Frontiers in Health Informatics www.healthinformaticsjournal.com
ISSN-Online: 2676-7104
2024; Vol 13: Issue 4 Open Access

Performance Analysis of InceptionV3, ResNet50 and VGG16 for Diabetic Retinopathy Detection

Rajendra Kumar
Department of Computer Science & Engineering,
School of Engineering & Technology, Sharda University, Greater Noida, India
rajendra04@gmail.com

Aman Anand
ITS Engineering College, Greater Noida, India
amananand0609@gmail.com

Nitin Jain
School of Computer Science Engineering & Technology,
Bennett University, Greater Noida, India
garg.nitin007@gmail.com

Rajat Sharma
School of Engineering and Sciences,
GD Goenka University, Gurugram, India
rajat.sharma@gdgu.org

Cite this paper as: Rajendra Kumar, Aman Anand, Nitin Jain, Rajat Sharma (2024) Performance Analysis of InceptionV3, ResNet50 and VGG16 for Diabeticn Retinopathy Detection. *Frontiers in Health Informatics, 13* (4), 320-335

Abstract **-** Diabetic retinopathy (DR) is an illness that has the potential to cause vision impairment in people who are diabetic by affecting the retina's blood vessels. The identification of DR using color fundus images typically relies on the expertise of trained medical professionals to manually detect lesions is labor-intensive and costly. This paper proposed a model based on InceptionV3, ResNet50, and VGG16 to detect and classify DR based on its severity level. The model incorporates the synthetic minority oversampling technique to address class imbalance issues commonly encountered in medical image analysis. Training and testing are conducted using the APTOS2019 and Messidor datasets, resulting in a validation accuracy of 73%, 60% and 52% for InceptionV3 and ResNet50 while VGG16 obtained a validation accuracy of 52%. Despite these promising results, the study identifies areas for improvement, such as mitigating underfitting for the VGG16 model. Strategies to enhance model generalization, optimize hyperparameters and diversify datasets should be prioritized to facilitate broader applicability across diverse clinical settings.

Keywords: Diabetes, Diabetic retinopathy, Convolutional neural network, Retinal images, Deep learning.

1. INTRODUCTION

As per the International Diabetes Federation (IDF), diabetes mellitus poses a major global health concern with projections indicating that by 2030, approximately six hundred and forty-three million people worldwide will be affected by the condition [1]. Around a third of individuals with diabetes experience an eye-related ailment associated with the condition, with DR being the most common among them [2]. It occurs when fractured blood vessels within the retina cause fluid to leak out. The presence of glucose and sugar in the bloodstream contributes to blood vessel blockage in the retina. The retina attempts to form new blood vessels in response as an effort to protect itself but they are fragile, resulting in fluid accumulation within the retina [3]. Examining the retina image for different types of lesions can be used to identify DR [4]. The lesions are composed of microaneurysms, hemorrhages, soft exudates, and hard exudates, and serve as disease indicators. Each of these lesions has its own distinct features which are elaborated upon in the following points:

- Microaneurysms are the initial indication of diabetic retinopathy, manifesting as small red spots with a size of less than 125 microns and well-defined edges. They result from the protrusion of weakened spots in the capillary walls, appearing as dots to the observer [5].
- Hemorrhages have a size exceeding 125 μm and display irregular borders [6].
- Hard exudates are characterized by their appearance as vivid-yellow dot caused by the leakage of plasma. These dots exhibit well-defined boundaries and are commonly observed in the outer retinal layers [7] .
- Soft exudates are white lesions on the retina caused by swelling of nerve fibers. They exhibit an oval or round shape [7].

1.1.1 Categorization of DR

The categorization of diabetic retinopathy into five stages is determined by the presence or absence of specific lesions. These stages are shown in Figure 1.

Figure1. The different stages of diabetic retinopathy.

Stage 1 is no DR which indicates that the retina is normal and that there are no lesions present and stage 2 is early-stage DR with minimal progression which contains solely microaneurysm. These microaneurysms may leak fluid into the surrounding retinal tissue, causing moderate retinal edema, as well as tiny dot and blot hemorrhages indicating localized bleeding caused by weakened blood arteries [8]. Stage 3 is intermediate stage DR with moderate severity, it has more lesions than microaneurysms but severity is less than stage 4.

Stage 4 represents an advanced stage DR with high severity indicating a critical stage where some signs point out severe retinal damage and higher potential for vision loss. These include more than 20 hemorrhages in each of the four quadrants, visible venous beading observed in 2 quadrants, and noticeable abnormalities in the microvasculature of the retina in 1 quadrant [9]. The final stage is proliferative DR, the retina starts secreting vaso-proliferative factors, which initiate the formation of new blood vessels. These newly produced vessels are in abnormal nature and highly fragile information [10].

1.1.2 DR screening

Patients with diabetes will need regular checks to detect DR in its early stages. Traditionally DR diagnosis is confirmed through the observation and examination of retina images by the doctor regarding the shape and appearance of lesions of different types. The amount of retina photographs that the screening program will produce will be proportional to the number of diabetic patients. It imposes huge labor-intensive costs on the medical professionals and escalates the prices of medical services; these are some of the issues and the increasing waiting list of ophthalmology consultations are the main issues facing the public health systems [11]. Such problems can be solved through an automated system for use either as an auxiliary tool for medical experts or as a full diagnostic device. Many studies have reported the adaptation of CNN. They are applied to process fundus images with the aim of real-time detection and classification of DR with a sensitivity of over 98% [12]. CNN is a type of multi-layer perceptron that takes inspiration from biological systems and can directly perceive visual patterns from raw image pixels. It utilizes an architecture comprising convolutional layers, which serve to identify specific local features present throughout the input images. The convolutional (CONV) layers in a neural network are designed to look for specific patterns or features in the input images.

Instead of looking at the whole image at once, these layers focus on small, neighboring regions and analyze them.

This helps the network to identify and understand local structures or details in the image. The significance of CNNs is their ability to share features among different parts of the image that have been analyzed. It is like having a team of detectives who learn to recognize specific things, such as shapes or patterns, and can apply that knowledge throughout the whole picture. This sharing of knowledge helps the CNNs to quickly find important features in the image and understand its different parts better [13].

Furthermore, CNN architectures often incorporate additional layers such as pooling layers and fully connected (FC) layers to enhance their capabilities. The most important information is captured by pooling layers, which help to diminish the spatial dimension of the discovered features. FC layers leverage the extracted features to make high-level predictions or classifications. The biologically inspired design of CNNs, combined with their shared weight approach, enables them to efficiently analyze large-scale image data and extract meaningful information contributing to advancements in artificial intelligence and pattern recognition. Therefore, it can help identify at-risk individuals earlier and provide timely interventions due to its effective extraction capabilities [7].

The main contributions of this study are:

- Employing pre-processing techniques to standardize and enhance image quality for improved feature extraction.
- Mitigating class imbalance by the use of synthetic minority over-sampling technique.
- Utilizing the architecture of InceptionV3 [14], ResNet50 [15], and VGG16 [16] models, incorporating diverse layers such as CONV and pooling layers, is essential for effectively processing image data. Through a fine-tuning strategy, the model balances leveraging prior knowledge from ImageNet and adapting to new data by selectively freezing and unfreezing layers to preserve learned representations.

The paper is organized as follows: Section 2 is a literature review of previous work conducted on the detection and classification of DR. Section 3 further describes the different steps involved in the proposed method. The results of the proposed models are provided in Section 4. After this, Section 5 is a discussion while Section 6 is the conclusion.

2. LITERATURE REVIEW

Due to the increasing prevalence of diabetes, there's a growing demand for reliable diagnostic methods. This has led numerous researchers to explore various algorithms and methodologies, particularly focusing on automated systems incorporating approaches like deep learning. Integration of CNNs with machine learning classifiers in the deep learning approach has enabled innovative approaches in image analysis. Gayathri et al. [17] study serves as a notable example of this collaboration. They introduced a CNN-based feature extractor combined with machine learning classifiers. Their CNN architecture comprised six CONV layers and two FC layers. Training of the network parameters involved backpropagation, and techniques such as batch normalization, dropout, and regularization were employed to address overfitting. Among the classifiers tested, the J48 decision tree-based classifier demonstrated superior performance by reducing misclassifications across datasets.

The authors attributed this success to the decision tree approach of the J48 classifier. By incorporating a CNN-based feature extractor, the study reduced computational demands and complexity, eliminating the necessity for labor-intensive handcrafted feature extraction and segmentation procedures. The CNN-based feature extractor directly captured pertinent features from retinal fundus images, consequently streamlining the computational workload and system complexity. Similarly, Das et al. [18] pioneered a genetic algorithm (GA) and support vector machine with CNN. To manually define the hyper-parameters in the CNN model can consume a significant amount of time and can cause errors. To tackle the issue, GA can be used to automatically identify the parameters. GA is inspired by natural selection that functions as a search-based optimization strategy to discover optimal solutions for problems. Therefore, it will explore an extensive range of potential parameter configurations to discover the most effective one that maximizes the CNN performance.

Developing and training CNN from scratch is a time-consuming endeavor that demands significant resources. Hence, some researchers opt for pre-trained models that employ transfer learning. Transfer learning is the process of reusing or adapting a model that has been taught to perform one task to another that is similar. Wan et al. [19], Shanthi et al. [20], Kassani et al. [21], Kajan et al. [22] and Samanta et al.[23] fine-tune the pre-trained models by making modifications to existing architectures to optimize them for specific tasks. Typically, this entails adjusting the network's hyperparameters and reinforcing the learned knowledge from the pre-trained model while training the model on task-specific data. By matching the model to the precise specifications of the intended activity, it enables the adaptation of complicated structures to new tasks.

Martinez-Murcia et al. [24] reuse the top layers of a pre-trained model while customizing the bottom layers for certain tasks. To prevent updates during the first step of this two-part training procedure, the upper layers' chosen weights are frozen. The lower layers are then fine-tuned using an optimization algorithm like Adadelta, with early stopping techniques used to establish the ideal weights based on the least loss iterations. Lu et al. [25] employed transfer learning by utilizing the ShuffleNetV2 model. They modified the ShuffleNetV2 model's last layer to provide outputs that correlate to various disease categories in order to match the requirements of the classification problem. They made an exception for the last linked layer, fine-tuning it to coincide with the intended classification objectives while utilizing pre-trained weights for the majority of layers to benefit from prior experience.

Multiple models can be combined using ensemble learning to achieve better results compared to those of single models. It enables researchers to enhance the functional capabilities of their system by making use of the diverse perspectives provided by the individual models. An ensemble learning framework employs the models on either the entire dataset or a subset of the provided data. The goal is to derive individual predictions from various models which will be combined to form the final prediction [26]. An example of ensemble learning is shown in the study proposed by Sandhya et al. [27] where multiple models are combined through the use of weighted averaging. In weighted averaging, different weights are assigned to each model based on their performance. Models with higher accuracy are given more weight and have a greater influence on the final prediction outcome whereas models with lower weights have less influence. The predictions of each model are combined by taking a weighted average.

3. METHODOLOGY

The method proposed to detect DR is illustrated in Figure 2. Initially, two datasets are gathered and merged together which undergoes preprocessing and synthetic minority oversampling technique (SMOTE) prior to training and testing. During preprocessing, the data undergoes various transformations to prepare it for analysis. SMOTE is used to address dataset imbalance by generating synthetic samples of the minority class.

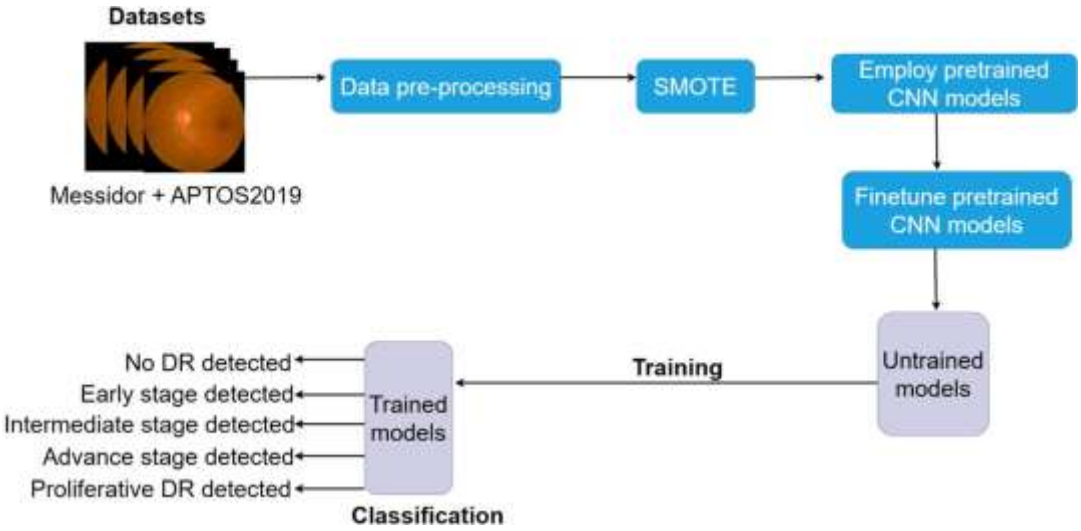

Figure 2. A workflow diagram of the proposed method.

Following SMOTE, separate pre-trained CNN models were utilized in the training phase to learn from the data and classify images into different stages of DR. Finally, the performance of each trained model is assessed on unseen data during the testing phase to determine its accuracy and reliability. Further details regarding each phase are elaborated in this section.

3.1 Datasets

The datasets used are the Kaggle APTOS2019 and Messidor datasets in which they are merged together into a total of 5, 406 images. In the APTOS2019 dataset [28], there are 3,662 images categorized from 0 to 4, aimed at assessing the severity of DR using the ICDRDSS scale [9]. Meanwhile, the Messidor-2 dataset [29] comprises 1744 images, also categorized from 0 to 4. Additionally, the Messidor-2 dataset labels stages of maculopathy or diabetic macular edema (DEM) as 'Referable DEM' (labeled 1) and 'No Referable DEM' (labeled 0). The class distribution of both datasets is shown in Table I.

TABLE I: APTOS2019 AND MESSIDOR CLASS DISTRIBUTION.

Class	**Severity level**	**APTOS 2019 dataset**		**Messidor dataset**	
		#images	***Percentage***	***#images***	***Percentage***
0	No DR	1,805	49.29%	1,017	58.31%
1	Early stage with minimal Progression.	370	10.10%	270	15.48%
2	Intermediate Stage DR with moderate severity.	999	27.28%	347	18.89%
3	Advanced stage DR with significant severity.	193	5.27%	75	4.3%
4	Proliferative DR	295	8.05%	35	2.06%

Both datasets inherently carry variations induced by distinct camera settings across centers, resulting in disparities in image quality, presence of artifacts, focus problems, and exposure discrepancies.

3.2 Pre-processing

Due to the variation exhibited by the datasets, pre-processing techniques play a key role in standardizing and enhancing the quality of the images in order for the CNN to extract meaningful features. Each of the images features a black backdrop and some images display additional dark pixels along their borders that do not contain relevant information about the retinal structures. Thus, the initial step entails cropping out the black border. Since the images vary in both width and height, the next step is to resize them to 224×224 pixels for uniformity.

After resizing, Gaussian blur was applied for removing the noise in the images. Gaussian blur creates smoother images by blending the intensity of neighboring pixels. It operates by using a Gaussian distribution to determine the significance of nearby pixels, assigning greater importance to those closest to the focal point. This prioritization ensures that distant pixels have less influence on the blurring process, resulting in a visually softer appearance.

A ratio of 40:40:20 is adopted for training, testing, and validation sets. Splitting the data, led to distribution whereby 2,162 images were allocated both to training and validation sets while 1,082 images comprised the testing set.

3.3 Synthetic Minority Over-sampling Technique

Table II showcases an evident class imbalance due to the unequal distribution of images across different DR severity levels. To mitigate this issue, SMOTE was used. SMOTE generates artificial images of the minority class to balance the class distribution. It accomplishes this by oversampling the minority class through synthetic cases rather than just copying existing examples.

In SMOTE, every instance belonging to the minority class identifies its k-nearest neighbors within the feature space (f_1 and f_2) based on the Euclidean distance metric as shown in Figure 3. By randomly choosing one neighbor, x_j, from the set of k nearest neighbors of x_i, a new sample x_{new} is generated [30]. This is done by created by combining x_i and x_j, with a random parameter, δ, determining its position between the two samples [31], based on the formula: $x_{new} = x_i + | x_i - x_j | \times \delta$.

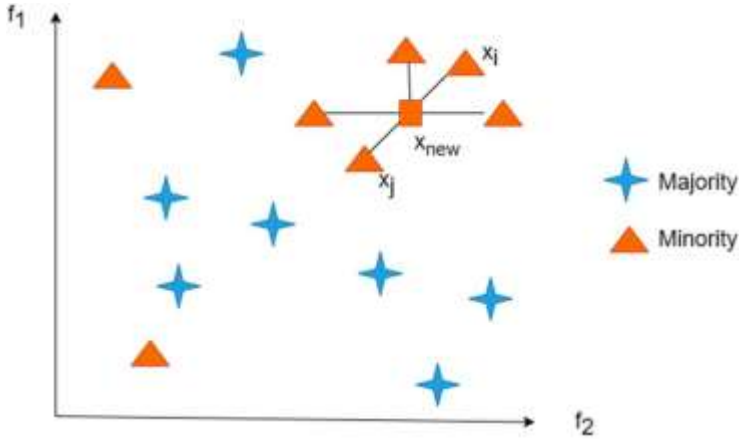

Figure 3. SMOTE schematic illustration.

In Figure 3, the stars and triangles represent the majority and minority class samples. The synthesized sample is depicted as a square shape, x_{new} emerges as the new sample situated between the existing samples x_i and x_j. After the application of SMOTE on the training dataset, it was observed that class 2 is considered to be the majority class which remained unchanged post-SMOTE. The remaining classes were balanced with each having exactly 1,127 images (Figure 4).

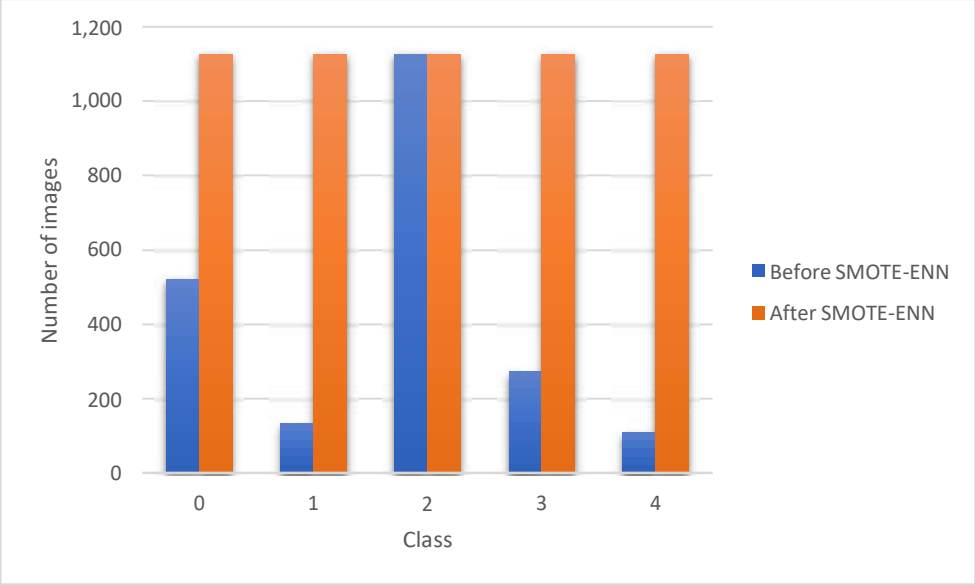

Figure 4. Number of images before and after SMOTE-ENN by class.

3.4 Pre-trained CNN models

A pre-trained CNN model is the CNN architecture that has been trained on big datasets mainly for image classification. In this study, three pre-trained CNN models are used: InceptionV3, ResNet50 and VGG16.

3.4.1 InceptionV3

The architecture of InceptionV3 consists of a deep stack of CONV layers, pooling layers, and FC layers with a total of 48 layers. One of the key features of InceptionV3 is its use of inception modules which are fundamental building blocks of the network as illustrated in Figure 5. These modules work by simultaneously processing picture data using a variety of CONV filter sizes.

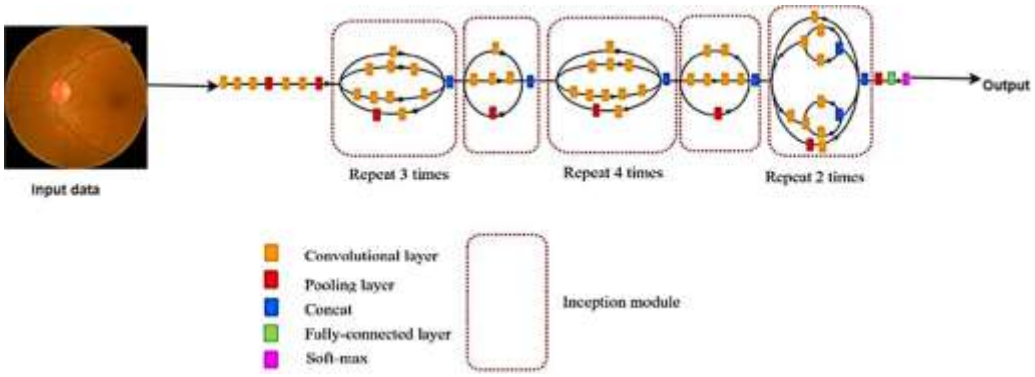

Figure 5. InceptionV3 architecture.

Parallel pathways within the network facilitate the capture of diverse spatial hierarchies and intricate patterns within images, thereby enhancing the model's capability to extract features across various scales [30]. In the training process of InceptionV3, additional supervision signals and gradients are introduced through auxiliary classifiers in intermediate layers, aiding the training of deeper networks, and classification is executed through the Softmax activation function (AF).

3.4.2 ResNet50

ResNet was introduced by Kaiming et al. [15], their approach allows layers to learn residual functions concerning the input data, which is particularly advantageous for training deeper networks. The residual block is the fundamental building block of ResNet. There are two CONV layers, followed by batch normalization and ReLU AF illustrated in Figure 6. Thereby, several layers of residual blocks are combined to produce a deep CNN.

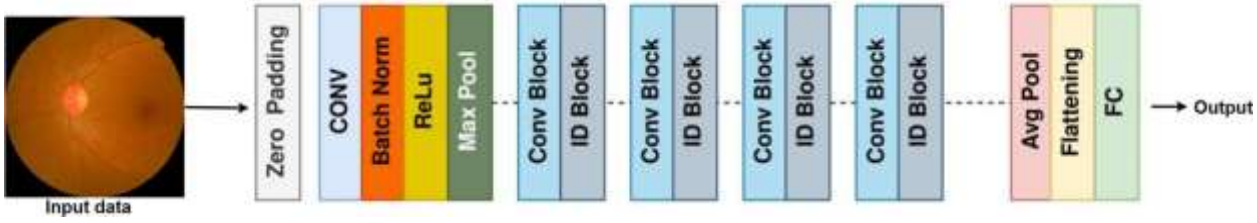

Figure 6. ResNet50 architecture.

Downsampling is used regularly within the network to lower the spatial dimensions of feature maps, which is commonly accomplished via stride CONV or pooling layers. At the network's end, global average pooling is frequently used to reduce the spatial dimensions of the feature maps to a single vector, followed by one or more FC layers for classification.

3.4.3 VGG16

VGG16 has 16 layers comprising 13 are CONV layers and 3 FC layers. The CONV layers are characterized by 3x3 filter sizes which are followed by ReLU AF [16]. As illustrated in Figure 7, the input layer accepts RGB images with dimensions of 224 × 224 pixels and following every pair of CONV layers, VGG16 includes max pooling layers that have layers have a 2 × 2 window size and a stride of 2, effectively reducing the spatial dimensions of the feature maps by half. Towards the end of the network, VGG16 features three FC layers labeled FC6, FC7, and FC8. Each fully connected layer, except the last one, is followed by a ReLU AF. The final FC layer, FC8, is followed by a softmax AF, transforming the network's output into class probabilities.

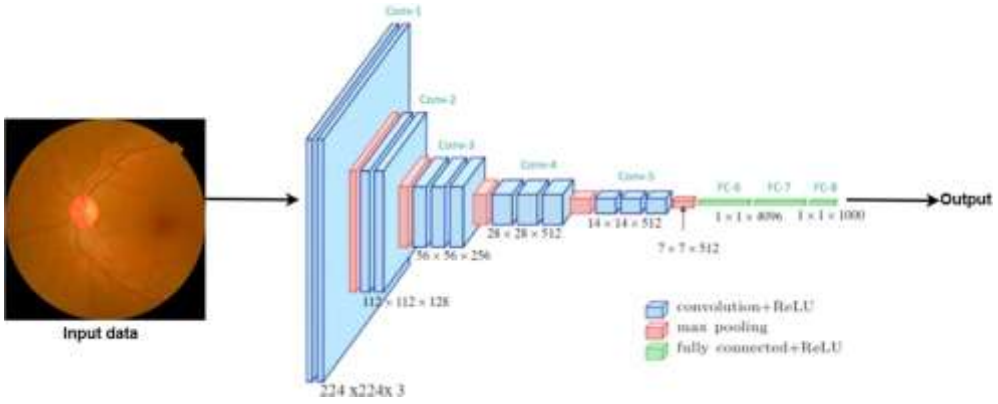

Figure 7. VGG16 architecture.

3.5 Fine-Tuning the pre-trained CNN Models

All of the model takes an input image of shape (224, 224, 3) and use the weights from ImageNet. To balance the use of pre-trained knowledge with the adaptation to new data, a fine-tuning strategy is employed. This strategy entails selectively freezing and unfreezing layers within the model architecture. All of the layers were frozen except for the last layer thereby maintaining their current weights and preventing updates during training. Freezing layers in a neural network maintain the learned representations from pre-trained weights, capturing valuable high-level features applicable across different domains. By retaining these representations, models benefit from previously acquired knowledge, enabling quicker training without demanding high-end hardware. However, frozen layers yield lower accuracy compared to unfrozen ones since they extract fewer features from the dataset. Unfreezing layers, on the other hand, extract richer hidden features but require more time and robust system hardware for training [32]

Subsequently, a sequential model serves as the overarching structure for organizing the architecture's components. Upon integrating the model into the sequential structure, the feature extraction process continues with the inclusion of a flatten layer. This layer reshapes the multidimensional output of the CONV layers into a flattened one-dimensional array. Furthermore, two dense layers with 128 units were integrated with a ReLU activation function to learn complex patterns from the flattened output. L2 regularization is applied to the kernel weights of these dense layers to mitigate overfitting and dropout layers with a dropout rate of 0.35 are introduced after each dense layer for additional regularization. The final layer is an output layer comprising 5 units and utilizes softmax activation that outputs the probabilities of the input image severity level.

3.6 Training of pretrained models

The training process involved utilizing the Stochastic Gradient Descent (SGD) optimizer. SDG is an iterative optimization algorithm that adjusts the model parameters using a gradient obtained from a single training example or a small subset. This adds randomness to the parameter updates, allowing SGD to efficiently navigate the datasets and avoid local minima [33]. The learning rate was set as 0.0001 and the loss function was categorical cross-entropy. This configuration was trained on the model for 10 epochs. In each epoch, the early stopping callback from Keras was called to track the validation loss during the time of training. You set patience 5, and the training stops when the validation loss does not improve for 5 consecutive epochs. In this

model, you are trying to minimize the difference between predicted and actual values by adjusting the weights based on gradients calculated from the loss function.

3.7 Testing of pretrained models

In the testing phase, the model's learned parameters are employed to make predictions on unseen data. These predictions are compared against the ground truth labels to assess the model's performance across various metrics such as accuracy, precision, recall, and specificity.

3.8 Performance measure

A confusion matrix is one of the commonly used evaluation techniques for a categorization model. It provides information about four possible scenarios that may arise from the output made by the model. True Positive, TP is when a model identifies a positive instance. A False Positive, FP is where a negative instance is presented as a positive one. It means True Negative (TN) when the model correctly identifies a negative occurrence. Finally, False Negative (FN) is when the model incorrectly predicts that a given instance is negative while there exists a positive class in reality. The following are metrics derived from the confusion matrix:

- Accuracy: It is the ratio of accurate predictions to the total predictions [34].
- Precision: It concentrates on accurately classifying negative instances and calculates the proportion of correctly classified negative classes [35].
- Sensitivity/recall: It measures how well the model identifies positive labels by focusing on reducing false negatives. It reflects the model's effectiveness in accurately recognizing and classifying positive instances [36].
- Specificity: It complements sensitivity by representing the percentage of normal images correctly identified as normal [37].

TABLE II: CONFUSION MATRIX FORMULA.

Metrix	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**
Formula	$\frac{TP + TN}{TN + TP + FN + FP}$	$\frac{TP}{TP + FP}$	$\frac{TP}{TP + FN}$	$\frac{TN}{TN + FP}$

4. RESULTS

The model training and testing process takes place in Kaggle's notebook environment, taking advantage of the platform's extensive datasets and well-known Python libraries such as TensorFlow and Keras. Kaggle's notebook execution takes place on cloud-based servers, with a Graphics Processing Unit (GPU) called GPU P100. The GPU P100 is known for its parallel processing capabilities and accelerates computational workloads, allowing algorithms to run quickly.

4.1 Train and validation results

For the InceptionV3 model, it obtains a training accuracy of 88% and a validation accuracy of 73%. Regarding the loss metrics, it reflects how much the model's predictions deviate from the actual target values. The training loss consistently decreases throughout the epochs which is the number of times the dataset is transmitted forward and backward through the model during training, converging to levels below 0.8 as shown Figure 8.

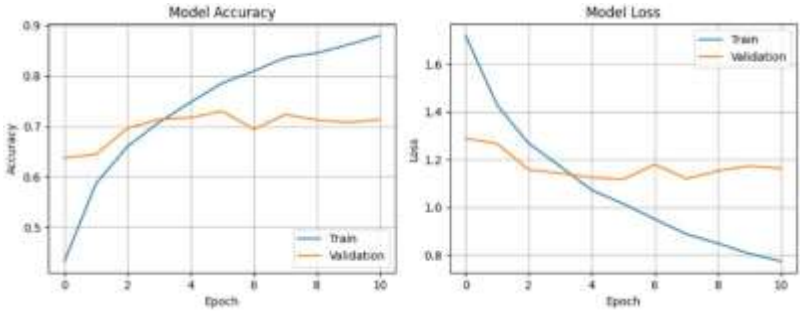

Figure 8. Inception model accuracy and loss curve.

It suggests that the model effectively learns from the training data. Conversely, the validation loss displays fluctuations within the range of 1.4 to 1.0, indicating instances of overfitting. The ResNet50 model obtained a training accuracy of 30% and a validation of 60%. The training and validation loss shown in Figure 9 consistently decreases throughout the epochs which is the number of times the dataset is transmitted forward and backward through the model during training, converging to levels below 1.4.

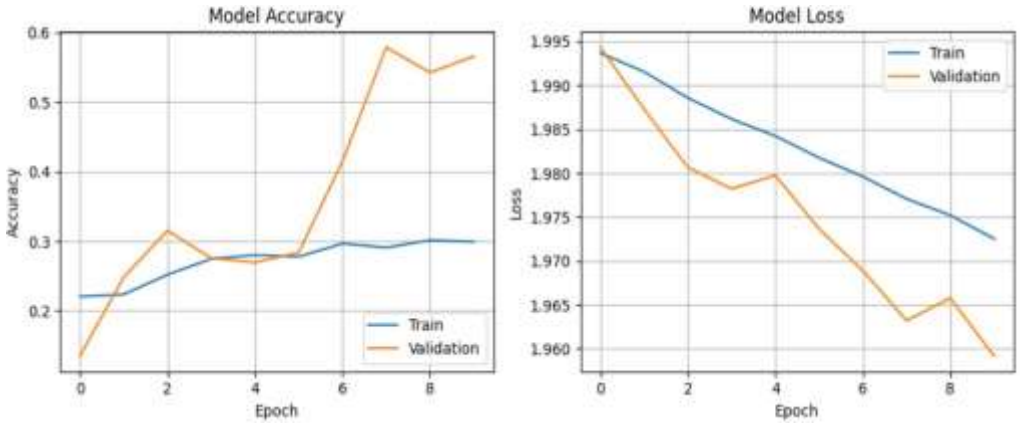

Figure 9. ResNet50 model accuracy and loss curve.

The training accuracy was 19%, and the validation accuracy was 60%. For the loss vs. epochs of training and validation, as shown in Figure 10, the loss values increase as the epochs get higher, which indicates that the model is underfitted.

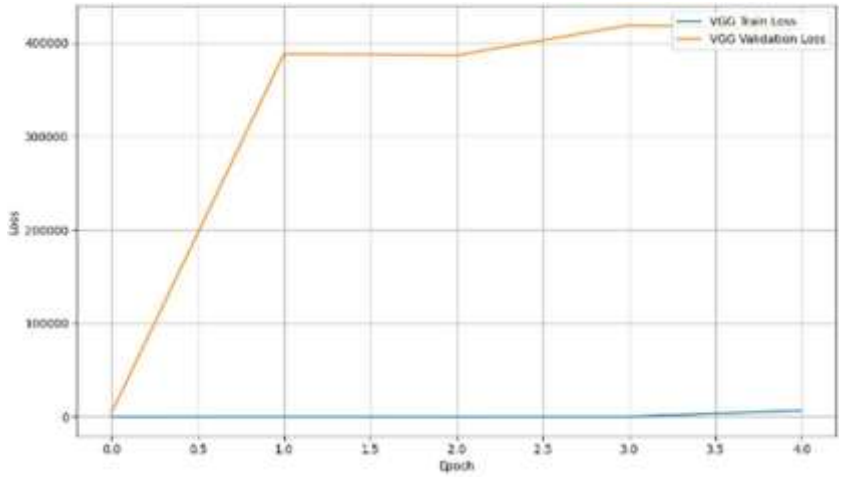

Figure 10. VGG16 loss curve.

4.2 Test results

The inceptionV3 model obtained a test accuracy of 72% and the confusion matrix in Figure 11 reveals notable diagonal values such as 350, 36, 107, 6, and 17. These values indicate precise predictions within each respective class, suggesting the model's capability to accurately classify instances. The existence of off-diagonal elements in the matrix signifies misclassifications, pointing out regions where the model may encounter difficulties or inconsistencies.

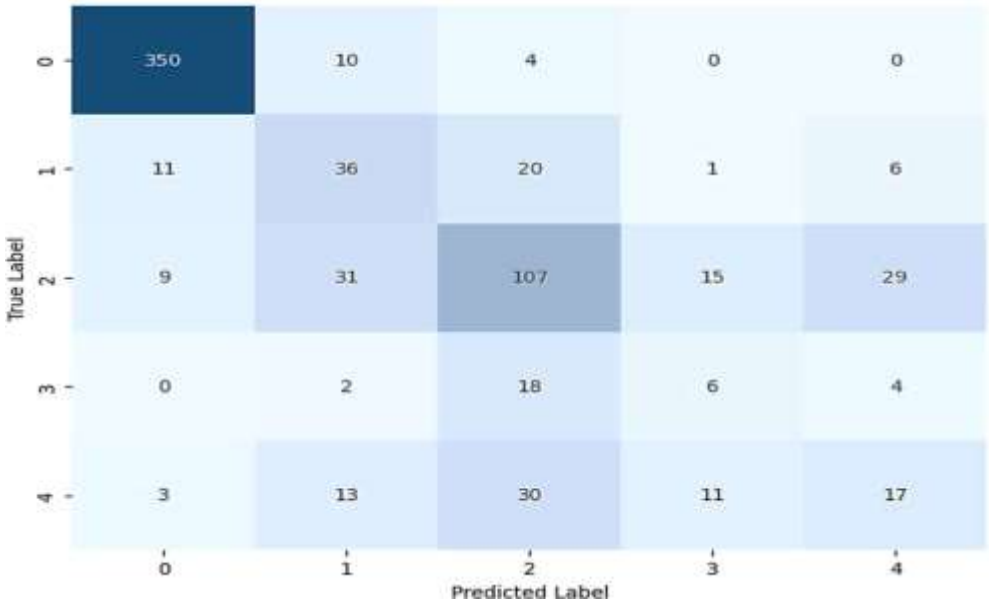

Figure 11. InceptionV3 confusion matrix.

The ResNet50 obtained a test accuracy of 57%.and the confusion matrix in Figure 12 shows high accuracy in correctly classifying instances from class 0 with 277 and 88 accurate predictions. However, it encounters challenges when classifying instances from the other classes. For example, it accurately identifies only 36 instances of class 1, while misclassifying a considerable number of class 1 instances as class 0 or class 4. Similarly, although correctly predicting 107 instances of class 2, the model misclassifies a significant portion as belonging to classes 0, 1,3, or 4. Moreover, class 4 presents difficulties for the model as none of the instances were able to be correctly classified.

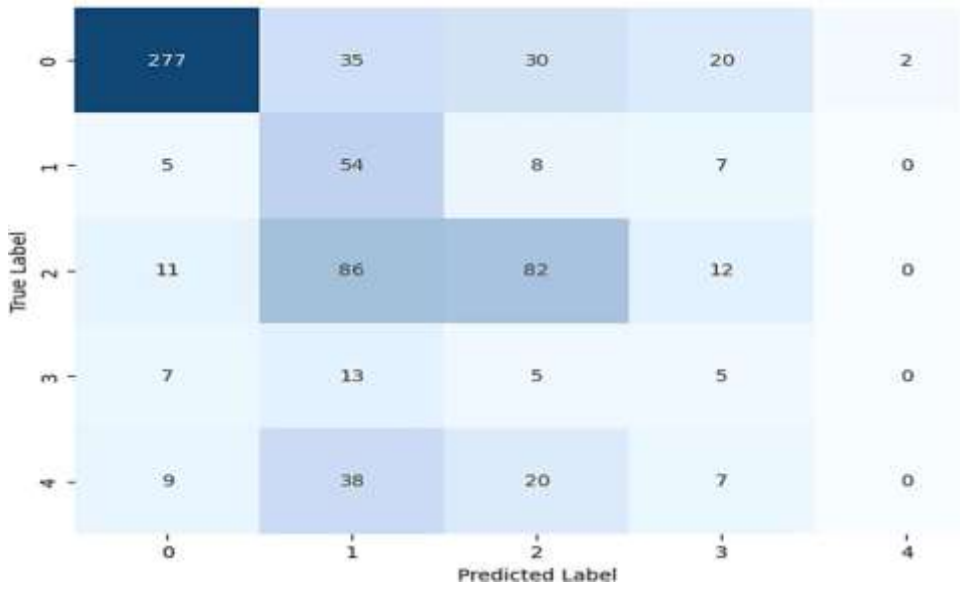

Figure 12. ResNet50 confusion matrix.

The confusion matrix for the VGG16 model in Figure 13 shows that the model consistently predicts only one class for all instances, resulting in all instances being classified into a single category. It predicts all instances to belong to a single class (Class 2 with 1122 instances) while making no predictions for the other classes. Therefore, it fails to differentiate between different classes and makes no correct predictions for any class other than the one it consistently predicts.

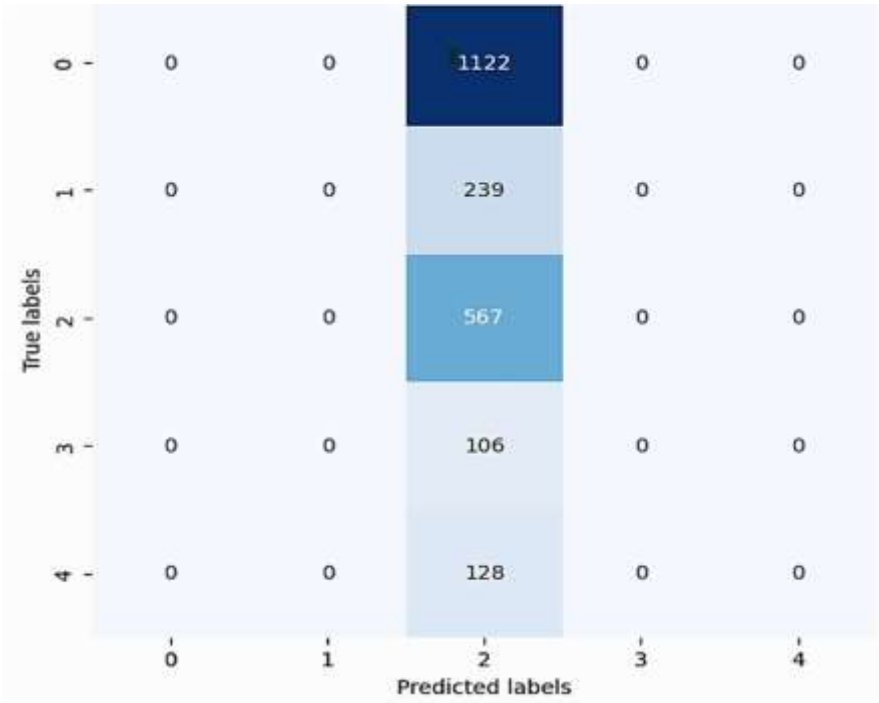

Figure 13. VGG16 confusion matrix.

The code uses values of the TP, TN, FP, and FN brought out by the confusion matrix for calculations of the performance measure as stated in section 3.8. Table III shows the overall performance result of each model.

TABLE III: PERFORMANCE EVALUATION.

Model	Accuracy	Precision	Sensitivity	Specificity
Training set				
InceptionV3	88%	88%	88%	97%
ResNet50	30%	32%	25%	89%
VGG16	19%	40%	20%	80%
Validation set				
InceptionV3	73%	51%	51%	93%
ResNet50	60%	36%	39%	94%
VGG16	52%	52%	20%	80%
Testing set				
InceptionV3	51%	34%	35%	87%
ResNet50	58%	37%	38%	88%
VGG16	23%	48%	20%	80%

5. DISCUSSION

This study delves into DR detection, using a merged dataset from APTOS2019 and Messidor, comprising 5,406 images. For example, SMOTE comes to address the problem of class imbalance, usually found in medical imaging applications, by the synthesis of samples from the minority class. Transfer learning lets the researchers take the knowledge learned from highly trained models such as ImageNet and be able to use it for other tasks. By transferring this learned knowledge, the learning process for a specific task can be enhanced, allowing for more efficient and effective model training [38]. By incorporating pre-trained models, researchers can capitalize on existing feature extraction capabilities and adapt them to specific diagnostic contexts.

In InceptionV3, the model suffered from overfitting which occurs when the model becomes too skilled at learning from the training data, leading to a decline in performance when faced with unfamiliar data. The occurrence of overfitting might be attributed to the generation of synthetic samples by SMOTE which may not accurately represent the underlying data distribution [39]. When SMOTE generates synthetic samples without considering the characteristics or distribution of the minority class, it can lead to unrealistic or irrelevant data points. Therefore, it can contribute to the formation of an overly complex decision boundary [40], increasing the likelihood of class overlap. It also plays a role in the underfitting observed in VGG16, which is attributed to its simpler architecture compared to ResNet50 and InceptionV3. VGG16 has only 16 layers, whereas InceptionV3 and ResNet50 boast 48 and 50 layers respectively. These deeper architectures leverage advanced features such as skip connections and inception modules, enhancing gradient flow and feature extraction to capture more intricate patterns Hence, it might pose a challenge for VGG16 to effectively manage such data distributions.

Although the ResNet50 model demonstrated a good fit, the disparity between the validation accuracy and training accuracy suggests a bias within the datasets. To address this, acquiring more data and extend the training duration can be used. The study has two drawbacks. Firstly, its reliance on existing datasets may limit its applicability to diverse clinical settings, highlighting the need for more varied datasets. Secondly, the lower validation and testing accuracies in VGG16 suggest the necessity for further refinement. To enhance the model's robustness, future research should prioritize acquiring diverse datasets that encompass a broad spectrum of imaging conditions encountered in clinical practice. Additionally, exploring alternative data resampling techniques and detection models to compare and evaluate their respective strengths and weaknesses, ultimately leading to informed decision-making in selecting the most suitable approach for a given task.

6. CONCLUSION

Diabetic retinopathy caused by diabetes mellitus can slowly damage the retina or even lead to loss of vision. Identifying the condition in the early stages can forestall the progression of DR and prevent permanent damage to the retina. The development of an automated DR detection system will minimize the time taken for DR screening. Leveraging techniques such as SMOTE for class balance and transfer learning. Legering pre-trained models like InceptionV3 and ResNet50, which makes it pretty easy to leverage the valuable insights from extensive datasets like ImageNet to detect and classify DR based on its severity level. However, challenges emerged during model implementation. Overfitting was observed in InceptionV3, possibly stemming from SMOTE-generated synthetic samples that may not accurately represent the true data distribution. Conversely, VGG16 exhibited underfitting due to its simpler architecture compared to InceptionV3 and ResNet50. The latter models, with their deeper architectures and advanced features, better manage complex data distributions. While ResNet50 demonstrated a good fit, the discrepancy between validation and training accuracies suggests dataset bias, highlighting the need for more extensive data acquisition and prolonged training durations. Future research directions can focus on diversifying the datasets, exploring alternative data resampling methods, and optimizing model architectures in order to further improve its generalization capabilities.

REFERENCES

[1] K. Ogurtsova , J .D. Fernandes, Y.Huang , U. Linnenkamp , L.Guariguata , N.H.Cho , D.Cavan , J .E .Shaw , L E.Makaroff, "IDF Diabetes Atlas: Global estimates for the prevalence of diabetes for 2015 and 2040," *Diabetes Research and Clinical Practice*, vol. 128, pp. 40–50, Jun. 2017, doi: 10.1016/j.diabres.2017.03.024.

[2] A. J. Jenkins, M. V. Joglekar, A. A. Hardikar, A. C. Keech, D. N. O'Neal, and A. S. Januszewski, "Biomarkers in Diabetic Retinopathy," *Rev Diabet Stud*, vol. 12, no. 1–2, pp. 159–195, 2015, doi: 10.1900/RDS.2015.12.159.

[3] A. V. Prasad, A. Gupta, L. J. Anbarasi, and M. Jawahar, "Current Trends, Challenges, and Future Prospects for automated detection of Diabetic Retinopathy," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, Coimbatore, India: IEEE, May 2021, pp. 340–343. doi: 10.1109/ICSPC51351.2021.9451643.

[4] N. K. Baskaran and T. R. Mahesh, "Performance Analysis of Deep Learning based Segmentation of Retinal Lesions in Fundus Images," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India: IEEE, Mar. 2023, pp. 1306–1313. doi: 10.1109/ICEARS56392.2023.10085616.

[5] S. S. Rahim, V. Palade, J. Shuttleworth, C. Jayne, and R. N. R. Omar, "Automatic Detection of Microaneurysms for Diabetic Retinopathy Screening Using Fuzzy Image Processing," in *Engineering Applications of Neural Networks*, vol. 517, L. Iliadis and C. Jayne, Eds., in Communications in Computer and Information Science, vol. 517., Cham: Springer International Publishing, 2015, pp. 69–79. doi: 10.1007/978-3-319-23983-5_7.

[6] S. Weis, M. Sonnberger, A. Dunzinger, E. Voglmayr, M. Aichholzer, R. Kleiser, P. Strasser, "Vascular Disorders: Hemorrhage," in *Imaging Brain Diseases*, Vienna: Springer Vienna, 2019, pp. 499–536. doi: 10.1007/978-3-7091-1544-2_19.

[7] W. L. Alyoubi, W. M. Shalash, and M. 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.

[8] A. J. Paul, "Advances in Classifying the Stages of Diabetic Retinopathy Using Convolutional Neural Networks in Low Memory Edge Devices," Ophthalmology, preprint, Jul. 2021. doi: 10.1101/2021.07.29.21261337.

[9] C. P. Wilkinson, F. L. Ferris, R. E. Klein, P. P. Lee, C. D. Agardh, M. Davis, D. Dills, A. Kampik, R. Pararajasegaram, J. T.Verdaguer, "Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales," *Ophthalmology*, vol. 110, no. 9, pp. 1677–1682, Sep. 2003, doi: 10.1016/S0161-6420(03)00475-5.

[10] W. L. Yun, U. R. Acharya, Y. V. Venkatesh, C. Chee, L. C. Min, and E. Y. K. Ng, "Identification of different stages of diabetic retinopathy using retinal optical images," *Information Sciences*, vol. 178, no. 1, pp. 106–121, Jan. 2008, doi: 10.1016/j.ins.2007.07.020.
[11] M. Z. Atwany, A. H. Sahyoun, and M. Yaqub, "Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey," *IEEE Access*, vol. 10, pp. 28642–28655, 2022, doi: 10.1109/ACCESS.2022.3157632.
[12] A.-O. Asia, C.Z. Zhu, S. A. Althubiti, D. Al-Alimi, Y. L. Xiao, P.B. Ouyang, M. A. A. Al-Qaness,"Detection of Diabetic Retinopathy in Retinal Fundus Images Using CNN Classification Models," *Electronics*, vol. 11, no. 17, p. 2740, Aug. 2022, doi: 10.3390/electronics11172740.
[13] N. Tajbakhsh, J. Y. Shin, S.R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, J. Liang, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016, doi: 10.1109/TMI.2016.2535302.
[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2015, doi: 10.48550/ARXIV.1512.00567.
[15] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, 2014, doi: 10.48550/ARXIV.1409.1556.
[17] G. S., V. P. Gopi, and P. Palanisamy, "A lightweight CNN for Diabetic Retinopathy classification from fundus images," *Biomedical Signal Processing and Control*, vol. 62, p. 102115, Sep. 2020, doi: 10.1016/j.bspc.2020.102115.
[18] S. Das and S. K. Saha, "Diabetic retinopathy detection and classification using CNN tuned by genetic algorithm," *Multimed Tools Appl*, vol. 81, no. 6, pp. 8007–8020, Mar. 2022, doi: 10.1007/s11042-021-11824-w.
[19] S. Wan, Y. Liang, and Y. Zhang, "Deep convolutional neural networks for diabetic retinopathy detection by image classification," *Computers & Electrical Engineering*, vol. 72, pp. 274–282, Nov. 2018, doi: 10.1016/j.compeleceng.2018.07.042.
[20] T. Shanthi and R. S. Sabeenian, "Modified Alexnet architecture for classification of diabetic retinopathy images," *Computers & Electrical Engineering*, vol. 76, pp. 56–64, Jun. 2019, doi: 10.1016/j.compeleceng.2019.03.004.
[21] S. H. Kassani, P. H. Kassani, R. Khazaeinezhad, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Diabetic Retinopathy Classification Using a Modified Xception Architecture," in *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Ajman, United Arab Emirates: IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/ISSPIT47144.2019.9001846.
[22] S. Kajan, J. Goga, K. Lacko, and J. Pavlovicova, "Detection of Diabetic Retinopathy Using Pretrained Deep Neural Networks," in *2020 Cybernetics & Informatics (K&I)*, Velke Karlovice, Czech Republic: IEEE, Jan. 2020, pp. 1–5. doi: 10.1109/KI48306.2020.9039793.
[23] A. Samanta, A. Saha, S. C. Satapathy, S. L. Fernandes, and Y.-D. Zhang, "Automated detection of diabetic retinopathy using convolutional neural networks on a small dataset," *Pattern Recognition Letters*, vol. 135, pp. 293–298, Jul. 2020, doi: 10.1016/j.patrec.2020.04.026.
[24] F. J. Martinez-Murcia, A. Ortiz, J. Ramírez, J. M. Górriz, and R. Cruz, "Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy," *Neurocomputing*, vol. 452, pp. 424–434, Sep. 2021, doi: 10.1016/j.neucom.2020.04.148.
[25] Z. Lu, J. Miao, J. Dong, S. Zhu, X. Wang, and J. Feng, "Automatic classification of retinal diseases with transfer learning-based lightweight convolutional neural network," *Biomedical Signal Processing and Control*, vol. 81, p. 104365, Mar. 2023, doi: 10.1016/j.bspc.2022.104365.
[26] J. P. Kandhasamy, S. Balamurali, S. Kadry, and L. K. Ramasamy, "Diagnosis of diabetic retinopathy using multi level set segmentation algorithm with feature extraction using SVM with selective features," *Multimed Tools Appl*, vol. 79, no. 15–16, pp. 10581–10596, Apr. 2020, doi: 10.1007/s11042-019-7485-8.
[27] M. Sandhya, M. K. Morampudi, R. Grandhe, R. Kumari, C. Banda, and N. Gonthina, "Detection of Diabetic Retinopathy (DR) Severity from Fundus Photographs: An Ensemble Approach Using Weighted Average," *Arab J Sci Eng*, vol. 47, no. 8, pp. 9899–9906, Aug. 2022, doi: 10.1007/s13369-021-06381-1.

[28] Asia Pacific Tele-Ophthalmology Society (APTOS), "APTOS 2019 Blindness Detection." [Online]. Available: https://www.kaggle.com/competitions/aptos2019-blindness-detection
[29] "Messidor." [Online]. Available: https://www.adcis.net/en/third-party/messidor/
[30] J. Luengo, A. Fernández, S. García, and F. Herrera, "Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling," *Soft Comput*, vol. 15, no. 10, pp. 1909–1936, Oct. 2011, doi: 10.1007/s00500-010-0625-8.
[31] Y. Yan, R. Liu, Z. Ding, X. Du, J. Chen, and Y. Zhang, "A Parameter-Free Cleaning Method for SMOTE in Imbalanced Classification," *IEEE Access*, vol. 7, pp. 23537–23548, 2019, doi: 10.1109/ACCESS.2019.2899467.
[32] S. Veeragandham and H. Santhi, "Effectiveness of convolutional layers in pre-trained models for classifying common weeds in groundnut and corn crops," *Computers and Electrical Engineering*, vol. 103, p. 108315, Oct. 2022, doi: 10.1016/j.compeleceng.2022.108315.
[33] N. Ketkar, "Stochastic Gradient Descent," in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 113–132. doi: 10.1007/978-1-4842-2766-4_8.
[34] T. J. Jebaseeli, C. A. Deva Durai, and J. D. Peter, "Retinal blood vessel segmentation from diabetic retinopathy images using tandem PCNN model and deep learning based SVM," *Optik*, vol. 199, p. 163328, Dec. 2019, doi: 10.1016/j.ijleo.2019.163328.
[35] S. Chavan and N. Choubey, "An automated diabetic retinopathy of severity grade classification using transfer learning and fine-tuning for fundus images," *Multimed Tools Appl*, vol. 82, no. 24, pp. 36859–36884, Oct. 2023, doi: 10.1007/s11042-023-15135-0.
[36] R. Kumar, R. C. Singh, and S. Kant, "Dorsal Hand Vein Recognition Using Very Deep Learning," *Macromolecular Symposia*, vol. 397, no. 1, p. 2000244, Jun. 2021, doi: 10.1002/masy.202000244.
[37] R. Kumar, R. C. Singh, and S. Kant, "Dorsal Hand Vein-Biometric Recognition Using Convolution Neural Network," in *International Conference on Innovative Computing and Communications*, vol. 1165, D. Gupta, A. Khanna, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds., in Advances in Intelligent Systems and Computing, vol. 1165., Singapore: Springer Singapore, 2021, pp. 1087–1107. doi: 10.1007/978-981-15-5113-0_92.
[38] N. Tsiknakis, D.Theodoropoulos, G. Manikis, E. Ktistakis, O. Boutsora, A. Berto, F. Scarpa, A. Scarpa, D. Fotiadis, K. Marias, "Deep learning for diabetic retinopathy detection and classification based on fundus images: A review," *Computers in Biology and Medicine*, vol. 135, p. 104599, Aug. 2021, doi: 10.1016/j.compbiomed.2021.104599.
[39] Haibo He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
[40] D. Elreedy and A. F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance," *Information Sciences*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.