Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer

Olyad Temesgen Beyene

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2023

Adama, Ethiopia

Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer

Olyad Temesgen Beyene

Advisor: Bahiru Shifaw Yimer (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2023

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Olyad Temesgen Beyene	_____	_____
Name	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer**" prepared under my/our guidance by Olyad Temesgen Beyene submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Bahiru Shifaw (Ph.D.) _____ _____

Major Advisor Signature Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "**Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer**" and developed by Olyad Temesgen Beyene, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Olyad Temesgen Beyene have read and evaluated the thesis entitled "**Transfer Learning based Multi Class Common Eye Disease Classification Using Enhanced Vision Transformer"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

Chair Person	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Graduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

Above all, I thank God, the creator and ruler of both the physical and spiritual realms for the blessings and support to undertake this research work. I am also thankful to Mother Saint Mary, and all the angels and saints for their countless and extraordinary blessings bestowed upon me.

I am extremely thankful to my advisor Bahiru Shifaw (Ph.D.), who made this thesis work possible. I appreciate for his support and guidance throughout the entire journey, from the initial stages to the final completion.

My Special Thanks to Dr. Adugna (ophthalmologist) and sister Tewodaj, who were involved in data collection process for this research work. Additionally, I am grateful to the esteemed faculty of the Computer Science and Engineering for imparting their invaluable knowledge throughout my studies. I sincerely thank all my friends who offered their support during my thesis period.

Finally and most importantly, I am very grateful to my parents for their boundless affection, constant prayers and support throughout my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SAMPLE CODE SNIPPETS

LIST OF ACRONYMS AND ABBREVIATIONS

AMD	Age-related Macular Degeneration
AUC	Area Under ROC Curve
CAD	Computer Aided Diagnosis
CCE	Categorical Cross Entropy
CNN	Convolutional Neural Network
DCGAN	Deep Convolutional Generative Adversarial Network
DR	Diabetic Retinopathy
GAN	Generative Adversarial Network
GeLU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
ICO	International Council of Ophthalmology
IDE	Integrated Development Environment
LN	Layer Normalization
MHSA	Multi Head Self-Attention
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
OCT	Optical Coherence Tomography
ODIR	Ocular Disease Intelligent Recognition
OpenCV	Open-Source Computer Vision Library
RAM	Random Access Memory
ReLu	Rectified Linear Unit
ResNet	Residual Neural Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
VGG	Visual Geometry Group

ViT	Vision Transformer
ViT-b16	Vision Transformer Base 16
ViT-b32	Vision Transformer Base 32
WHO	World Health Organization
2D	2-Dimension

LIST OF NOTATIONS AND SYMBOLS

Notation	Meaning
d_k	Dimension of keys
d_v	Dimension of values
X_{class}	Class embedding
E_{pos}	Positional Embedding
X_p^N	Indicates N patches ith P position
$QW_i^Q, KW_i^k, VW_i^v)$	Weight matrices of Query, Key and value for i^{th} input
Tt	Target learning task
Dt	Target Domain
Ds	Source Domain
Ts	Learning Task from Source
$\mathbb{R}$	Real Number
$\sum$	Summation
D_{pos}	Dimension of Positional Embedding
$\mathcal{F}T(\cdot)$	Predictive Function

ABSTRACT

Common Retinal diseases like diabetic retinopathy, glaucoma, age-related macular degeneration, and cataracts are leading causes of vision loss and blindness globally. In Ethiopia, the high prevalence coupled with limited number of ophthalmologists poses a major public health challenge. Early detection and diagnosis through fundus imaging can enable prompt treatment and prevent vision loss. However, manual analysis is time-consuming and prone to human error. While convolutional neural networks (CNNs) have shown promise in retinal image analysis, they have limitations in capturing global context in images. transformers have emerged as an alternative approach leveraging self-attention mechanism to model global context in images. This study investigates the potential of vision transformer model for multi-class classification of common retinal diseases using fundus photography. A custom vision transformer model is proposed by incorporating the standard architecture with convolutional and classifier layers tailored for retinal images. The study employed transfer learning from ImageNet pre-training. The model is trained on fundus datasets augmented through photometric and geometric transformations to expand the data size and improve generalization. Comparative assessment is conducted against Convolutional Neural Networks like ResNet50, VGG19 and InceptionV3. The custom vision transformer model achieves an accuracy of 95% in classifying normal retina and five disease classes outperforming vision transformer which scored 87-91% accuracy, as well as the CNN models(highest accuracy of 94%). Precision, recall and F1-score and AUC metrics are also improved over both baseline ViT and CNNs. Class activation mapping visualizations highlight discriminative regions focused on by the model, providing interpretability. The findings demonstrate the effectiveness of the custom transfer learning-based vision transformer model for retinal disease classification, demonstrating its superiority and generalization ability over convolutional neural networks. This work helps establish vision transformers as a promising approach toward developing automated screening systems to prevent vision impairment through early diagnosis of retinal diseases using fundus imaging.

Keywords: Fundus Images, Vision Transformer, CNN, Data Augmentation, Multi-class Classification.

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Retinal diseases, characterized by damage or abnormalities in the delicate tissue of the retina lining at the back of the eye, pose a significant health challenge worldwide. As per a report by WHO in 2022, over 2 billion people globally are affected by visual impairment or blindness, whether it is related to near or distant vision (*Blindness and Vision Impairment*, n.d.). As the report indicates, approximately half of these vision problems worldwide could have been prevented or are still untreated. The leading eye diseases that result in vision loss globally are glaucoma, diabetic retinopathy, age-related macular degeneration and cataracts (*Blindness and Vision Impairment*, n.d.; Bourne et al., 2021).

The prevalence of vision problems are four times higher in low and middle income countries than high income countries (World Health Organisation, 2019). Reports indicate that Ethiopia has very high rates of blindness and low vision in sub-Saharan Africa and globally (Cherinet et al., 2018; Flaxman et al., 2017). In Ethiopia, cataracts, glaucoma, and age-related macular degeneration are the primary factors leading to reduced vision and blindness. Cataracts account for 49.9% of blindness cases, glaucoma (5.2%), and age-related macular degeneration 4.8% (Berhane et al., 2006). Cataract (42.3%) and macular degeneration (4.6%) are also the leading causes of low vision. In addition, with the increasing prevalence of diabetes in Ethiopia, diabetic retinopathy is becoming more common, with a reported prevalence of 19.48% (Mersha et al., 2022).

There are various factors that contribute to an increased risk of retinal disease development over time. These risk factors include family history and genetics, cardiovascular disease, obesity, smoking, diabetes, aging process, and more (Akil et al., 2021). The burden of eye disease is a major issue that affects individuals, society, and the country as a whole negatively impacting both the economy and the society at large. As indicated in reports by WHO (*Blindness and Vision Impairment*, n.d.), the economic impact of reduced productivity due to vision impairment reaches an annual cost of 411 billion US dollars on a global scale. This financial burden is a

result of the reduced productivity and economic contributions of individuals with vision impairment, as well as the costs associated with medical treatment, assistive devices, and other support services required to manage the condition.

According to the International Council of Ophthalmology(ICO)[1],there are few ophthalmologists in Ethiopia. With the population of Ethiopia approaching 120,000,000, these professionals are very limited in number despite having one of the highest rates of blindness globally. This poses a serious health challenge in our country. Application of deep learning techniques in ophthalmology has shown promising results in automated retinal image analysis by assisting healthcare professionals in early detection and prompt treatment (Rahimy, 2018). Convolutional neural networks (CNNs) have not only gained significant attention but also dominated various image classification tasks, including retinal disease classification with their remarkable success. Several studies (Guo et al., 2021; Khan et al., 2022) have utilized CNNs to analyze retinal images and have shown promising results in accurate disease classification. However, CNNs rely on the local convolution features and have limitations when capturing contextual information in images. This can impact their performance in tasks where global patterns and visual feature relationships are crucial for accurate diagnoses, such as the classification of complex retinal diseases.

Recently, vision transformers have emerged as a promising alternative in the field of computer vision. Unlike CNNs, vision transformers utilize self-attention mechanisms to capture global context and relationships in images (Dosovitskiy et al., 2020). By modeling interactions between different image regions, vision transformers can effectively capture long-range dependencies and enhance their understanding of complex visual patterns. With the potential to alleviate and reduce vision loss and blindness, this study is aimed at developing a transformer model for the classification of common eye diseases from fundus Images. By using Vision Transformer (ViT), we developed a model with different data augmentation techniques to accurately and efficiently classify retinal diseases.

[1] https://icoph.org/advocacy/data-on-ophthalmologists-worldwide/

1.2. Motivation of the Study

The prevalence of eye diseases has been on the rise globally, posing significant challenges to public health. Ethiopia, having one of the highest prevalence rates of eye diseases globally, developing a deep learning technique for fundus image classification can potentially improve patient outcomes and assist ophthalmologists in the early treatment of eye diseases. With the emergence of deep learning methods, state-of-the-art performance has been achieved for binary classification of retinal fundus images, especially in differentiating between healthy and unhealthy retinal fundus images (Junayed et al., 2021). Combining of multiple common eye diseases in a single research project can provide a more comprehensive understanding of how different diseases can affect the eye and the interplay between them. This approach can help to develop better diagnostic and treatment strategies that consider the co-occurrence of different eye diseases.

The other motivation for this research is to explore the potential of vision transformers as an alternative approach to address lack of global context limitations of CNN models in retinal disease classification. Transformers have made significant advancements in the realm of NLP tasks, reaching state-of the- art results surpassing previous benchmarks. These models have also been adapted for use in computer vision field through the development of vision transformers. When trained on large datasets, vision transformer can exceed the state-of-the-art convolutional neural network models while using about four times fewer computing resources (Dosovitskiy et al., 2020). due to their ability to employ self-attention to attend to different patches in the image, for extracting local and global features that are relevant for classification, my research plan focuses on utilizing the vision transformer model for the multi-class classification of common eye diseases.

1.3. Statement of the problem

The leading causes of blindness and vision loss are glaucoma, diabetic retinopathy, cataract, and age-related macular degeneration, which account for around 80% of all impairment (*Blindness and Vision Impairment*, n.d.; Bourne et al., 2021). Ethiopia, having a high rate of blindness and low vision, there is a significant imbalance between the number of ophthalmologists and the population that needs eye care services. This scarcity of ophthalmologists has contributed to the high prevalence rate in the country (Hinkle et al., 2020).

Fundus imaging is a cost-effective diagnostic tool used by ophthalmologists for detecting and monitoring different eye diseases. However, manual analysis of fundus images is time-consuming, and prone to variability among different ophthalmologist, which can lead to misdiagnosis. In addition, the diagnostic procedure becomes costly as it requires specialized medical experts in identifying this ocular conditions. A deep learning method could assist ophthalmologists in distinguishing between different retinal pathologies and thus increase the accuracy of classifying fundus images.

CNNs have become very popular for classifying images in various tasks. The effectiveness of CNNs for image classification tasks has resulted in their broad adoption and success in computer vision. Most existing works in retinal disease classification utilized CNN models. However, classification is often limited to binary classification between normal and abnormal retina. There is also lack of comparative assessment between different deep learning approaches for multi-class retinal disease classification. CNNs are limited in capturing global context in images due to their local receptive fields. Other deep learning architectures such as attention-based models have recently gained attention owing to their superior performance over CNNs in certain tasks.

Recent works has explored integrating convolutional neural networks (CNNs) with self-attention mechanisms. Some studies have proposed replacing all convolution layers in CNNs with self-attention (Ramachandran et al., 2019), Others like (Bello et al., 2019) suggested replacing only certain convolution layers with attention for improving the performance of image classification models. However, these approaches had high computational costs. The large size of images increased the complexity of self-attention, making it computationally expensive.

Researchers have made efforts to apply transformers in images. (Dosovitskiy et al., 2020) proposed vision transformer architecture for computer vision task and achieved state-of-the-art results on image classification benchmarks. The self-attention mechanism in vision transformers allows them model global context and integrate information across all image patches, making them well suited for classification problems. Given the importance of automating fundus image and the success of self-attention in transformer models, this study develops a vision transformer model to classify common retinal diseases from fundus images.

1.4. Research Questions

The research explored and tried to address the following research questions.

RQ1**:** What is an efficient way of improving performance of the classification of most common eye disease from fundus images?

RQ2**:** To what extent can vision transformer accurately classify different types of eye diseases?

RQ3**:** How does the modified vision transformer model compare to existing vision transformer and CNN models in terms of classification accuracy and other performance metrics?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to develop and evaluate a vision transformer based deep learning model for multi-class classification of retinal diseases from fundus images.

1.5.2. Specific Objectives

To accomplish the general objective, this study addressed the following specific objectives:

- To review prior literature on retinal disease classification using deep learning techniques.
- To collect fundus image data.
- To perform pre-processing and implement data augmentation techniques to enhance model performance.
- To develop a customized vision transformer model architecture that incorporates a CNN block tailored for retinal images.
- To train and evaluate the proposed vision transformer model on the prepared dataset using appropriate evaluation metrics.
- To conduct comparative assessment of proposed model performance with CNN models.

1.6. Scope and Limitations of the Study

1.6.1. Scope of the Study

The scope of this research study focuses on the application of transformer model, specifically vision transformer model (ViT), for classification of common retinal diseases from fundus images. The study involves investigation of standard vision transformer architectures as well as proposing a customized model tailored for retinal images through additional convolutional and classifier layers. The study covers data collection, preprocessing, augmentation and model building on the input fundus image. Comparative assessment against CNN classifiers is also conducted. Furthermore, the scope includes exploring data augmentation techniques like transformational and synthetic augmentations to generate additional training data and also enhance model robustness.

1.6.2. Limitations of the Study

The study is limited to analysis of fundus imaging and classification task of most common retinal disease, including diabetic retinopathy, glaucoma, age-related macular degeneration and cataracts. It does not include incorporation of other retinal imaging modalities like optical coherence tomography (OCT). Despite the larger number of local patients with various retinal diseases, the local dataset used in this study is limited in size due to the scarcity of labeled fundus image data and inconsistencies. Testing model generalization across multiple datasets was also limited due to time constraints.

1.7. Significance of the Study

Undoubtedly better diagnosis brings better medical care**.** Early screening and detection of the diseases through the fundus technology can help to reduce the risk of vision loss and blindness. Automatic fundus image classification can help in the early detection of eye diseases, which can lead to better treatment outcomes and prevent vision loss.

Due to the lengthy examination process, lack of symptoms in early disease stages, and limited access to retinal specialists, significant portion of people with retinal diseases do not receive the recommended annual eye examination (Chou et al., 2014; Maclennan et al., 2014). The use of artificial intelligence techniques for identifying retinal disorders is one way to get around these obstacles. Minimizing false positive result is one of the crucial aspects in computer- aided

systems. This research work aims to investigate the potential of using vision transformer based deep learning model on retinal fundus images for classifying eye diseases.

In general this study is important for the following reasons:

- The study can help to improve the accuracy of retinal disease classification. This is important because accurate diagnosis is essential for early intervention and treatment.
- The study enables ophthalmologists to easily classify eye diseases from retinal fundus images, which can speed up detection and diagnosis process hence reduce the time and effort of patients and ophthalmologists.
- Improving the effectiveness of disease controlling mechanisms. This is significant because it can help to prevent the progression of eye diseases.

1.8. Organization of the Study

The study is organized into seven chapters and they are outlined as follows:

Chapter One – Introduction**:** The first chapter serve as an introduction to the research to be conducted. It covers the background, problem statement, research questions, objectives, scope, and limitations of the study.

Chapter Two – Literature Review and Related Works**:** The second chapter presents relevant literature and related works on retinal diseases. Augmentation techniques, and theoretical framework of deep learning models are discussed.

Chapter Three – Research Methodology**:** methodology employed in the study is discussed. It includes the data used in this work, data preparation, design tools, framework and evaluation methods used in the study.

Chapter Four – Proposed Solution**:** Chapter Four presents a detailed explanation of the proposed custom vision transformer model. It includes the proposed solution process flow, proposed model architecture, and model training pseudocode.

Chapter Five – Implementation**:** Chapter Five covers the experimental setup and implementation of. It includes the environment setup required for the implementation, code snippets of the proposed solution, and the experimental classes.

Chapter Six – Results and Discussion**:** Presents the results and discussion of the experiments conducted to answer the research question and achieve research objectives. Analysis of results with their descriptions, figures, graphs and tables to present the outcomes obtained from the experiments are presented.

Chapter Seven – Conclusion and Future Works**:** This chapter provides summaries of the study and the findings. It identifies and suggests potential next steps to advance the research.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In this chapter, we performed a review of literatures on classification of retinal diseases. It covers background information on retinal diseases and retinal fundus images. The chapter presents a review of related works that have focused on developing deep learning models for feature extraction, lesion detection, disease grading and classification using retinal photographs. Additionally, the chapter discusses different deep learning techniques used in medical Image recognition.

2.2. Common Eye Diseases

2.2.1. Cataract

Cataracts are an ocular condition affecting the lens of the eye[2]. In severe case cataract can result in permanent loss of vision (Askarian et al., 2021). Signs of cataract include, yellowish tint or graying of the lens, obscured macular area and a hazy appearance of the optic nerve. At first, the cataract will have a small size, but it will gradually grow over time resulting in vision loss or blindness. The only method for addressing a visually significant cataract involves implantation of new intraocular lens by removing the lens through surgery. Early screening is the best way to mitigate disease progression and reduce the risk of painful expensive surgeries and hence prevent vision loss.

2.2.2. Glaucoma

Glaucoma is an ocular condition that damages to the optic nerve head[3], which is responsible to transmit visual information from our eye to brain. High pressure within the eye or reduced blood supply to the optic nerve is a frequent causative factor in this nerve damage. Optic disc cupping, thinning of the neuroretinal rim, and abnormalities in the retinal nerve fiber layer are the signs of glaucoma seen in fundus images.

[2] https://www.ncbi.nlm.nih.gov/books/NBK539699/
[3] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

2.2.3. Age Related Macular Degeneration

Age-related macular degeneration (AMD) is an ocular disease caused as a result of damage to macula, the small central portion of the retina responsible for sharp central vision, (Lim et al., 2012). The formation of drusen, which are small yellow deposits made up of fatty protein under the retina, is an early indicator of AMD. Other signs include pigmentary changes, geographic atrophy, and choroidal neovascularization. Early stages of AMD are typically asymptomatic and do not cause vision loss, however the disease progression can lead to dark patches, shadows, or loss of the central vision[4].

2.2.4. Diabetic Retinopathy

Diabetic retinopathy (DR) is an ocular disease caused by the damages in the blood vessels of retina. as a result of high blood sugar level in the blood. DR can develop in any patient with diabetes, regardless of its severity (Janghorbani et al., 2009). The prevalence of diabetic retinopathy is expected to increase along with increasing rate of diabetes worldwide, largely driven by socio-demographic and economic changes (Mersha et al., 2022). Exudates, microaneurysms, and hemorrhages are lesions in a fundus image that are indication of abnormalities associated with diabetic retinopathy.

2.3. Retinal Fundus Image

The term “fundus” refers to the innermost part of the eyeball opposite to the pupil, where the macula, optic disc, retina, and blood vessels are located. A fundus camera is a device used to capture images of back portion of the interior of the eyeball. Fundus images obtained from fundus cameras provide detailed visualization of anatomical structures and pathology in the posterior of the eye[5]. These non-invasive images allow assessment of features important for screening, diagnosis, and monitoring of retinal diseases. Ophthalmologists commonly employ fundus cameras to detect and monitor eye diseases including cataracts, glaucoma, age-related macular degeneration, and diabetic retinopathy (Badar et al., 2020). The progression of retinal diseases are examined and monitored by observing abnormalities on different parts of the retina structure like the optic disc, fovea, blood vessels, exudates and microaneurysms.

[4] https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/age-related macular degeneration

[5] https://www.ncbi.nlm.nih.gov/books/NBK585111/funduscamera

In contrast to the expensive OCT imaging technology, fundus photography is a cost-effective means of screening and detecting eye conditions.

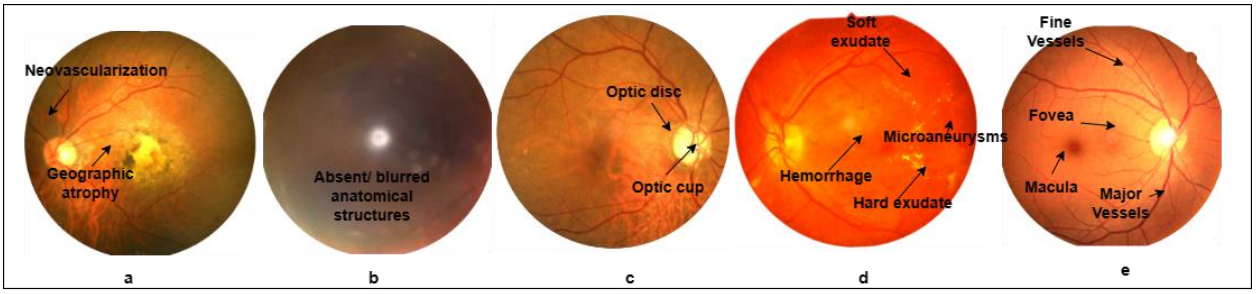


Figure 2.1: Fundus images showing retinal structures and ocular diseases:

(a) AMD, (b) cataract, (c) glaucoma, (d) DR, and (e) normal retina.

2.4. Data Augmentation Techniques

Augmentation techniques have now become indispensable for forging robust deep learning models, especially in domains like medical imaging where dataset constraints prevail. By artificially expanding the training dataset through various transformations and generation methods, data augmentation aims to introduce diversity and that closely resembles real-world scenarios. This exposes the model to altered iterations of the original images, thereby mitigating model overfitting and enhancing generalizability. Such techniques bestow immense value upon retinal imaging analysis tasks, where image datasets are constrained in both size and variability. Recent research has delved into the ways in which traditional augmentation methods, as well as more advanced synthetic techniques, can enhance model performance for retinal disease classification.

2.4.1. Transformation Augmentation

Transformation augmentation involves modifying retinal images through different geometric and photometric transformations to artificially expand and diversify the training dataset. Geometric augmentation modifies the geometric aspects of the image, such as its spatial arrangement, orientation, and position of pixels without changing pixel values whereas photometric augmentation modifies photometric properties of the image, such as color, brightness, saturation and noise addition (Shorten & Khoshgoftaar, 2019). (Rodriguez et al., 2022) used both transformational augmentation technique and found increase in robustness of

the model and scored state of the art result in multi label retinal disease classification. (Gangwar & Ravi, 2021) employed basic augmentation techniques like horizontal/vertical flips, rotations, zoom, width/height shifts on a small dataset yield improved accuracy and also prevented overfitting of data.

2.4.2. Synthetic Generation

Instead of transforming the input images directly, this method involves adjusting a generative model to produce new, synthetic data that mimics the original data. Recently Advanced generation techniques have been utilized using generative adversarial networks (GANs) to artificially generate new medical images.

GANs consist of two primary components, generator and discriminator. The generator network generates new synthetic images by taking in random noise as input, while the discriminator network act as a classifier to judge if the generated sample is real or fake (Creswell et al., 2018). This approach allows creating realistic new data from scratch rather than just modifying original images. Studies have shown GANs can be conditioned to generate retinal images for diabetic retinopathy lesions like hemorrhages, microaneurysms, and vessels (Devi & Kumar, 2022). Synthetic generation using (GANs) can improve model generalizability and robustness for retinal disease classification tasks.

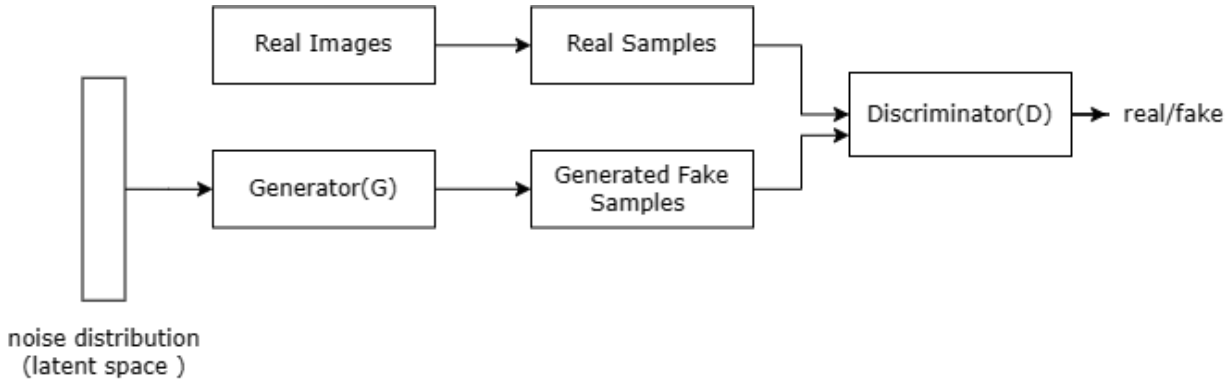


Figure 2.2: General Architecture of GAN[6]

The examination of augmentation techniques is relevant to this work on retinal image classification. Given limited datasets, augmentation is critical for training robust models. This

[6] https://www.slideshare.net/Artifacia/generative-adversarial-networks-and-their-applications

research will evaluate impacts of transformation techniques as well as synthetic method when training deep learning models. This will provide insights into which techniques are most effective for improving model generalization capability and robustness for retinal disease classification.

2.5. Machine Learning in Retinal Image Analysis

Prior to the development of deep learning methods, the majority of computer-aided diagnosis (CAD) systems for retinal image analysis relied on machine learning techniques. Hand-crafted features were manually engineered to analyze retinal images. vessel density and average vessel thickness features are manually extracted to analyze the optic nerve for classifying dry and wet age-related macular degeneration (Priya & Aruna, 2014). Optic disc measurements including diameter, cup-to-disc ratio and rim thinning are manual feature extractions used to assess glaucomatous damage. These features extracted from retinal images were then fed into machine learning classification models like support vector machine, naive bayes, logistic regression and k-nearest neighbor algorithms for predicting retinal diseases (Morales et al., 2017; Tang et al., 2013). However, extensive feature engineering was required and performance was limited compared to human experts. Deep learning methods now enable automatic hierarchical feature learning directly from retinal images, leading to improved retinal disease analysis.

2.6. Deep Learning in Retinal Image Analysis

Deep learning leverages deep neural networks to learn feature representations from an image, achieving human-level or superior performance on complex tasks. Deep learning techniques advanced medical image analysis for disease detection, segmentation, and classification (Biswas et al., 2019; Suganyadevi et al., 2022). Due to their effectiveness, current approaches for automated fundus image analysis leverage deep learning models as their core methodology.

Deep learning models have multiple hidden layers between the input and output layers. Each hidden layer consists of a group of neurons that are responsible to learn to distinct sets of features based on the outputs from previous layers. These multiple hidden layers allows them to learn hierarchical representations of data, where lower layers learn simpler features, and higher layers learn more complex features. This makes deep learning models capable of learning more abstract representations of data, resulting in higher accuracy and better performance on various

tasks. The basic structure of a deep neural network consists of an input layer, multiple hidden layers in the middle, and a final output layer (Figure 2.3).

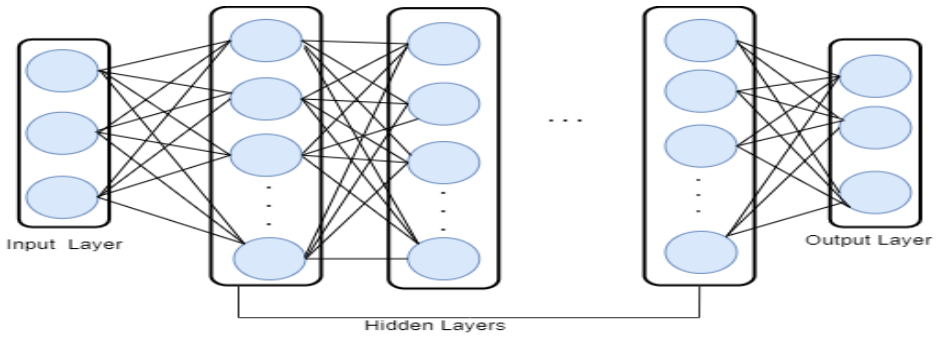


Figure 2.3: Basic Deep Learning Architecture

2.6.1. Convolutional Neural Network

CNNs revolutionized retinal image analysis by learning hierarchical feature representations directly from the pixels. A key aspect of CNN is their use of convolutional layers which employ convolution operations on the input image to extract spatial feature maps. Convolutional neural networks (CNNs) were developed based on insights into how the visual cortex in animal brains, especially cats, processes visual information (Hubel & Wiesel, 1962).

The basic building block of CNN is convolutional layer which consist of three operations namely: Convolution, ReLu, and Pooling. Convolution operation extracts local features by sliding a small filter over the input image they compute the dot product between the filter and the corresponding image pixels at each position.

Convolution operation can be expressed mathematically as:

$$f(x,y) = I(x,y) * W(x,y) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} w[u,v] I[x-u, y-v] \qquad eq.(2.1)$$

Convolution operation is performed between input image I and filter W to extract hierarchical feature representation f as an output.

The Rectified Linear Unit (ReLU) is applied on the features maps generated by convolutional layers. It provides nonlinear features of the decision function by performing element-wise operations, replacing all negative pixel values with zero while keeping the positive values

unchanged. This helps models learn complex patterns in data and makes the optimization process more efficient.

Pooling operations summarize the features and provide translation invariance by down sampling feature maps spatially across the input. This dimensionality reduction simplifies the representations learned by CNN, reducing computational requirements and overfitting. The general architecture of CNN is given in Figure 2.3.

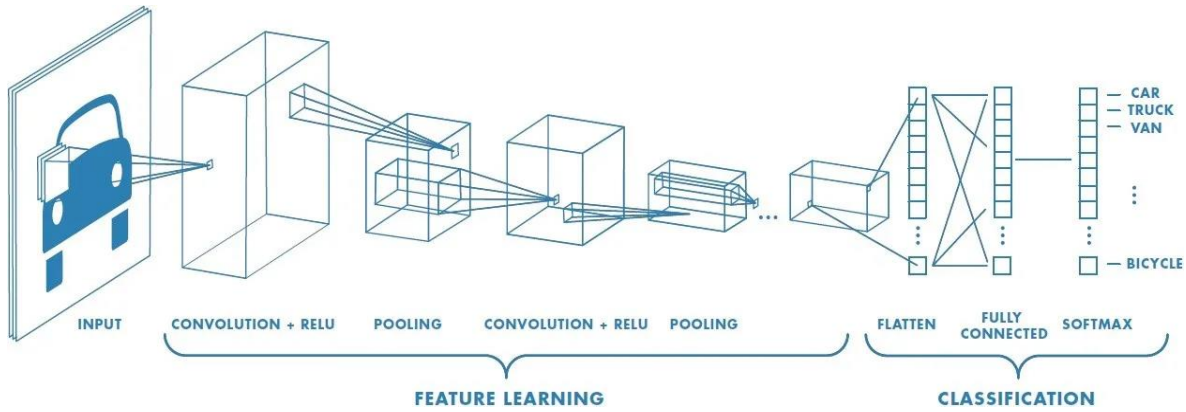


Figure 2.4: Architecture of CNN[7]

Fully connected layers are the final layers in a convolutional neural network (CNN). They are also known as dense layers or fully connected layers because these layers take the high-level features learned by the CNN and connect every neuron from the previous layer to every neuron in the next layer. These features are then used for classification or regression tasks.

The preceding layers in a CNN, are responsible for learning spatial hierarchies of features present in the input data. Whereas the fully connected layers are responsible for learning high-level features by combining the learned spatial features from the previous layers. CNNs have become the predominant approach for analyzing and processing fundus images. This involves utilizing them as either feature extractor or combined with a fully connected layer for the purpose of classification (Gulshan et al., 2016; Pratt et al., 2016).

[7] https://medium.com/analytics-vidhya/an-overview-of-convolutional-neural-network-cnn-a6a3d67ce543

2.6.2. Transformer Neural Network

A Transformer is a novel deep learning model architecture first introduced in 2017 introduced in the paper “Attention is all you need” by (Vaswani et al., 2017). It is an architecture for sequence-to-sequence modeling task, where the goal is to transform an input sequence into an output sequence.

Transformer model is unique that it does not use RNN or CNNs convolution operation for sequence modelling, instead it uses self-attention mechanism that allows it to focus on relevant parts of the input when modeling output sequence. This attention modeling approach has achieved state-of-the-art results in various natural language processing applications (Devlin et al., 2019).

2.7. Attention Mechanism

CNN has become the foundation for various computer vision applications since the advent of bigger datasets and improved computing resources. The focus of deep learning has moved to the development of CNN structures in order to enhance their performance on object detection, image segmentation and classification tasks. CNNs have a strong potential for capturing local features using convolution operation, but it is relatively weak in global feature representation due to their poor scaling properties for larger receptive fields (Bello et al., 2019; Ramachandran et al., 2019). The problem with CNNs have been addressed in sequence modeling by using attention mechanism.

Attention mechanism allows a neural network model to focus on specific parts of the input that are most relevant while performing computations (Soydaner, 2022). Transformer models use attention to assess the significance of each element in a series of inputs, allowing the model to concentrate on important information.

In Transformer models, Attention is computed using a scale dot-product attention operation. These operations involves converting the inputs in the form of query, key, and value to compute the attention weights (de Santana Correia & Colombini, 2022). First a dot product between each query vector and each key vector is computed to obtain attention weight score. This score signifies how well the query vector matches with the key vector. To normalize the score, they are divided by the square root of the key vector dimension. The normalized scores are softmaxed

to generate attention weights that sum to 1. A weighted sum is then computed using the attention weights and value vectors to concentrate on relevant values.

Scaled Dot-Product Attention is given as:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{dk}})V \qquad eq.(2.2)$$

Where, Queries (Q), keys (K) and values (V) are the three matrices. Q and K have a dimension of d_k , V have a dimension of d_v . To prevent dot products from reaching large magnitudes when d_k is large, the product is scaled by $\frac{1}{\sqrt{d_k}}$.

2.7.1. Vision Transformer

Vision transformer (ViT) is a deep learning model that is based on transformer architecture proposed for Computer Vision applications. ViT models apply the Transformer encoder to sequences of image patches, similarly to operating on word sequences. This allows attention-based modeling of global dependencies in images, just as in text.

ViT models achieved state-of-the-art results on image recognition benchmarks, outperforming CNNs given sufficient training data. They can match or exceed CNN performance with substantially fewer computational resources and parameters. The potential benefits of ViT make it appealing to explore for medical image analysis applications like retinal image classification. Applying ViT to such tasks could similarly improve efficiency and performance over CNN-based approaches. The self-attention mechanism of Transformers may allow ViT models to effectively learn from few labeled medical images.

In vision transformer an input image is divided in to small non overlapping patches, also known as tokens similar to words in a sentence. These image patch tokens are flattened and projected to an embedding space via linear transformation. A classifier token provides the model with information about class labels, whereas positional embedding holds the positional information of each patches. The sequence of patch embedding tokens along with the class token and positional embeddings makes a transformer encoder architecture. Multiple encoder layers process the sequence of patch embeddings allowing global communication between patches through self-attention layers. After the final Transformer encoder block, the output is passed to a multi-layer perceptron (MLP) to predict the image class.

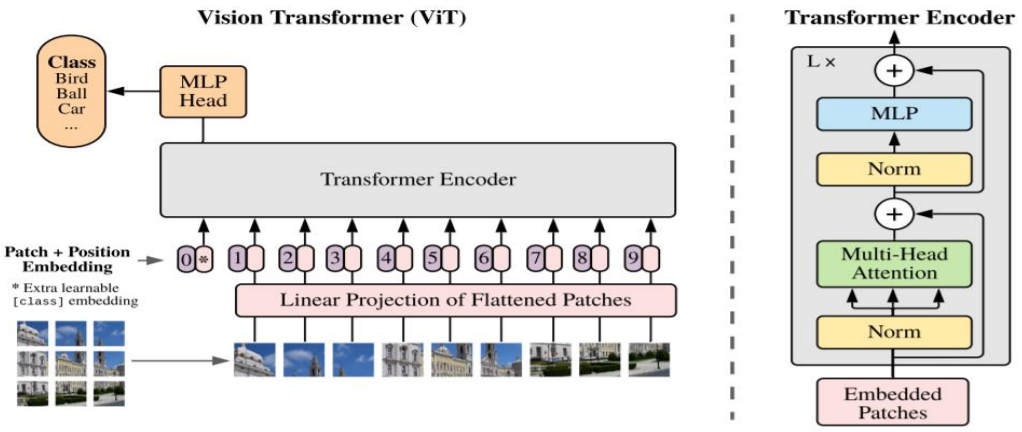


Figure 2.5: Architecture of Vision Transformer

Source: cited from (Dosovitskiy et al., 2020)

The key components of the Vision Transformer (ViT) architecture are:

Patch extraction – splits the input image into small non-overlapping patches which act as tokens. The original 2D input image of height H and width W with dimensions of $H \times W \times C$ are converted into sequence of N flattened 2D non overlapping patches with the size of $N \times (\mathrm{P}, \mathrm{P}, \mathrm{C})$,where (P, P) is resolution of each extracted patches and the total number of patches $N = \frac{H \times W}{P^2}$.

A learnable class token X_{class} is added before the embedded image patches. This class token represents the target class and guides the model to make accurate predictions.

Position embedding – responsible for encoding of the spatial position of each patch to retain positional information. They are added to patch embeddings.

$$Z0 = \left[Xclass; X_p^1; \ldots; X_p^N E\right] + Epos \quad E \in \mathbb{R}^{(P^2.C)\,x\,D} Dpos^{(N+1)\,x\,D} \qquad eq.(2.3)$$

The above equation represents an input to transformer encoder block, where X_{class} is class embedding, $X_p^1; \ldots; X_p^N E$ are patch embeddings and E_{pos} is a positional embedding.

Multi-head self-attention (MHSA) - is the main component of ViT encoder. The input patch embeddings are processed by multiple stacked self-attention layers to capture global and local

dependencies within the image MHSA allows global communication between image patches, processing them in parallel with several attention heads. MHSA is calculated as:

$$MultiHead(Q, K, V) = Concatenate(head_1, \ldots, head_h)\, W \qquad eq.(2.4)$$

Where $head_i = Attention(QW_i^Q, KW_i^k, VW_i^v)$, W_i is distinct weight matrices of the h-available attention head.

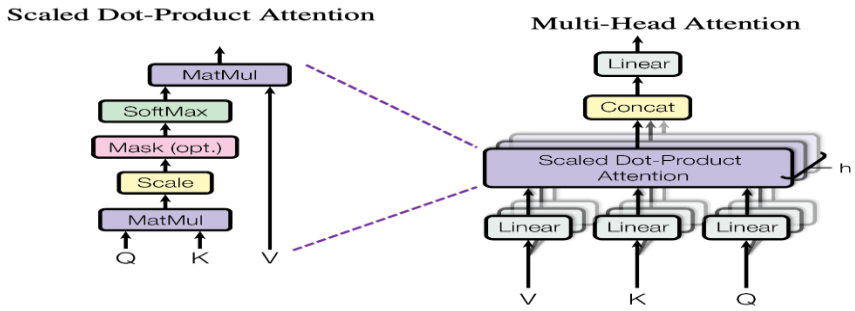


Figure 2.6: Attention and Multi Head Attention

Source: cited from (Vaswani et al., 2017)

Layer normalization (LN) - normalize the outputs of each attention layer in the transformer encoder.

Classification head (MLP) - is a linear layer that maps the output of the transformer encoder to a final classification score.

2.8. Transfer Learning

Transfer learning refers to the process of improving the training process by reusing a pretrained model as a base for a different task. It allows a model trained on one task to be adapted to a related but different task by leveraging the knowledge gained from the source task. When there is a scarcity of data resources, utilizing pre-trained deep neural network models is a commonly employed technique in computer vision tasks (Ghosh et al., 2018). This reduces the training data needed and computational resources required for training process. Transfer learning is especially useful in medical imaging, where available training data is often scarce. For retinal image analysis, utilizing pre-trained models and fine-tuning them on retinal datasets can serve as a strong baseline and also reduce computational requirements compared to training from scratch.

Transfer learning can be defined as:

For a Given a learning task Tt *that is based on* Dt *Transfer learning is concerned with enhancing the performance of a predictive function* $\mathcal{FT}(\cdot)$*.This is achieved by leveraging* knowledge from Ds *for the learning task* Ts*,Where* $Ds \neq Dt$ *and/or* $Ts \neq Tt$*. Typically,* Ds *is much larger than* Dt *in size, with* $Ns \gg Nt$ (C. Tan et al., 2018).

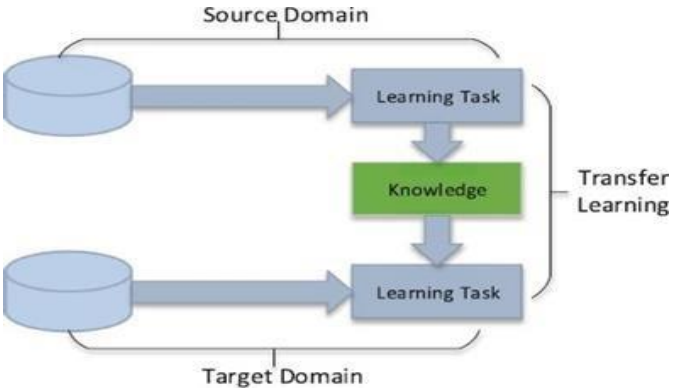


Figure 2.7: Transfer Learning Process

Source: cited from (C. Tan et al., 2018))

2.8.1. Deep CNN Architectures

Deep CNNs have dominated recent advancements in automated retinal disease diagnosis. Models pretrained on large-scale natural image datasets like ImageNet are fine-tuned on fundus images to effectively learn feature representations tailored for retinal abnormalities.

A. Visual Geometry Group (VGG)

VGG(Simonyan & Zisserman, 2015) is a deep CNN architecture that contains over 138 million parameters. It improved AlexNet performance achieving 7.4% error rate on the ImageNet dataset. VGG architecture is structured as a stack of convolutional layers, each followed by a max pooling layer. Each convolutional layers use small 3x3 filters. With 16 convolutional and 3 fully connected layers, VGG19 contains a total of 19 layers. The main advantage of this network lies in its sequential blocks, which consist of sequential convolutional layers that are inserted one after the other that decreases the amount of spatial information required.

B. Inception V3

Inception V3 is a deep CNN that was introduced in 2015 by researchers from Google. It is an enhanced version of original inception architecture (Szegedy et al., 2014). This 48-layer CNN employs several optimizations to reduce parameters and accelerate computation. The architecture is characterized by the use of inception modules. These modules consist of parallel convolutional layers of varying receptive field sizes (1x1, 3x3, 5x5) alongside pooling layers to capture both fine-grained and larger context features simultaneously. The main advantage of this model is gaining better accuracy by a modest increase of compute requirements compared to wide and shallow networks (Szegedy et al., 2015).

C. Residual Network (ResNet)

ResNet (Residual Network) is a deep CNN introduced by (He et al., 2016) to address the vanishing gradient issues in very deep neural networks. ResNet introduced skip connections to construct residual blocks where the input can take a shortcut to the output by being added directly after the convolutional layers. This residual structure allows gradients to flow directly through these shortcut connections, helping with the vanishing gradient problem. The key innovation of ResNet is the introduction of skip connections and residual blocks to enable training extremely deep CNNs. ResNet-50 is a specific variant of the ResNet architecture that consist of 50 layers and has won an ImageNet challenge in 2015 with a top-five accuracy rate of 94.29%.

Table 2.1: Deep CNN Architectures

Model	*Training params*	*No. of layers*
VGG19	143,667,240	19
InceptionV3	23,864,616	48
ResNet50	23,514,704	50

2.8.2. Pretrained Vision Transformer model

The ViT model provides an effective pretrained feature extractors for transfer learning in medical image analysis. They are pre-trained on ImageNet-21K, a large dataset of over 14 million images across 21,000 classes at 224x224 resolution. There are three variants of Vision Transformer (ViT) models depending on depth of the architecture: ViT Base, ViT Large, and

ViT Huge. In addition, ViT models come with different patch sizes of 16x16 or 32x32 pixels. The patch size relates to how the input image is split into embedded patches. Table2.2 shows ViT model variants from the paper (Dosovitskiy et al., 2020).

Table 2.2: Vision Transformer Model Variants

Model	*Encoder Layers*	*Hidden Size D*	*MLP size*	*Attention Heads*	*T.Params*
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

2.9. Related Works

Over the years, a numerous studies have been conducted on the detection and classification of retinal fundus images. Different researchers employed different methods, architectures, algorithms, and techniques to address the problem. This section presents and discusses previous works done on fundus image classification and detection.

The paper by (J. H. Tan et al., 2018) implemented a 14-layer Deep Convolutional Neural Network model for the detection and classification of age-related macular degeneration (AMD). The authors trained their model on a dataset of 402 normal and 708 AMD retinal fundus images collected from hospitals in India and achieved an average accuracy of 95.45% with ten-fold cross-validation. The proposed deep learning approach enables automated analysis of retinal images for detecting AMD without any manual feature engineering.

The paper by (Shinde, 2021) presented an automated system for glaucoma detection using a hybrid deep learning and machine learning techniques. Their approach involves segmentation of the optic disk and cup, manual feature extraction of cup-to-disc ratio and final classification. Evaluation on 6 datasets (666 images) shows 99% input validation accuracy, 98.67% ROI accuracy, 0.93 dice coefficient for optic disc segmentation, 0.87 for optic cup segmentation, and 100% classification accuracy, sensitivity and specificity. Testing on additional datasets confirms robustness. However, their method relies on manual feature extraction focused only on the optic-cup related features. Developing automatic feature extraction techniques and extracting more features by incorporating several retinal lesions, and extend the approach to classify other retinal diseases could help ophthalmologists as a diagnostic tool.

(Wassel et al., 2022) developed vision transformer models for automated classification of glaucoma from retinal fundus images, motivated by the potential of transformers to capture relevant global features. The authors combined multiple public fundus image datasets totaling over 9,000 images to train and evaluate various transformer architectures including ViT, DeiT, BEiT, CaiT, ResMLP, XCiT, CrossViT and Swin Transformer. Through comprehensive experiments with standalone models and ensembles, they found Swin Transformer achieved the best performance with 92.57% sensitivity and 97.9% AUC. The results demonstrate transformers are effective for glaucoma classification from fundus images likely due to transformers' ability to learn globally connected features. Key limitations is that their work is limited to binary glaucoma classification and also lack of model explainability.

(Bernabe et al., 2021) developed a 2-layer Convolutional Neural Network for classifying retinal images as either glaucoma or diabetic retinopathy. They trained and tested their model on small dataset of 168 glaucoma and 397 diabetic retinopathy images. The model attained a high classification accuracy of 99.89%, along with recall, specificity, precision and F1 scores. Limitations are the small dataset size and inclusion of only two disease classes. Increasing dataset size and more complex classifier models to capture complex retinal diseases features would be beneficial.

The paper by (Gangwar & Ravi, 2021) proposed a hybrid Inception-ResNet-v2 model to classify the severity level of diabetic retinopathy. The pretrained Inception-ResNet-v2 extracts high-level features and then fed to the custom convolutional blocks for final classification into 4 disease grades. They evaluated with 1200 images from Messidor-1 dataset and achieved an accuracy of 72.33%. The hybrid architecture provides improved performance by leveraging Inception-ResNet's multi-scale feature processing and custom block. Their proposed model demonstrates the effectiveness of hybrid models for accurate and efficient diabetic retinopathy detection.

(Tolstikhin et al., 2021) evaluated convolutional neural networks (CNNs), transformers, and multilayer perceptrons (MLPs) models for multi-class diabetic retinopathy (DR) detection using retinal fundus images. Models including EfficientNet, ResNet, Vision Transformer models (ViT, Swin Transformer) and MLP are trained on the diabetic retinopathy dataset without augmentation. The results demonstrate that transformer-based models, especially Swin-

transformer, achieved the highest accuracy of 86.4% and fast convergence comparable to CNNs. ViT also produces good accuracy of 83.8% with low training time. In contrast, CNN-based EfficientNet converges quickly but has much lower accuracy, while MLP performs worst with 78.5% accuracy and slower convergence. The findings indicate attention mechanisms in transformers benefited DR detection. The authors reported, transformer architecture is superior over CNN and MLP models for accurate and efficient diabetic retinopathy detection from fundus images.

(Raza et al., 2021) presented a deep learning framework for classifying retinal diseases into four categories - cataract, glaucoma, other retinal diseases and normal retina. Inception-V4 is trained on a 602 fundus images and achieved 96% accuracy. The authors highlighted the importance of retinal imaging in diagnosing various eye disorders and the challenges posed by the scarcity of ophthalmologists. Limitations include a small dataset and only four disease classes. Expanding the data diversity and classes would further validate the approach.

The paper by (Das et al., 2019) proposed a retinal disease classification system using transfer learning approach. The pretrained VGG19 model is leveraged as the base network and fine-tuned on a small dataset of healthy, glaucoma, and diabetic retinopathy retinal fundus images. Data augmentation by rotation, flipping, cropping is used to expand the limited dataset. The fully connected layers of VGG19 are retrained while retaining the feature extraction capabilities of the convolutional layers. Experiments with different learning rates and epochs show accuracy over 90% is achieved with 150 epochs and 0.001 learning rate. The paper highlighted, transfer learning approach provides an accurate and efficient solution for multi-class retinal disease classification with limited computational requirements.

The paper (Khan et al., 2022) proposed a transfer learning based deep CNN model using VGG19 for multi-class retinal disease classification from fundus images. To address the class imbalance, they transformed into multiple binary classification problem between normal and each disease type by balancing the retinal classes. This approach achieved high accuracy of 98.1% for binary classification. The work demonstrated how balancing datasets and transfer learning can effectively improve retinal disease classification performance.

(Sarki et al., 2022) proposed a convolutional neural network (CNN) architecture for multi-class classification of retinal fundus images into healthy, diabetic retinopathy, glaucoma, cataract and

macular edema, and they have achieved an accuracy of 81.33%. The paper utilized mathematical morphology-based image enhancement, and customized CNN architecture. Grad-CAM visualizations provided interpretation of pathological features detected. The integrated deep learning approach enables accurate multi-class classification and interpretation of diabetic eye diseases.

(Guo et al., 2021) proposed a MobileNetV2 transfer learning model for multi-class classification of four eye diseases along with normal fundus images. Transfer learning from ImageNet pretraining enables accurate classification with a small dataset of just 250 images. The model achieves 96.2% accuracy, 90.4% sensitivity and 97.6% specificity on the test set, outperforming InceptionV3 and AlexNet architectures. Grad-CAM visualization highlights anatomical regions correlated with each disease, providing model interpretability. The lightweight yet accurate MobileNetV2 model with transfer learning is shown to be effective for multi-class eye disease classification even with limited training data. The approach demonstrates the potential of leveraging transfer learning and efficient deep CNN architectures for robust medical image analysis and diagnosis.

(Gour & Khanna, 2021) implemented a transfer learning based a two-input VGG16 architecture for detection of ocular diseases from ODIR fundus image dataset. The proposed model with stochastic gradient descent (SGD) optimizer attained the best performance in terms of AUC and F1 score compared to other models. Class-wise analysis reveals that the model performs well for diseases with sufficient representation in the database but struggles with minority classes. The paper also discussed the advantages of the proposed method and highlighted the model behavior using activation maps.

The paper by (Chea & Nam, 2021) presented multi-class classification of retinal fundus images using deep convolutional neural networks. By combining and preprocessing several public fundus image datasets, the authors achieved peak accuracy of 91.16% in classifying normal retina versus the three disease classes using a 50-layer ResNet architecture. Through techniques like data augmentation and excluding noisy images, they were able to improve on baseline accuracy despite dataset inconsistencies.

Table 2.3: Summary of Related Works

Papers	**Dataset Used**	**Algorithm**		**Disease Labels**	**Evaluation metrics**	**Gaps/limitations**
		CNN	ViT			
(Wassel et al., 2022)	Combined public fundus datasets	--	✓	glaucoma, normal	Accuracy, Sensitivity, Specificity, AUC	The study performed binary classification; no comparison was done with CNN based models.
(Raza et al., 2021)	Kaggle (602)	✓	--	cataract, glaucoma, retinal diseases, and normal images	Accuracy	The study used a limited dataset; evaluation metrics were limited to accuracy only; no comparative evaluation with other models was performed.
(Das et al., 2019)	Kaggle	✓	--	healthy, glaucoma and DR	Accuracy	The study used a small dataset containing 3 disease classes. Comparative evaluation with other models was not done.
(Bernabe et al., 2021)	ORIGA, DIARETDB0(565)	✓	--	glaucoma, DR	F1-score, Accuracy, Precision, Recall	Limited dataset of glaucoma and DR image is used. the model used has limited capacity to capture complex features. Performance comparison with other models was not done.
(Khan et al., 2022)	Kaggle	✓	--	normal, diabetes, glaucoma, cataract, myopia, hypertension	F1-score, Recall Accuracy, Precision,	Only Binary classification for each disease was performed. effects of data augmentation and techniques are not investigated. No comparison with other models was done.

(Sarki et al., 2022)	Messidor, DRISHTI-GS, kaggle	✓	--	DR, DME, glaucoma, cataract, normal	Accuracy, Sensitivity, precision	Moderate performance compared with other related works. No comparison with other models was done.
(Guo et al., 2021)	Kaggle dataset (250)	✓	--	normal, myopia, glaucoma, pigmentosa, maculopathy	Accuracy, sensitivity, specificity	limited dataset was used; no image augmentation techniques were utilized.
(Chea & Nam, 2021)	Combined public dataset	✓	--	DR, glaucoma, and AMD	Accuracy	only 3 diseases classes; moderate result; no comparative model evaluation was done.

Prior studies have explored convolutional deep learning techniques for automated fundus image classification, but certain limitations remain. Most of the research papers have utilized convolutional neural networks. The application of attention-based transformers for multi class retinal disease classification is limited despite their success in other domains. Most works have focused on smaller datasets encompassing only 2-4 disease categories. Evaluation for some works is limited to accuracy metrics. More rigorous evaluation procedures using multiple metrics are also needed to thoroughly assess model capabilities. There is also a lack of comparative assessment between different model architectures like CNNs and transformers to determine the optimal approach. Overall, key limitations exist in multi-class classification, model comparison, use of proper evaluation metrics, model explainability, and handling data limitation challenges. Use of larger datasets by leveraging data augmentation techniques, expanding disease classes, comparative model evaluation, appropriate evaluation metrics, and model visualizations explored could help advance beyond current benchmarks. Also, there is significant potential for custom deep neural networks incorporating transformers together with CNNs to further enhance retinal disease diagnosis by effectively by enhancing global feature representation of the vision transformer model. This thesis tried to address the specified limitations.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

This chapter presents the methodology that was used to carry out this study. It provides an overview of the procedures followed to develop and evaluate multi-class retinal disease classification model. It discusses data collection and data augmentation techniques like transformational and synthetic augmentation utilized to expand the dataset. Finally, it presents evaluation metrics and, the software libraries and hardware utilized for model development and training.

The research process followed multiple stages to achieve the goal of developing vision transformer model for multi-class retinal disease classification. The process employed in carrying out this thesis involves identifying the research area, reviewing previous research by different researchers, investigating limitations in previous works, formulating research questions, selecting appropriate tools and methodologies, collecting and preparing datasets, and finally designing and implementing proposed solution. Figure 3.1 shows the high-level overview of research design process from identifying research area to model development, model evaluation and report writing.

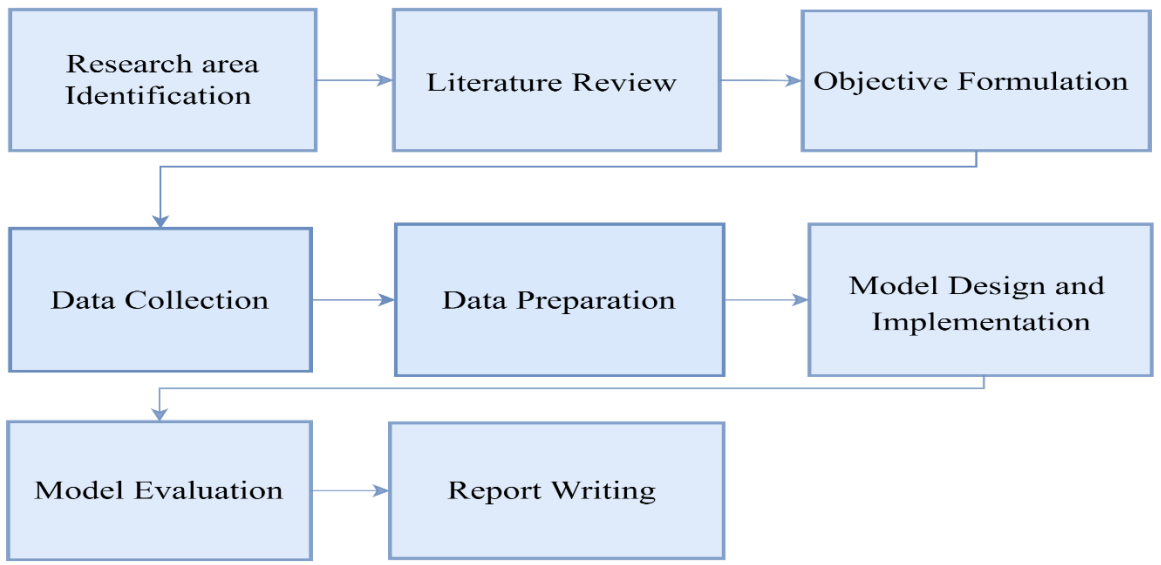


Figure 3.1: Overall Research Process

3.2. Data Collection

Deep learning algorithms depend heavily on data to acquire knowledge. Dataset is crucial for training robust models and evaluate their result. For this study both local fundus images and images from public datasets were used for training, validation and testing. This enables the model to acquire range of features from different sources which also leads to increased generizability and robustness of the model.

Local fundus images of Diabetic Retinopathy (DR), Glaucoma, Age related Macular Degeneration (AMD) and Cataract were collected from eye diagnosis center located in Addis Ababa namely Bruh Vision Eye Speciality Center and St. Paul's Hospital Millennium Medical College in collaboration with ophthalmologists at these centers. Approval was obtained from institutional ethics boards to use anonymized patient retinal images for research purposes. Images were included if they depicted retinal abnormalities indicative of diabetic retinopathy, age-related macular degeneration, glaucoma or cataracts. Low quality images were filtered out.

The data's are combined to better handle that as the prevalence and physiological aspects of some retinal diseases might differ among people of different races, genders, and ethnicities (Bourne, 2011; Malhotra et al., 2021; Peet & Brown, 2006). The local dataset reflects the actual target population that the model aims to serve in real-world clinical setting. A total of 400 local fundus images were collected for four diseases classes. The medical image format is in JPG (.jpg) and the dimensions of local fundus images range from 2124 x 2056 to 3456 x 2304. Table 3.1 shows the number of local and public dataset used for this study.

Table 3.1: Local and Public Fundus Image

dataset	Diabetic Retinopathy	Glaucoma	Cataract	AMD	Healthy	Other
Local	136	117	96	51	-	-
ODIR	300	207	211	171	600	271
Total	436	324	307	222	600	271

To expand the dataset additional ODIR (Ocular Disease Intelligent Recognition)[8] public dataset from kaggle was used for this study. It contains fundus images categorized into diseases like age-related macular degeneration, cataract, glaucoma, diabetes and other retinal abnormalities. Fundus images of the ODIR data set were in .jpg format and have dimension of 512 X 512. The final label in the ODIR dataset combines both left and right eye without distinguishing between potential diseases in each eye. For example, if left eye had cataract and right was normal, the label would just indicate cataract. To address this, the dataset was rearranged by mapping keywords to specific disease labels for each eye individually. Due to limited local dataset an additional 1760 fundus images from ODIR are combined with the local dataset.

The increased diversity and variability are key for developing robust medical imaging classifiers. The diagnostic diversity in ODIR images with additional fundus classes, and severity levels of the common retinal diseases, helps improve model robustness. The images captured using different fundus camera models also provide device variability, avoiding device bias. This aligns with the research objectives and provide complementary value in improving model robustness and generalization when combined with the local dataset.

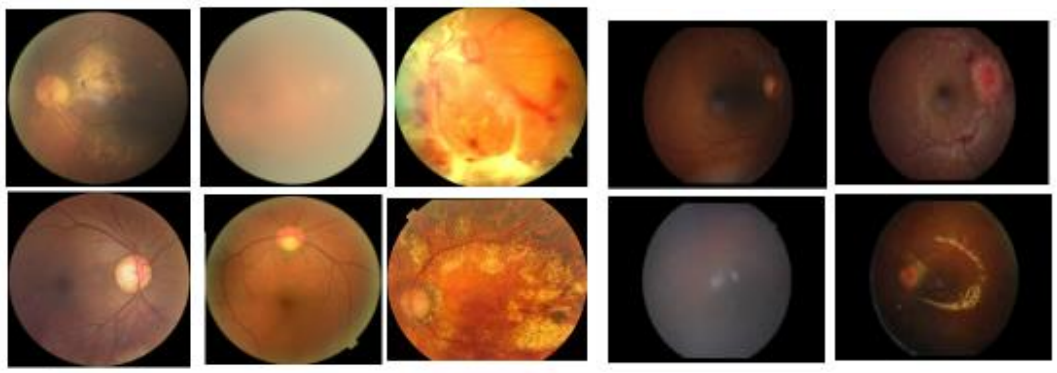


Figure 3.2: Sample images from Kaggle and local dataset.

[8] https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k

3.2.1. Data Augmentation

Deep learning models require large datasets to perform well on image classification tasks and also prevent overfitting. But obtaining extensive medical image data is challenging. To address this, data augmentation techniques were utilized in this study to expand the number and diversity of training images available. Data augmentation creates more training data from existing images through different transformation techniques. This helps the model to sufficient data variations and improves its ability to generalize well.

Three types of augmentations were utilized in this study for evaluating the effectiveness of augmentation techniques on retinal fundus images:

Photometric Transformations: Image brightness, contrast, blur and saturation were randomly altered using the Keras image generator to introduce realistic variability.

Geometric Transformations: Images were randomly rotated from -40 to +40 degrees, flipped horizontally and vertically to introduce realistic geometric variability. Keras image generator was used to implement these.

Table 3.2: Total number of images before and after augmentation

Retinal Diseases	Total number of original images		Total number of Images after augmentation
	Local	ODIR	
AMD	51	171	1400
Cataract	96	211	2160
Glaucoma	117	207	2400
DR	136	300	1910
Healthy	0	600	1840
Other	0	271	1820

Synthetic images using DCGAN: Augmentations can help to improve model performance by providing the model with more data to learn from. However excessive augmentations can lead to redundant features on the training data, which contribute toward model overfitting (Shorten & Khoshgoftaar, 2019).In this study, we employed a Deep Convolutional Generative Adversarial Networks (DCGAN) as an alternative to transformational augmentations to generate synthetic fundus images. DCGAN (Radford et al., 2016) is a specialized GAN architecture for images that uses convolutional neural networks for both the generator and discriminator, rather than multilayer perceptron used in the original GAN. DCGAN combines

GAN with CNN to obtain better training stability and stable training and synthesized image quality. DCGAN has convolutional layers in Generator and transpose-convolutional layers in the discriminator. Transpose convolution up samples the low-resolution noise representation into higher-resolution feature maps whereas convolutional layers down sample the real and fake images into feature maps.

3.2.2. Data Partitioning

The dataset was split into three subsets: training, validation, and test. The total amount of the data used for this study is 11530 fundus images. The training set, comprising 90% or 10377 images, was used to train the model, the validation set, which was split from the training data, helped fine-tune the model's hyperparameters, and a test of 10% consisting of 1153 images was used for final evaluation of the model's performance.

3.3. Evaluation Methods

Evaluation of a deep learning model is essential in every project to assess their performance. Relying on a single metrics for a classification problem is may not be sufficient in evaluating the performance of classifiers, so multiple evaluation metrics were used in this study: accuracy, F1-score, precision, recall, ROC-AUC.

A confusion matrix allows to evaluate the performance of a classification model on test data using a table of test results categorized into TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) to quantify how accurately the model classifies each class. These values are used to calculate accuracy, precision, recall and F1-score.

TP- True Positives, when the model accurately classifies the positive diseases class.

TN- True Negatives, when the model correctly classifies negative class.

FP- False Positives, when the model incorrectly classifies a negative instance as positive class.

FN- False Negatives, when the model wrongly predicts a positive sample as negative.

Table 3.3: Confusion Matrix

		Predicted
Actual	**True Positive** **TP**	**False Negative** **FN**
	False Positive **FP**	**True Negative** **TN**

Accuracy – measures the overall correctness of the model predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad eq.(3.1)$$

Precision – calculates the percentage of correct positive predictions (True positives) to all positive classes.

$$Precision \ = \frac{TP}{TP + FP} \qquad eq.(3.2)$$

Recall (Sensitivity) – the percentage of actual positives that are correctly identified.

$$Recall = \frac{TP}{TP + FN} \qquad eq.(3.3)$$

F1-score – Provides a single metric that captures overall performance across positive and negative cases. Combines both precision and recall.

$$F1 = 2 * \frac{Precision * Re\,c\,all}{Pr\,e\,cision + Re\,c\,all} \qquad eq.(3.4)$$

Receiver Operating Characteristic (ROC) –. Plots tradeoff between true positive rate (TPR) or recall against false positive rate (FPR), demonstrating models capability to differentiate between classes.

AUC – It computes the total two-dimensional region under the ROC curve. The AUC metric has a numerical range from 0 to 1. A higher AUC generally implies a better classifier, with a value of 1 representing a perfect classifier that has zero false positives and zero false negatives. An AUC of 0.5 is equivalent to randomly guessing the class.

In addition to the quantitative metrics discussed above, class attention map (CAM) were visually analyzed to provide qualitative evaluation of regions of the input image focused by the proposed vision transformer model when classifying the retinal diseases.

3.4. Design and Development Tools

Different tools were used to design and build the proposed work.

3.4.1. Design Tool

Draw.io: It is diagramming tool used to generate a variety of diagrams, including flowcharts, process diagrams, network diagrams, organizational charts, and more. It offers an intuitive interface and a diverse collection of shapes, symbols, and connectors that enable users to visually represent information and connections effectively.

3.4.2. Software Tools

Python[9]: Python is a programming language that is utilized extensively in data science and machine learning for its general-purpose and high-level features.

Tensorflow[10]**:** An open-source platform for numerical computing and machine learning. It provides libraries and tools for building and deploying ML models.

Keras[11]**:** A high-level Python API for developing deep learning models that runs on top of TensorFlow. It enables quick and easy model building and training.

Google Drive[12]**:** A cloud storage service used to store the datasets and share them for preprocessing and model training on Google Colab.

Google Colab[13]**:** is a free cloud-based Jupiter Notebook that needs no setup to use while offering free access to computation resources such as GPUs.

Anaconda[14]**:** A Python distribution for data science and machine learning that manages Python packages and environments. It includes many popular ML packages.

[9] https://www.python.org/
[10] https://www.tensorflow.org/
[11] https://keras.io/
[12] https://www.google.com/drive/
[13] https://colab.research.google.com/
[14] https://www.anaconda.com/

CUDA[15]**:** CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model developed by NVIDIA that allows to use GPUs for efficient computation of deep neural networks.

3.4.3. Hardware Tools

The hardware components utilized in this research are given in the table below.

Table 3.4: Hardware Tools

Tools	Description
GPU	an NVIDIA GeForce RTX GPU for accelerating training of the Deep learning models
RAM	For improving computer speed and performance by supporting efficient loading and patch processing of the image dataset.
Hard Drive (HDD)	For storing the fundus image datasets, trained models, experiment logs and results.

[15] https://developer.nvidia.com/cuda-toolkit

CHAPTER FOUR

4. PROPOSED SOLUTION

4.1. Chapter Overview

In this chapter, we present a detailed description of the proposed vision transformer model architecture. First, an overview of the general proposed solution process flow is presented. Next, preprocessing, augmentation techniques and the overall proposed vision transformer model architecture for classifying retinal disease are described. The chapter also discusses model selection, the model learning function, and pseudocode for training the model.

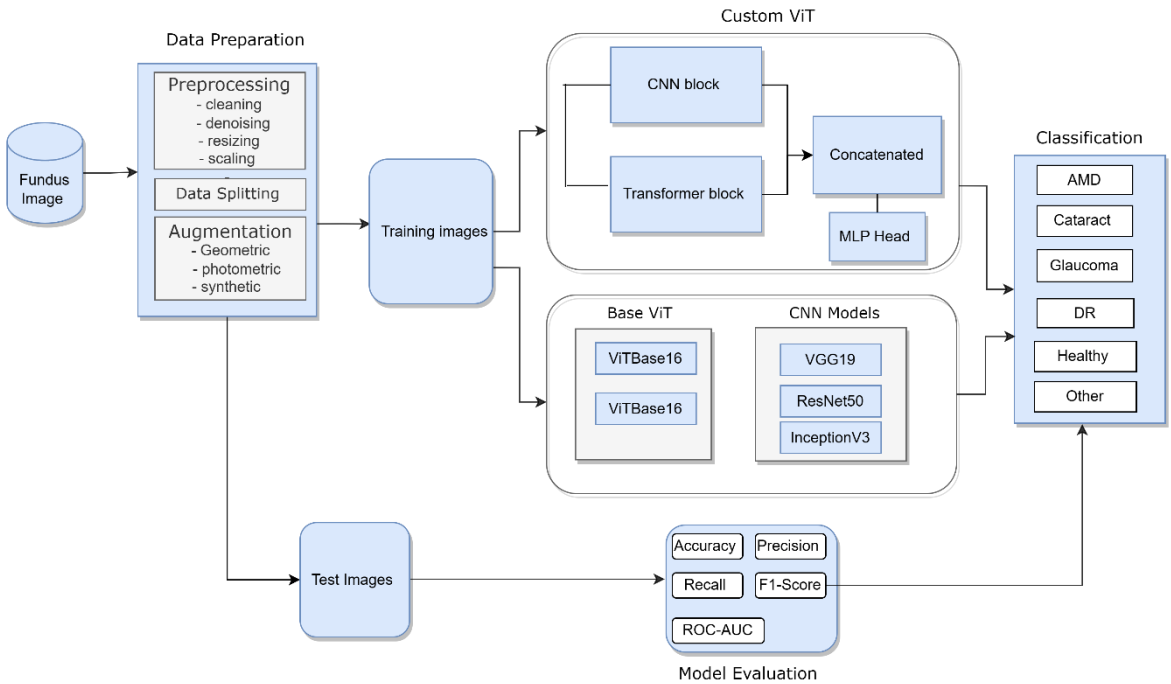


Figure 4.1: Overview of the Proposed Solution Process Flow

4.2. Preprocessing and Augmentation

The distribution and properties of retinal diseases might differ among different demographics due to genetic predisposition, environmental exposures, and access to care. Here local dataset is combined with public dataset for retinal disease classification. Local data reflects images from the actual target population, while public data provides more examples for generalization. Using a combination of local and public datasets provides more diversity and variability in the retinal images for training robust deep learning models.

The raw retinal fundus images collected from multiple sources can exhibit variability in aspects like image resolution, and noise levels. Different preprocessing is conducted to ensure the local data is in the same quality with the public data.

a. Image Cleaning

Low quality images lacking sufficient retinal information or were of extremely poor quality, making them unsuitable for classification, were excluded from the dataset.

b. Image Denoising

Noises in retinal image can be due to either the type or quality of imaging device used or environmental factors such as improper lighting, reflections. The major noises that affect fundus image is gaussian noise and salt and pepper noise (Sonali et al., 2019). Image denoising techniques, such as gaussian filtering and adaptive median filtering were employed to reduce noise artifacts while preserving important image details.

The following preprocessing steps were applied to both the local dataset and the public dataset:

4.2.1. Resize Image

Larger image sizes demand more computational resources. Resizing the images can substantially lower the computational cost of training. Additionally, since the images have black backgrounds, resizing reduces unnecessary background regions and focuses on relevant foreground areas during patch extraction. The preferred input image size for generating patches is 224 x 224 pixels. Therefore, all input images were resized to this dimension.

4.2.2. Normalization

Retinal fundus images can exhibit color imbalances due to differences in cameras or lighting conditions used. Normalization standardizes the input data by changing the range of pixel intensity values and improve model training performance. Normalization scales the pixel values from [0,255] to [0, 1] using the formula:

$$normalized\ image = (\frac{inputimage}{255}) \qquad eq.(4.2)$$

4.2.3. Data Augmentation

Supervised deep learning models require large training datasets to develop accurate and generalizable deep learning models. Augmenting the data by applying transformations can increase the number and diversity of training examples when limited data is available. This study employs two augmentation approaches - transformational and synthetic augmentations, to improve model performance and analyze the effect of augmentation on fundus images.

Synthetic Generation with DCGAN

The architecture of the DCGAN for synthetic retinal image generation is presented in Figure 4.2. The layers of DCGAN architecture are modified to output 224x224x3 dimension of images. The generator takes 100-dim noise vector as input. This is passed through a dense layer and reshaped to a higher dimensional 7x7x256 representation. Six transpose convolution layers then transform the input representation into a 224x224x3 image. Batch normalization after each layer except the last stabilizes training. LeakyReLU activation avoid vanishing gradients. Tanh output activation maps values to [-1, 1]. The discriminator takes real and generated 224x224 images as input, outputting a probability they are real/fake. It has five strided convolutional layers, reducing spatial size and increasing feature maps. Batch normalization follows each layer, with LeakyReLU activations. A final sigmoid activation produces the real/fake probability.

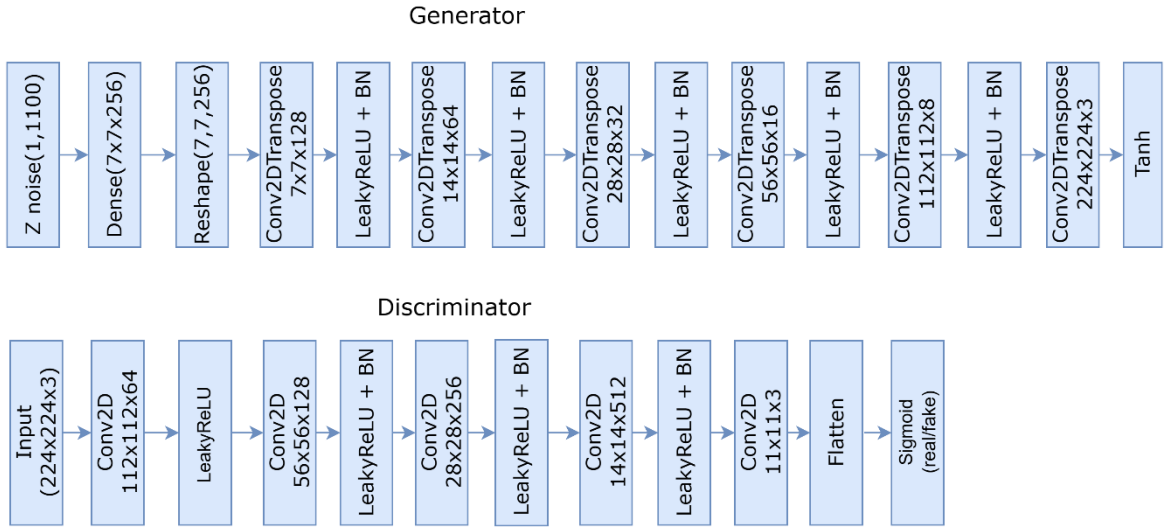


Figure 4.2: DCGAN Architecture for Synthetic Retinal Image Generation

4.3. Model Selection

4.3.1. ViT Model

Precisely locating retinal abnormalities is vital for diagnosing and treating retinal conditions. Nowadays, most research studies employed CNN for solving image classification tasks and it has become the state of-the-art algorithm. Despite having superior local feature extraction capability, CNNs do have certain limitations. A core limitation is that CNNs focus on local neighborhoods when applying convolution operations, and struggles to model global context and long-range dependencies in images (Li et al., 2022). This can limit their performance on medical images where identifying global patterns is important for diagnosis.

Unlike CNNs, which use convolutional layers to extract local features, vision transformers utilize self-attention mechanisms to capture both local and global information. Transformers excel at capturing long-range dependencies and global characteristics in sequences due to multiple self-attention layers, which makes them suitable for understanding relationships between different parts of an image. ViT models achieved state-of-the-art results on image recognition benchmarks, outperforming CNNs given sufficient training data (Dosovitskiy et al., 2020).

4.3.2. Deep CNN Models

Several CNN pre-trained models have been developed starting from 2012. Deep CNN architectures like ResNet, Inception and VGGNet have demonstrated strong performance on fundus image classification for retinal disease diagnosis (Guo et al., 2021; Khan et al., 2022). For performance comparison, we used three efficient pretrained CNN models ResNet50, VGG19, and Inception V3 as our CNN models. These specific models were chosen due to their top performance from CNN architectures and also because they have been widely employed as strong baselines in related works on retinal disease classification.

4.4. Proposed Vision Transformer Model

The architecture of the proposed vision transformer model tailored for retinal disease classification is illustrated in Figure 4.3. The proposed vision transformer architecture utilized ViT-B/16 and ViT-B/32 variants of ViT model. This base configuration contains a sequence of 12 identical transformer encoder blocks. The input retinal fundus image is split into small

non overlapping image patches, which are then embedded with positional encodings before feeding into a standard ViT-Base encoder block.

A modification on the Original ViT is the addition of convolutional blocks between each transformer encoder layer. The convolutional neural block (CNN), which runs in parallel with the ViT model, takes the input image and produces an original image embedding and they are added to the output of each of the encoder layers. At each encoder layer, before passing the output to the next layer, the original image embedding vector is concatenated to the output. This allows the network to remember the original input image after each encoder layers, enhancing global feature extraction capabilities of the ViT model. The custom MLP head classifier is used to further adapt the pretrained model to our target classification. Overall the proposed architecture aims to leverage the strengths of the transformer self-attention and CNN convolutions to boost diagnostic capabilities for retinal fundus image analysis.

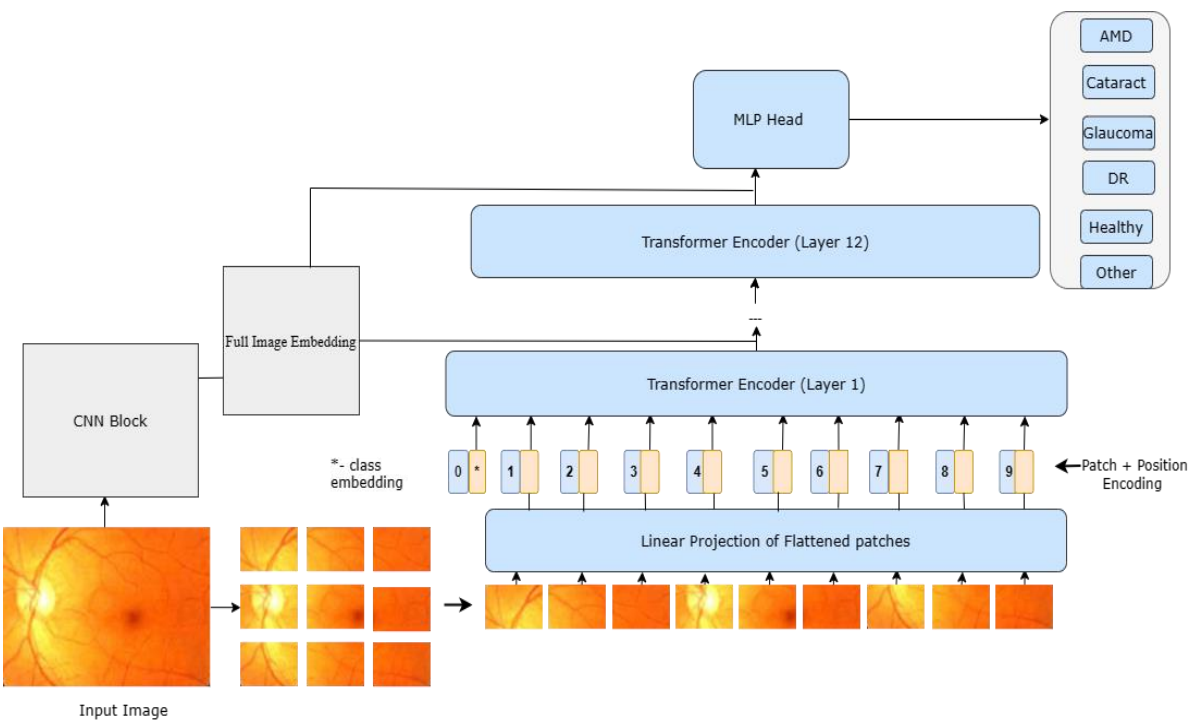


Figure 4.3: The Proposed Vision Transformer Model

Transformer Encoder

The transformer encoder contains a multi-headed self-attention mechanism to model interactions between input image patch tokens. The encoder takes embedded patches along

with position embeddings as input. Within each encoder block, multi-head self-attention computes query (Q), key (K), and value (V) matrices to model spatial relationships among patches. Using multiple heads captures different types of dependencies and relationships within the input patch sequences by computing multiple self-attention scores.

It achieves this by projecting the input into several different subspaces, each with its own set of learned weights or heads. Each head independently computes attention scores between the input representations, producing a set of attention outputs. These are concatenated into one vector, which is then added to the original input vector via a residual skip connection. This preserves information and prevents loss across transformations. The output is fed to a feedforward layer, combined with the previous output layers through another skip connection, and normalized again. Finally, the output of final transformer encoder layer is passed to custom MLP block for classification.

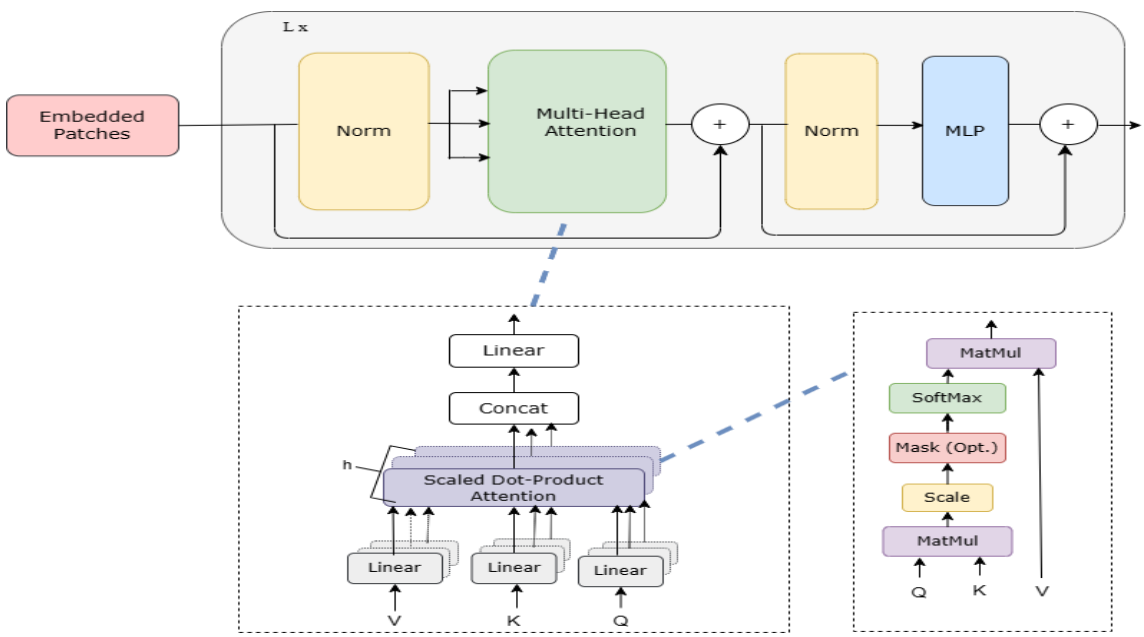


Figure 4.4: Transformer Encoder Architecture

(Dosovitskiy et al., 2020)

CNN Block

The CNN architecture consists of a sequence of three 2D Convolutional layers and 1D global max pooling layer. The first convolution has 16 filters of a 3x3 kernel, the second convolution layer with 256 filters and a 5x5 kernel size, and the third convolution layer with 768 filters,

which same as hidden dimension of the base ViT model and a size of 5x5 kernel is used. The image passes through the 2D convolutional layers, and then 1D max pooling condenses the convolutional feature maps into a single vector preserving the most salient features for each filter, finally representing the original input image embedding.

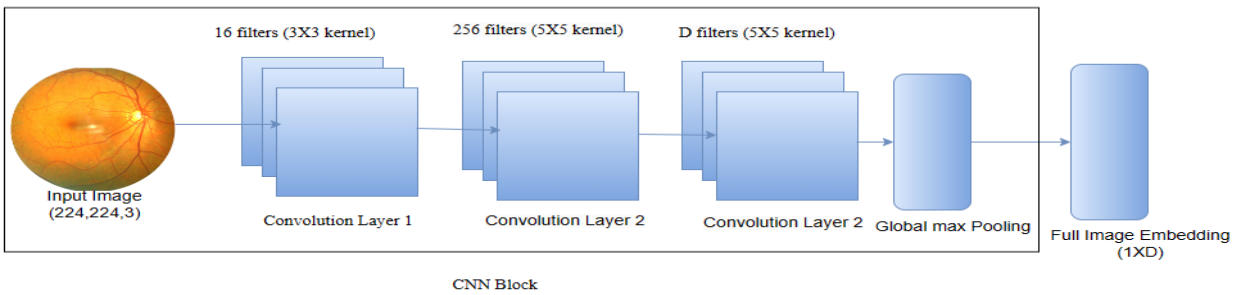


Figure 4.5: CNN Block

Custom MLP Head

The encoder output of the proposed Transfer Learning based ViT model is replaced by Flattening layer followed by batch normalization. Next, two dense layers (DL) of size 256 and 128 with GeLU (Gaussian Error Linear Unit) activation and L2 regularizer are applied. These add non-linearity and condense the encoded features. Batch normalization (BN) is used to normalize the data. Finally, a dense output layer with Softmax activation generates probability scores over the 6 fundus image classes.

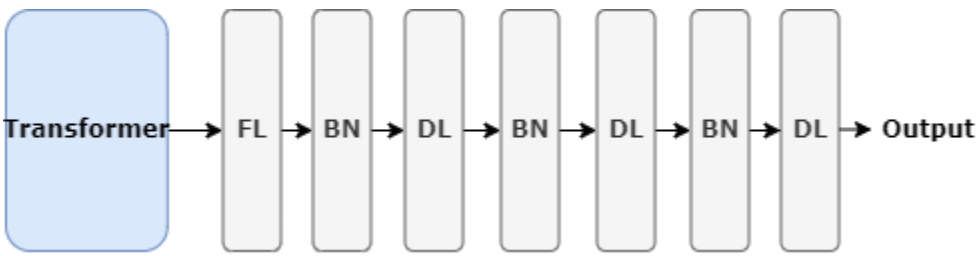


Figure 4.6: Custom MLP Block

Table 4.1: Proposed ViT Base16 Model Architecture Content

Layer (type)	Output Shape	Param #	Connected to
Input_1 (InputLayer)	[(None,224,224, 3)]	0	[]
embedding (Conv2D)	(None, 14, 14, 768)	590592	['input_1[0][0]']

par_cnn_0 (Conv2D)	(None, 222, 222, 16)	448	['input_1[0][0]']
patch_embed_to_1D (Reshape)	(None, 196, 768)	0	['embedding[0][0]']
par_cnn_1 (Conv2D)	(None, 218, 218, 25)	1026566	['par_cnn_0[0][0]']
class_token(ClassToken)	(None, 197, 768)	768	['patch_embed_to_1D[0][0]']
par_cnn_2 (Conv2D)	(None, 214, 214, 76)	4915968	['par_cnn_1[0][0]']
Transformer/posembed_input (AddpositionEmbs)	(None, 197, 768)	151296	['class_token[0][0]']
reshape_par_init (Reshape)	(None, 45796, 768)	0	['par_cnn_2[0][0]']
Transformer/encoderblock_0 (TransformerBlock)	((None, 197, 768), (None, 12, None, None))	7087872	['Transformer/posembed_input[0][0] ']
max_pooling1d(MaxPooling1D)	(None, 1, 768)	0	['reshape_par_init[0][0]']
concatenate (Concatenate)	(None, 198, 768)	0	['Transformer/encoderblock_0[0][0] ', 'max_pooling1d[0][0]']
-----	---	---	---
Transformer/encoder_norm (LayerNormalization)	(None, 209, 768)	1536	['concatenate_11[0][0]']
ExtractToken (Lambda)	(None, 768)	0	['Transformer/encoder_norm[0][0]']
dense (Dense)	(None, 2)	1538	['ExtractToken[0][0]']

Table 4.2: Proposed ViT Base32 Model Architecture Content

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None,224,224,3)]	0	[]
embedding (Conv2D)	(None, 7, 7, 768)	2360064	['input_1[0][0]']
par_cnn_0 (Conv2D)	(None, 222, 222, 16)	448	['input_1[0][0]']
patch_embed_to_1D (Reshape)	(None, 49, 768)	0	['embedding[0][0]']
par_cnn_1 (Conv2D)	(None, 218, 218, 2566)	102656	['par_cnn_0[0][0]']

class_token (ClassToken)	(None, 50, 768)	768	['patch_embed_to_1D[0][0]']
par_cnn_2 (Conv2D)	(None, 214, 214, 768)	4915968	['par_cnn_1[0][0]']
Transformer/posembed_input(AddPositionEmbs)	(None, 50, 768)	38400	['class_token[0][0]']
reshape_par_init (Reshape)	None, 45796, 768)	0	['par_cnn_2[0][0]']
Transformer/encoderblock_0(TransformerBlock)	((None, 50, 768), (None, 12, None,None))	7087872	['Transformer/posembed_input[0][0] ']
max_pooling1d (MaxPooling1D)	(None, 1, 768)	0	['reshape_par_init[0][0]']
concatenate (Concatenate)	(None, 51, 768)	0	['Transformer/encoderblock_0[0][0]', max_pooling1d[0][0]']
----	---	--	--
Transformer/encoder_norm(LayerNormalization)	(None, 62, 768)	1536	['concatenate_11[0][0]']
ExtractToken (Lambda)	(None, 768)	0	['Transformer/encoder_norm[0][0]']
pre_logits (Dense)	(None, 768)	590592	['ExtractToken[0][0]']
dense (Dense)	(None, 6)	4614	['pre_logits[0][0]']

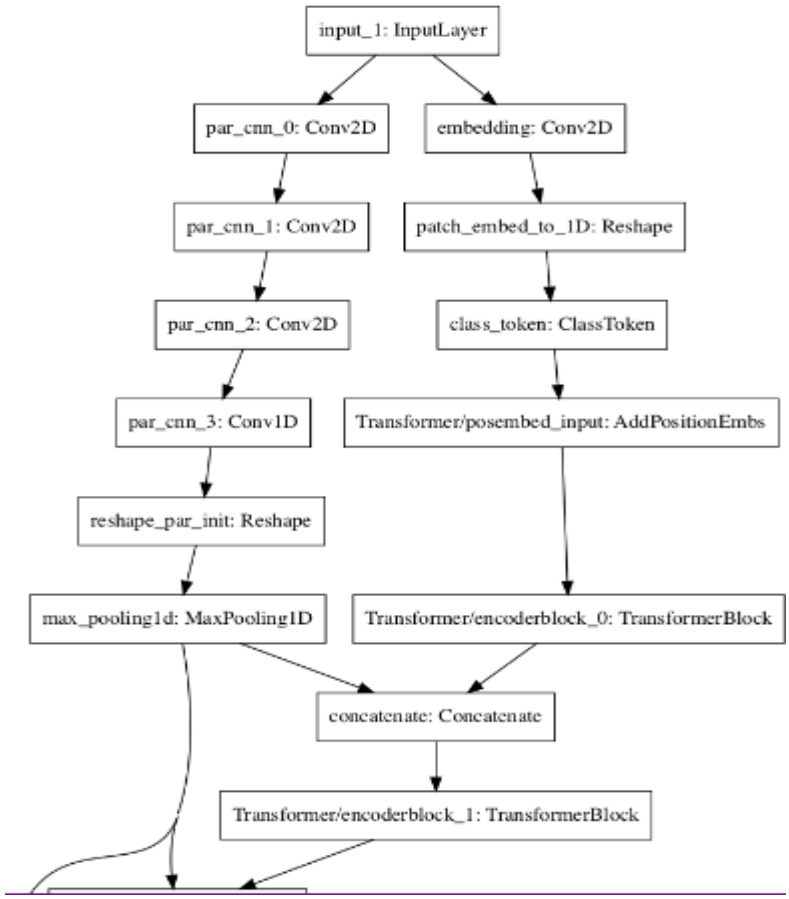


Figure 4.7: ViT Network Architecture with Concatenated CNN

4.5. Model Learning Function

4.5.1. Activation Function

Activation functions introduce nonlinearity into deep learning models to determine the models output from a given inputs. They transform the weighted sum of the total input nodes to an output neuron of a neural network. Softmax activation is used for this study.

Softmax: Softmax normalizes input vectors to calculate a probability distribution over multiple classis. The Softmax outputs sum to 1, representing probabilities of each class. Softmax activation is used for this study.

Softmax is given as $$Softmax(z)_i = \frac{e^{z_i}}{\Sigma_{j=1}^{k} e^{z_j}} \qquad eq.(4.3)$$

Where z is the input vector, k is the number of classes, e^{z_i} is exponential of input vector and e^{z_j} is exponential function for output vector.

4.5.2. Loss Function

For our multi-class classifier model, we used categorical cross-entropy (CCE) loss function. CCE calculates the divergence between the predicted class probabilities and the actual labels by taking the average negative log probability of the true class labels.

CCE is given as $CCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{C=1}^{C} y_i{}^{c} \log(\hat{y}_i{}^{c})$ $eq.(4.4)$

Where:

N - Number of samples

C - Number of classes

$y_i{}^{c}$ - True label (0 or 1) for a class

$\hat{y}_i{}^{c}$ - predicted probability for a class C

4.6. Training Process Pseudocode

The training process of the proposed vision transformer model is given in the following pseudocode.

Algorithm 1: *pseudocode for proposed model training and testing*

Input:

Augmented fundus image dataset

Data divided into: Train, Validation, and Test set

Output: classification score of the test set classes

Model parameters

1.Set Image dimension to: (224,224,3)

- *Use a batch size: 8*
- *Patch size:16,32*
- *Set optimizer: Adam*
- *Use learning rate of : 0.0001*
- *train for Number of epochs :50*

2. Set the number of mini-batches as: $nb = n / batchsize$

For iteration = 1: Number of epochs

For batch = $1:nb$ *(number of mini-batches)*

- *select a batch from the training set,*
- *Train the model on the image batch by minimizing the CCE loss.*
- *Back propagate the loss*
- *Update the model parameters*

4. Classify images in the test set using the trained model

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

This chapter provides an in-depth description of the tasks involved in implementing the proposed vision transformer architecture for retinal disease classification. It covers environment setup, data preprocessing, proposed model implementation, hyperparameter configuration and experimental class undertaken. Furthermore, this section provides some code sample snippets for the implementation of proposed solution.

5.2. Working Environments

The implementation of deep learning solutions requires selection of programming environments, frameworks, and hardware infrastructure. The following sections provide detailed specification of the hardware components and software tools utilized to implement proposed model. These tools enabled the experimentation of vision transformer architecture, CNN models, optimizer configurations, and training schemes.

5.2.1. Hardware Components

Two desktop computers were utilized with the following specifications:

- ✓ Desktop 1:
    - Windows 10 Pro operating system
    - Processor: Intel® Core i5-4590 CPU @ 3.3GHz
    - Installed Memory: 8GB RAM memory
    - 64-bit x64 processor
- ✓ Desktop 2 :
    - Windows 11 Pro operating system
    - Processor: Intel ® Core i7-8700 CPU @ 3.2GHz processor
    - Internal GPU: Intel® HD Graphics 4600
    - External GPU: NVIDIA GeForce RTX 2070 GPU with 8GB RAM
    - 64-bit x64 processor

The first desktop with i5 CPU was used for document writing. The second more powerful desktop with i7 CPU and dedicated GPU was leveraged to accelerate deep learning model training.

5.2.2. Software Components

The main software tools and libraries used for development and training were:

Anaconda: A distribution of Python and R that provides data science packages and IDEs like Jupyter Notebook and Spyder for coding. Anaconda Navigator version 2.3.1 is used.

Python: Python version 3.6 was used to implement the data preprocessing, augmentation, and deep learning models.

Jupyter Notebook: An interactive web-based IDE that allowed combined code execution, visualization, and documentation.

Google Colab: Is a free cloud-based notebook environment that gives free access to GPUs and TPU resources for accelerated deep learning training.

TensorFlow: An end-to-end open-source Python library for data preprocessing, model building, training and evaluation. Provides Keras API. TensorFlow version 2.6/2.12 was used for this study.

5.3. Training Procedure

To achieve the objectives of this study, the following processes were carried out.

- Data Collection: retinal image datasets were collected, such as from public repositories and local eye diagnosis centers.
- Data preprocessing: The image data was loaded, checked for quality, resized to standard dimensions.
- Data Augmentation: Data augmentation techniques like geometric and photometric augmentations, and synthetic generation were applied to expand the training data.
- GPU Configuration: The NVIDIA GPU server was configured, including installing CUDA/PyTorch and setting up drivers.
- Implementing the proposed solution.

5.4. Dataset Preparation

5.4.1. Data splitting and Loading

The data is split in to train (70%), validation (20%), and test (10%). To prevent data leakage, the test data is split from the dataset first, the validation is then separated from the training data.

Table 5.1: The number of fundus images for each class

Disease Type	Total Images	Training	Testing
AMD	1400	1260	140
Cataract	2160	1944	216
Glaucoma	2400	2160	240
DR	1910	1719	191
Healthy	1840	1656	184
Other	1820	1638	182

Resizing fundus input images during model training ensures consistent dimensions, improves computational efficiency, and also reduces the training time. All the input images are resized to size of 224 x 224 pixel.

Pixel values of the input images are normalized to values between 0 and 1 through the preprocessing step of rescaling=1/255. This process is known as normalization or standardization. This process improves numerical stability, enhances gradient-based optimization, facilitates faster convergence, treats all color channels equally, and can lead to improved model performance.

```
image_size = 224
batch_size = 8
EPOCHS = 50

train_data_dir = 'C:/Users/yared/Desktop/Olyad/FINAL TRAINING/OLO/TRAIN'
test_data_dir = 'C:/Users/yared/Desktop/Olyad/FINAL TRAINING/OLO/TEST'

train_generator = datagen.flow_from_directory(
        train_data_dir,
        target_size = (image_size, image_size),
        batch_size=batch_size,
        class_mode='categorical',
        subset='training')
validation_generator = datagen.flow_from_directory(
        train_data_dir,
        target_size = (image_size, image_size),
        batch_size=batch_size,
        class_mode='categorical',
        subset='validation',
        shuffle=False)
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
        test_data_dir,
        target_size=(image_size,image_size),
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=False
```

Snippet code 5.1: Dataset Loading and Splitting

5.4.2. Image Data Augmentation Implementation

As mentioned in chapter 3 two types of augmentation techniques are used in this study. Transformation augmentation and synthetic augmentation.

Keras ImageDataGenerator module is used to apply geometric augmentation techniques.

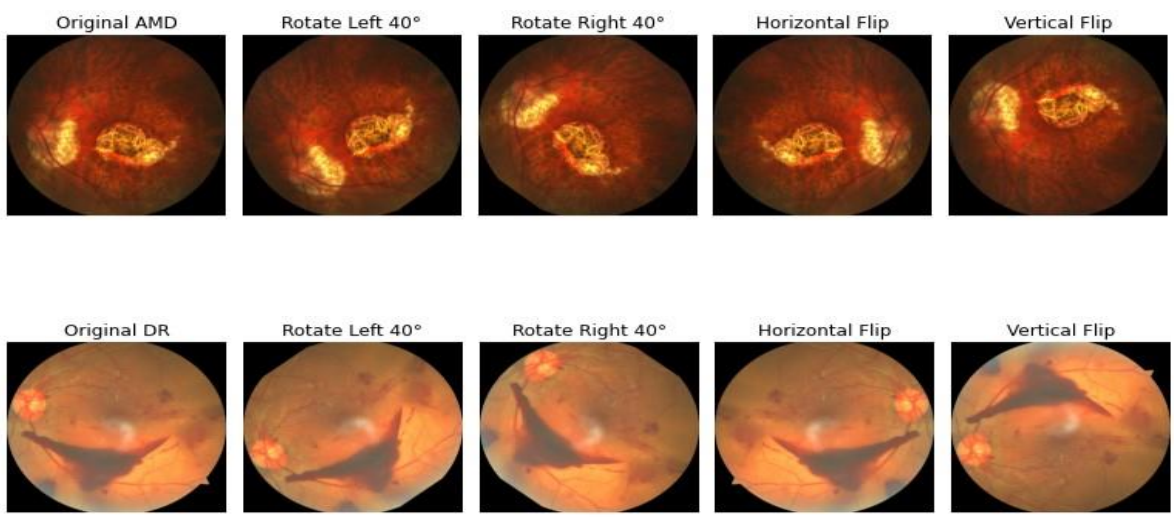


Figure 5.1: Sample Geometric Augmented Images for AMD and DR

```
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img,img_to_array

datagen = ImageDataGenerator(
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=40,
    shear_range=0.2
)
```

Snippet Code 5.2: Geometric Data augmentation

The OpenCV (cv2) library was used to generate photometric augmentations for my study**.**

```
import cv2
import os
import numpy as np
import matplotlib.pyplot as plt

def add_noise(image):
    noise = np.zeros_like(image)
    cv2.randn(noise, 0, 50)
    return cv2.add(image, noise)

def alter_contrast(image):
    alpha = np.random.uniform(0.5, 5)
    beta = np.random.uniform(-50, 50)
    return cv2.convertScaleAbs(image, alpha=alpha, beta=beta)

def alter_brightness(image):
    alpha =np.random.uniform(1, 1.5)
    beta = np.random.uniform(-50, 50)
    return cv2.convertScaleAbs(image, alpha=alpha, beta=beta)

def alter_saturation(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    alpha = np.random.uniform(0.5, 5)
    hsv[:,:,1] = cv2.convertScaleAbs(hsv[:,:,1], alpha=alpha)
    return cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

def blur(image):
    kernel_size = np.random.choice([3, 5, 7])
    return cv2.GaussianBlur(image, (kernel_size, kernel_size), 0)
```

Snippet Code 5.3: Photometric Data Augmentation

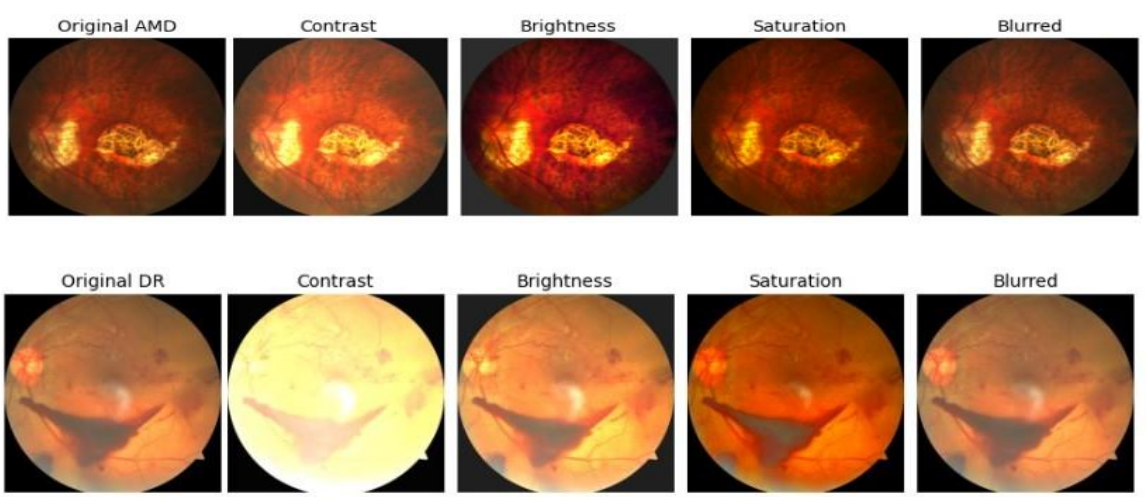


Figure 5.2: Sample Photometric Augmented Images for AMD and DR class

Synthetic Data Augmentation

In addition to geometric and photometric augmentations, synthetic retinal images were generated using a DCGAN as described in Section 4.2.3.The DCGAN generator and discriminator models were implemented in Keras. For synthetic retinal images generation, binary cross-entropy loss for the generator and discriminator network of the DCGAN model as well as least square loss for the discriminator were utilized. Adam optimizer with model a learning rate of 0.0001 were used for each network. Training proceeded for more than 500 epochs with a batch size of 16 images. Snippet 5.4 provides tensorflow code of the DCGAN generator and discriminator.

```
def generator():
    inputs = keras.Input(shape=(1, 1, 100), name='input_layer')

    x = layers.Dense(7*7*256, use_bias=False)(inputs)
    x = layers.Reshape((7, 7, 256))(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    x = layers.Conv2DTranspose(128, (5, 5), strides=(1, 1), padding='same', use_bias=False)(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    x = layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same', use_bias=False)(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    x = layers.Conv2DTranspose(32, (5, 5), strides=(2, 2), padding='same', use_bias=False)(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    x = layers.Conv2DTranspose(16, (5, 5), strides=(2, 2), padding='same', use_bias=False)(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    x = layers.Conv2DTranspose(8, (5, 5), strides=(2, 2), padding='same', use_bias=False)(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU()(x)

    outputs = layers.Conv2DTranspose(3, (5, 5), strides=(2, 2), padding='same', use_bias=False, activation='tanh')(x)

    model = tf.keras.Model(inputs, outputs, name="Generator")
    return model
```

```python
def discriminator():
    inputs = keras.Input(shape=(224, 224, 3), name='input_layer')

    x = layers.Conv2D(64, kernel_size=4, strides=2, padding='same', name='conv_1')(inputs)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(128, kernel_size=4, strides=2, padding='same', name='conv_2')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(256, kernel_size=4, strides=2, padding='same', name='conv_3')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(512, kernel_size=4, strides=2, padding='same', name='conv_4')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(3, kernel_size=4, strides=1, padding='valid', name='conv_5')(x)
    #x= layers.Dropout(0.2)
    x=layers.Flatten()(x)


   # outputs = layers.Activation('sigmoid')(x)
    outputs = layers.Dense(1, activation='sigmoid')(x)

    model = keras.Model(inputs, outputs, name="Discriminator")
    return model
```

Snippet Code 5.4: DCGAN Generator and Discriminator Model

5.5. Components of the Proposed ViT Model

5.5.1. VIT Model

This study employed the ViT model in two ways: the original ViT Model and custom ViT combined with CNN and custom MLP block. Both Model adheres to the architecture in (Dosovitskiy et al., 2020), consisting of the layers: patch creation, class tokens ,position embeddings a patch encoder layer, and the ViT encoder model.

Class Token

```python
class ClassToken(tf.keras.layers.Layer):
    """Append a class token to an input layer."""

    def build(self, input_shape):
        cls_init = tf.zeros_initializer()
        self.hidden_size = input_shape[-1]
        self.cls = tf.Variable(
            name="cls",
            initial_value=cls_init(shape=(1, 1, self.hidden_size), dtype="float32"),
            trainable=True,
        )
```

Snippet Code 5.5: Class Token implementation

Position Embeddings

```
class AddPositionEmbs(tf.keras.layers.Layer):
    """Adds (optionally learned) positional embeddings to the inputs."""

    def build(self, input_shape):
        assert (
            len(input_shape) == 3
        ), f"Number of dimensions should be 3, got {len(input_shape)}"
        self.pe = tf.Variable(
            name="pos_embedding",
            initial_value=tf.random_normal_initializer(stddev=0.06)(
                shape=(1, input_shape[1], input_shape[2])
            ),
            dtype="float32",
            trainable=True,
        )
```

Snippet Code 5.6: Positional Embedding Implementation

MutiHeadSelfAttention Layer

```
class MultiHeadSelfAttention(tf.keras.layers.Layer):
    def __init__(self, *args, num_heads, **kwargs):
        super().__init__(*args, **kwargs)
        self.num_heads = num_heads

    def build(self, input_shape):
        hidden_size = input_shape[-1]
        num_heads = self.num_heads
        if hidden_size % num_heads != 0:
            raise ValueError(
                f"embedding dimension = {hidden_size} should be divisible by number of heads = {num_heads}"
            )
        self.hidden_size = hidden_size
        self.projection_dim = hidden_size // num_heads
        self.query_dense = tf.keras.layers.Dense(hidden_size, name="query")
        self.key_dense = tf.keras.layers.Dense(hidden_size, name="key")
        self.value_dense = tf.keras.layers.Dense(hidden_size, name="value")
        self.combine_heads = tf.keras.layers.Dense(hidden_size, name="out")
```

Snippet Code 5.7: Multi Head Self Attention Implementation

Transformer Encoder Block

```
class TransformerBlock(tf.keras.layers.Layer):
    """Implements a Transformer block."""

    def __init__(self, *args, num_heads, mlp_dim, dropout, **kwargs):
        super().__init__(*args, **kwargs)
        self.num_heads = num_heads
        self.mlp_dim = mlp_dim
        self.dropout = dropout

    def build(self, input_shape):
        self.att = MultiHeadSelfAttention(
            num_heads=self.num_heads,
            name="MultiHeadDotProductAttention_1",
        )
        self.mlpblock = tf.keras.Sequential(
            [
                tf.keras.layers.Dense(
                    self.mlp_dim,
                    activation="linear",
                    name=f"{self.name}/Dense_0",
                ),
                tf.keras.layers.Lambda(
                    lambda x: tf.keras.activations.gelu(x, approximate=False)
                )
                if hasattr(tf.keras.activations, "gelu")
                else tf.keras.layers.Lambda(
                    lambda x: tfa.activations.gelu(x, approximate=False)
                ),
                tf.keras.layers.Dropout(self.dropout),
                tf.keras.layers.Dense(input_shape[-1], name=f"{self.name}/Dense_1"),
                tf.keras.layers.Dropout(self.dropout),
```

Snippet Code 5.8: Transformer Encoder Implementation

5.6. Hyperparameter Configuration

Hyper parameters are configurations that are defined for controlling training process of a model. Hyperparameters used in this study are discussed below.

- **Optimizers**: The Adam optimizer was used to update model weights during training, as it combines the advantages of AdaGrad and RMSprop (Kingma & Ba, 2015). It is efficient in terms of computations and memory requirements.
- **Learning Rate**: The learning rate controls how quickly weights update during optimization. A learning rate of 0.0001 was used.
- **Activation Function**: The Softmax was used for the multi-class classifier output layer, as it is well-suited for multi-class prediction tasks.
- **Number of Epochs**: The number of training epochs determines how many iterations through the full dataset. 50 epochs were chosen as optimal through experimentation.

- **Batch Size:** The batch size controls how many samples propagate through the network per iteration. A batch size of 8 was used for training, based on initial experiment and due to limited GPU resource.

```
def custom_loss(y_true, y_pred):
        return tf.keras.losses.categorical_crossentropy(y_true, y_pred, label_smoothing=0.2)
model.compile(optimizer=Adam(0.0001), loss=custom_loss, metrics=['accuracy'])
```

Snippet Code 5.9: Optimizer with Adam

Table 5.2: Summary of Hyperparameter Configuration

No	Hyperparameters	Types and Values
1	Learning rate	0.0001
2	Batch size	8
3	Epoch	50
4	Optimizer	Adam
5	Loss	Categorical cross-entropy loss
6	Patch size	16,32
7	Transformer layers	12
8	Number of attention heads	12
9	Transfer Learning	Yes
10	Weights	Pretrained on ImageNet

5.7. Network Training

Transfer learning from ImageNet pre-trained weights was utilized for the models. The original ViT architecture was trained on the augmented retinal fundus dataset using training from scratch and transfer learning approach. The custom ViT model with added CNN and MLP blocks was trained on the fundus data using transfer learning. The CNN models ResNet50, InceptionV3, and VGG19 were initialized with pre-trained weights. The models were trained using categorical cross-entropy loss optimized with Adam for 50 epochs. Extensive photometric, geometric, and synthetic augmentations were applied to the retinal fundus images during training.

5.8. Experiment Class

Experiments were conducted to assess the performance of vision transformer (ViT) model in the classification of multi-class retinal diseases. First, the original ViT model was trained from scratch and using transfer learning on the retinal fundus dataset augmented with photometric, geometric and synthetic transformations. Next, a custom ViT model was

proposed by adding custom CNN and MLP blocks. This model was trained on the three augmented datasets. The custom ViT model was compared to CNN models like ResNet50, InceptionV3 and VGG19. The results of these experiments are provided in the next section of chapter 6. Thorough analysis and discussion of the findings are also presented. Below table summarizes each experiment, the models evaluated, training approach used, and datasets.

Table 5.3: Summary of Experimental Class

Experiment	Models Evaluated	Training Approach	Dataset
Original ViT Evaluation	ViT-b16, ViT-b32	From scratch and transfer learning	Retinal fundus images (photometric, geometric, synthetic augmentations)
Custom ViT Evaluation	ViT with CNN and MLP blocks	Transfer learning	Retinal fundus images (photometric, geometric, synthetic augmentations)
Comparison with CNN Models	ResNet50, InceptionV3, VGG19	Transfer learning	Retinal fundus images (photometric, geometric, synthetic augmentations)

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1. Chapter Overview

This section presents the results of the experiments conducted. The experiments aimed to address the research questions around using vision transformer for classifying common eye disease from fundus images. The results are analyzed using tables, figures, and graphs to compare model performance across different experiments.

6.2. Augmentation Results

Due to the limited availability of large annotated retinal image datasets, data augmentation techniques were explored to expand the training data and improve model generalization. These study makes a comparison between Traditional transformation augmentations like photometric and geometric augmentations and synthetic generation when applied to retinal fundus images.

Traditional transformation augmentations helped increase dataset diversity and increase dataset size and hence enhancing model performance. For synthetic generation, however, the generated images lacked fine details and realistic lesions of retinal diseases. While DCGAN image quality slightly improved with more epochs, the results were not on par with real images to generate training data for classification models. Hyperparameter tuning and change of loss functions did not sufficiently improve quality. The GAN architecture, DCGAN had difficulties in effectively modeling realistic synthetic retinal image generation. Traditional transformation augmentations provided clear benefits, whereas DCGAN synthetic generation resulted in poor retinal disease classification accuracy due to challenges in generating realistic images. Figure 6.1 illustrates examples of original fundus images along with their corresponding synthetic images generated by the DCGAN model for different disease classes: two original images on top, and two synthetic images generated by the DCGAN underneath for each disease classes.

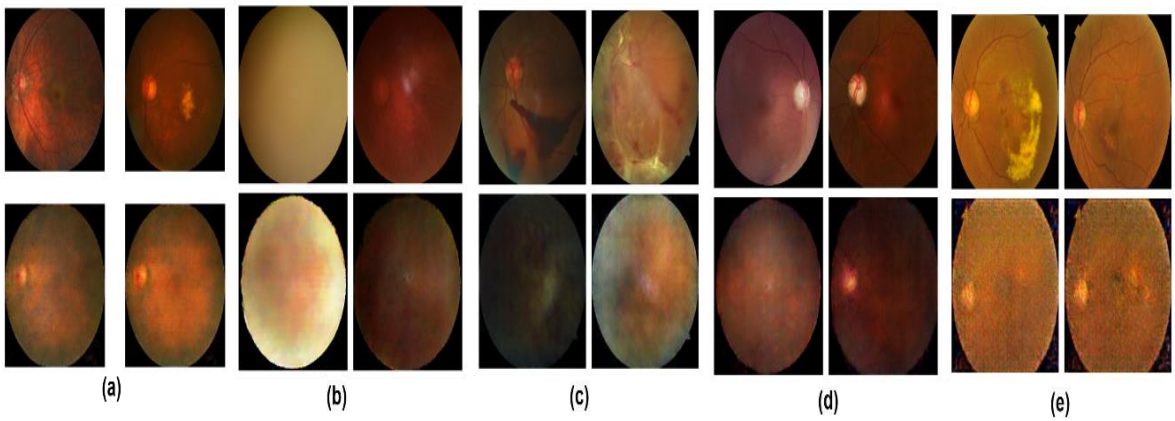


Figure 6.1: Sample original images and their corresponding generated fundus images

(a) AMD, (b) DR, (c) DR, (d) Glaucoma, and (e) Other.

6.3. Experimental Results

Multiple experiments were conducted to evaluate vision transformer (ViT) and CNN models for multi-class retinal disease classification. The existing ViT architectures of ViT-Base-16 and ViT-Base-32 were assessed given their promising results in image classification tasks. A custom ViT model was proposed by incorporating CNN block and MLP head aimed at improving capability on retinal fundus images. Additionally, widely used CNN models including VGG, ResNet, and InceptionNet were used for comparison.

6.3.1. Base Vision Transformer (ViT) Result

The first experiment was conducted using original ViT. The results of the base model variants: base16 and base32 considering the data augmentation techniques employed: geometric augmentation and photometric augmentation are presented. The goal is to evaluate the capabilities of standard ViT model for multi-class retinal disease classification when trained on retinal fundus dataset.

Table 6.1: Classification Performance for ViT Model Trained from Scratch

Augmentation Technique	Model	Accuracy	Precision	Recall	F1-Score
Geometric Transformation	ViTBase16	0.83	0.838	0.830	0.829
	ViTbase32	0.84	0.861	0.844	0.844
Photometric Transformation	ViTBase16	0.83	0.832	0.825	0.826
	Base32	0.85	0.859	0.855	0.854

Table 6.2: Classification Performance for ViT Model with Transfer Learning

Augmentation Technique	Model	Accuracy	Precision	Recall	F1-Score
Geometric Transformation	Base16	0.91	0.916	0.913	0.90
	Base32	0.89	0.895	0.890	0.891
Photometric Transformation	Base16	0.88	0.882	0.877	0.877
	Base32	0.87	0.877	0.871	0.871

The above table shows the benefits of using transfer learning when applying vision transformer model to retinal fundus image classification. As seen in Table 6.1, the Base-16 and Base-32 ViT model variants trained from scratch achieved reasonable but not exceptional performance, with accuracy ranging from 83-85%. Leveraging transfer learning with ImageNet pre-trained weights provided a significant boost as shown in Table 6.2. Accuracy improved to 87-91%, fine-tuning the pre-trained models elevated all the performance metrics including precision, recall, F1-score, and compared to training from scratch.

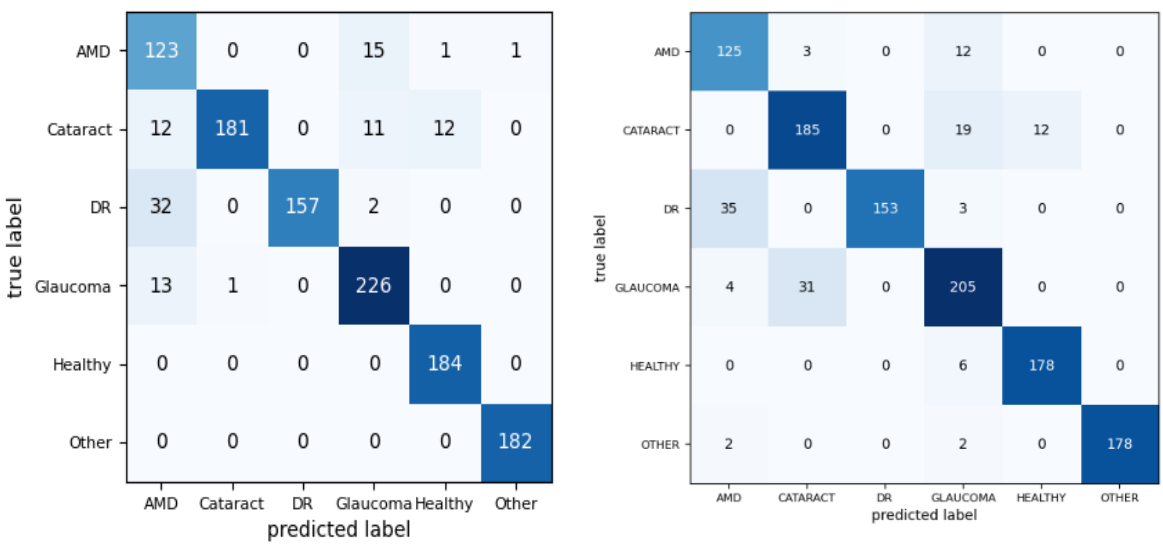


Figure 6.2: Confusion Matrix for Base ViT with Geometric Augmentation

The confusion matrix for the base ViT model trained with geometric augmentation and transfer learning indicates strong overall classification performance, with a highest accuracy of 91% across the six disease classes. The model shows capability in identifying conditions like Diabetic Retinopathy, Glaucoma, and Normal retina, with minimal misclassifications.

While the model achieves high accuracy overall, the confusion matrix reveals opportunities to improve discrimination of challenging diseases classes.

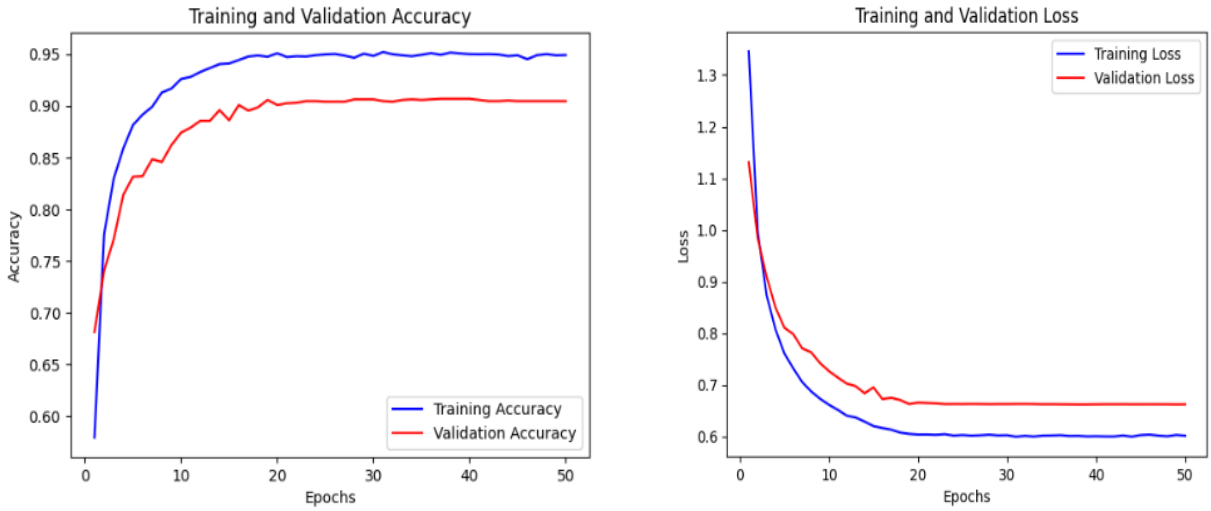


Figure 6.3: Training and Validation Accuracy/Loss for ViTBase32

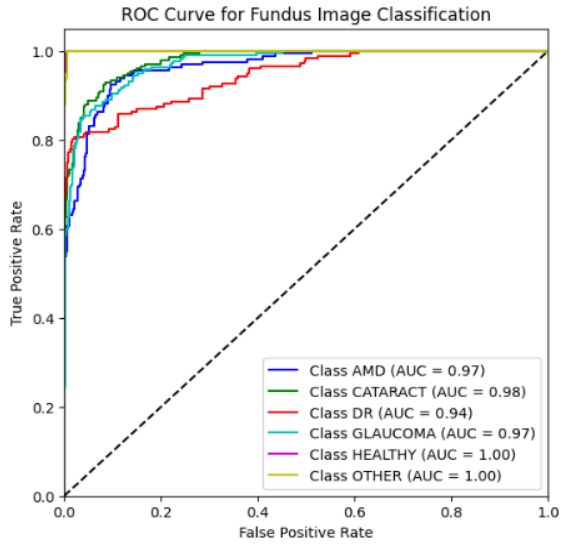


Figure 6.4: ROC for the Base 16 ViT models with Photometric Augmentation

6.3.2. Custom Vision Transformer Model Result

Building upon the initial promising results of the original ViT models, a custom ViT architecture was proposed aimed at improving performance on retinal fundus images: concatenating a convolutional block to each ViT encoder to enhance global feature learning of ViT model, and adding custom layers on top of the classifier to further refine and optimize

the classification. To assess the impact of these modifications, the custom ViT model was trained and evaluated on the same augmented retinal fundus dataset.

Table 6.3: Classification Performance for Custom ViT Model with Transfer Learning

Augmentation	Model	Accuracy	Precision	Recall	F1-Score
Geometric Transformation	Custom Base16	0.95	0.95	0.95	0.95
	Custom Base32	0.94	0.938	0.937	0.936
Photometric Transformation	Custom Base16	0.94	0.943	0.938	0.938
	Custom Base32	0.93	0.943	0.933	0.934

The table above presents the classification accuracy and other performance metrics achieved by the Modified Vision Transformer Model (CustomBase16 and CustomBase32) model trained on transformation augmentation techniques. The results demonstrate performance gains achieved by the custom ViT model over the original ViT architecture. Accuracy is improved, reaching up to 95% on the geometric augmented dataset using the Base-16 custom model. Precision, recall, F1-score are likewise elevated by 3-6% across both datasets and when compared to original ViT. For the custom Base16 variant, the model achieved an accuracy of 95% using geometric augmentation along with transfer learning. With photometric augmentation, the custom Base16 variant attained an accuracy of 94%. For the custom Base32 variant, the model achieved an accuracy of 93% using geometric transformation augmentation. And when trained on the photometric augmented data, the custom Base32 model scored 93% accuracy.

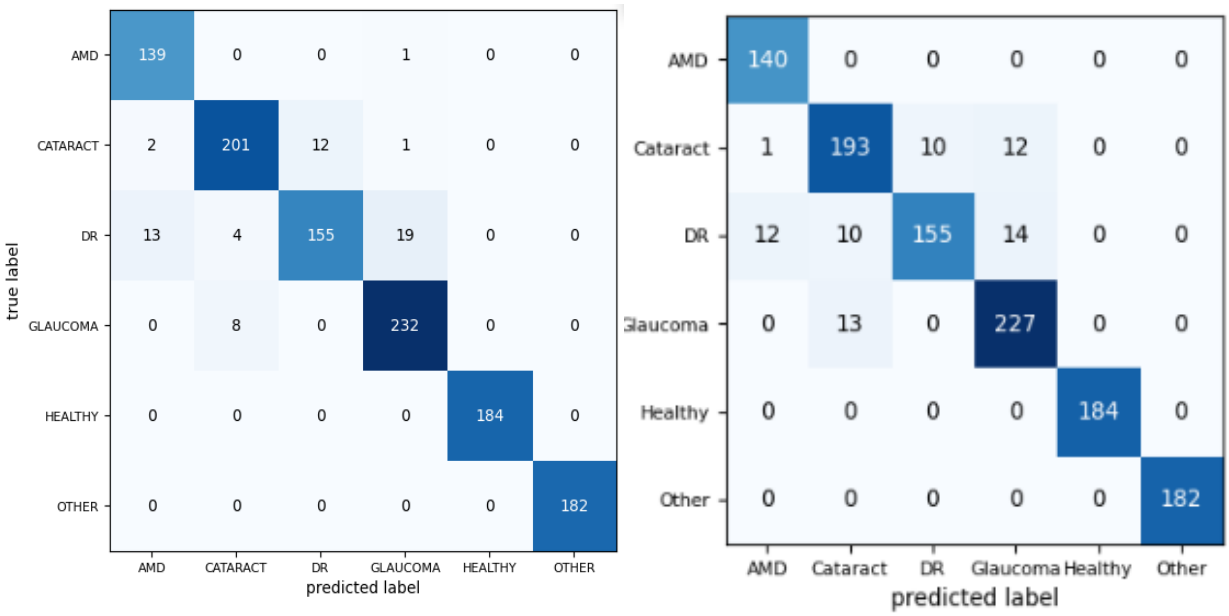


Figure 6.5: Confusion Matrix of Custom ViT with Geometric and Photometric

The confusion matrix above demonstrates the strong multi-class classification performance of the custom ViT model, achieving 95% overall accuracy. Compared to the original ViT, the proposed custom model shows improvements in distinguishing between diseases. There are minimal misclassifications across all disease types, with the diagonal representing high volumes of correct predictions. The high values along the diagonal and low misclassification counts highlight the custom ViT model's capabilities in accurately identifying retinal conditions and learning discriminative features tailored to fundus images through original image embedding of CNN integration and custom classifier head. The confusion matrix validates the performance gains from the proposed architectural improvements and their ability to boost ViT's suitability for retinal disease classification.

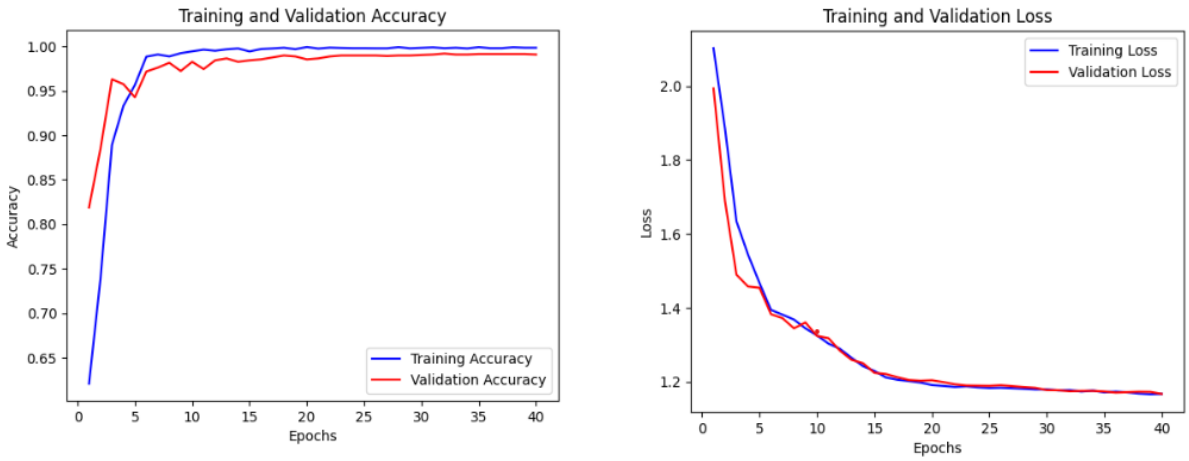


Figure 6.6: Training and Validation Accuracy/Loss for Custom ViT- Base32

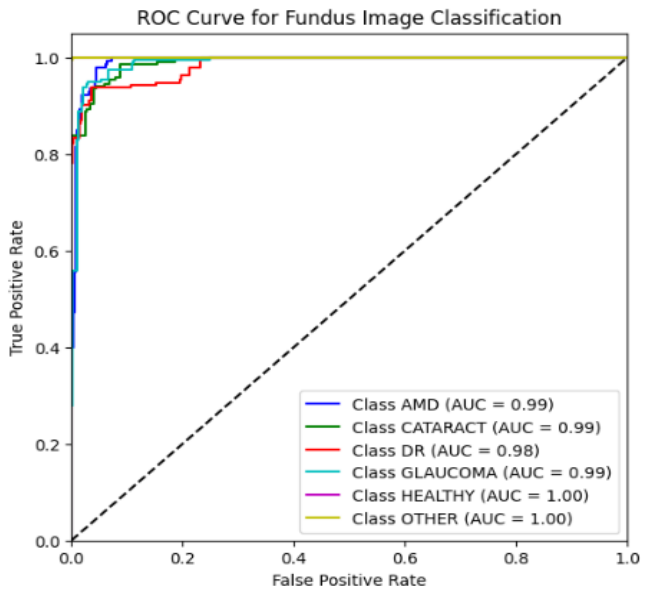


Figure 6.7: ROC for Custom ViT- Base16 with Photometric Augmentation

6.3.3. CNN Models Performance

We also evaluated the performance of CNN models, including ResNet50, InceptionV3, and VGG19 on the augmented retinal fundus images. The CNN models are trained and evaluated on the same augmented retinal fundus datasets using the same experimental settings for comparison. Transfer learning was leveraged by initializing the CNNs with ImageNet pre-trained weights and fine-tuning on the fundus images. Table 6.4 shows the results of the CNN models. The inceptionV3 model achieved the best accuracy, precision, recall, and F1-score, compared to ResNet50 and VGG19, with VGG19 having the lowest scores overall.

Table 6.4: Summary of Classification Performance of CNN Models

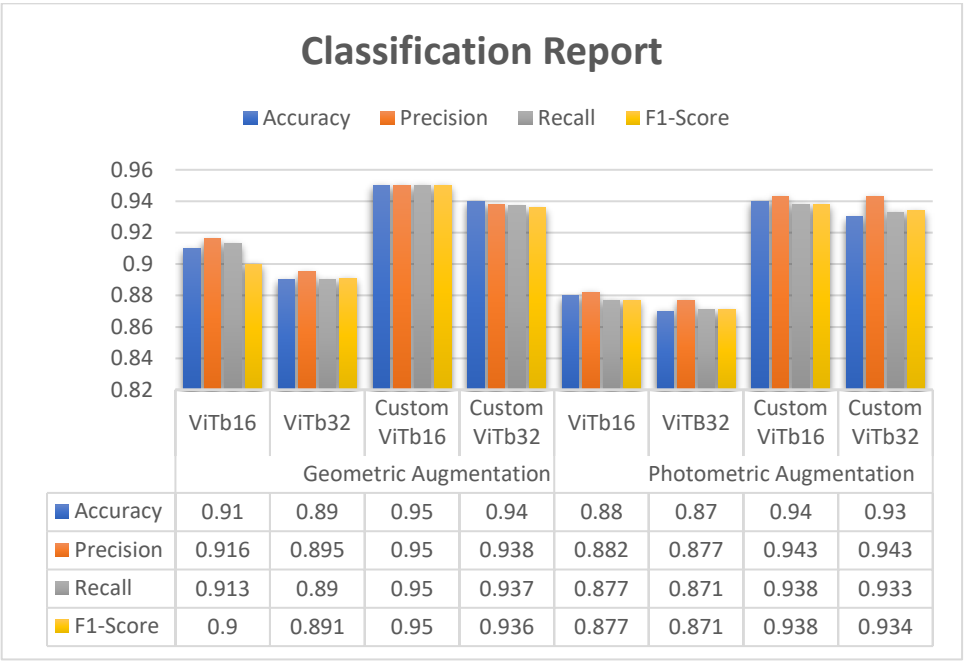
Model	Augmentation	Accuracy	Precision	Recall	F1-Score
ResNet50	Geometric	0.92	0.920	0.918	0.918
InceptionV3		0.94	0.949	0.943	0.943
VGG19		0.84	0.846	0.843	0.843
ResNet50	Photometric	0.89	0.896	0.889	0.889
InceptionV3		0.94	0.942	0.941	0.940
VGG19		0.85	0.849	0.845	0.845

6.3.4. Comparison With the Proposed Model

This section provides comparisons between the performances of the proposed vision transformer (ViT) model with original ViT as well as CNN models. The goal is to quantify the improvements afforded by the proposed modifications to ViT.

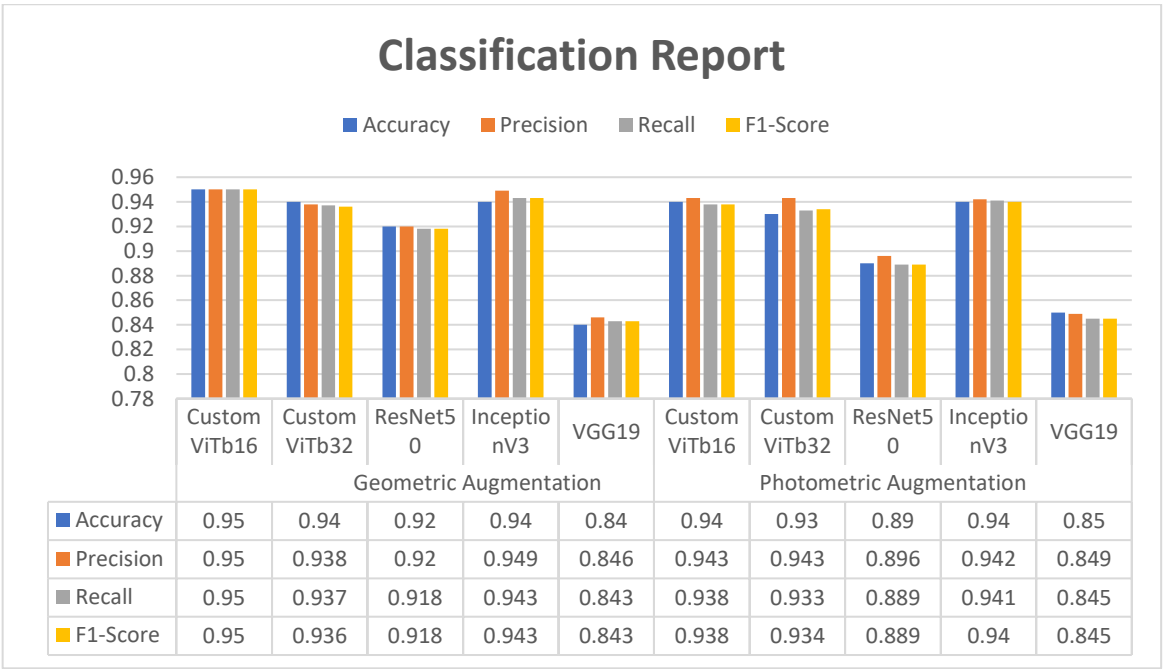
After evaluating their classification results, we found that the custom Vision Transformer Model achieved the highest accuracy of 95%. This surpasses the existing Vision Transformer model, which ranged from 87% to 91%, and the CNN models, which achieved the highest accuracy of 94%. In terms of precision, the custom Vision Transformer Model demonstrated a precision of 0.95, which is highest to the precision values of the existing Vision Transformers (ranging from 0.87 to 0.91) and outperformed the CNN models (precision ranging from 0.84 to 0.94). Regarding recall, the proposed Vision Transformer Model exhibited a recall of 0.95, which exceeded the recall values of the existing Vision Transformers (ranging from 0.87 to 0.91) and the CNN models (recall of 0.94). Furthermore, the custom ViT Model achieved F1-score of 0.95, indicating a balanced performance between precision and recall. This outperformed the F1-scores of the existing Vision Transformers (ranging from 0.87 to 0.90) and the CNN models (F1-score of 0.84-0.94). These experiments highlight the effectiveness of the Modified Vision Transformer Model in accurately classifying retinal diseases and suggest its potential as a promising approach for multi-class retinal disease classification. The choice between augmentation techniques had a slight impact on performance, with the geometric transformation generally achieving higher accuracy, precision, and F1-score. Figure 6.8 below shows a comparison of evaluation metrics

between the proposed custom ViT and the original ViT model and Figure 6.9 illustrates a comparison of metrics between the custom ViT and CNN models.



	ViTb16	ViTb32	Custom ViTb16	Custom ViTb32	ViTb16	ViTB32	Custom ViTb16	Custom ViTb32
	Geometric Augmentation				Photometric Augmentation			
Accuracy	0.91	0.89	0.95	0.94	0.88	0.87	0.94	0.93
Precision	0.916	0.895	0.95	0.938	0.882	0.877	0.943	0.943
Recall	0.913	0.89	0.95	0.937	0.877	0.871	0.938	0.933
F1-Score	0.9	0.891	0.95	0.936	0.877	0.871	0.938	0.934

Figure 6.8: Comparison of results of Proposed ViT and Original ViT model



	Custom ViTb16	Custom ViTb32	ResNet50	Inceptionv3	VGG19	Custom ViTb16	Custom ViTb32	ResNet50	Inceptionv3	VGG19
	Geometric Augmentation					Photometric Augmentation				
Accuracy	0.95	0.94	0.92	0.94	0.84	0.94	0.93	0.89	0.94	0.85
Precision	0.95	0.938	0.92	0.949	0.846	0.943	0.943	0.896	0.942	0.849
Recall	0.95	0.937	0.918	0.943	0.843	0.938	0.933	0.889	0.941	0.845
F1-Score	0.95	0.936	0.918	0.943	0.843	0.938	0.934	0.889	0.94	0.845

Figure 6.9: Comparison of the Proposed ViT Model and CNN Models

6.3.5. Class activation maps

To provide visualization of the model's predictions, Class Activation Maps (CAMs) are utilized as proposed by (Chefer et al., 2021). CAM provides a visualization of which parts of the input image provide the most evidence for the selected class, as understood by the model. To generate a CAM for a vision transformer, a model explanation technique called Layer-wise Relevance Propagation (LRP) is used. LRP calculates relevance scores for each patch of the input image by propagating relevance from the output layer to the input layer in a layer-wise manner. These relevance scores indicate the importance or contribution of each input feature to the final prediction made by the model. The relevance scores are then visualized as a heatmap, where the intensity of the colors represents the level of relevance. Figure 6.10 illustrates sample activation maps for images with different retinal conditions. The heat map highlights distinct features that the model pays attention to when making classification decisions based on relevant parts of the retinal images.

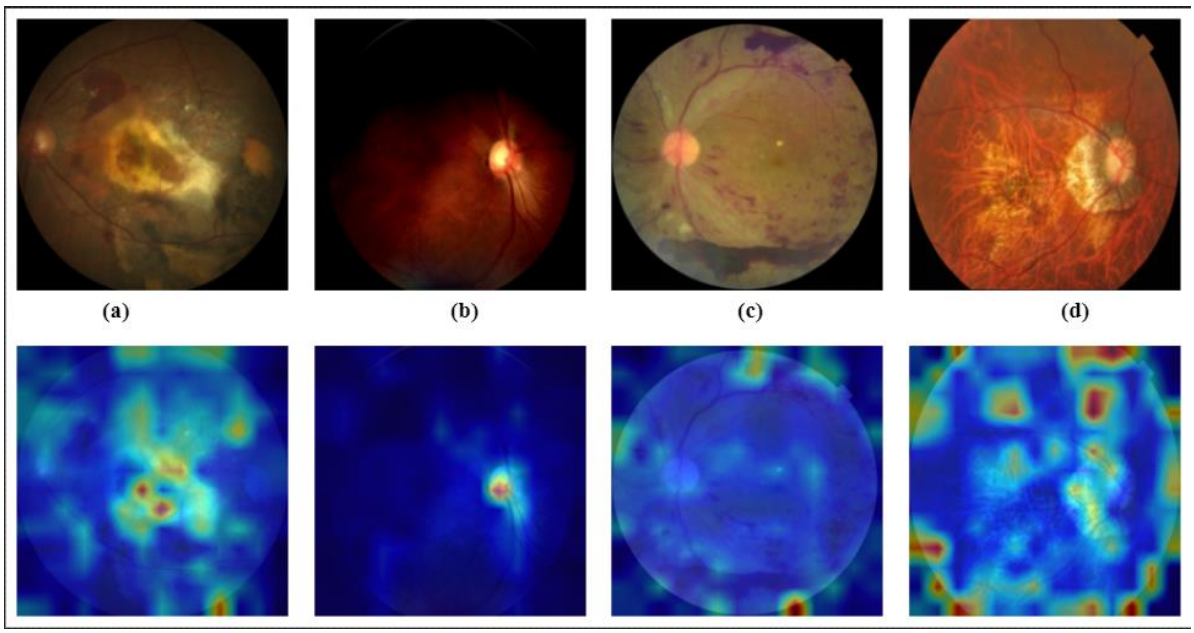


Figure 6.10: Heat map Visualization of Retinal Diseases.

(a) AMD, (b) Glaucoma, (c) DR, (d) Other.

Local Data Analysis

In order to analyze the model's capability when trained on local fundus image data, experiments were conducted training the custom model from scratch and using transfer learning on the augmented local dataset. The result demonstrated that the model is able to effectively learn from local data, achieving decent performance with 84-88% accuracy while trained from scratch and fine-tuning. The model demonstrated reasonably strong precision and recall across disease classes when trained on just the local dataset. Fine-tuning provides gains in precision for AMD, cataract and glaucoma. Confusion matrix in figure 6.11 shows some misclassifications distinguishing between AMD/cataract and DR/glaucoma, likely due to visual similarities between Ocular disease classes. Overall, the model exhibited reasonably strong performance when trained on local data but the gaps in accuracy and confusion between some classes illustrates the limitations of local dataset size and diversity. Combining it with public data is likely to further enhance model robustness, especially for minority classes. The public data provides increased diversity and increased dataset size to improve the model's ability to discriminate between visually similar classes. The result from the combined dataset highlights the value of supplementing local data with public data for enhanced the performance.

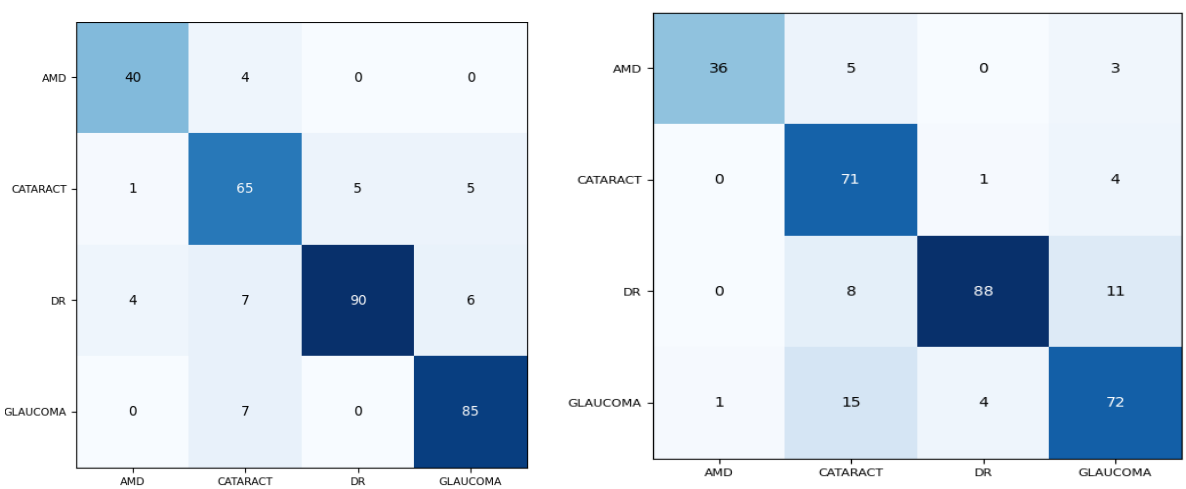


Figure 6.11: Confusion Matrix of Custom ViT with Local Dataset

Evaluation of Models Trained on Synthetic Images

In addition to transformational augmentation, DCGAN was implemented to generate synthetic retinal fundus images. However, an analysis of the generated images found that they lacked sufficient realism and fine details to accurately mimic real retinal photographs. Key pathological features and lesions were absent or poorly represented. As a result, models trained on the synthetic images performed very poorly compared to those trained on transformational augmented images. Specifically, the vision transformer model achieved only 69% accuracy when trained on synthetic images, compared to 95% accuracy when trained on real augmented images. Precision, recall and F1-scores were likewise significantly lower for the synthetically trained model.

The CNN models also demonstrated significantly reduced accuracy when trained on the synthetic fundus images.InceptionV3 accuracy dropped to 65% compared to 94% on transformation augmentation. Similarly, ResNet50's accuracy decreased using synthetic images. This poor performance indicates the synthetic images failed to provide useful representations for learning discriminative features needed for accurate classification.

6.3. Research Question and Answer Discussion

The findings of the experiments presented in this chapter are discussed to answer the research question.

RQ1: What is an efficient way of improving the performance of the classification of most common eye diseases from fundus images?

To answer this question, we conducted experiment evaluating Vision transformer (ViT) model. We proposed custom ViT model with concatenated custom CNN and MLP blocks tailored to fundus images. Additionally, we incorporated augmentation techniques to enhance the performance of eye disease classification. Augmentation techniques such as transformational and synthetic augmentation are utilized to increase the size and diversity of the training data and hence improve performance model generalizability. Based on the experiments, an efficient way to improve the performance of the classification of the most common eye disease from fundus images is by employing fine-tuned pretrained Vision Transformer (ViT) model with data augmentation techniques. The experiments showed that an efficient way to boost the accuracy of classifying the most prevalent eye diseases using fundus images is to fine-tune a pretrained Vision Transformer (ViT) model and use data

augmentation techniques. Our proposed custom ViT model, modified with custom CNN and MLP demonstrated strong performance, particularly ViTb16 with geometric transformation, achieving high accuracy, precision, recall, and F1-score. These models effectively learned and recognized retinal disease patterns, showcasing robustness in handling variations in position and scale. The use of geometric and photometric transformations enhanced the models' ability to generalize across different real-world variations. The results highlight the performance of vision transformer combined with augmentation techniques as an efficient approach for improving the generalization performance of classification of the most common eye disease from fundus images, ultimately contributing to accurate and reliable medical diagnostics in ophthalmology.

RQ2: To what extent can vision transformer (ViT) accurately classify different types of eye diseases?

To assess the classification capabilities of vision transformer (ViT), we conducted experimental classes specifically focused on two variants of the vision transformer model: ViTBase16 and ViTBase32 on a multi-class retinal disease dataset. To measure the classification performance we employed different evaluation metrics such as accuracy, precision, recall, F1-score and ROC curve. The results demonstrated over 80% accuracy for conditions like diabetic retinopathy, glaucoma, healthy retina and other classes. However, accuracy under 80% for identifying conditions like AMD and Cataracts.by using transfer learning the model achieved 88% accuracy across all classes confirming ViT models can accurately classify different eye disease types. The ViT models also achieved reasonably strong precision, recall and f1-score values for retinal disease classes, with scores over 0.80 when trained with transfer learning. The ROC curves plotted showcase strong discrimination ability, with curves skewed towards the upper left corner indicating indicates high true positive rate and low false positive rate. This indicates vision transformers show promising result for Multi class retinal disease classification.

RQ3: How does the modified vision transformer model compare to existing vision transformer and CNN models in terms of classification accuracy and other performance metrics?

Our third research question sought to compare our proposed custom vision transformer (ViT) model to the original ViT model and CNN models like ResNet50, VGG19 and InceptionV3 in terms of classification performance. The results demonstrated our custom ViT achieved higher accuracy of 95% compared to 87- 91% for existing ViT model and 84 - 94% for the CNN models. Precision, recall, and F1 metrics were also improved. The consistent gains can be attributed to our proposed enhancements of adding convolutional blocks and classifier layers, which effectively tailored ViT to the medical imaging domain. The CNN block allow the model to better capture high-level features, enhancing the ViT global feature extraction, and the classifier layers helped refine predictions. Together, they improved ViT's feature representations. This highlights the potential of customized vision transformer by combining CNN and ViT.

6.4. Hyperparameter Tuning Results

In our experiments, we tested different learning rates for the model between 0.0001 and 0.1 We found that slower learning rates like 0.0001, 0.0002 resulted in better model performance, even though they take longer to converge. As we increased the learning rate past around 0.001, the performance started degrading. This makes sense, since very high learning may cause the algorithm to converge faster but prevent the model from learning properly and result in poor performance. As a result, the value for learning rate is set to 0.0001.

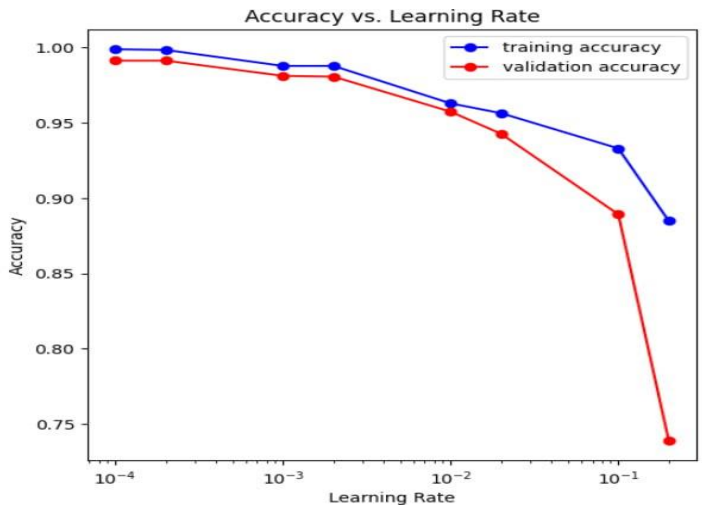


Figure 6.12: Training and Validation Accuracy against Learning rate

Hyperparameter tuning was also performed to determine the optimal activation function and number of epochs. The model was trained with two different optimizers (Adam and SGD) and two different epoch values (30 and 50). The results are shown in the table below. The model was trained using the Adam and SGD optimizers with the Softmax activation function. Training was done for 30 and 50 epochs to determine the impact of training duration. With the Adam optimizer, an accuracy of 93% was achieved after 30 epochs. Training for 50 epochs improved the accuracy to 95%. Using SGD, the accuracy after 30 epochs was lower at 91%. Increasing to 50 epochs increased the accuracy to 93%. Based on these results, the Adam optimizer achieved better accuracy compared to SGD. The accuracy also improved consistently with more training epochs for both optimizers.

Table 6.5: Optimizers and Epoch result

Optimizers	epochs	Accuracy
Adam	30	93%
	50	95%
SGD	30	91%
	50	93%

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

Finally, this chapter summarizes the key contributions and findings of this thesis work on developing enhanced vision transformer model for multi-class retinal disease classification. First, section 7.1 summarizes the main conclusions. Subsequently, section 7.2 presents research directions for future work to present opportunities to build on this research and further advance tailored vision transformer toward clinical impact for retinal disease screening and diagnosis.

7.1. Conclusion

The primary causes of Vision impairment worldwide include retinal diseases like diabetic retinopathy, cataract, glaucoma, and age relate macular degeneration, with high prevalence in the local context as well. Timely screening of these retinal conditions is crucial to mitigate vision loss and prevent blindness. In this research work, we developed a vision transformer deep learning model using transfer learning technique customized for multi-class retinal disease classification from using fundus images. In our approach we used the standard Vision Transformer (ViT) architecture and enhanced the input of the ViT with added convolutional and classifier layers to improve global feature learning for retinal images. Our approach employed transfer learning for classifying fundus images and achieved an accuracy of 93 - 95% on the experimental datasets.

The research also experimented retinal image augmentation techniques, including photometric, geometric, as a solution to address scarcity of training data, privacy issue associated with retinal image data and also to further increase diversity hence improve models performance. The results demonstrate the superior performance of the proposed custom ViT model in improving classification accuracies of retinal diseases outperforming both base original ViT variants and CNN models like VGG19, ResNet50, and Inceptionv3 highlighting the promise of customized vision transformers for advancing classification of retinal diseases.

In terms of model explainability, the research implements CAM (class activation mapping) visualization to understand which regions of the retinal image the model focuses on for making predictions. This helps in gaining insights into the decision-making process of the model. The

study also evaluated the performance of original ViT model trained from scratch and leveraging transfer learning, showed ViT models have significant promise for multi-class classification across different retinal disease types when appropriately customized and trained. Furthermore, the combination of local and public dataset shows that our proposed method is not constrained to fundus images taken using a specific device under certain conditions, instead it can be generalized. In summary, the custom vision transformer model fine-tuned through transfer learning surpassed both the original Vision Transformer and CNN-based transfer learning approaches on the task of classifying fundus images.

Contribution of the Study

- ✓ We thoroughly evaluated vision transformer capabilities for multi-class classification using fundus photography datasets augmented with various transformations.
- ✓ The proposed model architecture enhancements provide an effective way to tailor vision transformers to medical imaging, establishing strong performance surpassing ViT and CNNs.
- ✓ Insights from attention map visualizations and error analysis elucidate model decisions and limitations to guide future improvements.
- ✓ Findings conclusively demonstrate customized vision transformers are a competitive alternative to CNNs for medical image analysis tasks.

7.2. Feature Works

Possible extensions and improvements to our work include the following:

One area is exploring the combined effects of multiple data augmentation techniques like photometric, geometric and synthetic transformations to provide complementary benefits in expanding data diversity and model robustness. Additional work includes extending the model to classify a broader spectrum of disease types and incorporating multimodal retinal imaging data like OCT scans. This expansion would provide a more comprehensive diagnostic tool in real world clinical scenarios.

Another promising direction is exploring model ensembling by integrating vision transformers with state-of-the-art CNN models, for leveraging benefits of both global and local feature learning. Potentially pushing the boundaries of classification accuracy.

Expanding datasets from diverse sources in terms of populations, diseases, and imaging modalities would also boost model generalization capability. Ultimately, rigorous clinical validation studies are essential to evaluate real-world effectiveness and determine any gaps needing to be addressed prior to practical deployment. In summary, ample exciting avenues exist to advance vision transformers toward clinical impact in retinal disease.

* * *

This research project was funded by Adama Science and Technology University under the grant number: **ASTU/SM-R/818/23**

REFERENCES

Akil, M., Elloumi, Y., & Kachouri, R. (2021). Detection of retinal abnormalities in fundus image using CNN deep learning networks. *State of the Art in Neural Networks and Their Applications: Volume 1*, 19–61. https://doi.org/10.1016/B978-0-12-819740-0.00002-4

Askarian, B., Ho, P., & Chong, J. W. (2021). Detecting Cataract Using Smartphones. *IEEE Journal of Translational Engineering in Health and Medicine*, *9*. https://doi.org/10.1109/JTEHM.2021.3074597

Badar, M., Haris, M., & Fatima, A. (2020). Application of deep learning for retinal image analysis: A review. *Computer Science Review*, *35*, 100203. https://doi.org/10.1016/j.cosrev.2019.100203

Bello, I., Zoph, B., Le, Q., Vaswani, A., & Shlens, J. (2019). Attention Augmented Convolutional Networks. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*, 3285–3294. https://doi.org/10.1109/ICCV.2019.00338

Berhane, Y., Worku, A., & Bejiga, A. (2006). National Survey on Blindness , Low Vision and Trachoma in Ethiopia. *Ophthalmological Society of Ethiopia and Ethiopian Public Health Association*, *September*, 1–64.

Bernabe, O., Acevedo, E., Acevedo, A., Carreno, R., & Gomez, S. (2021). Classification of Eye Diseases in Fundus Images. *IEEE Access*, *9*, 101267–101276. https://doi.org/10.1109/ACCESS.2021.3094649

Biswas, M., Kuppili, V., Saba, L., Edla, D. R., Suri, H. S., Cuadrado-Godia, E., Laird, J. R., Marinhoe, R. T., Sanches, J. M., Nicolaides, A., & Suri, J. S. (2019). State-of-the-art review on deep learning in medical imaging. *Frontiers in Bioscience (Landmark Edition)*, *24*(3), 392–426. https://doi.org/10.2741/4725

Blindness and vision impairment. (n.d.). Retrieved February 22, 2023, from https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

Bourne, R. R. A. (2011). Ethnicity and ocular imaging. *Eye*, *25*(3), 297–300. https://doi.org/10.1038/eye.2010.187

Bourne, R. R. A., Steinmetz, J. D., Saylan, M., Mersha, A. M., Weldemariam, A. H., Wondmeneh, T. G., Sreeramareddy, C. T., Pinheiro, M., Yaseri, M., Yu, C., Zastrozhin, M. S., Zastrozhina, A., Zhang, Z. J., Zimsen, S. R. M., Yonemoto, N., Tsegaye, G. W., Vu, G. T., Vongpradith, A., Renzaho, A. M. N., … Vos, T. (2021). Causes of blindness and

vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: the Right to Sight: an analysis for the Global Burden of Disease Study. *The Lancet. Global Health*, *9*(2), e144–e160. https://doi.org/10.1016/S2214-109X(20)30489-7

Chea, N., & Nam, Y. (2021). *Classification of Fundus Images Based on Deep Learning for Detecting Eye Diseases*. https://doi.org/10.32604/cmc.2021.013390

Chefer, H., Gur, S., & Wolf, L. (2021). Transformer Interpretability Beyond Attention Visualization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 782–791. https://doi.org/10.1109/CVPR46437.2021.00084

Cherinet, F. M., Tekalign, S. Y., Anbesse, D. H., & Bizuneh, Z. Y. (2018). Prevalence and associated factors of low vision and blindness among patients attending St. Paul's Hospital Millennium Medical College, Addis Ababa, Ethiopia. *BMC Ophthalmology*, *18*(1), 10–12. https://doi.org/10.1186/s12886-018-0899-7

Chou, C. F., Sherrod, C. E., Zhang, X., Barker, L. E., Bullard, K. M. K., Crews, J. E., & Saaddine, J. B. (2014). Barriers to eye care among people aged 40 years and older with diagnosed diabetes, 2006-2010. *Diabetes Care*, *37*(1), 180–188. https://doi.org/10.2337/DC13-1507

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Deep learning for visual understanding: Part 2 generative adversarial networks. *IEEE SIgnal ProcESSIng MagazInE*, *January*, 53–65. https://twin.sci-hub.tw/6638/2bc534ddaf5c9240cfff3e1bb1697682/creswell2018.pdf#page=1

Das, A., Giri, R., Chourasia, G., & Bala, A. A. (2019). Classification of Retinal Diseases Using Transfer Learning Approach. *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, *Icces*, 2080–2084. https://doi.org/10.1109/ICCES45898.2019.9002415

de Santana Correia, A., & Colombini, E. L. (2022). Attention, please! A survey of neural attention models in deep learning. In *Artificial Intelligence Review* (Vol. 55, Issue 8). Springer Netherlands. https://doi.org/10.1007/s10462-022-10148-x

Devi, Y. S., & Kumar, S. P. (2022). *DR-DCGAN: A Deep Convolutional Generative Adversarial Network (DC-GAN) for Diabetic Retinopathy Image Synthesis*. *19*(2), 2022. http://www.webology.org

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://arxiv.org/abs/1810.04805v2

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020a). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. https://arxiv.org/abs/2010.11929v2

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020b). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. https://arxiv.org/abs/2010.11929v2

Flaxman, S. R., Bourne, R. R. A., Resnikoff, S., Ackland, P., Braithwaite, T., Cicinelli, M. V., Das, A., Jonas, J. B., Keeffe, J., Kempen, J., Leasher, J., Limburg, H., Naidoo, K., Pesudovs, K., Silvester, A., Stevens, G. A., Tahhan, N., Wong, T., Taylor, H., … Zheng, Y. (2017). Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis. *The Lancet Global Health*, *5*(12), e1221–e1234. https://doi.org/10.1016/S2214-109X(17)30393-5

Gangwar, A. K., & Ravi, V. (2021). Diabetic Retinopathy Detection Using Transfer Learning and Deep Learning. In *Advances in Intelligent Systems and Computing* (Vol. 1176). Springer Singapore. https://doi.org/10.1007/978-981-15-5788-0_64

Ghosh, B., Bhuyan, M. K., Sasmal, P., Iwahori, Y., & Gadde, P. (2018). Defect classification of printed circuit boards based on transfer learning. *Proceedings of 2018 IEEE Applied Signal Processing Conference, ASPCON 2018*, 245–248. https://doi.org/10.1109/ASPCON.2018.8748670

Gour, N., & Khanna, P. (2021). Multi-class multi-label ophthalmological disease detection using transfer learning based convolutional neural network. *Biomedical Signal Processing and Control*, *66*(May), 102329. https://doi.org/10.1016/j.bspc.2020.102329

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L.,

& Webster, D. R. (2016). *Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs*. *94043*, 1–9. https://doi.org/10.1001/jama.2016.17216

Guo, C., Yu, M., & Li, J. (2021). Prediction of different eye diseases based on fundus photography via deep transfer learning. *Journal of Clinical Medicine*, *10*(23). https://doi.org/10.3390/jcm10235481

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hinkle, J. W., Ansari, Z., & Tabin, G. C. (2020). Global Ophthalmology Insights for a Global Pandemic. *American Journal of Ophthalmology*, *216*, A9–A11. https://doi.org/10.1016/j.ajo.2020.05.011

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837

Janghorbani, M., Jones, R. B., & Allison, S. P. (2009). Incidence of and risk factors for proliferative retinopathy and its association with blindness among diabetes clinic attenders. *Http://Dx.Doi.Org/10.1076/Opep.7.4.225.4171*, *7*(4), 225–241. https://doi.org/10.1076/OPEP.7.4.225.4171

Junayed, M. S., Islam, M. B., Sadeghzadeh, A., & Rahman, S. (2021). CataractNet: An automated cataract detection system using deep learning for fundus images. *IEEE Access*, *9*, 128799–128808. https://doi.org/10.1109/ACCESS.2021.3112938

Khan, M. S., Tafshir, N., Alam, K. N., Dhruba, A. R., Khan, M. M., Albraikan, A. A., & Almalki, F. A. (2022). Deep Learning for Ocular Disease Recognition: An Inner-Class Balance. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/5007111

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Li, F., Zhou, L., Wang, Y., Chen, C., Yang, S., Shan, F., & Liu, L. (2022). Modeling long-range dependencies for weakly supervised disease classification and localization on chest X-ray.

Quantitative Imaging in Medicine and Surgery, *12*(6), 3364–3378. https://doi.org/10.21037/qims-21-1117

Lim, L. S., Mitchell, P., Seddon, J. M., Holz, F. G., & Wong, T. Y. (2012). Age-related macular degeneration. *The Lancet*, *379*(9827), 1728–1738. https://doi.org/10.1016/S0140-6736(12)60282-7

Maclennan, P. A., McGwin, G., Searcey, K., & Owsley, C. (2014). A survey of Alabama eye care providers in 2010-2011. *BMC Ophthalmology*, *14*(1), 1–10. https://doi.org/10.1186/1471-2415-14-44/TABLES/5

Malhotra, N. A., Greenlee, T. E., Iyer, A. I., Conti, T. F., Chen, A. X., & Singh, R. P. (2021). Racial, Ethnic, and Insurance-Based Disparities Upon Initiation of Anti–Vascular Endothelial Growth Factor Therapy for Diabetic Macular Edema in the US. *Ophthalmology*, *128*(10), 1438–1447. https://doi.org/10.1016/j.ophtha.2021.03.010

Mersha, G. A., Alimaw, Y. A., & Woredekal, A. T. (2022). Prevalence of diabetic retinopathy among diabetic patients in Northwest Ethiopia-A cross sectional hospital based study. *PLoS ONE*, *17*(1 January), 1–13. https://doi.org/10.1371/journal.pone.0262664

Morales, S., Engan, K., Naranjo, V., & Colomer, A. (2017). Retinal Disease Screening Through Local Binary Patterns. *IEEE Journal of Biomedical and Health Informatics*, *21*(1), 184–192. https://doi.org/10.1109/JBHI.2015.2490798

Peet, J., & Brown, G. C. (2006). Diabetic retinopathy in a multi-ethnic cohort in the United States. *Evidence-Based Ophthalmology*, *7*(4), 192–193. https://doi.org/10.1097/01.ieb.0000212053.47890.78

Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional Neural Networks for Diabetic Retinopathy. *Procedia - Procedia Computer Science*, *90*(July), 200–205. https://doi.org/10.1016/j.procs.2016.07.014

Priya, R., & Aruna, P. (2014). Automated diagnosis of age-related macular degeneration using machine learning techniques. *International Journal of Computer Applications in Technology*, *49*(2), 157–165. https://doi.org/10.1504/IJCAT.2014.060527

Radford, A., Metz, L., & Chintala, S. (2016). *U NSUPERVISED R EPRESENTATION L EARNING WITH D EEP C ONVOLUTIONAL*. 1–16.

Rahimy, E. (2018). Deep learning applications in ophthalmology. *Current Opinion in Ophthalmology*, *29*(3), 254–260. https://doi.org/10.1097/ICU.0000000000000470

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-Alone Self-Attention in Vision Models. *Advances in Neural Information Processing Systems*, *32*. https://github.com/google-research/google-research/tree/master/standalone_self_attention_in_vision_models

Raza, A., Khan, M. U., Saeed, Z., Samer, S., Mobeen, A., & Samer, A. (2021). Classification of Eye Diseases and Detection of Cataract using Digital Fundus Imaging (DFI) and Inception-V4 Deep Learning Model. *Proceedings - 2021 International Conference on Frontiers of Information Technology, FIT 2021*, 137–142. https://doi.org/10.1109/FIT53504.2021.00034

Rodriguez, M. A., AlMarzouqi, H., & Liatsis, P. (2022). Multi-label Retinal Disease Classification Using Transformers. *IEEE Journal of Biomedical and Health Informatics*, 1–13. https://doi.org/10.1109/JBHI.2022.3214086

Sarki, R., Ahmed, K., Wang, H., Zhang, Y., & Wang, K. (2022). Convolutional Neural Network for Multi-class Classification of Diabetic Eye Disease. *EAI Endorsed Transactions on Scalable Information Systems*, *9*(4). https://doi.org/10.4108/eai.16-12-2021.172436

Shinde, R. (2021). Glaucoma detection in retinal fundus images using U-Net and supervised machine learning algorithms. *Intelligence-Based Medicine*, *5*, 100038. https://doi.org/10.1016/j.ibmed.2021.100038

Shorten, C., & Khoshgoftaar, T. M. (2019). deep. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Sonali, Sahu, S., Singh, A. K., Ghrera, S. P., & Elhoseny, M. (2019). An approach for de-noising and contrast enhancement of retinal fundus image using CLAHE. *Optics and Laser Technology*, *110*, 87–98. https://doi.org/10.1016/j.optlastec.2018.06.061

Soydaner, D. (2022). Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, *34*(16), 13371–13385. https://doi.org/10.1007/s00521-022-07366-3

Suganyadevi, S., Seethalakshmi, V., & Balasamy, K. (2022). A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval*, *11*(1),

19–38. https://doi.org/10.1007/s13735-021-00218-1

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A Survey on Deep Transfer Learning Chuanqi. In *27th International Conference on Artificial Neural Networks* (Vol. 11141, Issue November). Springer International Publishing. https://doi.org/10.1007/978-3-030-01424-7

Tan, J. H., Bhandary, S. V., Sivaprasad, S., Hagiwara, Y., Bagchi, A., Raghavendra, U., Krishna Rao, A., Raju, B., Shetty, N. S., Gertych, A., Chua, K. C., & Acharya, U. R. (2018). Age-related Macular Degeneration detection using deep convolutional neural network. *Future Generation Computer Systems*, *87*, 127–135. https://doi.org/10.1016/j.future.2018.05.001

Tang, L., Niemeijer, M., Reinhardt, J. M., Garvin, M. K., & Abramoff, M. D. (2013). Splat feature classification with application to retinal hemorrhage detection in fundus images. *IEEE Transactions on Medical Imaging*, *32*(2), 364–375. https://doi.org/10.1109/TMI.2012.2227119

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., & Dosovitskiy, A. (2021). MLP-Mixer: An all-MLP Architecture for Vision. *Advances in Neural Information Processing Systems*, *29*, 24261–24272.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Wassel, M., Hamdi, A. M., Adly, N., & Torki, M. (2022). *Vision Transformers Based Classification for Glaucomatous Eye Condition Vision Transformers Based Classification for Glaucomatous Eye Condition*. *August*.

World Health Organisation. (2019). World report on vision. *World Health Organisation*, *214*(14), 180–235. https://www.who.int/publications-detail/world-report-on-vision

APPENDICES

Appendix A: Sample codes

embedded parallel Convolution layers

```
image_size_tuple = interpret_image_size(image_size)
assert (image_size_tuple[0] % patch_size == 0) and (
    image_size_tuple[1] % patch_size == 0
), "image_size must be a multiple of patch_size"
x = tf.keras.layers.Input(shape=(image_size_tuple[0], image_size_tuple[1], 3))
#================Parallel conv layer=====================
y2 = tf.keras.layers.Conv2D(filters=16,kernel_size=3,padding="valid",name="par_cnn_0",)(x)
y2 = tf.keras.layers.Conv2D(filters=256,kernel_size=5,padding="valid",name="par_cnn_1",)(y2)
y2 = tf.keras.layers.Conv2D(filters=hidden_size,kernel_size=5,padding="valid",name="par_cnn_2",)(y2)

y2 = tf.keras.layers.Reshape((y2.shape[1] * y2.shape[2], hidden_size),name="reshape_par_init")(y2)
y2 = tf.keras.layers.MaxPooling1D(pool_size=y2.shape[1], strides=None, padding="valid")(y2)
#============================Division to patches=========================
patch_hidden_size = hidden_size # Original was equal to hidden_size
y = tf.keras.layers.Conv2D(
    filters=patch_hidden_size,
    kernel_size=patch_size,
    strides=patch_size,
    padding="valid",
    name="embedding",
)(x)
y = tf.keras.layers.Reshape((y.shape[1] * y.shape[2], patch_hidden_size),name="patch_embed_to_1D")(y)

y = layers.ClassToken(name="class_token")(y)
y = layers.AddPositionEmbs(name="Transformer/posembed_input")(y)
#==================Loop that adds transformer layers=====================================
for n in range(num_layers):
    y, _ = layers.TransformerBlock(num_heads=num_heads,mlp_dim=mlp_dim,dropout=dropout,name=f"Transformer/encoderblock_{n}",)(y)
    concat = tf.keras.layers.Concatenate(axis=1)([y, y2])
    y = concat

y = tf.keras.layers.LayerNormalization(
    epsilon=1e-6, name="Transformer/encoder_norm"
)(y)
y = tf.keras.layers.Lambda(lambda v: v[:, 0], name="ExtractToken")(y)
if representation_size is not None:
    y = tf.keras.layers.Dense(
        representation_size, name="pre_logits", activation="tanh"
    )(y)
if include_top:
    y = tf.keras.layers.Dense(classes, name="head", activation=activation)(y)
return tf.keras.models.Model(inputs=x, outputs=y, name=name)
```

ViT model with custom embedding

```
    image_size_tuple = interpret_image_size(image_size)
    assert (image_size_tuple[0] % patch_size == 0) and (
        image_size_tuple[1] % patch_size == 0
    ), "image_size must be a multiple of patch_size"
    x = tf.keras.layers.Input(shape=(image_size_tuple[0], image_size_tuple[1], 3))
    #============================Division to patches==========================
    patch_hidden_size = hidden_size*4
    y = tf.keras.layers.Conv2D(
        filters=patch_hidden_size,
        kernel_size=patch_size,
        strides=patch_size,
        padding="valid",
        name="patch_embed",
    )(x)
    y = tf.keras.layers.Reshape((y.shape[1] * y.shape[2], patch_hidden_size),name="patch_embed_to_1D")(y)
    #=======================Custom embedding=================================
    y = tf.keras.layers.Conv1D(filters=hidden_size*4,kernel_size=(5),padding="same",name="embed_conv1")(y)
    y = tf.keras.layers.Conv1D(filters=hidden_size*2,kernel_size=(5),padding="same",name="embed_conv2")(y)
    y = tf.keras.layers.Dense(units=hidden_size*2,name="embed_dense1")(y)
    y = tf.keras.layers.Dense(units=hidden_size,name="embed_dense_final")(y)
    #========================================================================
    y = layers.ClassToken(name="class_token")(y)
    y = layers.AddPositionEmbs(name="Transformer/posembed_input")(y)
    for n in range(num_layers):
        y, _ = layers.TransformerBlock(
            num_heads=num_heads,
            mlp_dim=mlp_dim,
            dropout=dropout,
            name=f"Transformer/encoderblock_{n}",
        )(y)
    y = tf.keras.layers.LayerNormalization(
        epsilon=1e-6, name="Transformer/encoder_norm"
    )(y)
    y = tf.keras.layers.Lambda(lambda v: v[:, 0], name="ExtractToken")(y)
    if representation_size is not None:
        y = tf.keras.layers.Dense(
            representation_size, name="pre_logits", activation="tanh"
        )(y)
    if include_top:
        y = tf.keras.layers.Dense(classes, name="head", activation=activation)(y)
    return tf.keras.models.Model(inputs=x, outputs=y, name=name)
```

Custom ViT Model building

```
from tensorflow.keras.layers import Flatten, BatchNormalization, Activation
from tensorflow.keras.regularizers import l2
from vit_keras import vit

import sys
sys.path.append('C:/Users/yared/Desktop/Olyad/FINAL TRAINING/Custom - new Training/transformers')
from cvit.vitcustom_parallel import *

# Available models: vit_custom_b16, vit_custom_b32

vit_model = vit_custom_b32(
    image_size=224,
    activation='softmax',
    pretrained=True,
    include_top=False,
    pretrained_top=False,
    classes = 6,
    weights = 'imagenet21k')

x = vit_model.output

# custom MLP Head

x = Flatten()(x)
x = BatchNormalization()(x)
x = Dense(256,activation='gelu', kernel_regularizer=l2(0.001))(x)
x = BatchNormalization()(x)
x = Dense(128,activation='gelu', kernel_regularizer=l2(0.001))(x)
x = BatchNormalization()(x)
x = Dense(64, activation='gelu', kernel_regularizer=l2(0.001))(x)
x = BatchNormalization()(x)
x = Dense(32, activation='gelu', kernel_regularizer=l2(0.001))(x)
x = BatchNormalization()(x)

model_out = Dense(6, activation='softmax')(x)
model = Model(inputs=vit_model.input, outputs=model_out)

model.summary()
```

Appendix B: Data Augmentation Algorithm

Geometric Augmentation

Start:

1. Import the required modules

2. Create an instance of the ImageDataGenerator class and assign it to the variable datagen.

3. Set the desired augmentation parameters for datagen:

    a. horizontal_ flip = True : randomly flip the images horizontally.

    b. vertical_flip = True: randomly flip the images vertically.

    c. rotation_range: random rotation of images by a random degree between -40 and 40.

End

Photometric Augmentation

Start:

1. Import the required modules

2. Define the function alter_contrast that randomly adjusts contrast of the input image and returns the adjusted image

    function alter_contrast(image):

      return adjusted_image

3.Define the function alter_brightness that randomly alters the brightness of the input image and then returns the adjusted image

    function alter_brightness(image):

      return adjusted_image

4. Define the function alter_saturation that randomly alters the saturation by scaling the saturation channel, and returns the adjusted image in BGR color space

    function alter_saturation(image):

      return adjusted_image

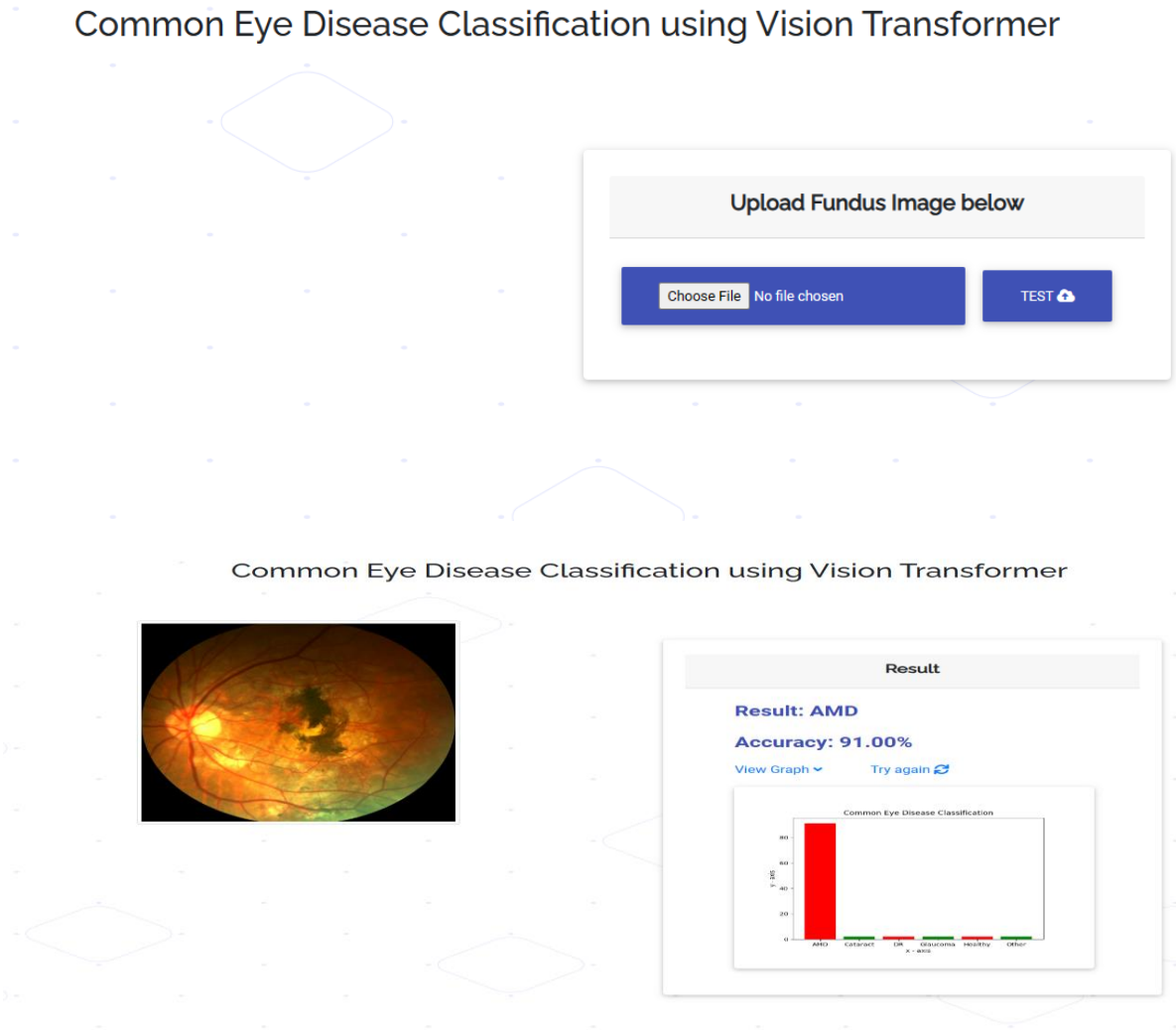
5: Define the function blur that applies gaussian blur to the input image with a randomly chosen kernel size and returns the resulting blurred image.

    function blur(image):

      return blurred_image

END

Appendix C: Sample Retinal Disease Classification Demo



```
from PIL import Image, ImageOps
import os
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.optimizers import Adam
import numpy as np

def process_img(img):
    def custom_loss(y_true, y_pred):
        return tf.keras.losses.categorical_crossentropy(y_true, y_pred, label_smoothing=0.1)
    model = keras.models.load_model('D:/demo_retinal_disease_classification/saved_model.h5',
                                    custom_objects={'custom_loss': custom_loss})
    model.compile(optimizer=Adam(0.0001), loss=custom_loss, metrics=['accuracy'])

    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
    image = Image.open(os.path.dirname(__file__) + '/test/' + img)
    size = (224, 224)
    image = ImageOps.fit(image, size, Image.ANTIALIAS)
    image_array = np.asarray(image)
    normalized_image_array = (image_array.astype(np.float32) / 255.0)
    data[0] = normalized_image_array
    prediction = model.predict(data)
    pred_new = prediction[0]
    pred = max(pred_new)

    print(pred_new)
    index = pred_new.tolist().index(pred)
    import matplotlib
    import matplotlib.pyplot as plt
    matplotlib.use('Agg')

    left = [1, 2, 3, 4, 5, 6]
    height = pred_new.tolist()
    new_height = []
    for i in height:
        new_height.append(round(i, 2) * 100)

    print(height)

    print(new_height)
    tick_label = ['AMD', 'Cataract', 'DR', 'Glaucoma', 'Healthy','Other']
    plt.clf()
    plt.bar(left, new_height, tick_label=tick_label,
            width=0.8, color=['red', 'green'])
    plt.xlabel('x - axis')
    # naming the y-axis
    plt.ylabel('y - axis')
    # plot title
    plt.title('Common Eye Disease Classification')
    plt.savefig(os.path.dirname(__file__) + '/output/graph.png')
    plt.show()
    result = []

    if index == 0:
        result.append("AMD")
    elif index == 1:
        result.append("Cataract")
    elif index == 2:
        result.append("DR")
    elif index == 3:
        result.append("Glaucoma")
    elif index == 4:
        result.append("Healthy")
    elif index ==5:
        result.append("Other")

    accuracy = round(pred, 2)
    result.append("-")
    result.append(accuracy * 100)

    return result
```