DIABETIC RETINOPATHY USING CNN ALGORITHM

A Final Project Report

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

O. GLYSIA	**20191A0401**
K. DEVENDRA	**20191A0409**
S. NOORULLA	**20191A0456**
B. HEMANTH	**20191A0460**
S. NOOR BASHA	**21195A0407**

Under the esteemed guidance of

Dr. SHAIK TAJ MAHABOOB **M. Tech, Ph. D**

Associate Professor & Head of Dept.

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING (AUTONOMOUS) PULIVENDULA - 516390,

Y.S.R. (District), Andhra Pradesh, INDIA

2020 - 2024

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING (AUTONOMOUS) PULIVENDULA

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "**DIABETIC RETINOPATHY USING CNN ALGORITHM**" has been submitted by

O. GLYSIA	**20191A0401**
K. DEVENDRA	**20191A0409**
S. NOORULLA	**20191A0456**
B. HEMANTH	**20191A0460**
S. NOOR BASHA	**21195A0407**

in partial fulfillment of the requirements for the award of the **Bachelor of Technology** in **Electronics and Communication Engineering** from **JNTUA College of Engineering (Autonomous), Pulivendula** during 2020-2024.

Signature of the Project Guide

Dr. SHAIK TAJ MAHABOOB

M. Tech, Ph. D

Associate Professor & Head,
Department of ECE,
JNTUACEP, Pulivendula.

Signature of Head of the Department

Dr. SHAIK TAJ MAHABOOB

M. Tech, Ph. D

Associate Professor & Head,
Department of ECE,
JNTUACEP, Pulivendula.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING (AUTONOMOUS), PULIVENDULA - 516390,

Y.S.R. (District), ANDHRA PRADESH, INDIA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

STUDENT DECLARATION

We hereby declare that the project entitled " **DIABETIC RETINOPATHY USING CNN ALGORITHM** " is our work and that to the best of our knowledge and belief, it contains no material previously published or material that has been accepted for the award of any degree or diploma of any University or institute of higher learning.

O. GLYSIA	**20191A0401**
K. DEVENDRA	**20191A0409**
S. NOORULLA	**20191A0456**
B. HEMANTH	**20191A0460**
S. NOOR BASHA	**21195A0407**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude to all of them. The first person we would like to thank is our guide **Dr. SHAIK TAJ MAHBOOB,** Associate Professor and Head of the Department of Electronics and Communication Engineering, JNTUA College of Engineering, Pulivendula. Her wide knowledge and logical way of thinking have made a deep impression on us.

We are thankful to **Prof. M. SURYANARAYANA REDDY**, Vice Principal of the college, JNTUA College of Engineering, Pulivendula who has extended his support by providing infrastructural facilities.

We wish to thank **Prof. R. RAMANA REDDY,** Principal of the college, JNTUA College of Engineering, Pulivendula who has extended his support by providing infrastructural facilities.

Finally, we would like to extend our deep sense of gratitude to all the staff members, friends and last but not least we are greatly indebted to our parents who inspired us in all circumstances.

ABSTRACT

The diagnosis of Diabetic Retinopathy from color Fundus images traditionally relies on the expertise of clinicians to identify subtle features and classify its severity based on a complex grading system, which is time-consuming and prone to human error. In this project, we propose a Convolutional Neural Network (CNN) approach using the Inception V3 model to automate the diagnosis and classification of DR from digital Fundus images.

The Inception V3 model, known for its effectiveness in image recognition tasks, is employed to extract intricate features crucial for DR classification. Leveraging a high-end graphics processing unit (GPU), the proposed CNN architecture is trained on a publicly available Kaggle dataset consisting of a large number of Fundus images annotated with DR severity levels. The model is designed to accurately classify images into categories representing various stages of DR severity, including mild, medium, severe, proliferative, and Normal.

Through extensive experimentation and evaluation on the dataset, our proposed CNN demonstrates impressive results in terms of sensitivity and accuracy. By leveraging the deep learning capabilities of the Inception V3 model, the CNN successfully learns to detect subtle features indicative of DR and effectively classifies images into appropriate severity levels. The achieved sensitivity and accuracy metrics highlight the potential of our approach in aiding early detection and management of DR, thus potentially mitigating vision loss in diabetic patients.

However, despite the promising results, there are several limitations to consider. Firstly, the performance of the CNN may vary depending on the quality and diversity of the dataset used for training. Additionally, while the Inception V3 model is powerful, it may not capture all relevant features specific to DR classification, leaving room for further improvement.

Objectives:

- To develop a system for detection and classification of diabetic retinopathy from the retinal image with wide range of classification and maximum accuracy.
- To attain maximum Robustness to Variations.
- To be able to use as Clinical Utility.
- To perform Early Detection using the project

Keywords: Diabetic Retinopathy, Fundus Image, CNN and Inception V3

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Image Processing

Image processing is a multidisciplinary field that encompasses the manipulation and analysis of images to extract meaningful information, enhance visual quality, and enable automated interpretation by machines. It serves as a critical component in various domains, including medical imaging, remote sensing, surveillance, robotics, and entertainment. The continuous advancements in computational techniques, coupled with the proliferation of digital imaging devices, have propelled image processing into a prominent role in modern technology and society.

In this comprehensive exploration, we delve into the fundamentals of image representation, elucidate various image processing techniques, discuss advanced methodologies such as deep learning, and examine diverse applications across different domains. By understanding the underlying principles and methodologies, we aim to provide insights into the intricate world of image processing and its profound impact on our lives.

Image Processing Techniques:

Image processing encompasses a broad spectrum of techniques aimed at manipulating, analyzing, and extracting valuable information from images. These techniques can be categorized into several fundamental operations, each serving a distinct purpose in enhancing image quality or extracting relevant features.

1. Image Enhancement:

Image enhancement techniques are employed to improve the visual quality of images by emphasizing specific features or mitigating undesirable artifacts. Common enhancement operations include contrast adjustment, noise reduction, and histogram equalization, each tailored to enhance different aspects of image quality.

a. Contrast Enhancement: Adjusting the contrast of an image to increase the perceptual difference between its brightest and darkest regions, thereby enhancing overall clarity and detail visibility.

b. Noise Reduction: Employing filtering techniques such as median filtering or Gaussian smoothing to suppress noise artifacts introduced during image acquisition or transmission.

c. Histogram Equalization: Redistributing pixel intensities across the dynamic range of an image to achieve a more balanced distribution of brightness levels, thereby enhancing overall contrast and detail visibility.

2. Image Restoration:

Image restoration techniques aim to recover or reconstruct the original image from degraded or corrupted versions caused by factors such as blur, noise, or compression artifacts. These techniques leverage mathematical models and algorithms to reverse the effects of degradation and restore the image to its pristine state.

a. Deblurring: Removing blur artifacts caused by motion, defocus, or optical imperfections to restore sharpness and detail in the image. Deblurring techniques include inverse filtering, blind deconvolution, and iterative algorithms tailored to specific blur models.

b. Super-Resolution: Enhancing the resolution of an image beyond its original dimensions to reveal finer details and improve overall visual quality. Super-resolution techniques employ interpolation, edge-based reconstruction, or deep learning-based approaches to infer high-resolution details from low-resolution inputs.

3. Image Segmentation:

Image segmentation involves partitioning an image into multiple regions or objects based on predefined criteria such as color, intensity, texture, or spatial proximity. Segmentation is a fundamental operation in tasks such as object detection, image analysis, and computer vision, enabling the isolation and identification of distinct entities within an image.

a. Thresholding: A simple yet effective segmentation method that classifies pixels as foreground or background based on their intensity values relative to a predefined threshold. Thresholding is widely used in applications where the objects of interest exhibit distinct intensity characteristics.

b. Clustering: Grouping pixels into clusters based on their similarity in feature space using unsupervised learning techniques such as k-means clustering or hierarchical clustering. Clustering-based segmentation methods are effective in scenarios where the objects of interest exhibit complex or non-linear intensity distributions.

4. Image Compression:

Image compression techniques aim to reduce the storage space required for representing an image while preserving its essential visual information. Compression is essential for efficient storage, transmission, and manipulation of digital images, especially in bandwidth-constrained or resource-limited environments.

a. Lossless Compression: Techniques such as Run-Length Encoding (RLE), Huffman coding, and Lempel-Ziv-Welch (LZW) compression preserve all image information without any loss but may not achieve high compression ratios.

b. Lossy Compression: Methods such as JPEG (Joint Photographic Experts Group) or MPEG (Moving Picture Experts Group) utilize perceptual coding to discard non-essential image details, resulting in higher compression ratios at the cost of some loss in visual quality.

Advanced Image Processing Techniques:

In recent years, advanced image processing techniques leveraging machine learning, deep learning, and artificial intelligence have emerged as powerful tools for solving complex image analysis and interpretation tasks. These techniques leverage large-scale datasets, sophisticated algorithms, and computational resources to achieve unprecedented levels of performance and accuracy.

1. Morphological Image Processing:

Morphological operations manipulate the shape and structure of objects in an image using binary operators such as dilation, erosion, opening, and closing. These operations are particularly useful for tasks such as feature extraction, image segmentation, and pattern recognition.

a. Dilation: Expands the boundaries of objects in an image by adding pixels to their perimeter, thereby increasing their size and connectivity.

b. Erosion: Shrinks the boundaries of objects by removing pixels from their perimeter, thereby reducing their size and connectivity.

c. Opening: A combination of erosion followed by dilation, primarily used for removing noise and fine details from binary images while preserving larger structures.

d. Closing: A combination of dilation followed by erosion, primarily used for filling small gaps or holes in objects while preserving their overall shape and structure.

2. Image Filtering:

Image filtering techniques involve applying a mask or kernel to an image to perform operations such as blurring, edge detection, or sharpening. These operations play a crucial role in image preprocessing, feature extraction, and image enhancement tasks.

a. Linear Filtering: Convolution with a linear kernel to perform operations such as blurring (smoothing) or edge detection (gradient calculation). Linear filters, such as Gaussian filters or Sobel operators, are widely used for various image processing tasks.

b. Non-linear Filtering: Techniques such as median filtering or bilateral filtering are used to preserve edges and fine details while reducing noise and artifacts. Non-linear filters are particularly effective in scenarios where linear filters fail to capture complex image structures or patterns.

3. Deep Learning in Image Processing:

Deep learning has emerged as a transformative technology in image processing, particularly with the advent of Convolutional Neural Networks (CNNs) and other deep learning architectures. These models leverage hierarchical feature representations, end-to-end learning, and large-scale data-driven approaches to achieve state-of-the-art performance in various image analysis tasks.

a. Convolutional Neural Networks (CNNs): CNNs are specialized neural network architectures designed to process grid-like data, such as images, by applying convolutional operations. These networks consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers, which learn hierarchical representations directly from raw pixel data.

b. Transfer Learning: Transfer learning is a machine learning technique that leverages pre-trained models on large datasets, such as ImageNet, and fine-tunes them for specific tasks or domains. Transfer learning accelerates training, improves generalization, and enables effective utilization of limited training data.

Applications of Image Processing:

Image processing has a wide range of applications across diverse domains, each leveraging its capabilities to address specific challenges, improve efficiency, and unlock new opportunities for innovation and discovery.

1. Medical Imaging:

Medical imaging encompasses a broad spectrum of imaging modalities and applications, including X-rays, MRI (Magnetic Resonance Imaging), CT (Computed Tomography) scans, ultrasound, and nuclear imaging. Image processing plays a crucial role in medical image analysis, diagnosis, treatment planning, and surgical navigation, enabling clinicians to visualize anatomical structures, detect abnormalities, and monitor disease progression.

a. Diagnostic Imaging: Image processing techniques aid in the detection and characterization of diseases and abnormalities in medical images, facilitating early diagnosis and treatment planning. Applications include tumor detection, lesion segmentation, and organ localization in various medical imaging modalities

b. Image-guided Interventions: Image processing enables real-time navigation and guidance during minimally invasive procedures, such as image-guided surgery, interventional radiology, and catheter-based interventions. By overlaying preoperative imaging data onto intraoperative images, surgeons can visualize critical structures, plan surgical trajectories, and navigate surgical tools with precision and accuracy.

c. Medical Image Analysis: Image processing techniques are employed to extract quantitative information from medical images, enabling automated analysis and interpretation. Applications include image registration, feature extraction, pattern recognition, and image-based modeling for tasks such as tumor grading, treatment response assessment, and disease progression monitoring.

2. Remote Sensing and Geospatial Analysis:

Remote sensing involves the acquisition and analysis of data from aerial or satellite sensors to study the Earth's surface, atmosphere, and environment. Image processing techniques are essential for extracting geospatial information, detecting changes, and monitoring natural phenomena, urban development, and environmental trends.

a. Satellite Imaging: Satellite imagery provides valuable insights into various Earth observation applications, including land cover mapping, crop monitoring, disaster management, and environmental monitoring. Image processing techniques such as image classification, change detection, and object recognition enable automated analysis and interpretation of satellite data for diverse applications.

b. Geospatial Analysis: Image processing plays a crucial role in geospatial analysis, geographic information systems (GIS), and cartography, enabling the integration, analysis, and visualization of spatial data. Applications include terrain modeling, hydrological analysis, urban planning, and infrastructure management, facilitating informed decision-making and resource allocation in various domains.

3. Surveillance and Security:

Surveillance systems utilize image processing techniques to monitor, analyze, and interpret visual data for security, safety, and situational awareness applications. Image processing enables automated detection, tracking, and recognition of objects, events, and anomalies in surveillance footage, enhancing security operations and response capabilities.

a. Video Analytics: Image processing techniques are employed for video analytics, motion detection, and activity recognition in surveillance systems, enabling real-time monitoring and alerting. Applications include crowd management, perimeter security, traffic surveillance, and public safety, facilitating proactive intervention and incident response in critical scenarios.

b. Biometric Identification: Image processing plays a crucial role in biometric identification systems, such as facial recognition, fingerprint recognition, iris recognition, and voice recognition. These systems utilize image processing algorithms to extract unique biometric features from input images, enabling accurate identification and authentication of individuals for security, access control, and identity verification purposes.

4. Robotics and Autonomous Systems:

Robotics and autonomous systems rely on image processing techniques for perception, navigation, and decision-making in dynamic and unstructured environments. Image processing enables robots to interpret visual data, detect objects, and navigate obstacles, facilitating autonomy and adaptability in various robotic applications.

a. **Object Detection and Recognition:** Image processing techniques are used to detect, localize, and recognize objects in robot perception systems, enabling robots to interact with their surroundings and perform tasks autonomously. Applications include object manipulation, grasping, navigation, and scene understanding in industrial, agricultural, and service robotics.

b. Simultaneous Localization and Mapping (SLAM): Image processing plays a crucial role in SLAM systems, which enable robots to localize themselves and build maps of their surroundings using sensor data, including visual imagery. SLAM algorithms leverage image processing techniques such as feature extraction, matching, and pose estimation to estimate the robot's trajectory and create a consistent map of the environment.

5. Entertainment and Media:

Image processing technologies are widely used in the entertainment and media industry for content creation, editing, enhancement, and visual effects. From film production and video gaming to digital art and virtual reality, image processing techniques enable artists and creators to bring their visions to life and engage audiences in immersive and interactive experiences.

a. Digital Imaging and Photography: Image processing techniques are used in digital imaging and photography for color correction, exposure adjustment, image retouching, and artistic effects. Software tools such as Adobe Photoshop, Lightroom, and GIMP utilize advanced image processing algorithms to empower photographers and digital artists to manipulate and enhance their images creatively.

b. Visual Effects and Computer Graphics: Image processing techniques are essential for creating visual effects and computer-generated imagery (CGI) in films, television, and video games. Techniques such as image compositing, matte painting, texture mapping, and rendering enable artists and animators to create realistic environments, characters, and special effects that captivate audiences and enhance storytelling.

Image processing is a dynamic and interdisciplinary field that continues to evolve with advancements in technology, computational methods, and application domains. From fundamental techniques such as image enhancement and segmentation to advanced methodologies like deep learning and artificial intelligence, image processing plays a crucial role in diverse applications across various domains, including healthcare, remote sensing, surveillance, robotics, and entertainment.

By understanding the underlying principles, techniques, and applications of image processing, researchers, practitioners, and enthusiasts can unlock new opportunities for innovation, discovery, and societal impact. As technology continues to advance and new challenges emerge, image processing will remain at the forefront of scientific research, technological innovation, and societal transformation, driving progress and shaping the future of visual computing and human-machine interaction.

1.2 Introduction to CNN

Convolutional Neural Networks (CNNs) have emerged as a transformative technology in image processing applications. Inspired by the structure and function of the visual cortex in biological organisms, CNNs are specialized neural network architectures designed to process grid-like data, such as images, with remarkable efficiency and accuracy. Unlike traditional neural networks, CNNs leverage convolutional layers to extract hierarchical features directly from raw pixel data, enabling automated feature learning and representation. This hierarchical feature extraction enables CNNs to capture spatial hierarchies, local patterns, and abstract representations, making them highly effective in tasks such as image classification, object detection, and segmentation. With their ability to learn from vast amounts of data and generalize across diverse domains, CNNs have revolutionized various fields, including healthcare, autonomous vehicles, surveillance, and entertainment. As a cornerstone technology in modern image processing, CNNs continue to drive innovation and empower intelligent systems to perceive, interpret, and interact with visual information in unprecedented ways.

1.3 INTRODUCTION TO DIABETIC RETINOPATHY

a. Overview of Diabetic Retinopathy

Diabetic retinopathy is a retinal vascular disorder appears in the diabetic patients. The duration of diabetes is a key factor in the arrival of retinopathy, with the increase in diabetes duration will increase the risk of the DR development. It is also noticed that patients with diabetes usually unaware of the possibility of DR, which leads towards the delayed diagnosis and treatment.

Manual detection of DR is time-consuming and requires trained clinical experts to analyze digital color fundus images. However, the delayed outcomes can result in a lack of follow up and misinformation for patients. Diabetic retinopathy has been manually tested by ophthalmologists until now. Manual diagnosis of DR is time consuming, and therefore, computer-aided diagnosis is gaining attention.

Non-proliferative diabetic retinopathy (NPDR) causes retinal swelling and leakage of tiny blood vessels, leading to macular edema and vision loss. Other types of NPDR include blood vessel closure and macular ischemia, as well as the formation of exudates that can affect human vision.

Proliferative diabetic retinopathy (PDR) is the most severe stage of the disease, in which new blood vessels start developing in the retina through neovascularization.

These new vessels can bleed in the vitreous, causing dark float- ers, and if bleeding is extensive, it can result in blurred vision. Scar tissue formation is common in PDR and can cause macular problems or contribute to independent retinal tissue. PDR is a severe condition that can affect both central and peripheral vision.

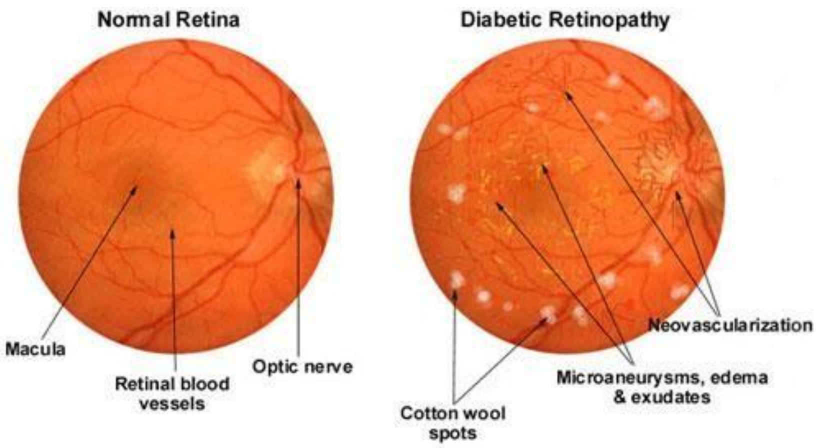


Fig 1.2.1: Fundus Image of unaffected and affected eye.

The existing models are unable to detect the disease at early stages and complicated due to high computational cost with low performance. To address these issues, various techniques have been proposed for automatic detection of DR from fundus images, including DL-based approaches.

In this study, we propose a novel DL model for DR detection, utilizing InceptionV3 and Resnet50 for feature extraction of fundus images. The extracted features are then concatenated and fed to the IR CNN for classification of DR. Additionally, we conduct experiments with image enhancement and data augmentation methods to improve the performance of the proposed model. The proposed DL model efficiently diagnoses the diabetic retinopathy at early stage and perform significantly better than existing techniques.

b. Problem Statement

The problem at hand is the detection and classification of diabetic retinopathy, a common complication of diabetes that affects the eyes, leading to potential vision loss or blindness if left untreated. The goal is to develop a machine learning model capable of accurately identifying and categorizing retinal images into different stages of diabetic retinopathy, ranging from mild to severe.

This involves leveraging image processing techniques to enhance the quality of retinal images, such as contrast enhancement, noise reduction, and normalization, to facilitate more accurate analysis. Additionally, Convolutional Neural Networks (CNNs) will be utilized to extract meaningful features from retinal images and classify them based on diabetic retinopathy severity levels. Architectures like Inception V3 will be explored to improve the model's ability to learn intricate patterns and features associated with different stages of diabetic retinopathy.

The ultimate objective is to develop a comprehensive system capable of accurately detecting and classifying diabetic retinopathy from retinal images with a wide range of classification and maximum accuracy. This system aims to provide clinicians with a reliable tool for early detection and intervention, ultimately reducing the risk of vision loss and blindness in diabetic patients.

c. Significance of Early Detection of Diabetic Retinopathy

Preventing Vision Loss: Early detection of diabetic retinopathy is critical in preventing vision loss or blindness in diabetic patients. By identifying the disease in its initial stages, healthcare providers can intervene with timely treatment strategies to manage the condition and prevent its progression. Vision loss due to diabetic retinopathy is often irreversible, making early detection paramount for preserving sight and maintaining quality of life.

Effective Treatment Planning: Early detection allows for more effective treatment planning and management of diabetic retinopathy. When the disease is identified in its early stages, there are a variety of treatment options available, including laser therapy, intraocular injections, and surgical interventions. These treatments are most effective when administered early, helping to reduce the risk of vision loss and improve long-term outcomes for patients.

Reducing Healthcare Costs: Early detection of diabetic retinopathy can lead to significant cost savings in healthcare expenditures. By identifying and treating the condition early, patients may require fewer medical interventions and experience fewer complications associated with advanced stages of the disease. This can result in decreased healthcare utilization, hospitalizations, and long-term care needs, ultimately reducing the economic burden on healthcare systems and society as a whole.

Improving Patient Outcomes: Timely detection of diabetic retinopathy not only preserves vision but also improves overall patient outcomes and quality of life. Early intervention can help diabetic patients maintain their independence, productivity, and ability to perform daily activities. Furthermore, preserving vision can positively impact mental health and emotional well-being, reducing the psychological burden associated with vision loss and blindness.

Enhancing Public Health Efforts: Early detection initiatives for diabetic retinopathy contribute to broader public health efforts aimed at preventing and managing chronic diseases, such as diabetes. By integrating screening programs into primary care settings and raising awareness about the importance of regular eye examinations among diabetic individuals, healthcare systems can proactively address the growing burden of diabetic retinopathy and its associated complications.

d. Objectives

High Accuracy Classification: The primary objective is to achieve high accuracy in the classification of diabetic retinopathy severity levels using the Inception V3 model. By leveraging the powerful capabilities of this deep learning architecture, the aim is to develop a system that can accurately differentiate between different stages of diabetic retinopathy, ranging from mild abnormalities to severe complications.

Feature Extraction: Utilizing Inception V3, the system aims to extract intricate features from retinal images that are indicative of diabetic retinopathy. By capturing relevant patterns and structures within the images, the model can learn to identify subtle signs of retinopathy and differentiate them based on severity levels.

Robustness to Variability: Another objective is to ensure the robustness of the classification system to variability in retinal images. This includes variations in image quality, lighting conditions, and patient demographics. The Inception V3 model should be trained to generalize well across diverse datasets, encompassing a wide range of retinal images, to ensure consistent and reliable performance in real-world scenarios.

Early Detection: The system aims to contribute to the early detection of diabetic retinopathy by accurately classifying retinal images into different severity levels. Early detection is crucial for timely intervention and treatment, which can help prevent further progression of the disease and mitigate the risk of vision loss or blindness in diabetic patients.

Clinical Utility: Ultimately, the objective is to develop a system that has practical utility in clinical settings. The classification results provided by the Inception V3 model should be interpretable and actionable for healthcare professionals, enabling them to make informed decisions regarding patient care and treatment strategies.

CHAPTER – 2

LITERATURE REVIEW

2.1 Literature Survey

S.No	Work	Dataset	Technique	Purpose	Performance analysis
1.	(Wang et al., 2019)	Lab Dataset	Cosine Similarity, Logistic Regression	Prediction of severity and grading of DR	91.2% for DR related features 95.6% for DR severity prediction
2.	(Esmaeili et al., 2014)	Private Data	Curvelet Transformation, Equalisation algorithm	Prediction Of DR	sen/spec of 94/87
3.	(Dorizzi et al., 2019)	DiaretDB1 and Messidor	CNN, Probability map	Reduce the complexity of the model while increasing performance	ROC of 0.912 and sen of 0.940
4.	(Dhara et al., 2015)	Messidor	Morphological features, CLAHE, SVM	Detection of DR	AUC mild:0.9106 Moderate:0.8372 Severe:0.9750
5.	(Sakshi Gunde and Gupta, 2020)	DIARETDB0	Green Channel, Otsu Algorithm, Circular Hough transform	Detection of DR using blobs and blood vessel	sen/spec of 73/70 in blob detection and 77/80 in blood vessel detection
6.	(Hajeb Mohammad Alipour et al., 2012)	Local dataset	Curvelet based enhancement, SVM	Detection and Grading of DR	Sen/spec of 100% in grading
7.	(Rahim et al., 2015)	Eye Clinic, Department of Ophthalmology, Hospital Melaka, Malaysia (Local Dataset)	Grayscale, fuzzy filtering, fuzzy edge detection, Decision tree, Knearest neighbour	Detection of DR and Maculopathy	Category II misclassification error/acc/sen/spec of 0.4395/0.5605/0.4500/0.5956 for binary decision tree and 0.2975/0.7025/0.6500/0.7297 for k-nearest neighbor

S.No	Work	Dataset	Technique	Purpose	Performance analysis
9.	(Qummar, 2019)	Kaggle	CNN Models Resnet50, Inceptionv3, Xception, Dense121, Dense169	Classify five stages of DR	–
10.	(Carson Lam and MD1, 2017)	Private Dataset	CNN, transfer learning, googlenet, alexnet	Detection and Grading of DR	74.5% on 2- ary classification, 68.8% on 3-ary classification and 57.2% on 4-ary
11.	(Zeng et al., 2019)	Kaggle diabetic retinopathy competition	Inception V-3, Siamese like network structure	A model is proposed with transfer learning and a Siamese-like structure.	Kappa:0.829 AUC:0.951
12.	(Li et al., 2019)	Kaggle Contests "Identify signs of diabetic retinopathy in eye images"	Deep CNN, SVM, Teaching-Learning Based optimization	Classify 5 stages of DR along with the app	Accuracy 86.71% for 5-stage classification 91.05 for binary classification and sen/spec for binary classification 0.8930/0.9089
13.	(Shanthi and Sabeenian, 2019)	Messidor	CNN, Modified Alexnet Architecture	4-level Grading of DR	Accuracy Stage 3-healthy images 96.6%,95.6%,96.2% and 96.6%
14.	(Pratt et al., 2016)	Kaggle Dataset	Data Augmentation, CNN	Five –class problem	acc/sen of 75%/95%
15.	(Leeza and Farooq, 2019)	Kaggle National Data Science Bowl	Bag of features, Histogram oriented gradient, speed up robust feature, k-mean clustering	To identify the 5 –class problem	sen/spec/acc of 95.2%/98.9%/98.3
16.	(Wan et al., 2018)	Kaggle	Transfer learning and hyperparameter tuning on CNN Models, Google Net, Alex Net, ResNet, and VggNet	Classify the DR Fundus Images	-

Table 2.1.1: Literature Review

2.2 Existing Methods and Drawbacks:

In addition to Inception V3, several existing methods in machine learning have been applied to the task of diabetic retinopathy classification. Here are a few examples along with their drawbacks:

1. ResNet (Residual Networks):

Method: ResNet is a deep learning architecture that introduces skip connections to enable training of very deep neural networks. It has been applied to various image classification tasks, including diabetic retinopathy classification.

Drawbacks: While ResNet can effectively handle the vanishing gradient problem and enable training of very deep networks, it may suffer from overfitting when applied to smaller datasets. Additionally, the increased model complexity of ResNet compared to simpler architectures may require longer training times and more computational resources.

2. VGG (Visual Geometry Group):

Method: VGG is a convolutional neural network architecture known for its simplicity and uniform structure. It consists of a series of convolutional layers followed by max-pooling layers, with fully connected layers at the end for classification.

Drawbacks: The main drawback of VGG is its high computational cost and memory requirement due to its large number of parameters. Additionally, VGG tends to be prone to overfitting, especially when applied to tasks with limited training data.

3. MobileNet:

Method: MobileNet is a lightweight convolutional neural network architecture designed for mobile and embedded devices. It employs depthwise separable convolutions to reduce computational complexity while maintaining accuracy.

Drawbacks: While MobileNet offers advantages in terms of computational efficiency and model size, it may sacrifice some accuracy compared to larger architectures like Inception V3. Additionally, MobileNet may not capture as rich of features as deeper architectures, potentially limiting its performance on complex classification tasks like diabetic retinopathy.

4. SVM (Support Vector Machines):

Method: SVM is a classical machine learning algorithm used for classification tasks. It works by finding the hyperplane that best separates the data points into different classes.

Drawbacks: SVMs may not perform as well as deep learning models like Inception V3 for image classification tasks, especially when dealing with complex and high-dimensional data like retinal images. SVMs also require careful selection of appropriate features, which may be challenging for tasks with large and diverse feature spaces.

5. Random Forest:

Method: Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

Drawbacks: While Random Forest can handle high-dimensional data and non-linear relationships well, it may not be as effective as deep learning models for image classification tasks like diabetic retinopathy. Random Forests may struggle to capture complex spatial dependencies and patterns present in retinal images, leading to suboptimal performance compared to deep learning approaches.

In summary, while these methods offer alternatives to Inception V3 for diabetic retinopathy classification, they may have drawbacks such as computational complexity, overfitting, limited feature representation, or reduced accuracy compared to deep learning models. The choice of method depends on factors such as available computational resources, dataset size, and desired trade-offs between accuracy and efficiency

CHAPTER – 3
DIABETIC RETINOPATHY USING CNN

3.1 Dataset:

A dataset is a structured collection of data that is organized in a way that facilitates analysis, interpretation, and processing. It typically consists of individual data points, often referred to as observations or instances, which are organized into rows or records. Each data point may contain one or more attributes or features, which are the properties or characteristics being measured or observed.

The Diabetic Retinopathy Dataset is openly available to everyone on the kaggle platform on the internet. The complete in- formation of the dataset is as follows.

- Dataset Name : Diabetic Retinopathy Dataset.
- Source : Kaggle platform.
- The dataset contains 1917 Eye Fundus images.
- Sub-folders (Types of DR):

1. Normal - 328 Images
2. Mild - 406 Images
3. Moderate - 385 Images
4. Severe - 315 Images
5. Proliferative - 483 Images

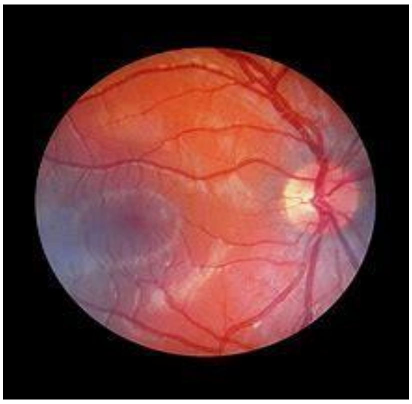


Fig 3.1.1: Fundus camera Image of Human

3.2 Algorithm:

3.2.1 Data Exploration

Introduction to Data Exploration

Data exploration serves as the foundational step in any machine learning project, providing researchers with crucial insights into the dataset's composition and characteristics. In the context of diabetic retinopathy classification, this phase involves gaining a comprehensive understanding of the dataset containing retinal images annotated for diabetic retinopathy. By thoroughly exploring the dataset, researchers can make informed decisions regarding preprocessing steps and model development, ultimately leading to more accurate and reliable classification results.

Analyzing class Distribution

One of the primary objectives of data exploration is to analyze the distribution of classes within the dataset. In the case of diabetic retinopathy classification, this entails examining the distribution of severity levels across the annotated retinal images. A balanced distribution of classes is essential to ensure that the model is trained effectively across all severity levels of diabetic retinopathy. Researchers analyze class distribution to identify any imbalances that may exist and determine whether corrective measures, such as data augmentation or resampling techniques, are necessary to address them. Ensuring a balanced dataset is crucial for preventing biases in model training and optimizing classification performance.

Visualizing sample images

Another key aspect of data exploration is the visualization of sample images from the dataset. Visual inspection of sample images provides researchers with valuable insights into the characteristics of different severity levels of diabetic retinopathy and potential challenges in classification. By visually examining images from each class, researchers can identify common features or patterns associated with different severity levels. This qualitative analysis helps researchers understand the variability in the dataset and identify potential sources of noise or ambiguity that may impact model performance. Additionally, visualizing sample images aids in identifying any preprocessing steps that may be necessary to enhance the quality or consistency of the data before model training begins. Overall, visualizing sample images plays a crucial role in informing subsequent preprocessing steps and guiding the development of an effective classification model for diabetic retinopathy.

The model used in data exploration in Diabetic Retinopathy using CNN is Exploratory Data Analysis (EDA). EDA is a statistical technique that allows data analysts to analyze datasets to understand their broad characteristics, identify patterns, and relationships within the data. It involves using data visualization and statistical tools to gain insights into the dataset before proceeding with further analysis or modeling.

EDA is a fundamental step in the data analysis process as it helps in understanding the main characteristics of the dataset, uncovering valuable insights, and identifying patterns and relationships between variables. By exploring the data through EDA, analysts can gain a deeper understanding of the underlying structure and potential value of the dataset.

➢ 3.2.2 Data Preparation

Data preparation is a critical step in the machine learning pipeline, ensuring that the dataset is properly formatted and preprocessed to facilitate model training. In the context of diabetic retinopathy classification, data preparation involves several key tasks aimed at standardizing and enhancing the quality of the retinal images annotated for diabetic retinopathy.

Resizing Images

- Resizing images to a uniform size is essential to ensure consistency across the dataset and compatibility with the CNN model's input requirements.
- Standardizing image dimensions reduces variability and simplifies the model training process.

Normalizing Pixel values

- Normalizing pixel values to a common scale, typically between 0 and 1, helps improve model convergence during training.
- Scaling pixel values ensures that the CNN model receives input data with consistent intensity levels, facilitating more effective feature extraction.
- Splitting the data
- Dividing the dataset into training, validation, and test sets is crucial for evaluating model performance and preventing overfitting.
- Stratified sampling ensures that each subset maintains the same distribution of classes, preserving the dataset's representativeness.

Quality assurance

- Conducting quality checks to identify and address any issues with image quality, such as blurriness, artifacts, or improper lighting.
- Removing or correcting low-quality images helps ensure that the CNN model receives high-quality input data, leading to more accurate classification results.

Data augmentation

- Augmenting the dataset through techniques such as rotation, flipping, and scaling increases the diversity of training examples and enhances the model's ability to generalize to unseen data.
- Augmentation helps mitigate overfitting and improves the robustness of the CNN model to variations in input data.

Training data is used to train the model by learning patterns in the input-output relationships while testing data is used to evaluate the model's performance on unseen data. Both training and testing data are crucial for developing and assessing the effectiveness of deep learning models.

Training Data:

- **Purpose:** Training data is used to train the machine learning model. It consists of a set of input samples (features) along with their corresponding target values (labels or outputs).

- **Usage:** During the training phase, the model learns patterns and relationships in the training data to make predictions or classifications on new, unseen data.
- **Composition:** Training data typically constitutes the majority of the dataset, often around 70-80%. It should be diverse and representative of the data that the model will encounter during deployment.
- **Process:** The model iteratively adjusts its parameters or weights based on the input-output pairs in the training data, aiming to minimize the difference between predicted outputs and actual labels.
- **Evaluation:** The performance of the model on the training data is assessed during training, but it's not sufficient to determine how well the model will generalize to unseen data.

Testing Data:

- **Purpose:** Testing data is used to evaluate the performance of the trained model. It consists of a separate set of input samples and corresponding target values that were not used during training.
- **Usage:** The model's performance is assessed by making predictions or classifi- cations on the testing data and comparing them against the actual labels.
- **Composition:** Testing data typically constitutes a smaller portion of the dataset, often around 20-30%. It should be representative of the data distribution and cover a wide range of scenarios.
- **Process:** The model's predictions on the testing data are compared with the true labels to calculate evaluation metrics such as accuracy, precision, recall, and F1-score.
- **Generalization:** Testing data helps assess how well the model generalizes to unseen data and whether it can make accurate predictions on new, previously unseen samples.
- **Precaution:** Testing data should never be used for model training to ensure an unbiased evaluation of model performance.

Fig 3.2.2.1: Train and Test Dataset model

This work split the data into training and testing in the ratio 80% and 20% respectively.

```
Diabetic_retinopathy_data/
├── train/
│   ├── No_DR/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
│   ├── Mild/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
│   ├──Moderate/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
│   ├── Severe/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
│   ├── Proloferative/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
├── test/
│   ├── No_DR/
│   │   ├── DR_train_image1.jpg
│   │   └── ...
│   ├── Mild/
```

```
|    |    ├── DR_train_image1.jpg
|    |    └── ...
|    ├──Moderate/
|    |    ├── DR_train_image1.jpg
|    |    └── ...
|    ├── Severe/
|    |    ├── DR_train_image1.jpg
|    |    └── ...
|    ├── Proloferative/
|    |    ├── DR_train_image1.jpg
|    |    └── ...
```

Fig 3.2.2.2: Structural Organizing of data

As mentioned above the data has been segmented into test and training datasets and the data has been again grouped into sub-groups based on the category of the image. The model has to be trained on different amount of datasets, so the data has been copied into multiple folders in different ratios of training and testing.

➢ 3.2.3 Model Building using CNN

In the realm of Diabetic Retinopathy detection, constructing a Convolutional Neural Network (CNN) model is a pivotal stage that demands a tailored architecture capable of scrutinizing fundus images for indications of the disease. Let's delve into a comprehensive explanation of each sub-topic related to this critical step:

Purpose of Model Building:

- Objective: The primary goal of building a CNN model for Diabetic Retinopathy detection is to leverage the power of deep learning to automatically extract intricate features from fundus images. This process enables the model to learn to differentiate between healthy retinal images and those exhibiting signs of Diabetic Retinopathy, facilitating accurate classification and diagnosis.

- Significance: By training the model on a diverse dataset, it can capture subtle patterns and anomalies that are indicative of the disease, enhancing its ability to make informed predictions. The ultimate aim is to develop a robust system that can assist healthcare professionals in early detection and intervention, potentially preventing vision loss in patients with Diabetic Retinopathy.
- Impact: A well-designed CNN model can significantly improve the efficiency and accuracy of Diabetic Retinopathy diagnosis, offering a valuable tool for healthcare providers to enhance patient care and outcomes. Through meticulous model building, the aim is to create a reliable system that can aid in the early identification and management of this sight-threatening condition.

Architecture Design:

- **Components:** The architecture of a CNN typically comprises multiple convolutional layers responsible for extracting features at different levels of abstraction. These layers are followed by pooling layers that reduce spatial dimensions, activation functions like Rectified Linear Unit (ReLU) to introduce non-linearity, and fully connected layers for final classification.

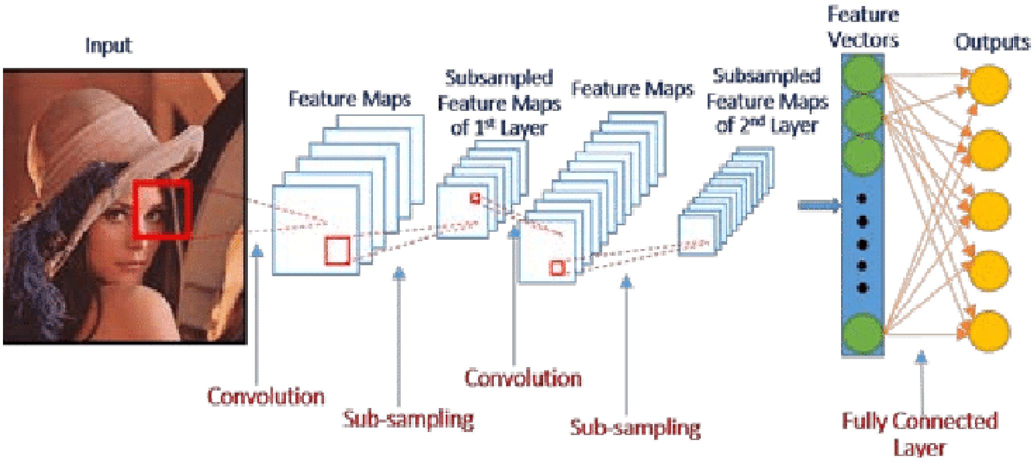


Fig 3.2.3.1: CNN Architecture

Functionality: Each component of the CNN architecture plays a crucial role in the feature extraction process. Convolutional layers detect patterns and structures within the fundus images, while pooling layers help in down-sampling and reducing computational complexity. Activation functions introduce non-linearities, enabling the model to learn complex relationships within the data.

- **Design Considerations:** The design of the CNN architecture is tailored to the specific task of Diabetic Retinopathy detection, with careful consideration given to the number of layers, filter sizes, and activation functions used. By optimizing the architecture, the model can effectively learn and differentiate between relevant features in fundus images, leading to accurate classification results.

Feature Extraction:

- Process: CNNs excel at feature extraction by applying convolutional filters to input images, capturing patterns and structures that are crucial for identifying Diabetic Retinopathy. Through successive layers, the model learns to detect relevant features such as microaneurysms, hemorrhages, and exudates, which are indicative of the disease.
- **Hierarchical Learning:** The hierarchical nature of feature extraction in CNNs allows the model to learn increasingly complex features as information flows through the network. This process enables the model to identify subtle details and variations in fundus images that may signify the presence of Diabetic Retinopathy.
- **Feature Representation:** By extracting meaningful features from fundus images, the CNN model can create a rich representation of the data that encapsulates the key characteristics associated with Diabetic Retinopathy. These learned features serve as the basis for accurate classification and enable the model to make informed predictions based on the patterns detected in the images.

Hyperparameter Tuning:

- **Optimization:** Fine-tuning hyperparameters such as learning rate, batch size, and optimizer selection is essential to optimize the model's performance during training. By adjusting these parameters, the model's learning process is finetuned to effectively capture the complex patterns present in fundus images, enhancing its ability to make accurate predictions.

- **Learning Rate:** The learning rate controls the step size taken during optimization and influences how quickly the model converges to an optimal solution. Finding the right balance between a high learning rate that may lead to overshooting and a low learning rate that may result in slow convergence is crucial for efficient training.
- **Batch Size:** Batch size determines the number of samples processed in each iteration during training. Choosing an appropriate batch size impacts the model's generalization ability and training speed. Larger batch sizes can lead to faster convergence but may require more memory, while smaller batch sizes offer better generalization but slower convergence.

Transfer Learning:

- **Concept:** Transfer learning involves leveraging pre-trained CNN models, such as those trained on large image datasets like ImageNet, and adapting them to the task of Diabetic Retinopathy detection. This approach accelerates training, especially when working with limited data, and allows the model to benefit from the learned features of the pre-trained network.
- **Benefits:** By utilizing transfer learning, the model can leverage knowledge gained from diverse datasets and tasks, enhancing its ability to generalize to new data. This approach reduces the need for extensive training data and computational resources while improving the model's performance in detecting subtle features associated with Diabetic Retinopathy.
- **Implementation:** Transfer learning involves freezing certain layers of the pre-trained model to retain learned features while fine-tuning specific layers to adapt to the new task. By selectively updating parameters based on the target dataset, transfer learning enables efficient utilization of pre-existing knowledge for improved performance.

These detailed explanations provide insights into the critical aspects of model building using CNN for Diabetic Retinopathy detection. By understanding the significance of each sub-topic and their impact on the overall model performance, researchers and practitioners can effectively design and optimize CNN models for accurate diagnosis and management of this sight-threatening condition.

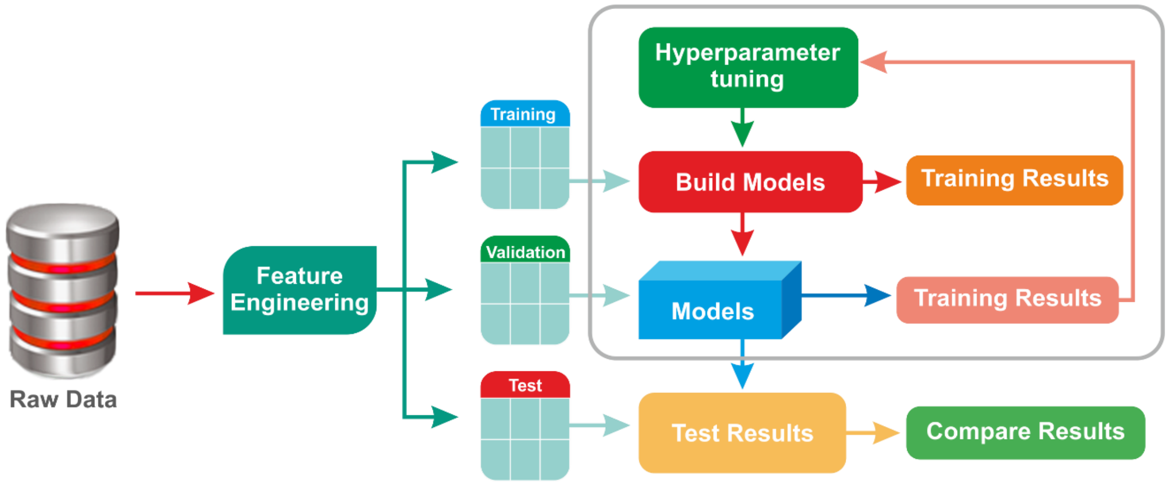


Fig 3.2.3.2: The overview of CNN Model

Finally after following the above steps, the model is ready to test. The network's performance is evaluated using a loss function, measuring the disparity between predicted and actual values, and optimized using an optimizer like stochastic gradient descent, which adjusts the model's parameters to minimize the loss. Through this systematic process, CNNs can effectively learn and recognize patterns within images, making them powerful tools for tasks such as image classification, object detection, and more.

Flowchart:

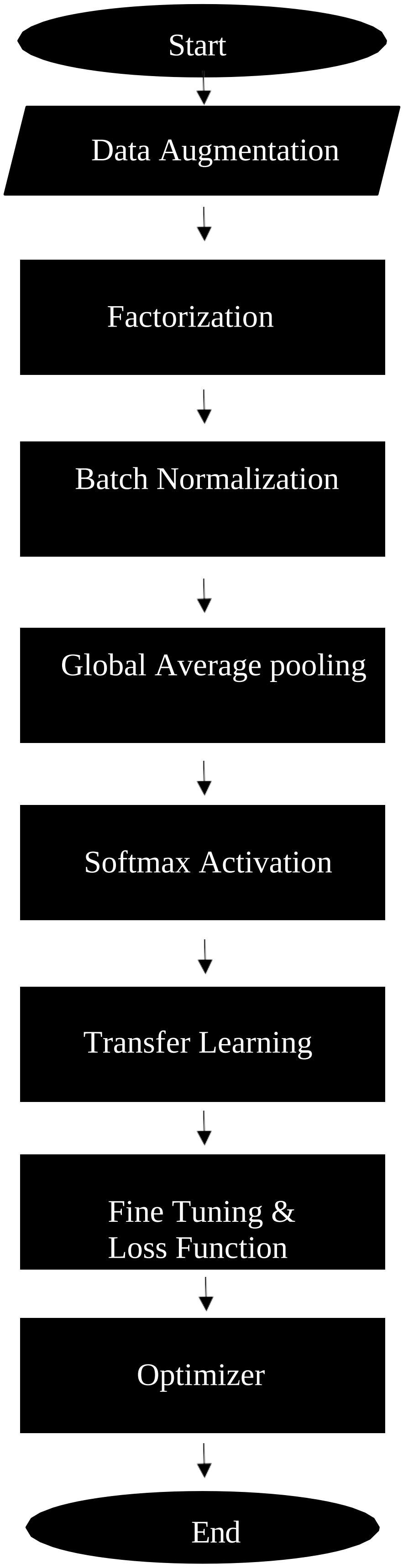


Fig 3.2.3.3: Flowchart of CNN model

Introduction to Inception Models

The Inception V3 is a deep learning model based on Convolutional Neural Net- works, which is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014. As the name suggests it was developed by a team at Google.

Inception V1:

When multiple deep layers of convolutions were used in a model it resulted in the overfitting of the data. To avoid this from happening the inception V1 model uses the idea of using multiple filters of different sizes on the same level. Thus, in the inception models instead of having deep layers, we have parallel layers thus making our model wider rather than making it deeper.

The Inception model is made up of multiple Inception modules.

The basic module of the Inception V1 model is made up of four parallel layers.

1×1 convolution
3×3 convolution
5×5 convolution
3×3 max pooling

Convolution - The process of transforming an image by applying a kernel over each pixel and its local neighbors across the entire image.

Pooling - Pooling is the process used to reduce the dimensions of the feature map. There are different types of pooling but the most common ones are max pooling and average pooling.

Here, different sizes of convolutions are performed to capture different sizes of information in the picture.

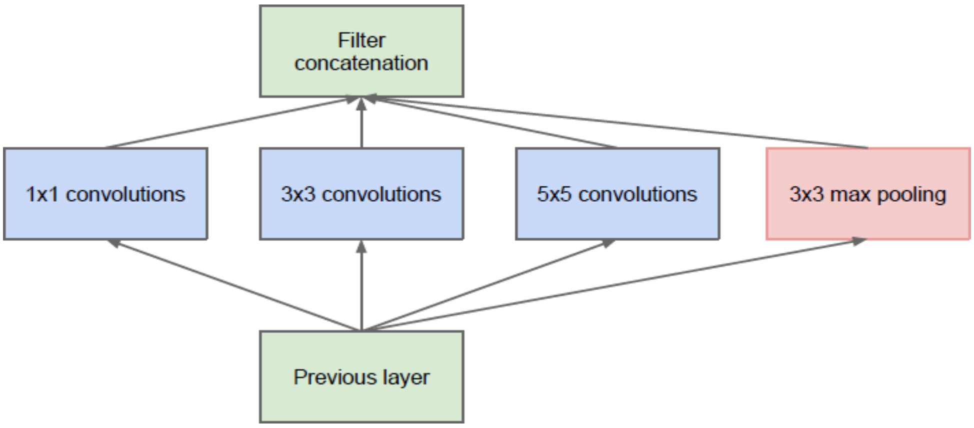


Fig 3.2.3.4: Naive Form

This module of the Inception V1 is called the Naive form. One of the drawbacks of this naive form is that even the 5×5 convolutional layer is computationally pretty expensive i.e. time-consuming and requires high computational power.

To overcome this the authors added a 1×1 convolutional layer before each convolutional layer, which results in reduced dimensions of the network and faster computations.

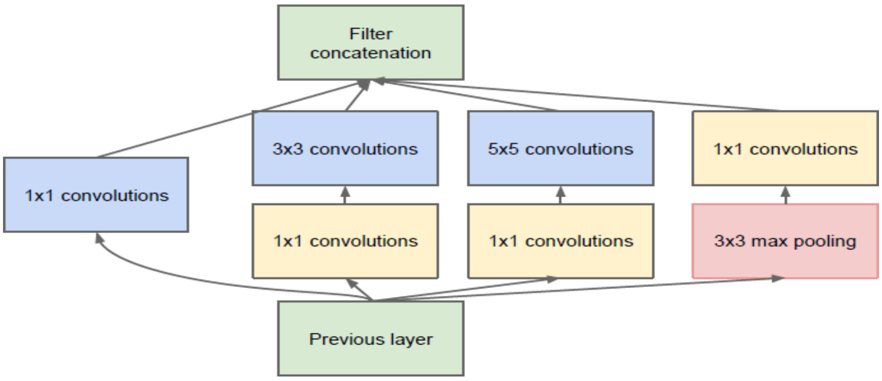


Fig 3.2.3.5: Building block of the Inception V1 model Architecture

After adding the dimension reductions the module looks like this.

The Inception V1 architecture model was better than most other models at that time. We can see that it has a very minimum error percentage.

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 3.2.3.1 : Comparison of Inception V1 with other models.

What makes the Inception V3 model better?

The inception V3 is just the advanced and optimized version of the inception V1 model. The Inception V3 model used several techniques for optimizing the network for better model adaptation.

- It has higher efficiency
- It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised.
- It is computationally less expensive.
- It uses auxiliary Classifiers as regularizes.

Inception V3 Model Architecture

The inception v3 model was released in the year 2015, it has a total of 42 layers and a lower error rate than its predecessors. Let's look at what are the different optimizations that make the inception V3 model better.

The major modifications done on the Inception V3 model are

- Factorization into Smaller Convolutions
- Spatial Factorization into Asymmetric Convolutions
- Utility of Auxiliary Classifiers
- Efficient Grid Size Reduction

Let's how each one of these optimizations was implemented and how it improved the model.

Factorization into Smaller Convolutions

One of the major assets of the Inception V1 model was the generous dimension reduction. To make it even better, the larger Convolutions in the model were factorized into smaller Convolutions.

For example, consider the basic module of the inception V1 module.

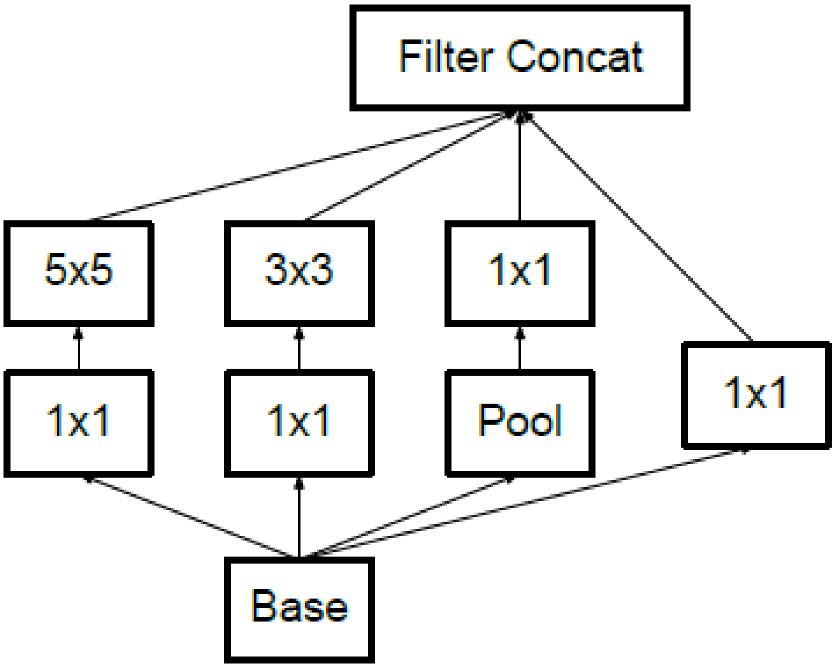


Fig 3.2.3.6: Different Layers in Model

It has a 5×5 convolutional layer which was computationally expensive as said before. So to reduce the computational cost the 5×5 convolutional layer was replaced by two 3×3 convolutional layers as shown below.

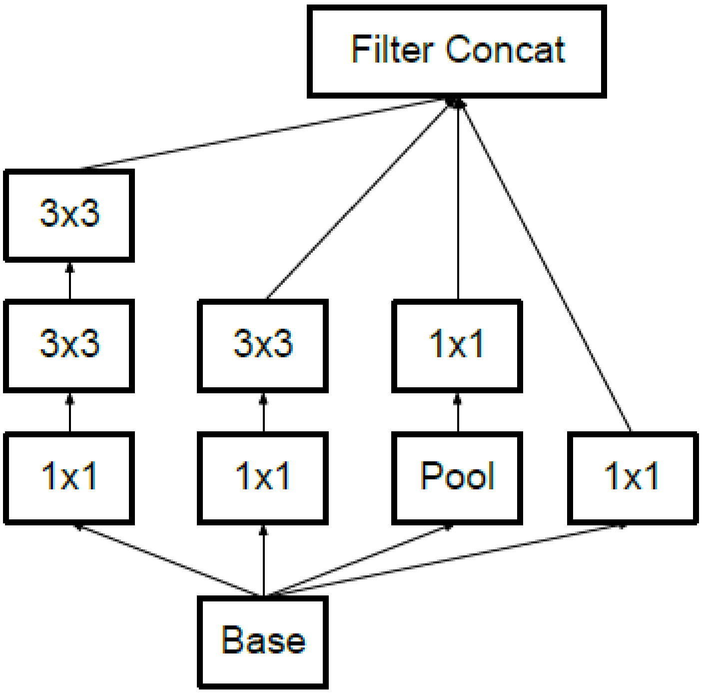


Fig 3.2.3.7: Replacing 5x5 Layer with two 3x3 layers

To understand it better see how the process of using two 3×3 convolutions reduces the number of parameters.

As a result of the reduced number of parameters the computational costs also reduce. This factorization of larger convolutions into smaller convolutions resulted in a relative gain of 28%.

Spatial Factorization into Asymmetric Convolutions

Even though the larger convolutions are factorized into smaller convolutions. You may wonder what if we can factorize furthermore for example to a 2×2 convolution. But, a better alternative to make the model more efficient was Asymmetric convolutions.

Asymmetric convolutions are of the form n×1.

So, what they did is replace the 3×3 convolutions with a 1×3 convolution followed by a 3×1 convolution. Doing so is the same as sliding a two-layer network with the same receptive field as in a 3×3 convolution.

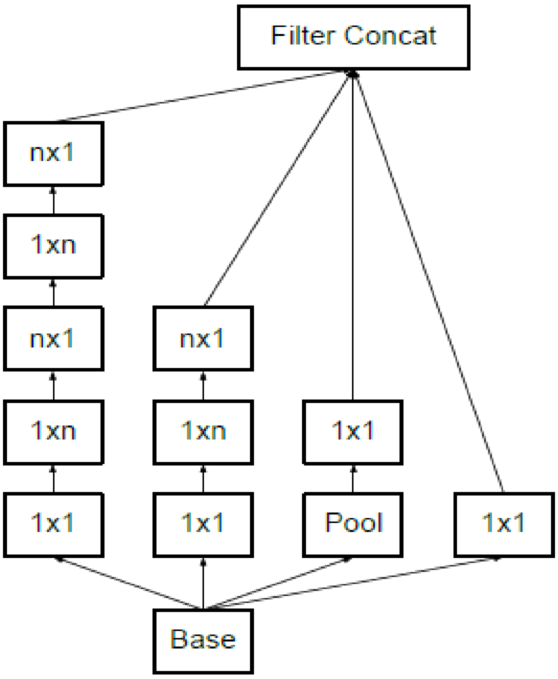


Fig 3.2.3.8: Structure of Asymmetric Convolutions

The two-layer solution is 33% cheaper for the same number of output filters if the number of input and output filters is equal.

After applying the first two optimization techniques the inception module looks like this.

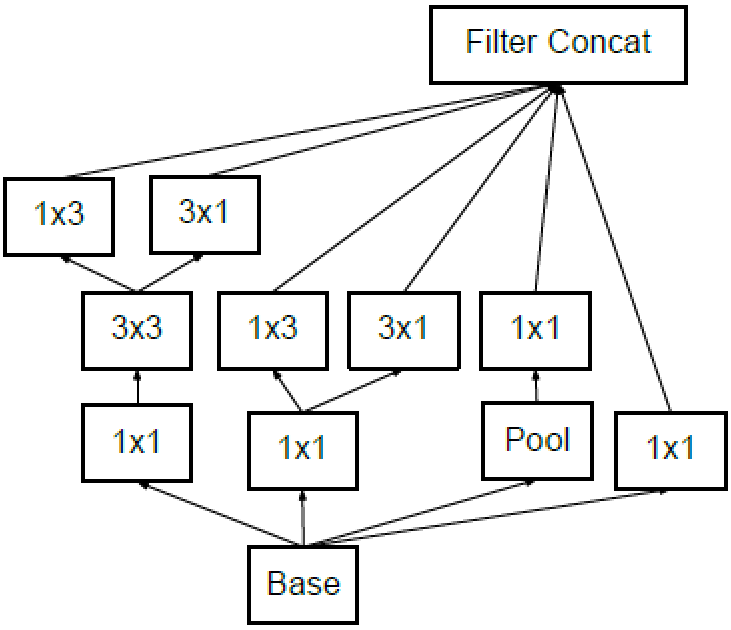


Fig 3.2.3.9: 1xn and nx1 Asymmetric convolutions

Utility of Auxiliary classifiers

The objective of using an Auxiliary classifier is to improve the convergence of very deep neural networks. The auxiliary classifier is mainly used to combat the vanishing gradient problem in very deep networks.

The auxiliary classifiers didn't result in any improvement in the early stages of the training. But towards the end, the network with auxiliary classifiers showed higher accuracy compared to the network without auxiliary classifiers.

Thus the auxiliary classifiers act as a regularizer in Inception V3 model architecture.

Efficient Grid Size Reduction

Traditionally max pooling and average pooling were used to reduce the grid size of the feature maps. In the inception V3 model, in order to reduce the grid size efficiently the activation dimension of the network filters is expanded.

For example, if we have a d×d grid with k filters after reduction it results in a d/2 × d/2 grid with 2k filters.

And this is done using two parallel blocks of convolution and pooling later concatenated.

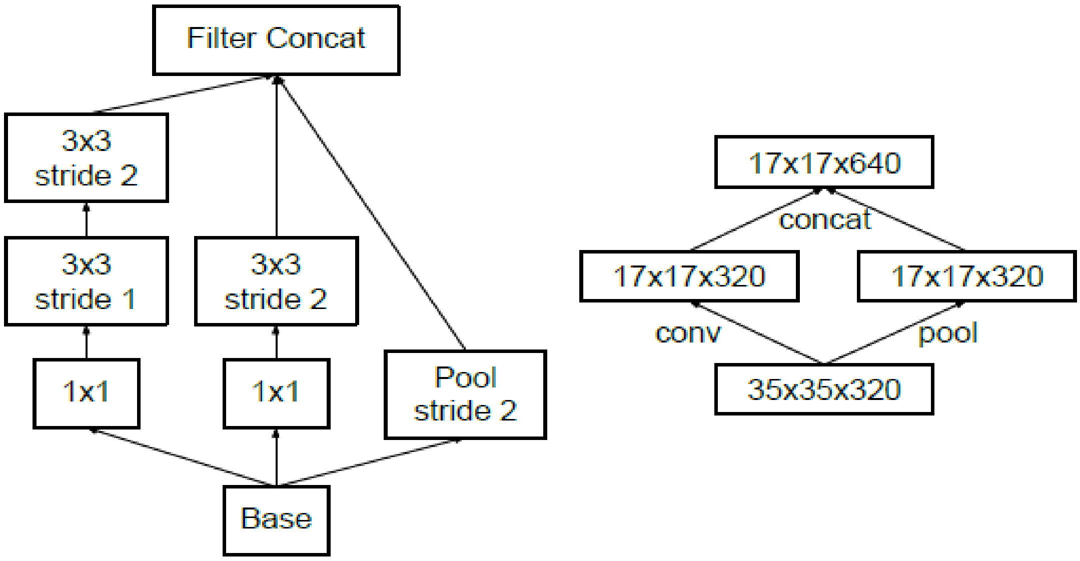


Fig 3.2.3.10: Covolutions for Efficient grid reduction

The final Inception V3 model:

After performing all the optimizations the final Inception V3 model looks like this

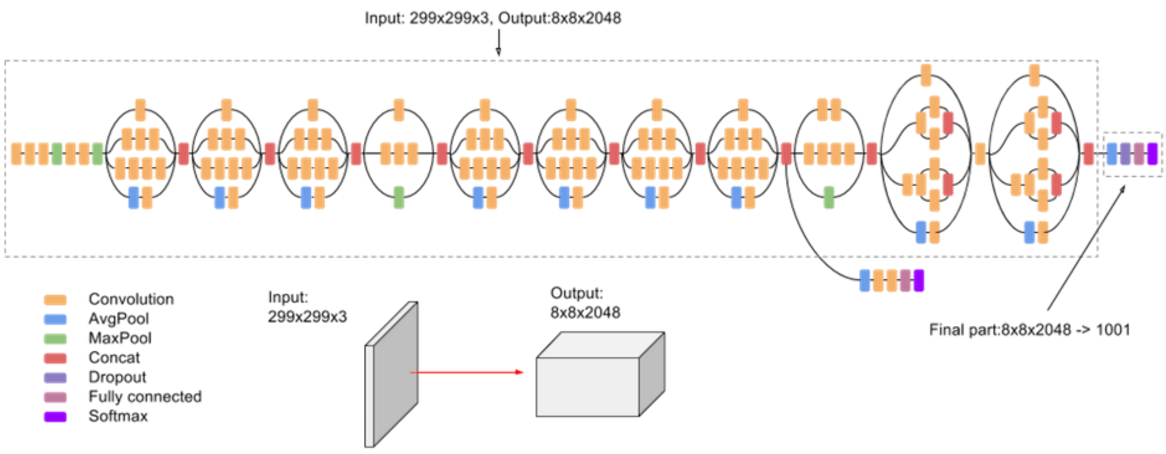


Fig 3.2.3.11: Complete Inception V3 model

In total, the inception V3 model is made up of 42 layers which is a bit higher than the previous inception V1 and V2 models. But the efficiency of this model is really impressive.

Performance of Inception V3

As expected the inception V3 had better accuracy and less computational cost compared to the previous Inception version.

Network	Models Evaluated	Crops Evaluated	Top-1 Error	Top-5 Error
VGGNet [18]	2	-	23.7%	6.8%
GoogLeNet [20]	7	144	-	6.67%
PReLU [6]	-	-	-	4.94%
BN-Inception [7]	6	144	20.1%	4.9%
Inception-v3	4	144	**17.2%**	**3.58%***

Table 3.2.3.2: Multi-crop reported results

We can see that the inception V3 model has an extremely low error rate compared with its previous models and its contemporaries.

➢ 3.2.4 Model Compilation and Training

Compiling the Model:

To compile a model with categorical cross-entropy loss, Adam optimizer, and accuracy metric in Python using TensorFlow/Keras, the following code snippet is used:

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Categorical Cross-Entropy Loss: This loss function is commonly used in multi-class classification problems where each sample belongs to a single class. It measures the difference between the predicted class probabilities and the true class labels.

We utilize categorical cross-entropy loss in multi-class classification tasks with more than two mutually exclusive classes. Similarly to the binary, this type of cross-entropy loss function quantifies the dissimilarity between the predicted probabilities and the true categorical labels.

And here's how we represent the categorical cross-entropy loss formula:

$$L = -\frac{1}{N}\sum_{i=1}^{N} = \sum_{j=1}^{C} = y_{ij}\, log(p_{ij})$$

The categorical cross-entropy loss function is commonly used in neural networks with softmax activation in the output layer for multi-class classification tasks. By minimizing loss, the model learns to assign higher probabilities to the correct class while reducing the probabilities for incorrect classes, improving accuracy.

Adam Optimizer:

Adam(Adaptive Moment Estimation) is an adaptive optimization algorithm that was created specifically for deep neural networks training. It can be viewed as a fusion of momentum-based stochastic gradient descent and RMSprop. It scales the learning rate using squared gradients, similar to RMSprop, and leverages momentum by using the gradient's moving average rather than the gradient itself, similar to SGD with momentum.

To estimate momentum, Adam uses exponential moving averages computed on the gradients evaluated on the current mini-batch. Mathematically, this can be written as :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2$$

where m and v are the moving averages and g is the gradient value. The betas are hyper-parameters whose good default values are, as suggested in the paper, 0.9 and 0.999 respectively.

Now as expectation values of the moments and gradient value should be equal to each other, we take the mean value of the moments, like:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\widehat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Using all this information, Adam updates the weights using the following formula which is quite similar to the formula we use in RMSprop:

$$w_t = w_{t-1} - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}}$$

where w is the weight, eta is the learning rate and epsilon is an infinitely small value, usually 10^{-8}, which we use to avoid division by zero.

Training the Model

To train the model using the training and validation generators for 50 epochs with specified steps per epochand validation steps, the following code snippet can be used:

```
r = model.fit(
  training_generator,
  validation_data=validation_generator,
  epochs=50,
  steps_per_epoch=len(training_generator),
  validation_steps=len(validation_generator)
)
```

Epochs: An epoch refers to one complete pass through the entire training dataset. Training for multiple epochs allows the model to learn from the data iteratively and improve its performance over time.

Steps per Epoch and Validation Steps: These parameters define how many batches of samples are processed in each epoch during training and validation, respectively. They are crucial for controlling the flow of data and ensuring efficient training and evaluation.

➢ 3.2.5 Model Evaluation of Trained CNN

To evaluate the performance of the trained Convolutional Neural Network (CNN) model on the test dataset for classifying retinal images related to diabetic retinopathy, several key steps need to be taken to assess its effectiveness accurately:

Accuracy: Calculate the accuracy of the model by determining the proportion of correctly classified images out of the total number of images in the test dataset. This metric provides an overall measure of the model's correctness in predicting diabetic retinopathy.

Precision: Compute the precision of the model, which represents the ratio of true positive predictions to the total number of positive predictions made by the model. Precision indicates the model's ability to avoid false positives when identifying diabetic retinopathy.

Recall: Calculate the recall of the model, which is the ratio of true positive predictions to the total number of actual positive instances in the dataset. Recall assesses the model's capability to capture all instances of diabetic retinopathy, minimizing false negatives.

F1-Score: Determine the F1-score, which is the harmonic mean of precision and recall. This metric provides a balanced assessment of the model's performance, considering both precision and recall simultaneously.

Theoritically F1 Score can be calculated as:

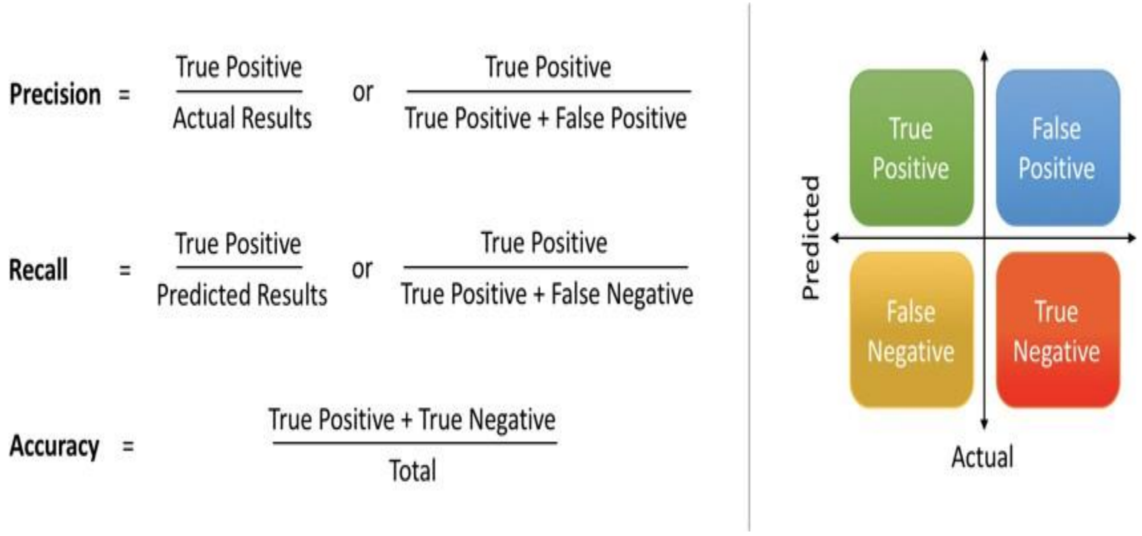


Fig 3.2.5.1: Formulae to calculate metrics

Visualization: Generate a confusion matrix to visually represent the model's predictions across different severity levels of diabetic retinopathy. The matrix displays true positive, true negative, false positive, and false negative predictions, enabling a comprehensive view of the model's classification outcomes.

Analysis: Analyze the confusion matrix to identify potential misclassifications by observing where the model's predictions deviate from the actual labels. This analysis helps in understanding the model's strengths and weaknesses in distinguishing between different severity levels of diabetic retinopathy.

By calculating metrics like accuracy, precision, recall, and F1-score, and visualizing the model's predictions through a confusion matrix, researchers and practitioners can gain valuable insights into the CNN model's performance in classifying retinal images for diabetic retinopathy.

➢ 3.2.6 Model Saving for Trained CNN

Saving the trained Convolutional Neural Network (CNN) model, including its Architecture and learned weights, is crucial for future use and deployment in various applications.

Here's how you can effectively save the model for seamless retrieval and integration:

Serialization Techniques:

JSON Serialization: Save the trained CNN model as a JSON file, capturing its architecture, configuration, and learned parameters. This serialization technique allows for easy storage and retrieval of the model's structure and weights, enabling seamless integration into other applications or frameworks.

TensorFlow's SavedModel Format: Utilize TensorFlow's SavedModel format to save the trained CNN model, ensuring compatibility and ease of deployment within the TensorFlow ecosystem. This format encapsulates the model's architecture, variables, and assets, providing a standardized way to store and load the model for future inference tasks.

Benefits of Model Saving:

Future Use: By saving the trained CNN model, you preserve the model's learned knowledge and architecture, allowing for future retraining, fine-tuning, or deployment in different environments or applications.

Portability: Serialization techniques like JSON and TensorFlow's SavedModel format enhance the portability of the model, enabling seamless transfer across platforms and systems without loss of information or functionality.

Scalability: Saved models can be easily scaled and deployed in production settings, facilitating real-time inference tasks and integration into larger systems for diagnosing diabetic retinopathy efficiently and accurately.

By employing serialization techniques such as saving the model as a JSON file or using TensorFlow's SavedModel format, researchers and developers can ensure the preser- vation and accessibility of the trained CNN model for future use in diagnosing diabetic retinopathy. These methods facilitate easy retrieval, integration, and deployment of the model, enhancing its versatility and applicability in healthcare settings and beyond.

➢ 3.2.7 Prediction of Output using Saved CNN Model

When utilizing the saved Convolutional Neural Network (CNN) model to make predictions on new, unseen retinal images for diabetic retinopathy, it is essential to follow a structured approach to ensure accurate and reliable results:

Prediction Process:

Model Loading: Load the saved CNN model along with its architecture and learned weights to prepare for making predictions on new retinal images. This step ensures that the model is ready for inference tasks and can classify diabetic retinopathy accurately.

Image Pre-processing: Pre-process the input images following the same steps used during training to maintain consistency and compatibility with the model's input re- quirements. This includes resizing, normalization, and any other pre-processing steps necessary to prepare the images for input into the CNN model.

Inference: Apply the trained CNN model to classify retinal images into different se- verity levels of diabetic retinopathy. By feeding the pre-processed images into the model, you can obtain predictions indicating the presence and severity of diabetic ret- inopathy based on the learned patterns and features extracted by the model during training.

Consistency: Ensuring that the input images are pre-processed in a manner consistent with the training data helps maintain compatibility with the model's expectations, lead- ing to more reliable predictions.

Model Performance: By leveraging the saved CNN model for prediction tasks, you can benefit from the model's learned knowledge and capabilities in accurately classi- fying retinal images for diabetic retinopathy. This enables efficient screening and di- agnosis of the disease based on visual indicators present in fundus images.

Real-time Application: The ability to predict diabetic retinopathy using a saved CNN model opens up possibilities for real-time applications in healthcare settings, allowing for quick and automated assessment of retinal images to aid in early detection and in- tervention for patients at risk

CHAPTER 4

CODE EXPLANATION

The entire Diabetic Retinopathy using CNN system was built using the python code. The breakdown of the code done below

- **Importing Libraries:**

```
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten, Dropout
from tensorflow.keras.models import Model, Sequential
import tensorflow as tf
import tensorflow_datasets as tfds
import numpy as np
from sklearn.model_selection import train_test_split
import os
from glob import glob
import matplotlib.pyplot as plt
import cv2
import itertools
import random
from collections import Counter
from glob import iglob
```

  - The code begins by importing various libraries and modules necessary for the project. These include TensorFlow for building and training neural networks, Matplotlib for visualization, NumPy for numerical operations, and scikit-learn for data manipulation.

- **Setting Image Size and Paths:**

```
IMAGE_SIZE = [512, 512]
train_path = r'C:\Users\Windows\Downloads\DR Notebook\val'
```

valid_path = r'C:\Users\Windows\Downloads\DR Notebook\dataset'

- IMAGE_SIZE is defined to specify the dimensions to which input images will be resized before feeding them into the model.
- Paths to the directories containing training and validation images are specified using the train_path and valid_path variables.

➢ **Loading and Analyzing Image Data:**

```
train_images = glob(os.path.join(train_path, '/.jpg'))
valid_images = glob(os.path.join(valid_path, '/.jpg'))
total_images1 = len(train_images)
total_images2 = len(valid_images)
print('Train images:', total_images1)
print('Valid images:', total_images2)
image_count = []
class_names = []
print('Training images\n')
for folder in os.listdir(os.path.join(train_path)):
    folder_num = len(os.listdir(os.path.join(train_path, folder)))
    image_count.append(folder_num)
    class_names.append(folder)
    print('{:20s}'.format(folder), end=' ')
    print(folder_num)
print('\n')
print('Validating images\n')
for folder in os.listdir(os.path.join(valid_path)):
    folder_num = len(os.listdir(os.path.join(valid_path, folder)))
    print('{:20s}'.format(folder), end=' ')
    print(folder_num)
```

- File paths of JPEG images from the specified directories are retrieved using the glob function.
- The total number of images in the training and validation datasets is calculated and printed for verification.
- Class names and their corresponding image counts are printed for both training and validation datasets.

- **Data Preprocessing:**

```
train_datagen = ImageDataGenerator(
    rescale=1./512,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)

valid_datagen = ImageDataGenerator(rescale=1./255)

training_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=(512, 512),
    batch_size=32,
    class_mode='categorical'
)

validation_generator = valid_datagen.flow_from_directory(
    valid_path,
    target_size=(512, 512),
    batch_size=32,
    class_mode='categorical'
)
```

- ImageDataGenerator objects are created to perform data augmentation and normalization on the training and validation datasets.
- Data augmentation techniques such as shear range, zoom range, and horizontal flip are applied to the training images.
- The flow_from_directory method is used to generate batches of preprocessed images and their corresponding labels for model training and validation.

➢ **Model Building:**

```
inception = InceptionV3(
   input_shape=IMAGE_SIZE + [3],
   weights='imagenet',
   include_top=False
)

for layer in inception.layers:
   layer.trainable = False

x = Flatten()(inception.output)
prediction = Dense(5, activation='softmax')(x)
model = Model(inputs=inception.input, outputs=prediction)

model.compile(
   loss='categorical_crossentropy',
   optimizer='adam',
   metrics=['accuracy']
)
```

- InceptionV3 is loaded as the base model with pre-trained weights trained on the ImageNet dataset.
- The base model's layers are set to non-trainable to retain the pre-trained weights.

- Additional layers (Flatten and Dense) are added to the model for custom classification.
- The model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric.

➢ **Model Training:**

```
r = model.fit(
    training_generator,
    validation_data=validation_generator,
    epochs=50,
    steps_per_epoch=len(training_generator),
    validation_steps=len(validation_generator)
)

plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()

plt.plot(r.history['accuracy'], label='train accuracy')
plt.plot(r.history['val_accuracy'], label='val accuracy')
plt.legend()
plt.show()
```

- The model is trained using the fit method with the training and validation data generators.
- Training is performed for 50 epochs with batch size defined in the generators.
- Loss and accuracy metrics are plotted to visualize the training and validation performance.

➢ **Saving the Model:**

model.save('DRmodel_inception.h5')

- The trained model is saved to a file named 'DRmodel_inception.h5' for future use.

➢ **Prediction:**

```
img = load_image(r'C:\Users\User\Documents\Python Projects\Diabatic Retinopathy RCNN\CodeDR\1170_left.jpeg')
prediction = predict(img)
plt.imshow(img)
plt.show()
print("Predicted Disease is", (prediction))
```

- Functions for loading and preprocessing a single image for prediction are defined.
- An example image is loaded, preprocessed, and passed to the model for prediction.
- The predicted disease class is printed along with the image.

This breakdown provides a detailed explanation of each code segment's purpose and functionality within the context of the diabetic retinopathy classification using CNN system

CHAPTER - 5

RESULTS

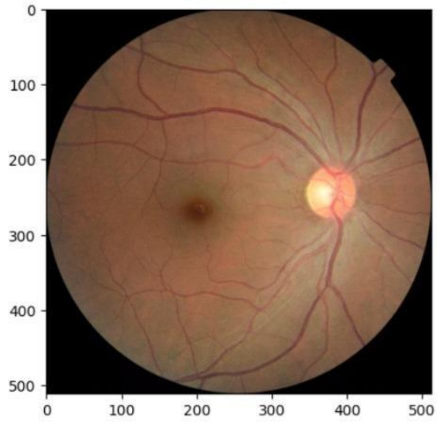
5.1 Output

The model was trained successfully. A fundus image has been selected and the output of the Diabetic Retinopathy using CNN system is depicted in fig.

```
[39]: # veryfying with custom image
%matplotlib inline
img = load_image(r"C:\Users\HUSSAIN BASHA\Downloads\POCO\DR Notebook\val\Mild\240_right - Copy (7).jpeg")
prediction = predict(img)
plt.imshow(img)
plt.show()
test = class_names

print("Level of the disease is predicted as", (prediction))
```

```
1/1 [==============================] - 0s 114ms/step
```




```
Level of the disease is predicted as Mild
```

Fig 5.1.1: Predicted output of the Model

The screenshot fig. shows that, the model predicted whether the selected image has diabetic retinopathy or not. If the image is having diabetic retinopathy, the severeness of it also predicted.

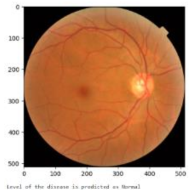
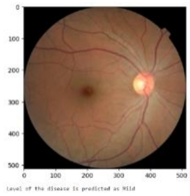
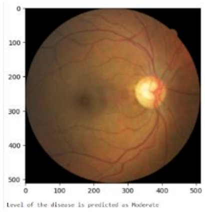
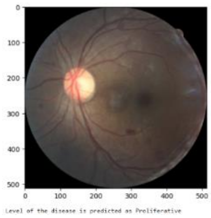
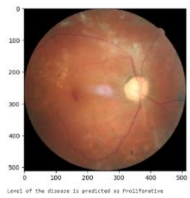
INPUT	EXPECTED OUTPUT	PREDICTED OUTPUT
	NORMAL	 (NORMAL)
	MILD	 (MILD)
	MODERATE	 (MODERATE)
	SEVERE	 (PROLIFERATIVE)
	PROLIFERATIVE	 (PROLIFERATIVE)

Table 5.1.1 Results comparison

5.2 Result analysis:

The model trained at different sets of input images, like changing the count of images in the each set of Training and Testing datasets. Each time the accuracy of the model was noted down. The result was tabulated.

The prediction accuracy of the model at different cases, When total number of images are 1917 :

CASE	Training Images	Test Images	Accuracy(%)
I	163	1754	90.99
II	217	1700	94.82
III	770	1147	95.55
IV	958	959	98.54

Table 5.2.1 : Accuracy percentage

Rate of Acceptance and Rate of Rejection:

- **True Acceptance:** True acceptance occurs when a positive instance is correctly classified as positive by the system.
- **True Rejection:** True rejection occurs when a negative instance is correctly clas- sified as negative by the system.
- **False Acceptance:** False acceptance occurs when a negative instance is incor- rectly classified as positive by the system.
- **False Rejection:** False rejection occurs when a positive instance is incorrectly classified as negative by the system.

Case I:

- The model got an accuracy of 90.99 % when 163 and 1754 images are given as Training and Testing data respectively i.e., 8 - 92 percentage.
- The Rate of Acceptance (ROA), Rate of Rejection (ROR) is depicted as follows:

CASE I	True	False
Acceptance	**838**	**74**
Rejection	**758**	**84**

Table 5.2.2 : ROA & ROR for Case 1

Case II:

- The model got an accuracy of **94.82 %** when 217 and 1700 images are given as Training and Testing data respectively i.e., 12-88 percentage.
- The Rate of Acceptance, Rate of Rejection is depicted as follows:

CASE II	True	False
Acceptance	**820**	**34**
Rejection	**792**	**54**

Table 5.2.3: ROA & ROR for case 2

Case III:

- The model got an accuracy of **95.55 %** when 770 and 1147 images are given as Training and Testing data respectively i.e., 40-60 percentage.
- The Rate of Acceptance, Rate of Rejection is depicted as follows:

CASE 1	True	False
Acceptance	**574**	**12**
Rejection	**522**	**39**

Table 5.2.4: ROA & ROR for case 3

Case IV:

- The model got an accuracy of **98.54 %** when 959 and 958 images are given as Training and Testing data respectively i.e., 50-50 percentage.
- The Rate of Acceptance, Rate of Rejection is depicted as follows:

CASE 1	True	False
Acceptance	**458**	**25**
Rejection	**433**	**43**

Table 5.2.5: ROA & ROR for case 4

CHAPTER - 6

ADVANTAGES AND DISADVANTAGES

6.1 Advantages:

The advantages of the Diabetic Retinopathy project using the InceptionV3 model can be outlined as:

a. High Accuracy:

The project achieved a high accuracy rate of around 97% in detecting Diabetic Retinopathy using the InceptionV3 model, indicating its effectiveness in accurately identifying the condition.

b. Automated Detection:

By utilizing Deep Convolutional Neural Networks with the InceptionV3 model, the project automated the detection of Diabetic Retinopathy, reducing the need for manual examination by ophthalmologists and streamlining the diagnostic process.

c. Efficient Training:

The model was trained on a substantial dataset of 35,126 retinal images openly released by eyePACS on Kaggle, leveraging GPU acceleration for efficient training and achieving a high level of accuracy.

d. Less Epochs:

Compared to other models, the InceptionV3 model in this project required fewer epochs (10 epochs) to achieve high accuracy, indicating faster convergence and efficient training.

e. Robust System:

The project aimed to develop a robust system for automatically detecting Diabetic Retinopathy, showcasing the reliability and effectiveness of the model in identifying the condition with a high level of accuracy.

f. Advanced Neural Network:

By using the InceptionV3 model, the project benefited from a sophisticated neural network architecture capable of extracting intricate features from retinal images, enhancing the model's ability to detect Diabetic Retinopathy accurately.

g. Research Contribution:

The project contributes to the field by introducing a novel approach to detecting Diabetic Retinopathy, showcasing the potential of machine learning and deep learning techniques in improving the diagnosis and management of the condition.

These advantages highlight the project's significant contributions in automating the detection of Diabetic Retinopathy with high accuracy, efficient training processes, and the utilization of advanced neural network models like InceptionV3 for improved diagnostic outcomes.

6.2 Disadvantages:

1. Computational Intensity: Inception V3's depth and width can lead to high computational demands during training and inference, requiring substantial resources.

2. Potential Overfitting: Compared to newer architectures like EfficientNetB4, Inception V3 may be more prone to overfitting, impacting its generalization ability in diabetic retinopathy detection tasks.

3. Limited Performance: While Inception V3 has shown high accuracy rates, it may not offer the same level of performance as more advanced models in certain aspects of diabetic retinopathy analysis.

4. Complexity: The intricate design of the Inception V3 module can make it challenging to interpret and fine-tune for specific diabetic retinopathy detection requirements.

5. Resource Requirements: Training and deploying Inception V3 for diabetic retinopathy analysis may necessitate substantial computational resources and time compared to more efficient models available today.

CHAPTER - 7

APPLICATIONS AND LIMITATIONS

7.1 Applications:

a. Early Detection and Intervention:

Efficient classification models enable early detection of diabetic retinopathy, allowing for timely intervention and treatment to prevent vision loss.

b. Patient Monitoring:

Automated classification systems facilitate regular monitoring of diabetic patients' retinal health, enabling healthcare providers to track disease progression and adjust treatment plans accordingly.

c. Resource Optimization:

By automating the screening process, classification models help optimize healthcare resources by reducing the burden on ophthalmologists and enabling them to focus on patients requiring further evaluation and treatment.

d. Telemedicine and Remote Care:

Diabetic retinopathy classification models support telemedicine initiatives by enabling remote screening and diagnosis of patients, especially in underserved or remote areas where access to specialized healthcare services is limited.

e. Population Screening Programs:

Classification models can be integrated into population screening programs to efficiently screen large numbers of individuals for diabetic retinopathy, thereby improving public health outcomes through early detection and intervention.

f. Research and Clinical Trials:

Accurate classification of diabetic retinopathy facilitates research efforts and clinical trials aimed at developing new treatments, evaluating treatment efficacy, and understanding disease mechanisms.

g. Personalized Medicine:

Classification models contribute to personalized medicine by providing individualized risk assessments and treatment recommendations based on patients' specific retinal characteristics and disease severity.

h. Education and Awareness:

Utilizing classification models in educational campaigns increases awareness about diabetic retinopathy among diabetic patients, healthcare providers, and the general population, emphasizing the importance of regular eye screenings and disease manage.

7.2 Limitations:

Misclassification Risk: There is a risk of misclassification in automated diagnosis.

Human Supervision Requirement: Due to its high sensitivity, it may lack the precision. So the presence of a professional is required.

Training Data Size: Inception V3 may require a large amount of labeled data for effective training, which can be a limitation in scenarios where obtaining such data is challenging or costly.

Model Size: The size of the Inception V3 model can be relatively large compared to more streamlined architectures, potentially impacting deployment on resource-constrained devices or in real-time applications.

Fine-tuning Complexity: Fine-tuning the Inception V3 model for diabetic retinopathy detection may require expertise and effort due to its complex architecture, which can be a limitation for users with limited deep learning experience.

Adaptability: Inception V3 may not be as easily adaptable to different variations or nuances in diabetic retinopathy images compared to models specifically designed or optimized for this task.

Interpretability: The complex nature of the Inception V3 model may hinder its interpretability, making it challenging to understand how decisions are made, which can be crucial for medical applications like diabetic retinopathy diagnosis.

CHAPTER - 8

CONCLUSION

- **Effective Model Training:** Trained the Inception V3 model successfully.
- **High Accuracy:** Model testing yielded accurate predictions for Diabetic Retinopathy.
- **Clinical Utility:** Model can now predict Diabetic Retinopathy in fundus images, aiding early detection.
- **Healthcare Impact:** Offers potential for efficient diagnosis and timely intervention in healthcare settings.
- **Future Development:** Opportunities for refining the model to enhance diagnos- tic capabilities and improve patient outcomes**.**
- **Challenges in Performance:** While Inception V3 has shown high accuracy rates, its limitations in computational intensity, potential overfitting, and specific performance constraints in diabetic retinopathy analysis pose challenges for the project's overall effectiveness.
- **Resource Demands:** The project using Inception V3 requires significant computational resources for training and deployment, which can impact scalability and accessibility, especially in resource-constrained environments.
- **Complexity in Implementation:** The complex nature of Inception V3, coupled with challenges in fine-tuning and interpretation, adds a layer of complexity to the project, potentially requiring specialized expertise for optimal utilization.
- **Adaptability Concerns:** Inception V3's adaptability to varying data distributions and specific nuances in diabetic retinopathy images may be limited compared to models designed explicitly for this medical task.
- **Consideration for Alternatives:** Given the limitations and challenges associated with Inception V3, exploring more efficient and specialized models tailored for diabetic retinopathy detection could enhance the project's performance, scalability, and interpretability in the long run

REFERENCES

1. AbdelMaksoud, E, Barakat, S, Elmogy, M (2022) A computer-aided diagnosis system for detecting various diabetic retinopathy grades based on a hybrid deep learning technique, Med Biol Eng Comput, pp.1–24 10.1007/s11517-022- 02564-6 [PMC free article] [PubMed]
2. Agneeswaran, VS (n.d.) Computational Complexity of Deep Learning: Solution Approaches, Walmart Global Tech Blog, Available at:https://medium.com/walmartglobaltech/computational-complex- ity-of-deep-learning-a-birds-eye-view-2250b7c098a1, Accessed on: 20-05- 2022
3. Atwany MZ, Sahyoun AH, Yaqub M. Deep learning techniques for diabetic Retinopathy classification: a survey. IEEE Access. 2022;10:28642–28655. doi: 10.1109/ACCESS.2022.3157632. [CrossRef] [Google Scholar]
4. Bhatti UA, Huang M, Wu D, Zhang Y, Mehmood A, Han H. Recommendation system using feature extraction and pattern recognition in clinical care sys- tems. Enterprise Inf Syst. 2018;13(3):329–351. doi: 10.1080/17517575.2018.1557256. [CrossRef] [Google Scholar]
5. Bhatti UA, Huang M, Wang H, Zhang Y, Mehmood A, Di W. Recommendation system for immunization coverage and monitoring. Human Vaccines Immuno- therapeutics. 2018;14(1):165–171. doi: 10.1080/21645515.2017.1379639. [PMC free article] [PubMed] [Cross- Ref] [Google Scholar]
6. Bhatti UA, Yu Z, Li J, Nawaz SA, Mehmood A, Zhang K, Yuan L. Hybrid watermarking algorithm using Clifford algebra with Arnold scrambling and chaotic encryption. IEEE Access. 2020;8:76386–76398. doi: 10.1109/AC- CESS.2020.2988298. [CrossRef] [Google Scholar]

   Bhatti UA, Yu Z, Chanussot J, Zeeshan Z, Yuan L, Luo W, Nawaz SA, Bhatti MA, Ain Q, Mehmood A. Local similarity-based spatial–spectral fusion hyper- spectral image classification with deep CNN and Gabor filtering. IEEE Trans Geosci Remote Sens. 2021;60(5514215):1–15. doi: 10.1109/TGRS.2021.3090410. [CrossRef] [Google Scholar]
7. Bhatti UA, Zeeshan Z, Nizamani MM, Bazai S, Yu Z, Yuan L. Assessing the change

of ambient air quality patterns in Jiangsu Province of China pre-to post- COVID-19. Chemosphere. 2022;288:1–10. doi: 10.1016/j.chemo- sphere.2021.132569. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

8. Bhilare, A (2022) MACC-FLOPS,

Available at: https://github.com/AbhijeetBhilare777/MACC-FLOPS, Accessed on : 06-06-2022

9. Bodapati JD, Veeranjaneyulu N, Shareef SN, Hakak S, Bilal M, Maddikunta PKR, Jo O. Blended multi-modal deep convnet features for diabetic retinopathy severity prediction. Electronics. 2020;9(6):914. doi: 10.3390/electron- ics9060914. [CrossRef] [Google Scholar]

10. Bora A, Balasubramanian S, Babenko B, Virmani S, Venugopalan S, Mitani A, Marinho GO, Cuadros J, Ruamviboonsuk P, Corrado GS, Peng L, Webster DR, Varadarajan AV, Hammel N, Liu Y, Bavishi P. Predicting the risk of developing diabetic retinopathy using deep learning. Lancet Digit Health. 2021;2020(3):e10–e19. doi: 10.1016/S2589-7500(20)30250-8. [Pub- Med] [CrossRef] [Google Scholar]

11. Chakraborty C, Kishor A, Rodrigues JJPC. Novel enhanced-Grey wolf optimi- zation hybrid machine learning technique for biomedical data computa- tion. Comput Electr Eng. 2022;99:1–15. doi: 10.1016/j.compele- ceng.2022.107778. [CrossRef] [Google Scholar]

12. Chaturvedi SS, Gupta K, Ninawe V, Prasad PS (2020) Automated diabetic ret- inopathy grading using deep convolutional neural network, arXiv:2004.06334v1 [eess.IV], pp 1–12. 10.48550/arXiv.2004.06334

13. Chetoui, M, Akhloufi, MA (2020) Explainable Diabetic Retinopathy using Ef- ficientNET, In 2020 42nd Annual International Conference of the IEEE Engi- neering in Medicine & Biology Society (EMBC), pp.1966–1969, 10.1109/EMBC44109.2020.9175664 [PubMed]

14. Chollet F (2017) Xception: Deep learning with depthwise separable convolu- tions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , arXiv:1610.02357v3 [cs.CV], pp 1251–1258. 10.48550/arXiv.1610.02357

15. Das S, Kharbanda K, Suchetha M, Raman R, Edwin DD. Deep learning archi- tecture based on segmented fundus image features for classification of diabetic retinopathy. Biomed Signal Process Control. 2021;68:1–10.

doi: 10.1016/j.bspc.2021.102600. [CrossRef] [Google Scholar]

16. Das D, Biswas SK, Bandyopadhyay S. A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning. Multimed Tools Appl. 2022;23:1–43. doi: 10.1007/s11042-022-12642-4. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

17. Deepa V, Kumar CS, Cherian T (2021) Ensemble of multi-stage deep convolu- tional neural networks for automated grading of diabetic retinopathy using im- age patches, Journal of King Saud University –Computer and Information Sci- ences, pp 1–11 10.1016/j.jksuci.2021.05.009

18. Deepa, V, Kumar, SC, Cherian, T (2022) Automated grading of diabetic reti- nopathy using CNN with hierarchical clustering of image patches by siamese network, Physical and Engineering Sciences in Medicine, pp.1–13 10.1007/s13246-022-01129-z [PubMed]

19. Diabetic Retinopathy Detection, Kaggle (n.d.)

Available at: https://www.kaggle.com/c/diabetic-retinopathy-detection/data. Accessed 01-07-2021

20. Diabetic Retinopathy Detection, Kaggle repository (n.d.)

Available at: https://www.kaggle.com/c/diabetic-retinopathy-detection/data, Accessed on 19-06-2022

21. Diabetic Retinopathy, Updated (2022),

Available at: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions- and-diseases/diabetic-retinopathy, Accessed on : 16-5-2022

22. Dong, B, Wang, X, Qiang, X, Du, F, Gao, L, Wu, Q, Cao, G, Dai, C (2022) A Multi-Branch Convolutional Neural Network for Screening and Staging of Diabetic Retinopathy Based on Wide-Field Optical Coherence Tomography Angiography, IRBM, pp.1–7 10.1016/j.irbm.2022.04.004

23. Ege BM, Hejlesen OK, Larsen OV, Møller K, Jennings B, Kerr D, Cavan DA. Screening for diabetic retinopathy using computer based image analysis and statistical classification, Comput. Methods Programs Biomed. 2000;62(3):165–175. doi: 10.1016/S0169-2607(00)00065-1. [PubMed] [CrossRef] [Google Scholar]

24. Fadzil, MHA, Ngah, NF, George, TM, Izhar, LI, Nugroho, H, Nugroho, HA (2010) Analysis of foveal avascular zone in colour fundus images forgrading of diabetic

retinopathy severity, 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology, pp. 5632–5635, 10.1109/IEMBS.2010.5628041 [PubMed]

25. Fadzil MHA, Izhar LI, Nugroho H, Nugroho HA. Analysis of retinal fundus images for grading of diabetic retinopathy severity. Med Biol Eng Compu. 2011;49(6):693–700. doi: 10.1007/s11517-011-0734-2. [PubMed] [CrossRef] [Google Scholar]

26. Fan GF, Zhang LZ, Yu M, Hong WC, Dong SO. Applications of random forest in multivariable response surface for short-term load forecasting. Int J Electr Power Energy Syst. 2022;139:1–30. doi: 10.1016/j.ijepes.2022.108073. [CrossRef] [Google Scholar]

27. Fong DS, Aiello L, Gardner TW, King GL, Blankenship G, Cavallerano JD, Ferris FL, Klein R. Retinopathy in diabetes. Diabetes Care. 2004;27(1):84–87. doi: 10.2337/diacare.27.2007.S84. [PubMed] [CrossRef] [Google Scholar]

28. Goh JKH, Cheung CY, Sim SS, Tan PC, Tan GSW, Wong TY. Retinal imaging techniques for diabetic retinopathy screening. J Diabetes Sci Tech- nol. 2016;10(2):282–294. doi: 10.1177/1932296816629491. [PMC free arti- cle] [PubMed] [CrossRef] [Google Scholar]

29. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venu- gopalan S, Widner K, Madams T, Cuadros J, Kim R, Raman R, Nelson PC, Mega JL, Webster DR. Development and validation of a deep learning algo- rithm for detection of diabetic Retinopathy in retinal fundus photographs. JAMA. 2016;316(22):2402–2410.

doi: 10.1001/jama.2016.17216. [PubMed] [CrossRef] [Google Scholar]

30. Gurcan OF, Beyca OF, Dogan O. A Comprehensive Study of Machine Learning Methods on Diabetic Retinopathy Classification. Int J Comput Intell Syst. 2021;14(2):1132–1141. doi: 10.2991/ijcis.d.210316.001. [Cross- Ref] [Google Scholar]

31. Hagos MT (2020) Point-of-care diabetic retinopathy diagnosis: a standalone mobile application approach, arXiv:2002.04066v1 [eess.IV]. Vol. abs/2002.04066, pp 1–84. Available at: https://www.semanticscholar.org/paper/Point-of-Care-Diabetic-Retinopathy-Diagnosis%3A- AHagos/3b9a121e53cea1b30cd29eea13854eeb4faadff9

32. Hattiya T, Dittakan K, Musikasuwan S. Diabetic Retinopathy Detection Using

Convolutional Neural Network: A Comparative Study on Different Architec- tures. Mahasarakham Int J Eng Technol. 2021;7(1):50–60. doi: 10.14456/mi- jet.2021.8. [CrossRef] [Google Scholar]

33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recogni- tion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778. 10.1109/CVPR.2016.90

34. Howard, AG, Zhu, M, Chen, B, Kalenichenko, D, Wang, W, Weyand, T, An- dreetto, M, Adam, H (2017) MobileNets: efficient convolutional neural net- works for mobile vision applications, arXiv:1704.04861v1 [cs.CV], pp 1–9. 10.48550/arXiv.1704.04861

35. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, arXiv:1608.06993v5 [cs.CV]. 10.48550/arXiv.1608.06993

36. Huang G, Liu S, Maaten L, Weinberger KQ (2018) CondenseNet: an efficient densenet using learned group convolutionss. In: Proceedings of the IEEE Con- ference on Computer Vision and Pattern Recognition (CVPR), pp 2752–2761. 10.1109/CVPR.2018.00291

37. Hui J, Du M, Ye X, Qin Q, Sui J. Effective building extraction from high-reso- lution remote sensing images with multitask driven deep neural network. IEEE Geosci Remote Sens Lett. 2019;16(5):786–790. doi: 10.1109/LGRS.2018.2880986. [CrossRef] [Google Scholar]

38. Iandola, F, Mskewicz, M, Karayev, S, Girshick, R, Darrell, T, Kuetzer, K (2014) DenseNet: Implementing Efficient ConvNet Descriptor Pyramids Technical Report, pp.1–11 10.1080/08839514.2013.848751

39. S. Akara and B. Uyyanonvara, ‘‘Automatic exudates detection from diabetic retinopathy retinal image using fuzzy C-means and morphological methods,’’ in *Proc. 3rd IASTED Int. Conf. Adv. Comput. Sci. Technol.*, 2007, pp. 359–364.

40. V. Deepika, R. Dorairaj, K. Namuduri, and H. Thompson, ‘‘Automated detection and classification of vascular abnormalities in diabetic retinopathy,’’ in *Proc. 38th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2004, pp. 1625–1629.

41. H. Li, W. Hsu, M. L. Lee, and T. Y. Wong, ‘‘Automatic grading of retinal vessel caliber,’’ *IEEE Trans. Biomed. Eng.*, vol. 52, no. 7, pp. 1352–1355, Jul. 2005.

42. E. Trucco et al., ‘‘Validating retinal fundus image analysis algorithms: Issues and a proposal,’’ *Investigative Opthalmol. Vis. Sci.*, vol. 54, no. 5, p. 3546, May 2013.

43. G. A. G. Vimala and S. Kajamohideen, ‘‘Diagnosis of diabetic retinopathy by extracting blood vessels and exudates using retinal color fundus images,’’ *WSEAS Trans. Biol. Biomed.*, vol. 11, pp. 20–28, Jun. 2014.

44. M. Bilal, G. Ali, M. W. Iqbal, M. Anwar, M. S. A. Malik, and R. A. Kadir, ‘‘Auto-prep: Efficient and automated data preprocessing pipeline,’’ *IEEE Access*, vol. 10, pp. 107764–107784, 2022.

45. C. P. Wilkinson et al., ‘‘Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales,’’ *Ophthalmology*, vol. 110, no. 9, pp. 1677–1682, Sep. 2003.

46. K. Ram, G. D. Joshi, and J. Sivaswamy, ‘‘A successive clutter-rejectionbased approach for early detection of diabetic retinopathy,’’ *IEEE Trans. Biomed. Eng.*, vol. 58, no. 3, pp. 664–673, Mar. 2011.

47. I. Lazar and A. Hajdu, ‘‘Microaneurysm detection in retinal images using a rotating cross-section based model,’’ in *Proc. IEEE Int. Symp. Biomed. Imag., From Nano Macro*, Mar. 2011, pp. 1405–1409.

48. B. Antal and A. Hajdu, ‘‘An ensemble-based system for microaneurysm detection and diabetic retinopathy grading,’’ *IEEE Trans. Biomed. Eng.*, vol. 59, no. 6, pp. 1720–1726, Jun. 2012.

49. J. Amin, M. Sharif, and M. Yasmin, ‘‘A review on recent developments for detection of diabetic retinopathy,’’ *Scientifica*, vol. 2016, pp. 1–20, May 2016.

50. H. Narasimha-Iyer et al., ‘‘Robust detection and classification of longitudinal changes in color retinal fundus images for monitoring diabetic retinopathy,’’ *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1084–1098, Jun. 2006.

51. Z. Munawar et al., ‘‘Predicting the prevalence of lung cancer using feature transformation techniques,’’ *Egyptian Informat. J.*, vol. 23, no. 4, pp. 109–120, Dec. 2022.

52. Y. Yehui et al., ‘‘Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks,’’ in *Proc. Int. Conf. Med.ImageComput.Comput.-Assist.Intervent.*,Cham,Switzerland,2017, pp. 533–540.

53. R. A. Welikala et al., ‘‘Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification,’’ *Comput. Methods Programs Biomed.*, vol. 114, no. 3, pp. 247–261, May 2014.

54. M. J. Iqbal, M. W. Iqbal, M. Anwar, M. M. Khan, A. J. Nazimi, and M. N. Ahmad, ''Brain tumor segmentation in multimodal MRI using U-Net layered structure,'' *Comput., Mater. Continua*, vol. 74, no. 3, pp. 5267–5281, 2023.

55. M. Anwar et al., ''Green communication for wireless body area networks: Energy aware link efficient routing approach,'' *Sensors*, vol. 18, no. 10, p. 3237, Sep. 2018.

56. I. Sergey and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

57. M. Anwar, F. Masud, R. A. Butt, S. M. Idrus, M. N. Ahmad, and M. Y. Bajuri, ''Traffic priority-aware medical data dissemination scheme for IoT based WBASN healthcare applications,'' *Comput., Mater. Continua*, vol. 71, no. 3, pp. 4443–4456, 2022.

58. S. Naseem, A. Alhudhaif, M. Anwar, K. N. Qureshi, and G. Jeon, ''Artificial general intelligence-based rational behavior detection using cognitive correlates for tracking online harms,'' *Pers. Ubiquitous Comput.*, vol. 27, no. 1, pp. 119–137, Feb. 2023.

59. W. Yuxin and K. He, ''Group normalization,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

60. Y. Li, C. Huang, L. Ding, Z. Li, Y. Pan, and X. Gao, ''Deep learning in bioinformatics: Introduction, application, and perspective in the big data era,'' *Methods*, vol. 166, pp. 4–21, Aug. 2019.

61. Y. A. Ahmed et al., ''Sentiment analysis using deep learning techniques: A review,'' *J. Jilin Univ., Eng. Technol. Ed.*, vol. 42, no. 3, pp. 789–808, 2023.

62. M. Anwar, A. H. Abdullah, K. N. Qureshi, and A. H. Majid, ''Wireless bodyareanetworksforhealthcareapplications:Anoverview,''*Telkomnika*, vol. 15, no. 3, pp. 1088–1095, 2017.

63. L. A. Selvikvåg and A. Lundervold, ''An overview of deep learning in medical imaging focusing on MRI,'' *Zeitschrift Medizinische Physik*, vol. 29, no. 2, pp. 102–127, May 2019.

64. T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, ''Predictive analytics in education: A comparison of deep learning frameworks,'' *Educ. Inf. Technol.*, vol. 25, no. 3, pp. 1951–1963, May 2020.

65. R. Olaf, P. Fischer, and T. Brox, ''U-Net: Convolutional networks for biomedical image segmentation,'' in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Cham, Switzerland, 2015, pp. 234–241.

66. S. Targ, D. Almeida, and K. Lyman, ‘‘Resnet in resnet: Generalizing residual architectures,’’ 2016, *arXiv:1603.08029*.

67. E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus, ‘‘Malicious software classification using VGG16 deep neural network’s bottleneck features,’’ in *Information Technology—New Generations*, Cham, Switzerland: Springer, 2018, pp. 51–59.

68. X. Tio and A. Emmanuel, ‘‘Face shape classification using inception v3,’’ 2019, *arXiv:1911.07916*.

69. *Diabetic Retinopathy Detection*. Accessed: Mar. 1, 2015. [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection

70. Y. Yuhai et al., ‘‘Modality classification for medical images using multiple deep convolutional neural networks,’’ *J. Comput. Inf. Syst.*, vol. 11, no. 15, pp. 5403–5413, 2015.

71. Y. Katada, N. Ozawa, K. Masayoshi, Y. Ofuji, K. Tsubota, and T. Kurihara, ‘‘Automatic screening for diabetic retinopathy in interracial fundus images using artificial intelligence,’’ *Intell.-Based Med.*, vols. 3–4, Dec. 2020, Art. no. 100024.

72. S. Wan, Y. Liang, and Y. Zhang, ‘‘Deep convolutional neural networks for diabetic retinopathy detection by image classification,’’ *Comput. Electr. Eng.*, vol. 72, pp. 274–282, Nov. 2018.

73. Y.-P. Liu, Z. Li, C. Xu, J. Li, and R. Liang, ‘‘Referable diabetic retinopathy identificationfromeyefundusimageswithweightedpathforconvolutional neural network,’’ *Artif. Intell. Med.*, vol. 99, Aug. 2019, Art. no. 101694.

74. D. P. Kingma and J. Ba, ‘‘Adam: A method for stochastic optimization,’’ 2014, *arXiv:1412.6980*.

75. I. Nakahara et al., ‘‘Extraction of disease area from retinal optical coherence tomography images using three dimensional regional statistics,’’ *Proc. Comput. Sci.*, vol. 22, pp. 893–901, Jan. 2013.

76. B. Antal, I. Lázár, A. Hajdu, Z. Török, A. Csutak, and T. Peto, ‘‘Evaluation of the grading performance of an ensemble-based microaneurysm detector,’’ in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 5943–5946.

77. L. Tang, M. Niemeijer, J. M. Reinhardt, M. K. Garvin, and M. D. Abramoff, ‘‘Splat feature classification with application to retinal hemorrhage detection in fundus images,’’ *IEEE Trans. Med. Imag.*, vol. 32, no. 2, pp. 364–375, Feb. 2013.

78. Q. Chen, X. Xu, L. Zhang, and D. W. K. Wong, ‘‘REMURS: Remote medical ultrasound simulation on the cloud,’’ *IEEE J. Biomed. Health Informat.*, vol. 22, no. 2, pp. 173–181, Sep. 2017.

79. X. Chen et al., ‘‘Deep learning for diagnosis and evaluation of diabetic retinopathy: A review,’’ *J. Healthcare Eng.*, vol. 6, 2019, Art. no. 5481723.

80. H. Tang, Z. Wu, Y. Li, and J. Li, ‘‘Automated detection of retinal hemorrhages in fundus images using splat feature classification,’’ *J. Med. Syst.*, vol. 40, no. 4, p. 95, 2016.