model accurately classifies the severity of DR into its corresponding stages, such as no DR, mild, moderate, severe, and proliferative DR. The model attained a remarkable accuracy of 97.56% on the test dataset, not only displaying high precision but also stability across various image categories.

The central strength of ResNet-50 is its residual learning method, which supports training deep networks without the vanishing gradient issue typically experienced in standard deep neural networks. This helps the model learn intricate patterns and fine features in high-resolution retinal images that human eyes or standard algorithms might not detect. Such a level of detail in feature extraction becomes critical in separating neighboring DR stages, where differences may be subtle but clinically relevant.

Apart from high classification performance, our model also shows great potential in facilitating clinical workflows. With automated screening, the system has the potential to alleviate the diagnostic workload of ophthalmologists and allow them to concentrate on more

severe and complex cases. Identification at the early stages enables timely intervention, with the potential to arrest disease progression and save vision. Integration into clinical decision support tools can standardize diagnosis, minimize inter-observer variability, and simplify patient management.

The ramifications of this work go beyond the accuracy of classification. This research lays the foundation for subsequent innovation in AI-supported ophthalmic diagnosis. A potential path is the growth of the dataset to encompass more representative and diverse samples, thereby addressing existing constraints related to demographic and device heterogeneity. A bigger dataset would increase the model's resilience, reduce bias, and provide for usability across a wide range of clinical settings.

In addition, network architecture optimization for real-time operation is an important area of future research. Obtaining lower inference times without a reduction in accuracy can facilitate live diagnosis and monitoring, especially valuable in teleophthalmology. Incorporating multimodal clinical data — e.g., patient history, blood glucose, and optical coherence tomography (OCT) scans — may further improve prediction accuracy and enable comprehensive patient profiling.

A key factor for clinical uptake is interpretability of the model. Improving the transparency of predictions via visual explanations like Class Activation Mapping (CAM) or Grad-CAM can facilitate clinicians' understanding of the reasons behind each classification. This interpretability not only instills confidence in the AI system but also supports educational and training applications among medical personnel.

Moreover, ethical and regulatory compliance, especially on matters of data privacy, fairness, and accountability, will need to be emphasized as the system transitions to real-world use. The development of a platform for ongoing learning, where the model can adjust to new data and changing diagnostic criteria, will similarly be important to keeping the system relevant and effective in the long term.

In summary, this research highlights the great potential of employing the ResNet-50 deep learning model for efficient, accurate, and scalable diabetic retinopathy stages classification.

Its effectiveness is proved in retina scan analysis, and it presents a feasible tool to support clinicians in diagnosis and decision-making. Through resolution of critical issues in ophthalmic diagnosis and alignment with directions such as explainable AI and real-time deployment, this study makes a valuable contribution to the revolution of diabetic retinopathy screening and, by extension, to other uses of AI in medical imaging. The embedding of such technologies holds the potential not only for better patient outcomes but also for greater efficiency and equity in global healthcare delivery systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DR	Diabetic Retinopathy
CNNs	Convolutional Neural Network
Grad-CAM	Gradient-weighted Class Activation Mapping
MVDRNet	Minimum Variance Distortionless Response Neural Network
ROIs	Region of Interests
HPTI-v4	Hyperparameter Tuning Inception-v4
DL	Deep Learning

CHAPTER 1

INTRODUCTION

1.1 Diabetic Retinopathy

A vision disorder called "diabetic retinopathy" becomes more pronounced in patients suffering from diabetes. Diabetic retinopathy is a significant cause of blindness and serious eye-related issues among diabetics. According to the International Diabetes Federation's report, the prevalence of DR is rising day by day. As far as conditions that threaten individuals' eye health are concerned, DR is viewed to be in the number one position. Diabetic retinopathy can have no symptoms whatsoever. Some of the symptoms are night blindness, difficulty distinguishing colors, and others. The disease can lead to chronic degeneration and eventually permanent blindness if it is not diagnosed in its early stages. In its early stages, it also has no noticeable symptoms. It could lead to irreversible loss of eyesight. In DR detection, deep learning is used for categorization.

Although early stages of diabetic retinopathy are predominantly symptomless, neuronal retinal damage and clinically unsuspected microvascular changes take place at this time [1]. Thus, diabetic patients need regular eye examinations since early detection and treatment of the disease are very important [2]. Early detection of DR is especially important because control of hyperglycemia, hyperlipidemia, and hypertension is the only prevention method. In addition, if eyes are treated early in the disease, therapies already available, like laser photocoagulation, significantly decrease the possibility of blindness in proliferative retinopathy and diabetic maculopathy in as many as 98% of cases [3]. It is evident that early detection and proper treatment are crucial to postpone or even avoid blindness due to diabetic retinopathy [4].

This paper uses a deep learning-based model and an attention mechanism for automated diagnosis of diabetic retinopathy. This attention mechanism enables the model to selectively pay attention to the most relevant sections of the retinal images to improve its ability to accurately ascertain the stage of DR. Drawing on the power of the collaboration between deep learning and the attention mechanism, this work seeks to offer a more accurate, efficient, and convenient tool for the screening of DR.

1.2 Project Description

This project focuses on the conceptualization and development of an advanced AI-driven framework for detecting DR using the ResNet-50 CNN architecture. DR is a microvascular complication of diabetes per se, which involves the retina, and continues to be the most common cause of avoidable blindness in the majority of regions across the globe. Early diagnosis is vital for avoiding progression and facilitating early intervention. The suggested system uses high-resolution images of the retinal fundus as input and uses the deep feature extraction ability of ResNet-50 to diagnose DR in its different levels of severity, from no DR to proliferative DR. After rigorous training and validation, the model obtained an impressive classification accuracy of 97.56%, which reflects its capability to detect subtle as well as overt pathological structures like microaneurysms, hemorrhages, and neovascularization.

To extend diagnostic precision, an attention mechanism was incorporated into the network architecture. The module enables the model to distribute computational resources to clinically significant areas of interest (ROIs) in the retina, making it more sensitive and specific. The attention module learns to dynamically weigh spatial features, increasing the interpretability and salience of learned representations during classification. This automated solution has a number of practical advantages. By curtailing dependence on ophthalmologist manual review, it minimizes clinical workload and diagnostic delay. The system also provides enhanced access to DR screening, especially among the remote or underserved population where resources for ophthalmology may be scarce. The AI model is also scalable, with potential for integration into teleophthalmology systems and mobile screening trucks. In the future, system optimization will be aimed at real-time inference on edge-computing platforms and embedded systems to facilitate deployment in point-of-care settings. Integration with larger and more diverse datasets from multiple imaging sources will improve model robustness and generalizability. In addition, methods such as Grad-CAM and saliency maps will be employed to enhance model explainability, and this will foster more clinical trust and transparency. In total, these innovations will result in the creation of a robust, low-cost, and clinically viable AI-based diagnostic system for DR early detection.

CHAPTER 2

LITERATURE REVIEW

For effective feature extraction, a multi-view DRD based on combining DCNNS and attention mechanisms (MVDRNet) was proposed in [14]. Lesion explanation training of the network using proposed methods did not give better results. A shallow CNN-based model including multi-scale for end-to-end early DR detection in retinal image arrangement was proposed by Chen [15]. The proposed model could not improve the categorization accuracy any further. For the routine diagnosis of DR, a CNN was created [16]. However, the inefficient architecture of the network hindered it from enhancing the automated diabetic retinopathy grading systems currently being used in clinical environments. The deep learning architecture was created in [17] based on scheduled segmented fundus image features for the purpose of classifying DR. A capsule network modified for the classification of DR using fundus images and feature extraction [19]. Yet, a more accurate and timely diagnosis was not set to prevent vision loss. A new DRD approach was designed using top fundus imaging and deep learning methods [20]. Yet, the suggested approach did not produce a region of interest for retinopathy detection from the whole area captured, in an effort to reduce complexity. A deep learning method for the automatic detection of diabetic retinopathy lesions based on ROI extraction [21]. The algorithms could not employ the minimum amount of system resources. Automatic retinopathy detection was generated with the machine learning architecture of the dynamic deep learning technique [22]. But the multi-layered architecture could not provide support to a range of multimedia applications as dreamt. The deep symmetric CNN approach was designed to identify diabetic retinopathy [23]. However, the anticipated process was unable to precisely identify diabetic retinopathy from the photos. A multitask deep learning model was proposed to detect different types of diabetic retinopathy [24]. But the real-time achievement of this model was not implemented. They developed a multi-scale attention network to categorize diabetic retinopathy [25]. To classify the fundus pictures, a novel automated model called Hyperparameter Tuning Inception-v4 (HPTI-v4) was developed [40].

An independent DR detection method was formulated to enhance the efficiency of DR detection [41]. Deep learning algorithms are applied in screening to achieve the optimal degree of accuracy [42]. Features were not extracted with the intended method. To enhance the diagnostic efficiency, the general deep learning model was used [43]. Nevertheless, the false-positive rate did not decrease. Machine intelligence-based automated detection was proposed to achieve accurate diagnostic results [44]. The noisy pixel, though, could not be removed. Hybrid DL was proposed for automatically detecting DR [45]. Noisy images were removed with the help of preprocessing. To minimize the time taken, an independent deep learning-based bleeding detection system was developed [46].

CHAPTER 3

PROPOSED METHODOLOGY

This paper suggests an efficient and automated multi-stage classification approach to diabetic retinopathy (DR) with the aid of a deep learning mechanism supported by an attention mechanism. The backbone of this model is based on the ResNet-50 model, selected due to its robust hierarchical feature extraction capabilities and resistance to vanishing gradients in deep network training. The below elements form the overall pipeline of the proposed system:

3.1 Feature Selection and Preprocessing

The dataset utilized consists of 2,760 retinal fundus images obtained from Kaggle, and they represent five different DR stages: healthy, mild, moderate, proliferative, and severe. With the inconsistency in image resolution and quality, preprocessing became a necessity to normalize input and enhance model learning. The procedures involved:

The procedures involved:

Resizing: The images were resized to a standard size of 224×224 pixels to match the input specifications of the ResNet-50 model.

Normalization: The pixel intensities were normalized to the [0,1] interval through min-max scaling (i.e., dividing by 255), which enables faster convergence in training.

Format Filtering: Images of only compatible formats (e.g., JPG, PNG, TIFF) were kept in order to preserve dataset integrity and compatibility with OpenCV processing libraries.

Each image went through these transformations prior to inclusion in the dataset employed for training and testing.

It is required for the improvement of the efficiency of the deep models by prioritizing the saliencies. In the diabetic retinopathy detection context, the ResNet50 model is able to learn the features hierarchically from the images automatically, eliminating the need for manual engineering features. Images from the dataset for the diabetic retinopathy were preprocessed for homogeneity of the sizes and the distribution of the pixels for the enhancement of the capability of the model for the extraction of the relevant features. Images from the dataset for the inputs were not the same sizes, putting the risk for the fixed-size inputs for the convolutional neural network (CNN). To counter this risk, all the images were resized into the same target size of 224 * 224 pixels, the default for the ResNet50 model. Resizing operation was carried out using the following function, as seen from the expression equation (1)

resized_image = cv2.resize(image,(224,224)) (1)

Where image is the input array, cv2.resize is a function from the OpenCV library used to resize the image to the target size. To ensure that the pixel values are within a range that accelerates the model's convergence, min-max normalization was applied in equation (2)

normalized_image = (resized_image)/255.0 (2)

In this step, each value (in pixels) in the resized image is divided by 255, transforming the integer pixel values (ranging from 0 to 255) into floating-point values within the range of 0 to 1. The dataset contains images in various formats, such as PNG, JPG, BMP, and TIFF. The pre-processing script filters specific formats and processes only files with the specified extensions to ensure compatibility, as outlined in Equation (3)

if file.endswith(('.png','.jpg','.jpeg','.bmp','.tiff')) (3)

The entire pre-processing process was applied iteratively to all images in the dataset, ensuring that each image is resized, normalized, and stored for further use in model training. The processed images were then stored in a list for use in the subsequent steps.

3.2 Deep Learning Architecture

The core of the classification framework is based on ResNet-50 architecture, a 50-layer convolutional neural network widely recognized for its residual learning technique. In contrast to standard CNNs that are prone to degradation and vanishing gradients as they are deepened, ResNet-50 uses identity shortcut connections through which the gradients pass unobstructed. Every residual block of the network calculates a function that gets added to the input, enhancing the reuse of features and the flow of information.

The structure of the model with layers includes:

A starting 7×7 convolutional layer followed by max pooling.

Several residual blocks with three convolutional layers (1×1, 3×3, and 1×1).

Batch normalization following every convolution to stabilize learning.

Down-sampling and spatial dimension reduction using strided convolutions.

Global average pooling close to the output for minimizing overfitting.

A last fully connected layer followed by a softmax classifier to estimate the probability of each DR class.

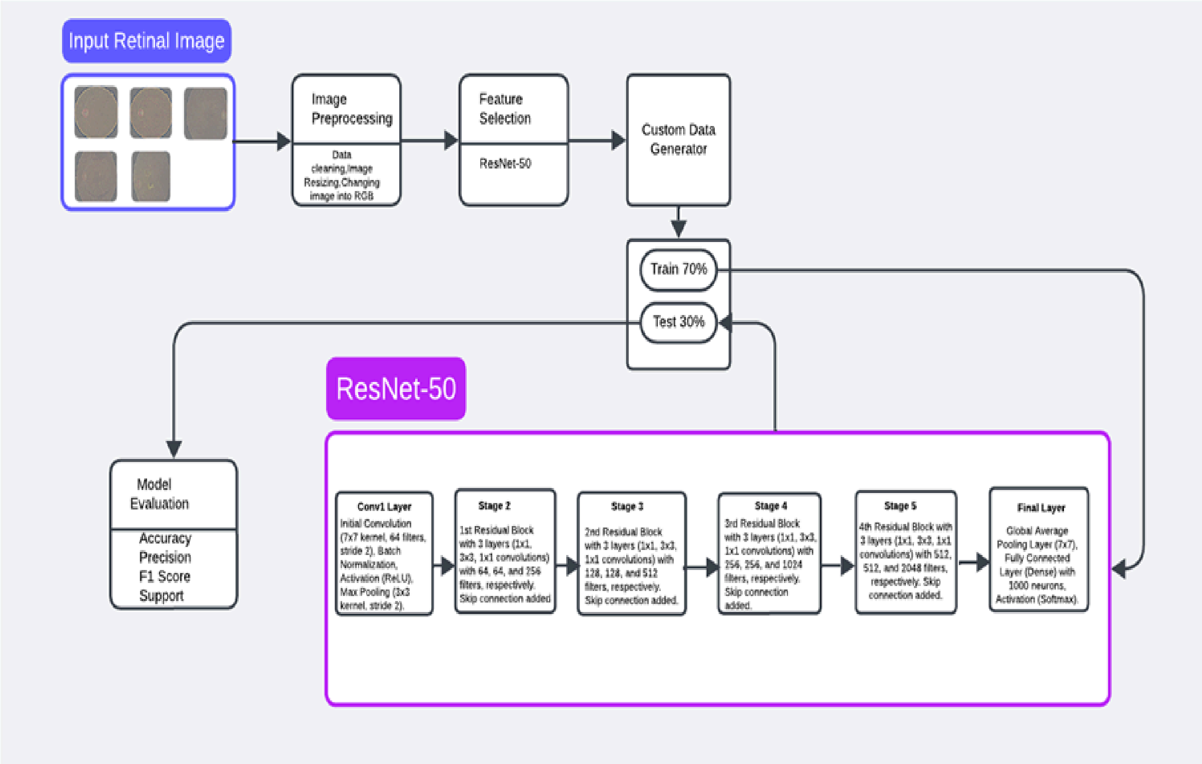

Input Retinal Image
Image Preprocessing
Data cleaning,Image Resizing,Changing image into RGB
Feature Selection
ResNet-50
Custom Data Generator
Train 70%
Test 30%
ResNet-50
Model Evaluation
Accuracy
Precision
F1 Score
Support
Conv1 Layer
Initial Convolution (7x7 kernel, 64 filters, stride 2), Batch Normalization, Activation (ReLU), Max Pooling (3x3 kernel, stride 2).
Stage 2
1st Residual Block with 3 layers (1x1, 3x3, 1x1 convolutions) with 64, 64, and 256 filters, respectively. Skip connection added
Stage 3
2nd Residual Block with 3 layers (1x1, 3x3, 1x1 convolutions) with 128, 128, and 512 filters, respectively. Skip connection added.
Stage 4
3rd Residual Block with 3 layers (1x1, 3x3, 1x1 convolutions) with 256, 256, and 1024 filters, respectively. Skip connection added.
Stage 5
4th Residual Block with 3 layers (1x1, 3x3, 1x1 convolutions) with 512, 512, and 2048 filters, respectively. Skip connection added.
Final Layer
Global Average Pooling Layer (7x7), Fully Connected Layer (Dense) with 1000 neurons, Activation (Softmax).

Fig 1: Proposed Architecture for Diabetic Retinopathy Detection

3.3 Attention Mechanism Integration

In order to further optimize classification performance, the model adds an attention mechanism that allows the network to focus on pertinent areas of the retinal image. This is particularly useful in medical imaging where localized features (microaneurysms, hemorrhages, exudates) play an important role in diagnosis.

The attention layer calculates alignment scores between the feature maps through scaled dot-product attention. The mechanism works in three steps:

Score Calculation: Calculates a dot product of query (Q) and key (K) vectors.

Scaling: The output scores are scaled by the square root of the key dimension to prevent very large gradients.

Softmax Application: Softmax function transforms the scores to attention weights.

Weighted Sum: These weights are utilized to value (V) vectors to produce attention-concentrated outputs.

This mechanism makes the model focus on regions of high diagnostic utility and thereby enhance its interpretability as well as its accuracy.

3.4 Model Training and Optimization

The network was trained by a supervised learning method, where the categorical cross-entropy loss function was optimized through a variation of the stochastic gradient descent (SGD) algorithm. In order to prevent overfitting as well as enhance generalization, methods such as data augmentation, dropout, and learning rate scheduling were employed.

Performance was measured through metrics like:

Accuracy: Number of correctly predicted instances out of all instances.

Precision: Capability of the model to provide correct positive predictions.

Recall: Number of positive cases correctly identified out of total positive cases.

F1-Score: Harmonic mean between precision and recall.

Support: Number of instances per class.

These metrics yield a comprehensive performance measurement of the model's predictability, particularly in class-unbalanced environments.

CHAPTER 4

RESULTS AND DISCUSSION

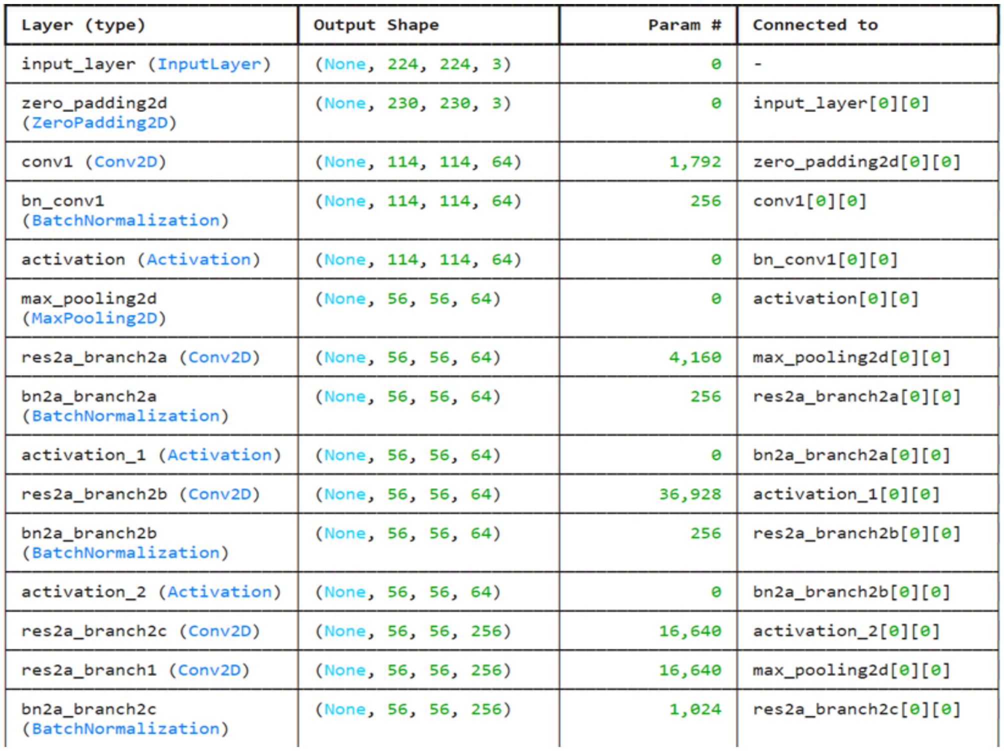
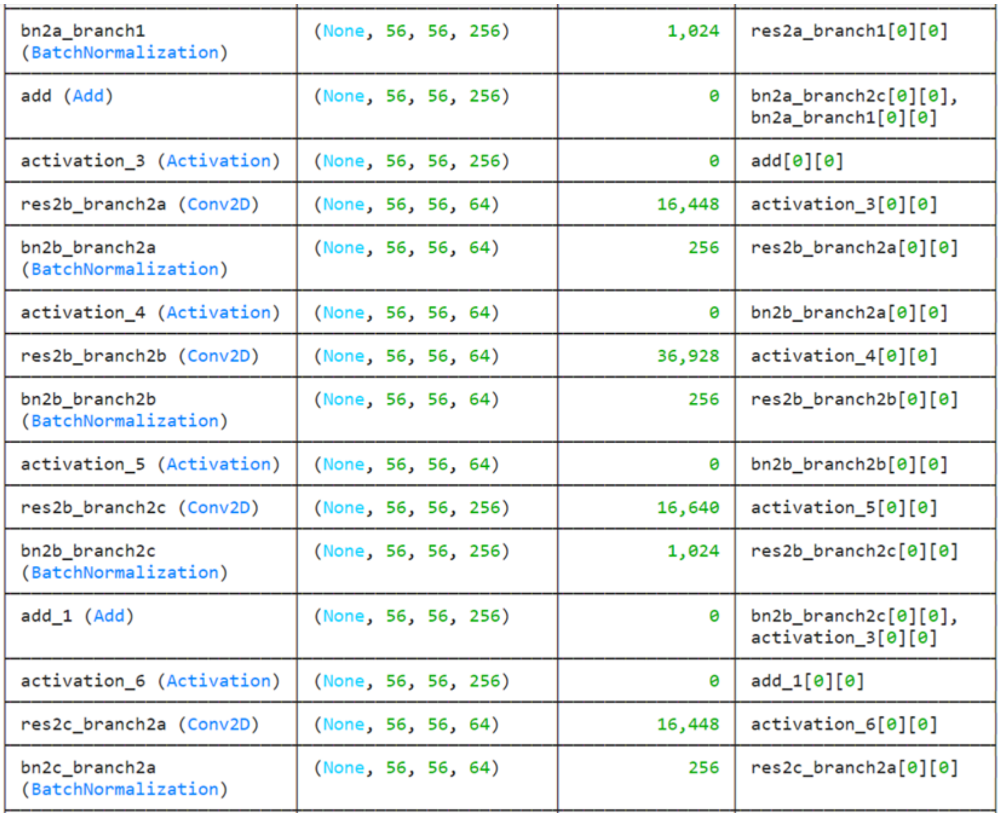
The use of the ResNet-50 deep learning model in detecting diabetic retinopathy (DR) has been shown to have a spectacular classification accuracy rate of 97.56%, highlighting the revolutionizing potential of deep learning methodologies in medical image analysis. With a residual architecture as a convolutional neural network (CNN), ResNet-50 is specifically geared to capturing and learning subtle patterns in high-resolution images. With regard to retinal imaging, the model effectively extracts and examines fine pathological features like microaneurysms, exudates, and hemorrhages in the retina, all of which are crucial predictors of DR progression. By correctly classifying DR into its multiple stages—from mild non-proliferative to severe proliferative—the model presents an effective diagnostic tool that enables timely clinical decision-making.

The model summary presents a deep convolutional neural network based on the ResNet-50 architecture, widely known for its exceptional image classification performance. It accepts 224x224 RGB images as input, which are first processed through a 7x7 convolution layer with 64 filters and a stride of 2, followed by a 3x3 max-pooling layer. This combination reduces the spatial dimensions while capturing low-level features like edges and textures.

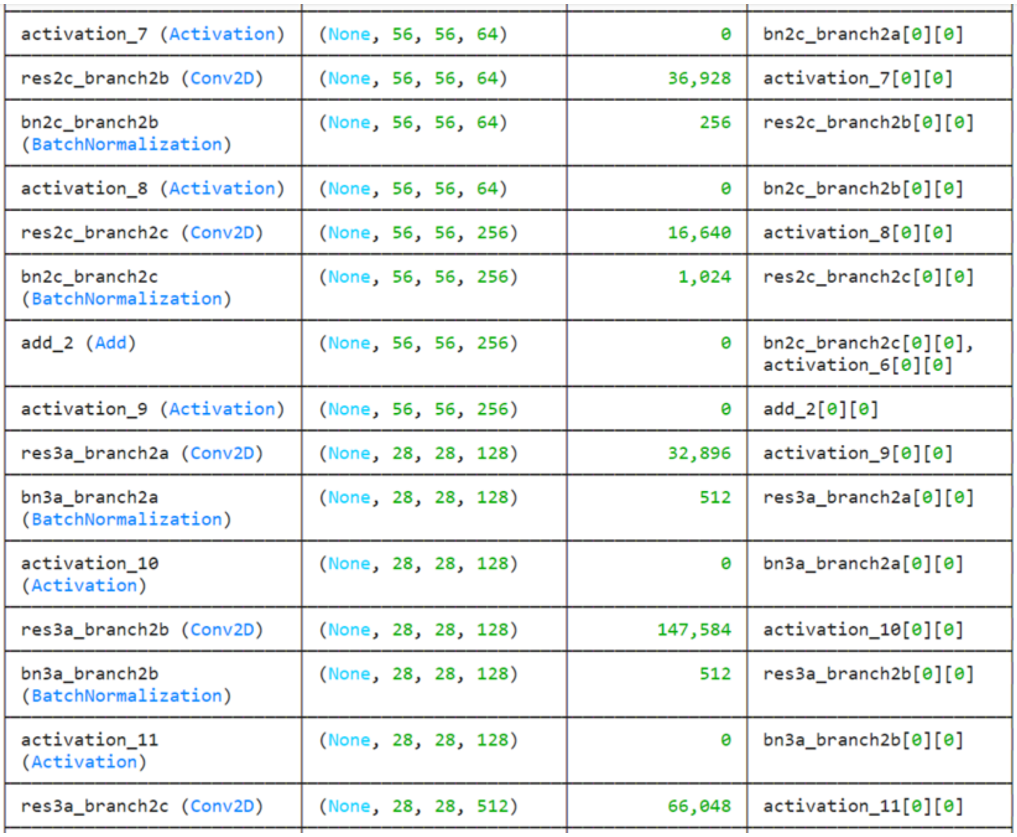
4.1 Results

The core of the network consists of four main stages of residual blocks. Each block contains a sequence of convolutional layers with 1x1, 3x3, and 1x1 filters, using BatchNormalization and ReLU activation to enhance learning. The first stage has 64 filters, the second 128, the third 256, and the final stage 512 filters, allowing the network to capture progressively complex and abstract features. The residual connections within each block help alleviate the vanishing gradient problem, improving training stability and allowing for deeper networks.

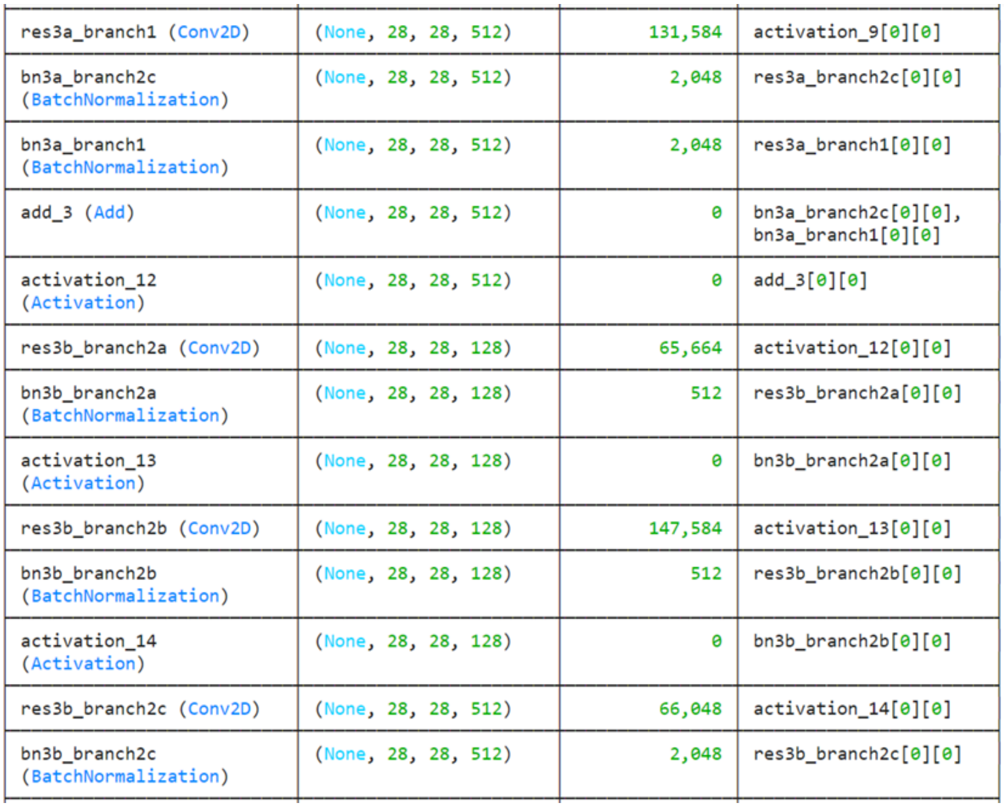
After these convolutional blocks, the model applies a global average pooling layer to reduce the feature maps into a single vector, followed by a fully connected dense layer for final classification. This architecture, with over 23 million trainable parameters, is highly effective for large-scale image recognition.

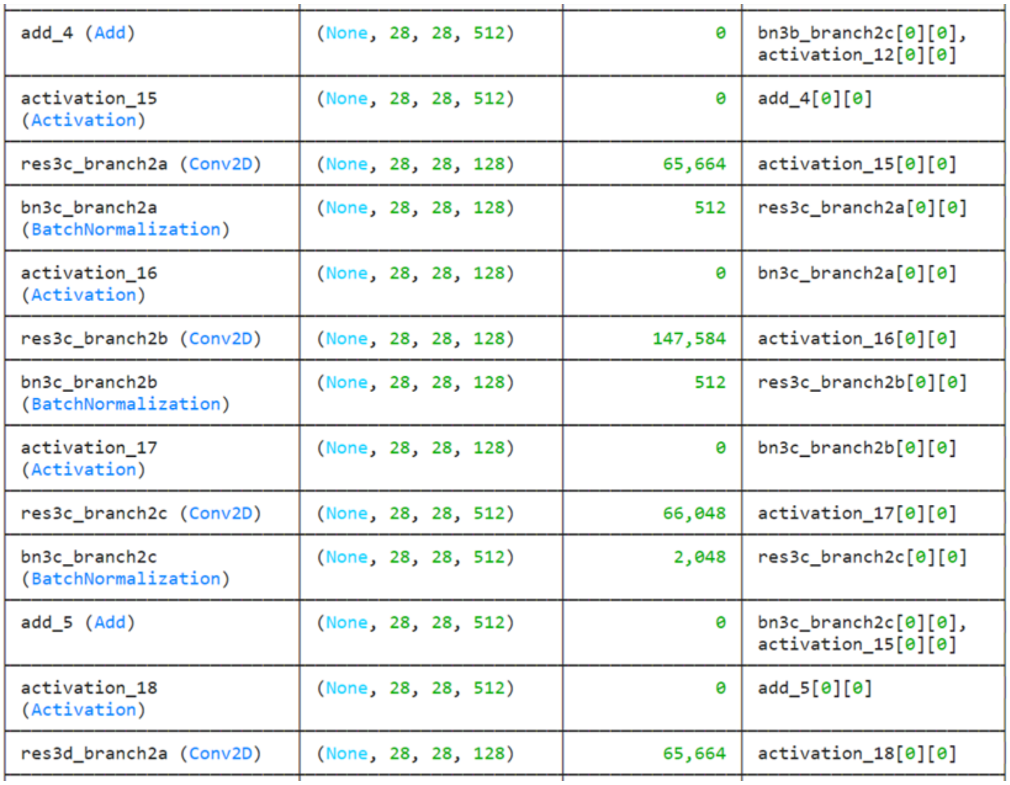
Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
zero_padding2d (ZeroPadding2D)	(None, 230, 230, 3)	0	input_layer[0][0]
conv1 (Conv2D)	(None, 114, 114, 64)	1,792	zero_padding2d[0][0]
bn_conv1 (BatchNormalization)	(None, 114, 114, 64)	256	conv1[0][0]
activation (Activation)	(None, 114, 114, 64)	0	bn_conv1[0][0]
max_pooling2d (MaxPooling2D)	(None, 56, 56, 64)	0	activation[0][0]
res2a_branch2a (Conv2D)	(None, 56, 56, 64)	4,160	max_pooling2d[0][0]
bn2a_branch2a (BatchNormalization)	(None, 56, 56, 64)	256	res2a_branch2a[0][0]
activation_1 (Activation)	(None, 56, 56, 64)	0	bn2a_branch2a[0][0]
res2a_branch2b (Conv2D)	(None, 56, 56, 64)	36,928	activation_1[0][0]
bn2a_branch2b (BatchNormalization)	(None, 56, 56, 64)	256	res2a_branch2b[0][0]
activation_2 (Activation)	(None, 56, 56, 64)	0	bn2a_branch2b[0][0]
res2a_branch2c (Conv2D)	(None, 56, 56, 256)	16,640	activation_2[0][0]
res2a_branch1 (Conv2D)	(None, 56, 56, 256)	16,640	max_pooling2d[0][0]
bn2a_branch2c (BatchNormalization)	(None, 56, 56, 256)	1,024	res2a_branch2c[0][0]
bn2a_branch1 (BatchNormalization)	(None, 56, 56, 256)	1,024	res2a_branch1[0][0]
add (Add)	(None, 56, 56, 256)	0	bn2a_branch2c[0][0], bn2a_branch1[0][0]
activation_3 (Activation)	(None, 56, 56, 256)	0	add[0][0]
res2b_branch2a (Conv2D)	(None, 56, 56, 64)	16,448	activation_3[0][0]
bn2b_branch2a (BatchNormalization)	(None, 56, 56, 64)	256	res2b_branch2a[0][0]
activation_4 (Activation)	(None, 56, 56, 64)	0	bn2b_branch2a[0][0]
res2b_branch2b (Conv2D)	(None, 56, 56, 64)	36,928	activation_4[0][0]
bn2b_branch2b (BatchNormalization)	(None, 56, 56, 64)	256	res2b_branch2b[0][0]
activation_5 (Activation)	(None, 56, 56, 64)	0	bn2b_branch2b[0][0]
res2b_branch2c (Conv2D)	(None, 56, 56, 256)	16,640	activation_5[0][0]
bn2b_branch2c (BatchNormalization)	(None, 56, 56, 256)	1,024	res2b_branch2c[0][0]
add_1 (Add)	(None, 56, 56, 256)	0	bn2b_branch2c[0][0], activation_3[0][0]
activation_6 (Activation)	(None, 56, 56, 256)	0	add_1[0][0]
res2c_branch2a (Conv2D)	(None, 56, 56, 64)	16,448	activation_6[0][0]
bn2c_branch2a (BatchNormalization)	(None, 56, 56, 64)	256	res2c_branch2a[0][0]



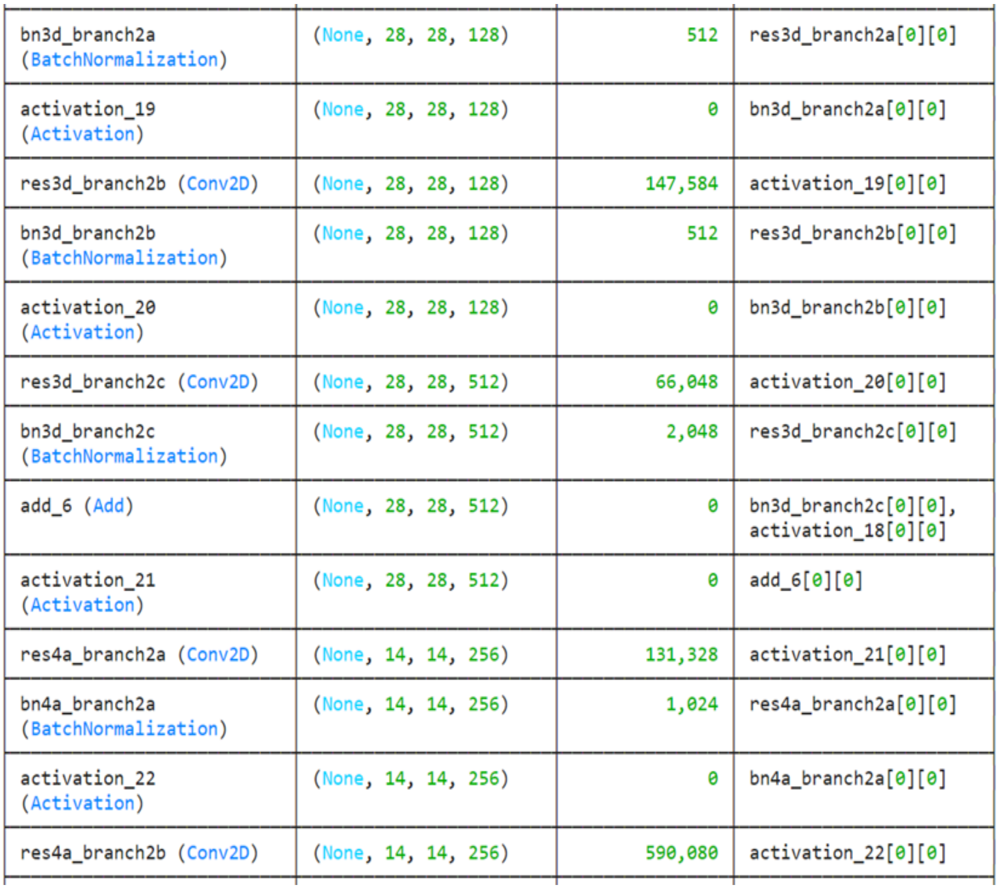
activation_7 (Activation)	(None, 56, 56, 64)	0	bn2c_branch2a[0][0]
res2c_branch2b (Conv2D)	(None, 56, 56, 64)	36,928	activation_7[0][0]
bn2c_branch2b (BatchNormalization)	(None, 56, 56, 64)	256	res2c_branch2b[0][0]
activation_8 (Activation)	(None, 56, 56, 64)	0	bn2c_branch2b[0][0]
res2c_branch2c (Conv2D)	(None, 56, 56, 256)	16,640	activation_8[0][0]
bn2c_branch2c (BatchNormalization)	(None, 56, 56, 256)	1,024	res2c_branch2c[0][0]
add_2 (Add)	(None, 56, 56, 256)	0	bn2c_branch2c[0][0], activation_6[0][0]
activation_9 (Activation)	(None, 56, 56, 256)	0	add_2[0][0]
res3a_branch2a (Conv2D)	(None, 28, 28, 128)	32,896	activation_9[0][0]
bn3a_branch2a (BatchNormalization)	(None, 28, 28, 128)	512	res3a_branch2a[0][0]
activation_10 (Activation)	(None, 28, 28, 128)	0	bn3a_branch2a[0][0]
res3a_branch2b (Conv2D)	(None, 28, 28, 128)	147,584	activation_10[0][0]
bn3a_branch2b (BatchNormalization)	(None, 28, 28, 128)	512	res3a_branch2b[0][0]
activation_11 (Activation)	(None, 28, 28, 128)	0	bn3a_branch2b[0][0]
res3a_branch2c (Conv2D)	(None, 28, 28, 512)	66,048	activation_11[0][0]



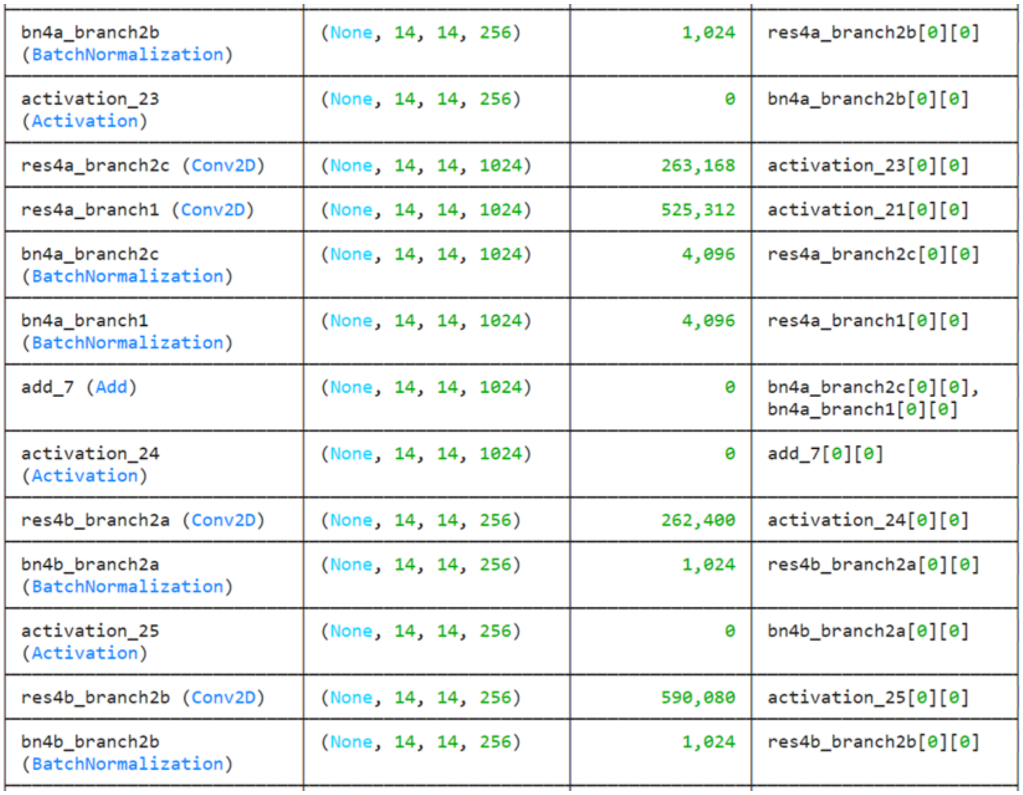
res3a_branch1 (Conv2D)	(None, 28, 28, 512)	131,584	activation_9[0][0]
bn3a_branch2c (BatchNormalization)	(None, 28, 28, 512)	2,048	res3a_branch2c[0][0]
bn3a_branch1 (BatchNormalization)	(None, 28, 28, 512)	2,048	res3a_branch1[0][0]
add_3 (Add)	(None, 28, 28, 512)	0	bn3a_branch2c[0][0], bn3a_branch1[0][0]
activation_12 (Activation)	(None, 28, 28, 512)	0	add_3[0][0]
res3b_branch2a (Conv2D)	(None, 28, 28, 128)	65,664	activation_12[0][0]
bn3b_branch2a (BatchNormalization)	(None, 28, 28, 128)	512	res3b_branch2a[0][0]
activation_13 (Activation)	(None, 28, 28, 128)	0	bn3b_branch2a[0][0]
res3b_branch2b (Conv2D)	(None, 28, 28, 128)	147,584	activation_13[0][0]
bn3b_branch2b (BatchNormalization)	(None, 28, 28, 128)	512	res3b_branch2b[0][0]
activation_14 (Activation)	(None, 28, 28, 128)	0	bn3b_branch2b[0][0]
res3b_branch2c (Conv2D)	(None, 28, 28, 512)	66,048	activation_14[0][0]
bn3b_branch2c (BatchNormalization)	(None, 28, 28, 512)	2,048	res3b_branch2c[0][0]



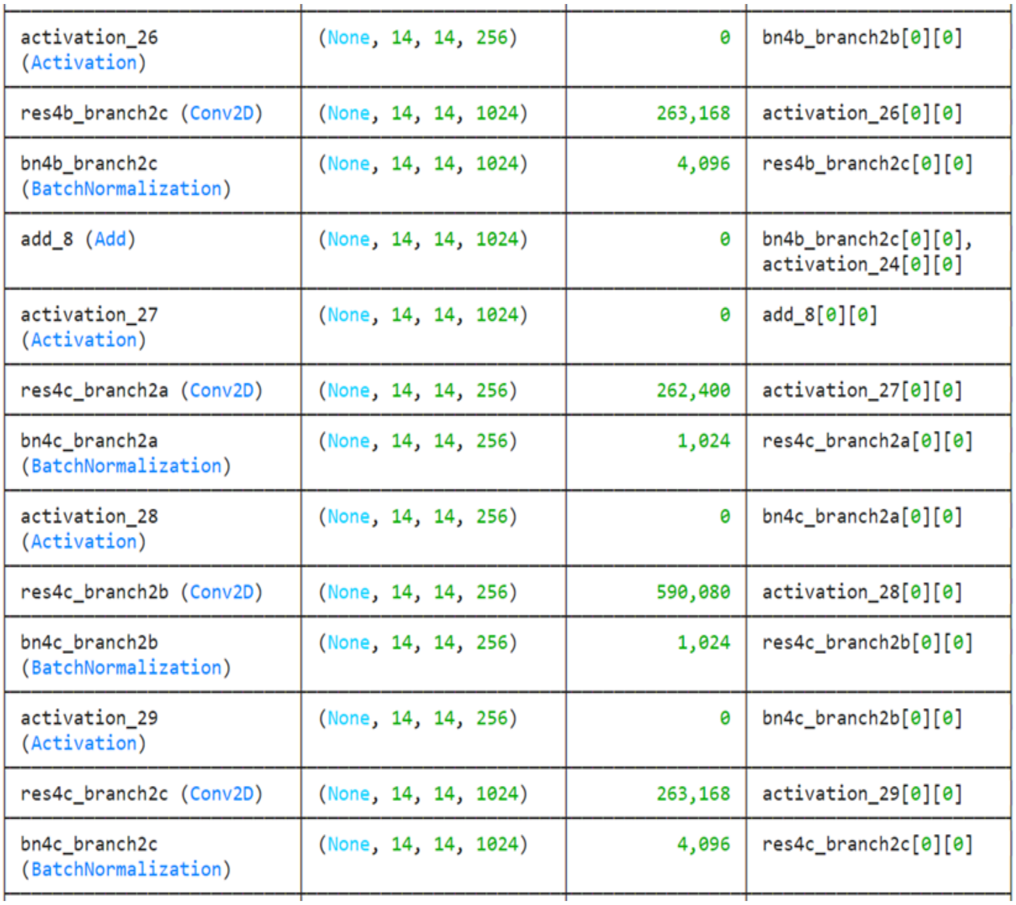
add_4 (Add)	(None, 28, 28, 512)	0	bn3b_branch2c[0][0], activation_12[0][0]
activation_15 (Activation)	(None, 28, 28, 512)	0	add_4[0][0]
res3c_branch2a (Conv2D)	(None, 28, 28, 128)	65,664	activation_15[0][0]
bn3c_branch2a (BatchNormalization)	(None, 28, 28, 128)	512	res3c_branch2a[0][0]
activation_16 (Activation)	(None, 28, 28, 128)	0	bn3c_branch2a[0][0]
res3c_branch2b (Conv2D)	(None, 28, 28, 128)	147,584	activation_16[0][0]
bn3c_branch2b (BatchNormalization)	(None, 28, 28, 128)	512	res3c_branch2b[0][0]
activation_17 (Activation)	(None, 28, 28, 128)	0	bn3c_branch2b[0][0]
res3c_branch2c (Conv2D)	(None, 28, 28, 512)	66,048	activation_17[0][0]
bn3c_branch2c (BatchNormalization)	(None, 28, 28, 512)	2,048	res3c_branch2c[0][0]
add_5 (Add)	(None, 28, 28, 512)	0	bn3c_branch2c[0][0], activation_15[0][0]
activation_18 (Activation)	(None, 28, 28, 512)	0	add_5[0][0]
res3d_branch2a (Conv2D)	(None, 28, 28, 128)	65,664	activation_18[0][0]



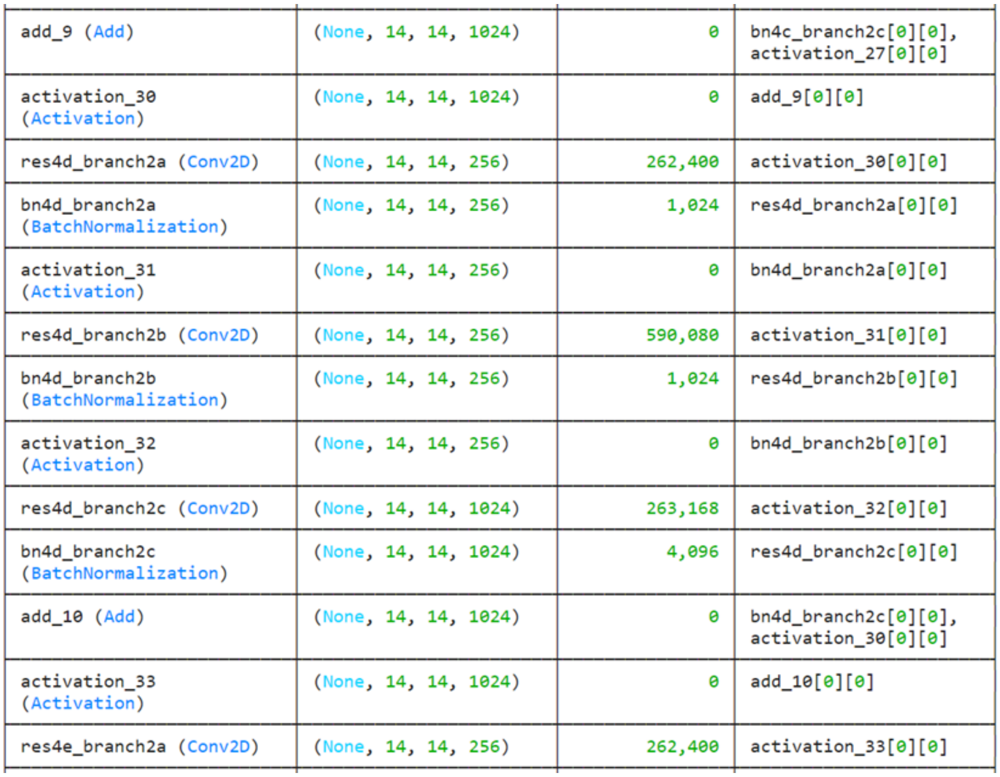
bn3d_branch2a (BatchNormalization)	(None, 28, 28, 128)	512	res3d_branch2a[0][0]
activation_19 (Activation)	(None, 28, 28, 128)	0	bn3d_branch2a[0][0]
res3d_branch2b (Conv2D)	(None, 28, 28, 128)	147,584	activation_19[0][0]
bn3d_branch2b (BatchNormalization)	(None, 28, 28, 128)	512	res3d_branch2b[0][0]
activation_20 (Activation)	(None, 28, 28, 128)	0	bn3d_branch2b[0][0]
res3d_branch2c (Conv2D)	(None, 28, 28, 512)	66,048	activation_20[0][0]
bn3d_branch2c (BatchNormalization)	(None, 28, 28, 512)	2,048	res3d_branch2c[0][0]
add_6 (Add)	(None, 28, 28, 512)	0	bn3d_branch2c[0][0], activation_18[0][0]
activation_21 (Activation)	(None, 28, 28, 512)	0	add_6[0][0]
res4a_branch2a (Conv2D)	(None, 14, 14, 256)	131,328	activation_21[0][0]
bn4a_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4a_branch2a[0][0]
activation_22 (Activation)	(None, 14, 14, 256)	0	bn4a_branch2a[0][0]
res4a_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_22[0][0]



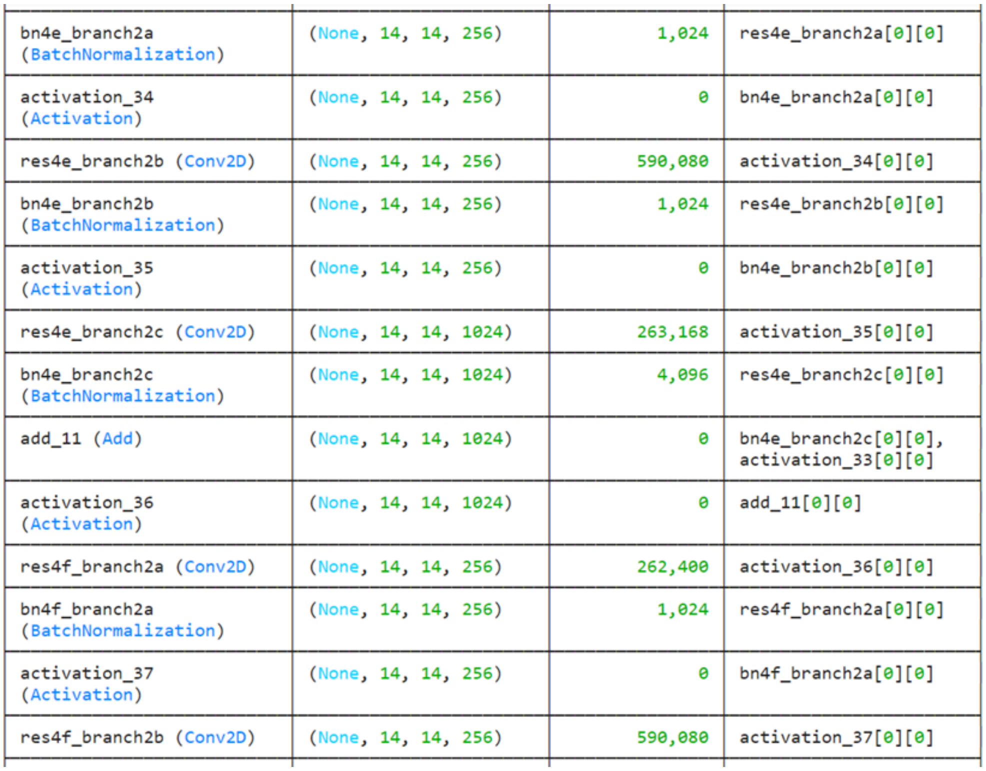
bn4a_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4a_branch2b[0][0]
activation_23 (Activation)	(None, 14, 14, 256)	0	bn4a_branch2b[0][0]
res4a_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_23[0][0]
res4a_branch1 (Conv2D)	(None, 14, 14, 1024)	525,312	activation_21[0][0]
bn4a_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4a_branch2c[0][0]
bn4a_branch1 (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4a_branch1[0][0]
add_7 (Add)	(None, 14, 14, 1024)	0	bn4a_branch2c[0][0], bn4a_branch1[0][0]
activation_24 (Activation)	(None, 14, 14, 1024)	0	add_7[0][0]
res4b_branch2a (Conv2D)	(None, 14, 14, 256)	262,400	activation_24[0][0]
bn4b_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4b_branch2a[0][0]
activation_25 (Activation)	(None, 14, 14, 256)	0	bn4b_branch2a[0][0]
res4b_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_25[0][0]
bn4b_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4b_branch2b[0][0]



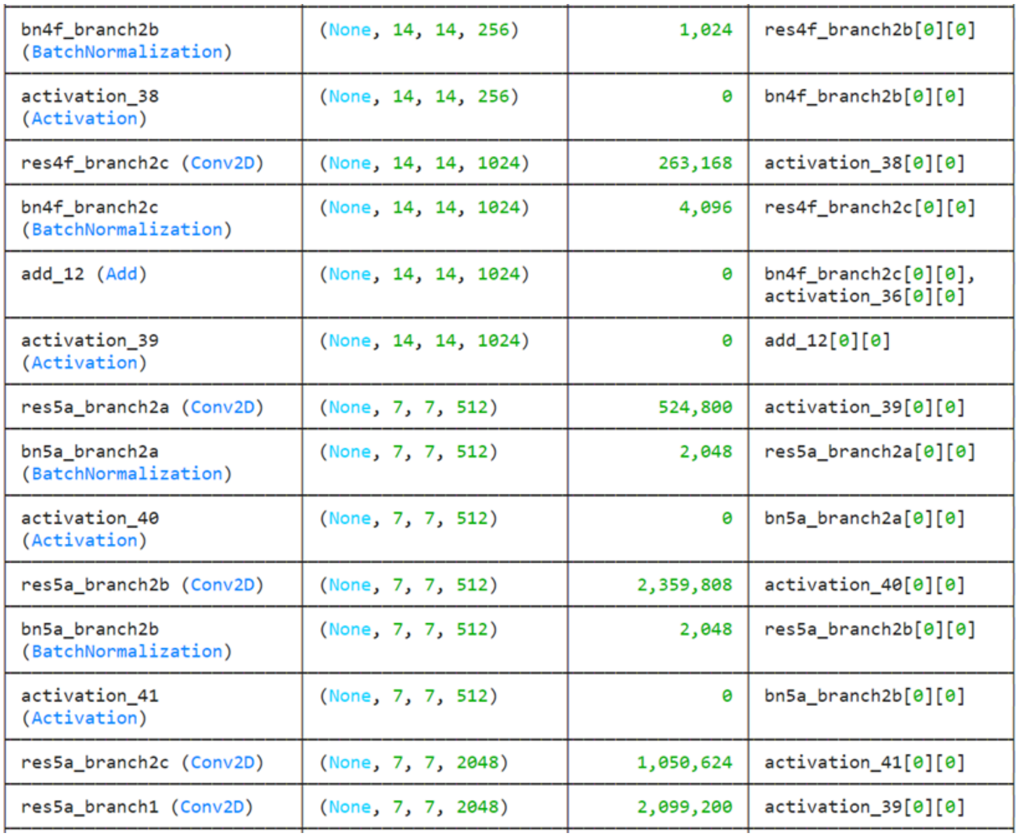
activation_26 (Activation)	(None, 14, 14, 256)	0	bn4b_branch2b[0][0]
res4b_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_26[0][0]
bn4b_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4b_branch2c[0][0]
add_8 (Add)	(None, 14, 14, 1024)	0	bn4b_branch2c[0][0], activation_24[0][0]
activation_27 (Activation)	(None, 14, 14, 1024)	0	add_8[0][0]
res4c_branch2a (Conv2D)	(None, 14, 14, 256)	262,400	activation_27[0][0]
bn4c_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4c_branch2a[0][0]
activation_28 (Activation)	(None, 14, 14, 256)	0	bn4c_branch2a[0][0]
res4c_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_28[0][0]
bn4c_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4c_branch2b[0][0]
activation_29 (Activation)	(None, 14, 14, 256)	0	bn4c_branch2b[0][0]
res4c_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_29[0][0]
bn4c_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4c_branch2c[0][0]



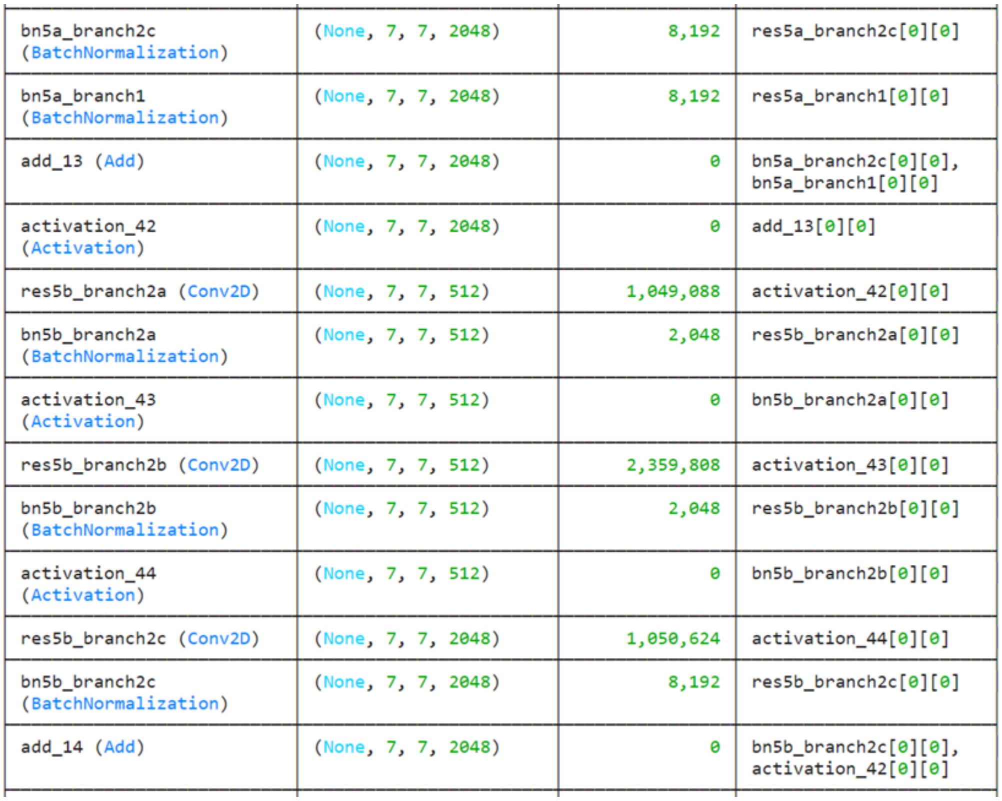
add_9 (Add)	(None, 14, 14, 1024)	0	bn4c_branch2c[0][0], activation_27[0][0]
activation_30 (Activation)	(None, 14, 14, 1024)	0	add_9[0][0]
res4d_branch2a (Conv2D)	(None, 14, 14, 256)	262,400	activation_30[0][0]
bn4d_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4d_branch2a[0][0]
activation_31 (Activation)	(None, 14, 14, 256)	0	bn4d_branch2a[0][0]
res4d_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_31[0][0]
bn4d_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4d_branch2b[0][0]
activation_32 (Activation)	(None, 14, 14, 256)	0	bn4d_branch2b[0][0]
res4d_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_32[0][0]
bn4d_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4d_branch2c[0][0]
add_10 (Add)	(None, 14, 14, 1024)	0	bn4d_branch2c[0][0], activation_30[0][0]
activation_33 (Activation)	(None, 14, 14, 1024)	0	add_10[0][0]
res4e_branch2a (Conv2D)	(None, 14, 14, 256)	262,400	activation_33[0][0]



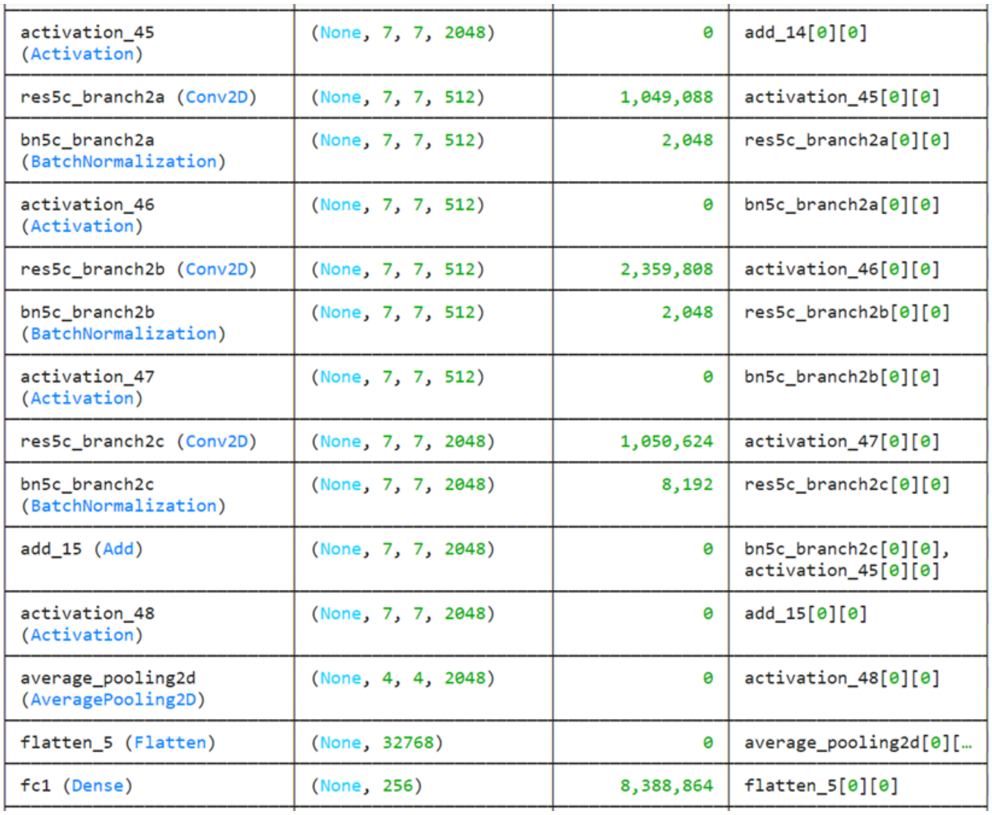
bn4e_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4e_branch2a[0][0]
activation_34 (Activation)	(None, 14, 14, 256)	0	bn4e_branch2a[0][0]
res4e_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_34[0][0]
bn4e_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4e_branch2b[0][0]
activation_35 (Activation)	(None, 14, 14, 256)	0	bn4e_branch2b[0][0]
res4e_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_35[0][0]
bn4e_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4e_branch2c[0][0]
add_11 (Add)	(None, 14, 14, 1024)	0	bn4e_branch2c[0][0], activation_33[0][0]
activation_36 (Activation)	(None, 14, 14, 1024)	0	add_11[0][0]
res4f_branch2a (Conv2D)	(None, 14, 14, 256)	262,400	activation_36[0][0]
bn4f_branch2a (BatchNormalization)	(None, 14, 14, 256)	1,024	res4f_branch2a[0][0]
activation_37 (Activation)	(None, 14, 14, 256)	0	bn4f_branch2a[0][0]
res4f_branch2b (Conv2D)	(None, 14, 14, 256)	590,080	activation_37[0][0]



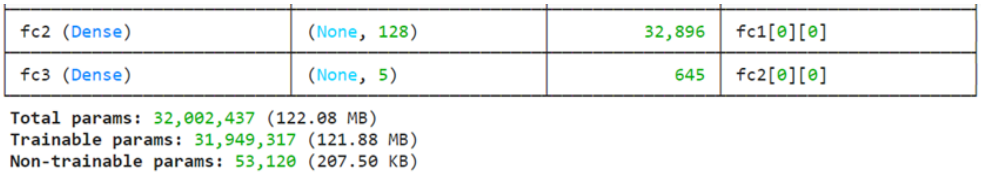
bn4f_branch2b (BatchNormalization)	(None, 14, 14, 256)	1,024	res4f_branch2b[0][0]
activation_38 (Activation)	(None, 14, 14, 256)	0	bn4f_branch2b[0][0]
res4f_branch2c (Conv2D)	(None, 14, 14, 1024)	263,168	activation_38[0][0]
bn4f_branch2c (BatchNormalization)	(None, 14, 14, 1024)	4,096	res4f_branch2c[0][0]
add_12 (Add)	(None, 14, 14, 1024)	0	bn4f_branch2c[0][0], activation_36[0][0]
activation_39 (Activation)	(None, 14, 14, 1024)	0	add_12[0][0]
res5a_branch2a (Conv2D)	(None, 7, 7, 512)	524,800	activation_39[0][0]
bn5a_branch2a (BatchNormalization)	(None, 7, 7, 512)	2,048	res5a_branch2a[0][0]
activation_40 (Activation)	(None, 7, 7, 512)	0	bn5a_branch2a[0][0]
res5a_branch2b (Conv2D)	(None, 7, 7, 512)	2,359,808	activation_40[0][0]
bn5a_branch2b (BatchNormalization)	(None, 7, 7, 512)	2,048	res5a_branch2b[0][0]
activation_41 (Activation)	(None, 7, 7, 512)	0	bn5a_branch2b[0][0]
res5a_branch2c (Conv2D)	(None, 7, 7, 2048)	1,050,624	activation_41[0][0]
res5a_branch1 (Conv2D)	(None, 7, 7, 2048)	2,099,200	activation_39[0][0]



bn5a_branch2c (BatchNormalization)	(None, 7, 7, 2048)	8,192	res5a_branch2c[0][0]
bn5a_branch1 (BatchNormalization)	(None, 7, 7, 2048)	8,192	res5a_branch1[0][0]
add_13 (Add)	(None, 7, 7, 2048)	0	bn5a_branch2c[0][0], bn5a_branch1[0][0]
activation_42 (Activation)	(None, 7, 7, 2048)	0	add_13[0][0]
res5b_branch2a (Conv2D)	(None, 7, 7, 512)	1,049,088	activation_42[0][0]
bn5b_branch2a (BatchNormalization)	(None, 7, 7, 512)	2,048	res5b_branch2a[0][0]
activation_43 (Activation)	(None, 7, 7, 512)	0	bn5b_branch2a[0][0]
res5b_branch2b (Conv2D)	(None, 7, 7, 512)	2,359,808	activation_43[0][0]
bn5b_branch2b (BatchNormalization)	(None, 7, 7, 512)	2,048	res5b_branch2b[0][0]
activation_44 (Activation)	(None, 7, 7, 512)	0	bn5b_branch2b[0][0]
res5b_branch2c (Conv2D)	(None, 7, 7, 2048)	1,050,624	activation_44[0][0]
bn5b_branch2c (BatchNormalization)	(None, 7, 7, 2048)	8,192	res5b_branch2c[0][0]
add_14 (Add)	(None, 7, 7, 2048)	0	bn5b_branch2c[0][0], activation_42[0][0]

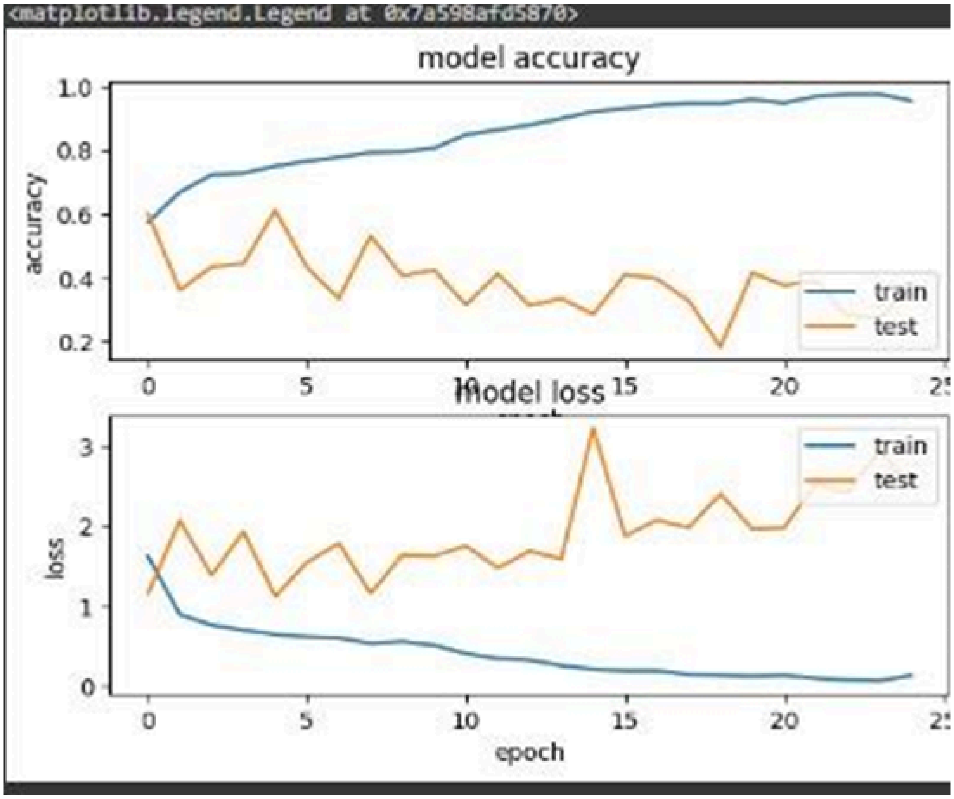


activation_45 (Activation)	(None, 7, 7, 2048)	0	add_14[0][0]
res5c_branch2a (Conv2D)	(None, 7, 7, 512)	1,049,088	activation_45[0][0]
bn5c_branch2a (BatchNormalization)	(None, 7, 7, 512)	2,048	res5c_branch2a[0][0]
activation_46 (Activation)	(None, 7, 7, 512)	0	bn5c_branch2a[0][0]
res5c_branch2b (Conv2D)	(None, 7, 7, 512)	2,359,808	activation_46[0][0]
bn5c_branch2b (BatchNormalization)	(None, 7, 7, 512)	2,048	res5c_branch2b[0][0]
activation_47 (Activation)	(None, 7, 7, 512)	0	bn5c_branch2b[0][0]
res5c_branch2c (Conv2D)	(None, 7, 7, 2048)	1,050,624	activation_47[0][0]
bn5c_branch2c (BatchNormalization)	(None, 7, 7, 2048)	8,192	res5c_branch2c[0][0]
add_15 (Add)	(None, 7, 7, 2048)	0	bn5c_branch2c[0][0], activation_45[0][0]
activation_48 (Activation)	(None, 7, 7, 2048)	0	add_15[0][0]
average_pooling2d (AveragePooling2D)	(None, 4, 4, 2048)	0	average_pooling2d[0][…
flatten_5 (Flatten)	(None, 32768)	0	average_pooling2d[0][…
fc1 (Dense)	(None, 256)	8,388,864	flatten_5[0][0]



fc2 (Dense)	(None, 128)	32,896	fc1[0][0]
fc3 (Dense)	(None, 5)	645	fc2[0][0]

```
Total params: 32,002,437 (122.08 MB)
Trainable params: 31,949,317 (121.88 MB)
Non-trainable params: 53,120 (207.50 KB)
```



```
<matplotlib.legend.Legend at 0x7a598afd5870>
```

Fig 2: Accuracy and Loss Curves for the DR Classification Model's Training and Validation

4.2 Comparative Assessment

This section presents the comparative analysis for the different reported methods from the literature against the resulting achieved using the ResNet-50 architecture from the current study.

(a) Priya, R., and P. Aruna (2013) [47] used machine learning algorithms, Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), and Support Vector Machines (SVM) in the diagnosis of diabetic retinopathy. According to the application of the lesion-based criterion, the Multi-Layer Perceptron (MLP) gave a mean sensitivity (SE) value of 88.14% and the positive predictive value (PPV) value of 80.72%. For comparison purposes, the Support Vector Machine (SVM) indicated the value of SE as 87.61% and the value of PPV as 83.51%. From the image-based criterion, the general accuracy for the MLP indicated the value of 97.01%, and the Radial Basis Function (RBF) and the SVM models indicated the value for the accuracies as 92.54% and 91.04%, respectively.

(b) Usman, Tiwalade Modupe (2023) [48] used ResNet-50, ResNet-152, and Squeezenet-1 deep models for classification and multi-label feature extraction for diabetic retinopathy detection via principal component analysis. ResNet-152 scored the best among them by having the highest rate of accuracy, 94.40%, followed by ResNet-50 with 93.67% and Squeezenet-1 by having the highest rate of 91.94.

(c) Alabdulwahhab, K.M (2021) [49] employed traditional machine learning classifiers like Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Random Forest (RF) in their "Automated detection of diabetic retinopathy using machine learning classifiers." The more advanced version of RF, referred to as RRF, performed better, classifying 86% of the patients compared to 80% accuracy for LDA and SVM in their base forms.

(d) Qummar, Sehrish (2019) [50], in their article "A deep learning ensemble approach for diabetic retinopathy detection," used an ensemble model to detect diabetic retinopathy. Their method had comparatively lower performance metrics of accuracy at 58.08%, recall at 58.10%, and precision at 70.3%.

(e) Butt, Muhammad Mohsin (2022) [51], in the article "Diabetic retinopathy detection from fundus images of the eye using hybrid deep learning features," researched hybrid deep learning features and obtained an average accuracy of 97.39% based on SVM's use. They found class-wise accuracies of 97.52% for "No DR" and 97.26% for "DR" class along with high precision, recall, and F1-scores of 97.40%.

	Proposed	[12]	[13]	[14]	[15]	[16]
Model	ResNet-50	MLP	ResNet-152	RRF	Ensemble Approach	SVM
		RBF	ResNet-50	LDA		
		SVM	SqueezeNet-1	SVM		
Accuracy	97.56	97.01	94.40	86	58.08	97.39
		92.54	93.67	80		
		91.04	91.94	80		

Table 1 : Comparative assessment of different work

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This work showcases the strength of combining deep learning with attention mechanisms for automatic diabetic retinopathy stage classification. Utilizing the ResNet-50 model, coupled with the addition of an attention layer, the network achieves excellent classification accuracy with an accuracy of 97.56%. The method efficiently detects important retinal features like microaneurysms, hemorrhages, and exudates, which are predictors of disease advancement.

Applying residual connections in ResNet-50 facilitates effective deep network training, facilitating gradient flow even to numerous layers. The inclusion of the attention mechanism further enhances the model to identify important regions in fundus images, leading to more interpretable and accurate predictions. The system also decreases the reliance on manual screening, thus making it a dependable instrument to help ophthalmologists and medical practitioners diagnose DR, particularly in its initial stages where signs are not so apparent.

The performance of the framework confirms its utility in actual clinical practice, with advantages of accelerated diagnosis, decreased clinical workload, and assistance in timely intervention to avoid loss of vision. In general, this model is a promising development toward the construction of AI-based medical diagnostic systems, especially for vision-compromising diseases like diabetic retinopathy.

In order to facilitate widespread clinical application, it is critical to fine-tune the model for real-time deployment on edge-computing devices and handheld retinal imaging devices. Such fine-tuning would permit eye care professionals to perform DR screening in rural or underserved communities without an extensive infrastructure, thus facilitating community-based screening programs and extending healthcare outreach.

Another area of prime importance is model interpretability. Increasing transparency using visualization methods like Gradient-weighted Class Activation Mapping (Grad-CAM) and attention mechanisms can assist clinicians in comprehending the model's decision-making process better. This not only establishes trust in AI-aided diagnostics but also facilitates the detection of possible sources of error or misclassification.

This technology is especially important in low-resource or rural areas, where there is limited access to trained ophthalmologists and specialized medical technology. An AI-driven diagnostic tool can be used as an upfront screening device, allowing early identification, enhancing triage effectiveness, and lowering the potential for permanent vision loss or blindness. Furthermore, the reproducibility and scalability of deep learning-driven diagnostics can decrease the load on health systems and enable large-scale public health programs targeting diabetes complication management.Although its remarkable accuracy, more work is needed to make the model more robust and generalizable across a variety of clinical settings. Increasing the training dataset to include a broad spectrum of demographic populations, imaging conditions, and device types will ensure that the model can be used reliably in practice. Moreover, incorporation of multimodal clinical information, including HbA1c values, blood pressure, patient history, and other pertinent biomarkers, can enhance the accuracy of diagnosis and allow for more tailored risk stratification and treatment planning.

5.2 Future Scope

In the future, this work can be taken in various influential directions to increase both its clinical impact and scalability. One major advancement would be optimizing the model for real-time computation, allowing its implementation in mobile or handheld diagnostic devices for use in remote or resource-limited environments. Increasing the dataset size with more representative and diverse retinal images across various demographics and imaging devices can enhance model robustness and generalizability. Including multimodal clinical information such as patient history, blood glucose levels, or OCT scans could potentially enhance diagnostic accuracy. Augmenting explainability using visual aids such as attention heatmaps or saliency maps can facilitate clinicians in comprehending and accepting the model's outputs. Future research can also investigate releasing the model through cloud-based platforms or applying federated learning for protecting the privacy of patient data while enabling large-scale use. Lastly, clinical validation and regulatory approval through clinical trials will be crucial for its use in real-world practices, which might open the door for its application in the diagnosis of other ophthalmic or systemic conditions through retinal imaging.

Looking forward, coordination with clinicians, hospitals, and regulatory agencies will be critical to validate the clinical utility of the model, achieve compliance requirements, and determine cost-efficient deployment methods. These activities will provide assurance of safe incorporation of AI-based technologies into clinical pathways and expedite the transition from research to useful healthcare applications. Finally, the integration of such AI-based tools as ResNet-50 in ophthalmology has great potential for enhancing the early detection of DR, facilitating clinical outcomes, and shaping the future of accessible, efficient, and automated healthcare.

REFERENCES

[1]Safi Hamid, Sare Safi, Ali Hafezi-Moghadam, Ahmadieh Hamid, Early detection of diabetic retinopathy, Surv. Ophthalmol.

[2]Scottish Intercollegiate Guideline Network, Management of Diabetes: a National Clinical Guideline, Scottish Intercollegiate Guidelines Network, Edinburgh, 2014.

[3]H Bresnick George, Dana B. Mukamel, John C. Dickinson, David R. Cole, A screening approach to the surveillance of patients with diabetes for the presence of vision-threatening retinopathy, Ophthalmology

[4]Linda Hill, E. Lydia, Makaroff. Early detection and timely treatment can prevent or delay diabetic retinopathy, Diabetes Res. Clin. Pract.

[5]Noemi Lois, Rachel V. McCarter, Christina O'Neill, J. Reinhold, Medina, Alan W Stitt, Endothelial progenitor cells in diabetic retinopathy, Front. Endocrinol.

[6]Ronald Klein, B.E. Klein, Scot E. Moss, The Wisconsin epidemiological study of diabetic retinopathy: a review, Diabetes Metab. Rev.

[7]L.Z. Heng, O. Comyn, T. Peto, C. Tadros, E. Ng, S. Sivaprasad, P.G. Hykin, Diabetic retinopathy: pathogenesis, clinical grading, management and future developments, Diabet. Med.

[12]Early Treatment Diabetic Retinopathy Study Research Group, Early treatment diabetic retinopathy study design and baseline patient characteristics: etdrs report number 7, Ophthalmology

[14] Luo X, Pu Z, Xu Y, Wong WK, Su J, Dou X, Ye B, Hu J, Mou L (2021) MVDRNet: Multi-view diabetic retinopathy detection by combining DCNNs and attention mechanisms. Pattern Recognition

[15] Chen W, Yang B, Li J, Wang J (2020) An Approach to Detecting Diabetic Retinopathy Based on Integrated Shallow Convolutional Neural Networks. IEEE Access

[16]Martinez-Murcia FJ, Ortiz A, Ramírez J, Górriz JM, Cruz R (2021) Deep Residual Transfer learning for Automatic Diagnosis and Grading of Diabetic Retinopathy. Neurocomputing,

[17]Das S, Kharbanda K, Suchetha M, Raman R, Dhas E (2021) Deep learning architecture based on segmented fundus image features for classification of diabetic retinopathy. Biomed Signal Processing Control

[19]Kalyani G, Janakiramaiah B, Karuna A (2023) Diabetic retinopathy detection and classification using capsule networks.

[20]Oh K, Kang HM, Leem D, Lee H, Seo KY, Yoon S, (2021) Early detection of diabetic retinopathy based on deep learning and ultra-wide-field fundus images, Scientific Reports.

[21]Erciyas A, Barısci N (2021) An Effective Method for Detecting and Classifying Diabetic Retinopathy Lesions Based on Deep Learning. Computational and Mathematical Methods in Medicine.

[22]Qureshi I, Ma J, Abbas Q (2021) Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. Multimedia Tools Applications

[23]Liu T, Chen Y, Shen H, Zhou R, Zhang M, Liu T, Liu J (2021) A Novel Diabetic Retinopathy Detection Approach Based on Deep Symmetric Convolutional Neural Network.

[24] Majumder S, Kehtarnavaz N (2021) Multitasking Deep Learning Model for Detection of Five Stages of Diabetic Retinopathy.

[25]Al-Antary MT, Arafa Y (2021) Multi-Scale Attention Network for Diabetic Retinopathy Classification.

[28] Khan Z, Khan FG, Khan A, Rehman ZU, Shah S, Qummar S, Ali F, Pack S (2021) Diabetic Retinopathy Detection Using VGG-NIN a Deep Learning Architecture.

[40]Shankar K, Zhang Y, Liu Y, Wu L, Chen CH (2020) Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification.

[41]Kumar G, Chatterjee S, Chattopadhyay C (2021) DRISTI: a hybrid deep neural network for diabetic retinopathy diagnosis. Signal, Image and Video Processing,

[42]Bhuiyan A, Govindaiah A, Deobhakta A, Hossain M, Rosen R, Smith T (2021) Automated diabetic retinopathy screening for primary care settings using deep learning.

[43]Chen PN, Lee CC, Liang CM, Pao SI, Huang KH, Lin KF (2021) General deep learning model for detecting diabetic retinopathy.

[44]Nawaz F, Ramzan M, Mehmood K, Khan HU, Khan SH, Bhutta MR (2021) Early Detection of Diabetic Retinopathy Using Machine Intelligence through Deep Transfer and Representational Learning. Computers

[45] Menaouer B, Dermane Z, El Houda Kebir N, Matta N. (2022) Diabetic Retinopathy Classification Using Hybrid Deep Learning Approach.

[46]Aziz T, Charoenlarpnopparut C, Mahapakulchai S (2023) Deep learning-based hemorrhage detection for diabetic retinopathy screening.

[47] Priya, R., and P. Aruna. "Diagnosis of diabetic retinopathy using machine learning techniques." ICTACT Journal on soft computing

[48] Usman, Tiwalade Modupe, et al. "Diabetic retinopathy detection using principal component analysis multi-label feature extraction and classification." International Journal of Cognitive Computing in Engineering

[49] Alabdulwahhab, K. M., et al. "Automated detection of diabetic retinopathy using machine learning classifiers." European Review for Medical & Pharmacological Sciences

[50] Qummar, Sehrish, et al. "A deep learning ensemble approach for diabetic retinopathy detection."

[51] Butt, Muhammad Mohsin, et al. "Diabetic retinopathy detection from fundus images of the eye using hybrid deep learning features."

APPENDIX

Dataset Sample Details

The dataset used in the project has been taken from Kaggle. This dataset consists of a total of 2,760 retinal images, each with a resolution of 256 x 256 pixels. The images represent various stages of diabetic retinopathy, categorized into five distinct classes:

Healthy: Images with no observable signs of diabetic retinopathy.

Mild DR: Images exhibiting mild symptoms of diabetic retinopathy, such as small lesions or microaneurysms.

Moderate DR: Images showing moderate signs of diabetic retinopathy, including more pronounced lesions and exudates.

Proliferative DR: Images depicting advanced diabetic retinopathy with significant neovascularization and other severe pathological features.

Severe DR: Images with severe diabetic retinopathy characteristics, such as extensive hemorrhages and retinal changes.

Healthy

Mild DR

Moderate DR

Severe DR Proliferative DR

Fig 3 : Retinal Images of different Stages of DR

Image Size (Post-Processing): 224 × 224 pixels

Software Configration

Programming Language: Python

Libraries Used: TensorFlow, Keras, OpenCV, NumPy, Matplotlib

Deep Learning Framework: TensorFlow (with Keras)

Evaluation Matrix Equation

Accuracy reflects the percentage of instances that have been correctly classified compared to the total number of instances, serving as a comprehensive indicator of performance. However, in imbalanced datasets, it may give a skewed perspective, such as diabetic retinopathy classification, as it may overestimate performance in scenarios where one class dominates. The accuracy is calculated using the following Equation (4)

$$Overall\ Classification\ Accuracy\ = \frac{Ttp+Ttn}{Ttp+Ttn+Fp+Fn} \qquad (4)$$

Ttp is true positive, Ttn is true negative, Fp is false positive, Fn is false negative.

Precision, also known as Positive Predictive Value, indicates how accurately the model identifies positive cases. It is particularly important in medical diagnoses, as a high precision reduces the rate of false positives, given by Equation (5)

$$Positive\ Predicted\ Value\ = \frac{Ttp}{Ttp+Fp} \qquad (5)$$

Recall, also known as the sensitivity or the true positive rate, measures how well the model detects actual positive cases. In diabetic retinopathy detection, a higher recall ensures that most of the cases with retinopathy are detected, followed by Equation (6)

$$True\ Positive\ Rate\ = \frac{Ttp}{Ttp+Fn} \qquad (6)$$

The F1-score provides a fair assessment of both precision and recall by representing the harmonic mean of the two. It is very useful in scenarios with imbalanced class distributions and is calculated using Equation (7)

$$F1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} \qquad (7)$$

Support plays a crucial role when dealing with imbalanced datasets, which is common in diabetic retinopathy detection. Support provides insight into the distribution of the target classes, helping to understand how the model's performance varies across different classes, especially when some classes have significantly fewer samples than others, given by Equation (8)

$$Support = Ttp + Fn \qquad (8)$$