Effects of Image Augmentation on Multi-Class Retinal Disease Prediction

Submitted By

MINHAJUL ISLAM SHITOL

221-35-887

Department of Software Engineering

Daffodil International University

Supervised by

MD FAZLA ELAHE

Assistant Professor & Associate Head

Department of Software Engineering

Daffodil International University

A thesis submitted in partial fulfillment of the requirement for the degree of

Bachelor of Science in Software Engineering

Spring-2024

Effects of Image Augmentation on Multi-Class Retinal Disease Prediction

MINHAJUL ISLAM SHITOL

Bachelor of Science

DAFFODIL INTERNATIONAL UNIVERSITY

APPROVAL

This thesis titled on "**Effects of Image Augmentation on Multi-Class Retinal Disease Prediction**", submitted by **MINHAJUL ISLAM SHITOL (ID: 221-35-887)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Fazla Elahe

____________________________ **Chairman**

Dr. Fazla Ealhe
Assistant Professor & Associate Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

____________________________ **Internal Examiner 1**

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

____________________________ **Internal Examiner 2**

Dr. Shabnom Mustary
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

____________________________ **Internal Examiner 3**

Md. Rajib Mia
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

____________________________ **External Examiner**

Mohammod Abul Kashem, PhD
Professor
Department of Computer Science and Engineering
DUET, Bangladesh

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	MINHAJUL ISLAM SHITOL
Date of Birth	:	08 May, 2003
Title	:	Effects of Image Augmentation on Multi-Class Retinal Disease Prediction
Academic Session	:	Fall 2025

I declare that this thesis is classified as:

☑	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☑	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Thesis is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the thesis for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the thesis for academic exchange.

Certified by:

Shitol	Fazla Elahe
(Student's Signature)	(Supervisor's Signature)
MINHAJUL ISLAM SHITOL	MD FAZLA ELAHE
Student ID: 221-35-887	Name of Supervisor MD FAZLA ELAHE
Date: 27/11/2025	Date: 27/11/2025

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR's DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Bachelor of Science/ Master of Science.

Fazla Elahe

(Supervisor's Signature)

Full Name : MD FAZLA ELAHE

Position : Assistant Professor & Associate Head

Date : 27 November, 2025

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

Shitol

(Student's Signature)

Full Name : MINHAJUL ISLAM SHITOL

ID Number : 221-35-887

Date : 27 November, 2025

Effects of Image Augmentation on Multi-Class Retinal Disease Prediction

MINHAJUL ISLAM SHITOL

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science/Master of Science

Department of Software Engineering (Major in Data Science)

DAFFODIL INTERNATIONAL UNIVERSITY

November, 2025

ACKNOWLEDGEMENT

I've always been interested in how machines can learn from images, and that interest gradually pointed me toward retinal diseases and deep learning. It has been difficult but also incredibly rewarding to attempt to develop models that could aid in early detection of eye diseases for me. I thank the Almighty for giving me health, strength and patience to continue this work to the end.

Thank you to my family and parents for always being there with me. My parents who have loved, supported and encouraged me tirelessly through my life– I own them more than I can ever repay. I owe much of my drive and perseverance for getting to this point and finishing this thesis to their prayers and faith in me.

I must also acknowledge my cordial thanks to Dr. Imran Mahmud, HoD, SE & all of my Hon'ble teachers in Daffodil International University. Their advice, knowledge and constant support has played an important role throughout my academic journey.

A word of thank goes to my guide, Fazla Elahe. The guidance he has provided me spanning the entire duration, and constructive comments along the way have been particularly helpful at each stage of this work. For your continuous guidance, great ideas and motivation for keeping myself focused and alert during the hard times of this thesis.

Finally, I am really grateful to my friends and classmates in DIU. Their support, late night talks and never-ending motivation made this trip not just easier but totally enjoyable. I am very grateful for the incredible support they have given to me, and their good vibes during this episode.

ABSTRACT

Small datasets and device-to-device variations in image quality frequently limit deep learning for color fundus images. In this thesis, I use a nine-class dataset of 4,500 images (500 per class) to examine the impact of image augmentation strength on multi-class retinal disease prediction. Four convolutional models are used in the experiments: EfficientNet-B4, MobileNetV3-Large, DenseNet-121, and Custom CNN.

Here every model is trained with four different augmentation settings. Those are: no augmentation, mild, strong, and an advanced. All runs share the same train–validation–test split and training setup, so differences in performance can be linked mainly to the model and the augmentation level.

The results show that the effect of augmentation is model-dependent. The Custom CNN performs best without augmentation, while DenseNet-121 reaches its peak with mild augmentation. EfficientNet-B4 performs best with strong augmentation. MobileNetV3-Large benefits the most from heavy augmentation: with the advanced setting, it achieves the highest overall performance, with accuracy around 0.851 and macro-F1 around 0.849.

These findings suggest that augmentation strength should not be chosen as a single fixed recipe for all backbones. Instead, it needs to be tuned per model when designing retinal disease classification systems based on deep learning.

Table of Contents

CHAPTER 1
INTRODUCTION

Introduction

Retinal diseases are one of the major causes of vision problems and many progress without clear any early symptoms. Fundus photography is mainly used for screening, but manual review is very slow. As screening programs grow, the automated support tools are becoming more valuable.

Deep learning models handle retinal images well, but they usually require large and diverse datasets. Medical datasets often fall short of that. The dataset we used here includes 4,500 images across nine classes and we sorted (500 image) per class, which is balanced but still small for modern CNNs.

Image augmentation can be a method of diminishing overfitting. Basic levels of augmentation (such as, flipping or changing brightness) will also aid in reduction of overfitting; however, greater amounts of augmentation will provide more change to the images' contents. The performance impact of the type or amount of augmentation performed will vary from model to model.

To support this investigation, four different neural network architectures (Custom CNN, DenseNet-121, MobileNetV3-Large, and EfficientNet-B4) were trained on four conditions of image augmentation (None, Mild, Strong, and Advanced). The objective was to examine how the amount or strength of image augmentation, is able to promote improved accuracy in the classification of Retina as a multi-class disease.

The outcomes of this study demonstrated the relationship between the augmentation method(s), were dependent on the specific model being trained. For example, MobileNetV3 significantly improved its classification accuracy when provided with aggressive augmentations, whereas DenseNet-121 primarily benefitted from less aggressive (mild) augmentations. Custom CNN performed best with very minimal augmentations. The findings suggest, therefore, that augmentation methods will require individual adjustment for each unique model to yield optimal results.

1.2 Background

Color fundus photography which is one of the common methods for examining the retina. It shows the blind spot, macula, vessels, and surrounded tissue, which help identifying diseases like the diabetic retinopathy, glaucoma, and macular disorders. Even though valuable, the quality of those images varies relying on lighting, camera type, and patient collaboration.

Deep learning has been becoming a strong choice for analyzing fundus images. CNN-based models such as Dense Net, Mobile Net, and Efficient Net normally work well, especially when beginning from ImageNet pretraining. Here the difficulty is that they still need diverse data to generalize appropriately. There many medical datasets do not offer enough variation.

Augmentation has been widely used to compensate for this problem. The types of techniques for augmentation range from simple flipping to more complicated transformations such as color, shape, and noise. Although augmentation will improve the strength of the data set, it is also possible that too much distortion of the data may obscure important retinal characteristics. This is particularly true in diseases where the presence of thin or differently-shaped patterns is very significant.

Due to these reasons, there is a critical need to test the effects of augmentations on retinal images. A multi-class retinal disease classification represents an excellent example of this process. Each category of disease has a unique pattern, and individual categories may respond differently to parameters such as augmentations.

1.3 Problem Statement

Although Deep Learning Models (DLMs) are good at analysing retinal images, the overall performance of these models was lower when trained with a limited or uniform data set of images. When using a balanced dataset with 4,500 images, there is still limited colour variance, contrast, and image conditions, increasing the risk ofOverfitting and affecting the ability to classify multiple classes of diseases; All of these diseases display very similar visual characteristics.

Augmentation is one solution to help reduce this issue, but its effect may vary. Some DLMs show improvement when a small degree of augmentation is applied, while others must be augmented to a greater extent. However, in some instances, a large amount of augmentation results in inferior performance by altering structures necessary to identify a specific retinal image.

The main problem addressed in this study is understanding how different augmentation strengths affect multi-class retinal disease classification across several CNN architectures. Since each model has its own design and capacity, the same augmentation policy may not work equally well for all of them.

This study aims to identify which combinations of model and augmentation yield the most reliable performance for nine retinal disease classes, and to determine whether stronger augmentation always leads to better generalization or not.

1.4 Research Gaps

Most deep-learning studies on retinal images use augmentation, but many treat it as a routine preprocessing step rather than a variable that can influence performance. The choice of augmentation strength is often not discussed, and comparisons across different levels are limited.

Another gap is model-specific behavior. Architectures such as MobileNet, DenseNet, and EfficientNet answer in a different way to spatial and color transformations, but a little works directly compare them under controlled augmentation settings. As a result, it is not so cleared whether the certain models benefit more from mild or aggressive transformations.

Finally, multi-class retinal disease classification is less explored than binary tasks. Subtle differences between classes—such as between macular scar, myopia, and central serous chorioretinopathy—make the problem sensitive to both model design and augmentation choices. Current literature offers limited guidance on how augmentation affects these finer distinctions.

This study addresses these gaps by systematically evaluating four augmentation levels across four CNN architectures on a balanced nine-class dataset.

1.5 Objectives

The main goal of this thesis is to understand how far we can push deep learning models for retinal disease classification when the dataset is small but balanced, and when augmentation is carefully controlled. Instead of only training one model with one type of augmentation, the work is designed to compare several combinations in a systematic way.

First, the study aims to evaluate four convolutional architectures—CustomCNN, DenseNet-121, MobileNetV3-Large, and EfficientNet-B4—on the same nine-class retinal dataset. All models use the same training–validation–test split so that differences in performance can be traced back to the model design and the augmentation strategy rather than to the data split.

Second, the study will examine the effects of different strengths of augmentation. Four sets of conditions have been established by the researchers for this study, which include:

(A) no augmentation; (B) slight changes to images; (C) moderate image transformations; (D) complex policy with combined image transformations.

Controlling for all other variables allows researchers to evaluate whether incrementing augmentation strength yields steady improvement in performance or whether there is a level after which performance will actually begin to fall off as augmentation strength continues to increase.

Additionally, while researchers will be evaluating overall accuracy across all diseases, they will also be looking closely at how augmentation can impact performance on each disease. Within the dataset, two diseases, glaucoma and macular scars, are visually less distinct and obvious than some others, and the goal is to see if augmentation can improve performance for these more subtle disease classes vs the less subtle types of disease classes.Finally, the thesis aims to identify a practical "best choice" for this dataset: a specific model and augmentation setup that deliver strong and stable performance. This can serve as a baseline for future work on retinal disease prediction using similar image collections.

1.6 Motivation

This work started from a natural inspection: in many eye-care center, doctors are always on under pressure, but the number of patients is kept growing day by day. Fundus cameras are available in many hospitals but making those images into a reliable decision still depends on a human. Sitting in front of a screen and checking every cases. Deep learning can help, but in practice it is not always clear how to train these models in a way that is both stable and realistic for medical data.

When I began experimenting with this nine-class retinal dataset, it was clear that the amount of data was not very large by deep learning standards. There are we have 4,500 images in total, which is good for a beginning project, but still little for four different neural networks. This obviously raised the question: how much can we gain from data augmentation, and what is the limit where it will stops helping and starts hurting?

Another motivation came from the behavior I saw during training. Some models improved a lot when I increased augmentation strength, while others became unstable or even lost performance. MobileNetV3, for example, responded very well to advanced augmentation, whereas the custom CNN seemed to prefer simpler training conditions. This highlighted that "just turn on augmentation" is not always the right answer; the effect clearly depends on the model.

Because of these experiences, I wanted the thesis to do more than just report a single accuracy value. The inspiration is to document, in a clear and honest way that how the four different models react to the four levels of augmentation. Also, on the same retinal dataset. The hope is that these people who want to build similar systems and need practical guidance instead of trial-and-error. So that the findings will be useful for others.

1.7 Summary

This chapter will introduce the total context and purpose of the thesis. It began with the idea of that retinal diseases are common and often progress silently, while the fundus photography is main tool used to detect them. At the same time, manually grading is so slow and depends on the expert availability, which creates the opportunity for deep learning models to help with screening.

The chapter also discussed why training these models is not straightforward. Even though the dataset used here is balanced with 4,500 images across nine classes, it is still relatively small for modern CNNs. This raises concerns about the overfitting and motivates the uses of image augmentation. However, the effect of augmentation is not guaranteed to be positive for every model, and strong transformations can sometimes damage important retinal patterns.

based totally on those lookouts, the chapter outlined the main targets and the incentive for take a look at,to evaluate 4 different architectures under 4 augmentation degrees. additionally to recognize how these picks have an impact on equally ordinary overall performance and class-smart behavior. The paintings is driven by means of practical questions that got here up at some stage in experimentation, specially the version-established reaction to augmentation energy. the following chapter reviews associated studies and suggests how this undertaking suits in the current literature on retinal photograph evaluation and deep getting to know.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this part of the paintings, I study what has already been done with retinal photographs, deep getting to know fashions, and information augmentation. The goal is easy: to peer how other human beings approached similar troubles and to recognize where my undertaking is comparable or special.

Retinal images are being used for a long time to find diseases such as diabetic retinopathy and glaucoma. Earlier systems are being usually depended on hand-crafted features. Researchers designed rules based on blood vessel shape, bright or dark spots, or measurements around the optic disc. These methods could work on the dataset they were built on, but they often struggled when the images came from a different camera or hospital.

With deep learning, this has changed. Convolutional networks can learn the features directly from pixels, so there is fewer need for manual feature design. Many recent papers fine-tune pre-trained models, such as ResNet, DenseNet, MobileNet or EfficientNet, on retinal datasets. Most of them also apply some form of image augmentation, mainly to deal with the small size and limited diversity of medical data.

However, when reading the literature, I noticed that augmentation is usually described in only one or two lines. The strength of the transformations, and how they interact with different backbones, is rarely analyzed in detail. This is the gap that motivates my experiments. The next sections briefly review previous work on retinal image analysis, deep learning backbones, and augmentation in medical imaging, focusing only on the parts that are relevant for this thesis.

2.2 Previous Literature

Work on retinal image analysis began a long before deep learning became famous. Previous systems concentrated on specified tasks such as finding microaneurysms, exudates, or segmenting blood vessels. Majority of these methods used hand-crafted features. Researchers designed filters, edge detectors, and texture measures, and then combined those all with standard classifiers like SVMs or random forests. These approaches could work well on small, controlled datasets, but they are often sensitive to changes in illumination, camera type, or noise.

With the success of convolutional neural networks in natural image classification, deep learning quickly moved into the retinal domain. Many studies applied architectures like VGG-style networks, ResNet, DenseNet, and later EfficientNet to problems such as diabetic retinopathy grading, glaucoma screening, and AMD detection. A common pattern in these works is to take a model pre-trained on ImageNet and fine-tune it on retinal images. This transfer learning strategy usually gives better performance than training from scratch, especially when the medical dataset is not very large.

Several papers report strong results on binary or low-class problems, for example “disease vs. healthy” or a small set of severity levels. Multi-class settings with more than four or five categories are less common, and when they do appear, they are often focused on a single disease family (for example, only diabetic retinopathy stages). Studies that cover a broader range of retinal conditions, like the nine-class setup used in this thesis, are fewer in number.

Augmentation is present in almost all deep learning papers on retinal images, but it is usually treated as part of the standard training recipe. Typical transformations include horizontal flips, small rotations, and modest brightness or contrast changes. Some recent works also experiment with stronger methods such as mixup or cutmix, especially when they want to improve robustness. However, most of these studies report only the final performance of their chosen setup. They do not systematically compare different augmentation strengths or analyze how each backbone responds to these choices.

There is also a growing interest in lighter models, such as MobileNet and other efficient architectures, for deployment on lower-power devices. These models are attractive for screening programmes, but the published results often focus on one backbone at a time. Direct comparisons between several backbones under the same augmentation and training settings on a multi-class retinal dataset are still limited.

Overall, the literature shows that deep learning is effective for retinal image classification and that augmentation is widely used, but it does not clearly answer how augmentation strength should be chosen for different models and disease sets. This is the space where the present work aims to contribute, by running controlled experiments across four architectures and four augmentation levels on a nine-class dataset.

2.3 Summary

This chapter briefly reviewed how retinal images have been used in earlier research and how the field has moved from hand-crafted features to deep learning. Older methods relied on manually designed measurements and classical classifiers, which often worked only under specific conditions and did not transfer well to new datasets.

With the arrival of CNNs and transfer learning, models such as ResNet, DenseNet, MobileNet, and EfficientNet became the standard choice for retinal image analysis. Most recent studies report good performance, especially on binary or low-class problems, and almost all of them use some form of image augmentation to fight overfitting on small medical datasets.

However, the review also highlighted a gap: augmentation is usually described very briefly, and its strength is rarely treated as a parameter to be studied on its own. Direct comparisons across multiple backbones and multiple augmentation levels on a multi-class retinal dataset are limited.

The work in this thesis is designed to address that point. By training four different models under four augmentation settings on a nine-class retinal disease dataset, it aims to provide a clearer picture of how augmentation strength interacts with architecture choice. The next chapter explains the dataset, preprocessing, models, and training setup used to run these experiments.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter has explained how the experiments in this thesis were carried out. It describes the dataset, the preprocessing steps, the augmentation settings, the models, and the training procedure. The goal is to give a clear picture of what was actually done, so that the results in the next chapter are easy to understand and, in principle, possible to reproduce.

All experiments are based on the same nine-class retinal dataset with 4,500 color fundus images. Those data are structured into folders by the name of disease, and each of those folders contains with 500 images. The images are loaded using a custom PyTorch dataset class and then those are split into training, validation, and test sets while keeping the class balanced intact.

Before being fed into the models, each image will go through a set of preprocessing steps, including resizing and contrast enhancement. On top of this, different augmentation policies are applied depending on the experiment: no augmentation, mild, strong, or advanced. These policies are implemented with Albumentations and are the main factor being studied in this work.

There four convolutional backbones are been used: a Custom CNN, DenseNet-121, MobileNetV3-Large, and EfficientNet-B4. All models are trained and evaluated under standard framework using PyTorch also with early stopping, learning-rate scheduling, and class-weighted loss. The relaxation of this bankruptcy is going via those components in greater element, beginning from the dataset after which moving to augmentation, version design, and the training setup.

3.2 Overall Framework of the Approach

In this thesis, a simple and consistent workflow was created to allow any changes in performance to be attributable primarily to either the model or the augmentation level, with limited influence from external or other factors affecting the results.

The folder structure of the dataset serves as the basis of the initial step for the program to develop. The program will then generate an ordered list of all images, along with the classification associated with each image, followed by dividing those images into an 80-10-10 stratified random distribution of images to use during training, validation, testing. By using one random seed throughout the project, all experiments will allow the exact same images to be used during each trial.

For each run, I choose:

one backbone model (Custom CNN, DenseNet-121, MobileNetV3-Large, or EfficientNet-B4), and

one augmentation policy (None, Mild, Strong, or Advanced).

The chosen augmentation is applied only on the training set through Albumentations, while validation and test images only go through the basic preprocessing (resize, CLAHE, normalization). This keeps the evaluation clean and comparable.

All runs are handled by the same training loop. The loop takes care of loading batches, moving data to the GPU (when available), computing the loss, updating the model, and tracking metrics. Early stopping and a learning-rate programmer are used for avoid the wasting epochs when the model is stops improving.

At the end time when each run, the best model checkpoint (based on validation macro-F1) is reviewed on the test set. The script saves those accuracy, macro-F1, weighted-F1, and per-class statistics into the JSON file that been created. And then repeating this for all 4 models × 4 augmentation levels gives the 16 results that are later analyzed in the Chapter 4.

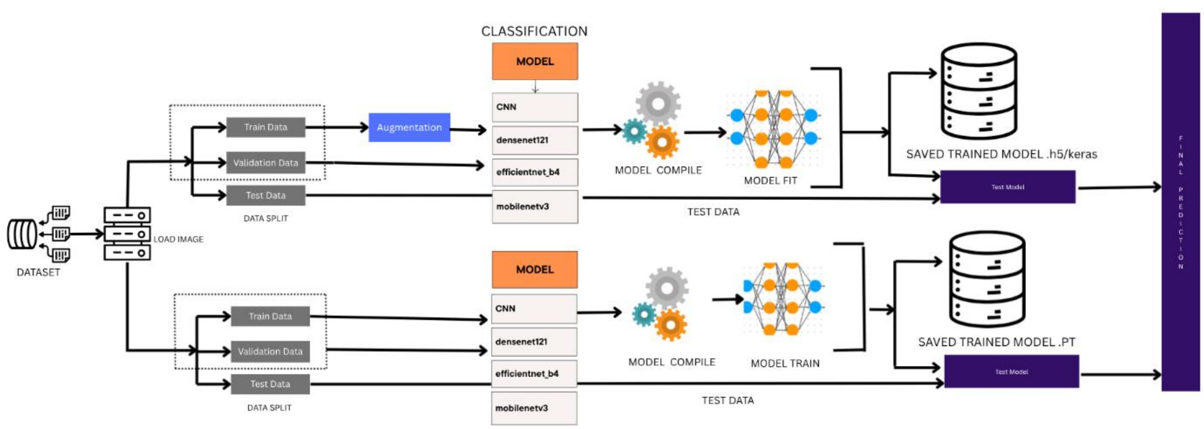


Figure 3.2: Synchronous Augmentation workflow for retinal eye diseases

3.3 Dataset and Splitting

For this work I used a retinal disease dataset that contains 4,500 color fundus images. The structure is very straightforward, there is one folder for every disease, and those each folder has 500 images. Those nine classes are:

Central Serous Chorioretinopathy

Diabetic Retinopathy

Disc Edema

Glaucoma

Healthy

Macular Scar

Myopia

Retinal Detachment

Retinitis Pigmentosa

In the code, the dataset class simply walks through these folders, reads the image paths, and assigns a numeric label based on the folder name. This creates a list of all images with their corresponding class index. Nothing fancy happens at this stage; it is just a clean way to map from directory names to integer labels.

For training and evaluate those models equally, I separate those data into training, validation, and test sets using an 80–10–10 ratio. Because every class has exactly 500 images, that means 400 images are per class for training, 50 for the validation, and 50 for testing. I used a stratified break up, so the class balance is preserved in all three units.

The same random seed is utilized in every time, so the cut up does no longer get modified among runs. that is very crucial because it ensures that any distinction in consequences comes from the version or the augmentation setting, not from a specific information split. After the indices are created, they're wrapped lower back into 3 separate dataset objects, every with its very own rework pipeline (schooling with augmentation, validation and check with out heavy randomness).

Retinal Class	Total Images	Train (80%)	Validation (10%)	Test (10%)
Central Serous Chorioretinopathy	500	400	50	50
Diabetic Retinopathy	500	400	50	50
Disc Edema	500	400	50	50
Glaucoma	500	400	50	50
Healthy	500	400	50	50
Macular Scar	500	400	50	50
Myopia	500	400	50	50
Retinal Detachment	500	400	50	50
Retinitis Pigmentosa	500	400	50	50
Total	4500	3600	450	450

Figure 3.3: Class-wise distribution of images and stratified 80/10/10 train–validation–test split.

3.4 Preprocessing and Color Handling

Before any model sees an image, it goes through a small but important preprocessing pipeline. The goal is to make the inputs consistent and clean, without doing anything too aggressive that might remove useful medical information.

All of the main experiments in this thesis are done with RGB fundus images. In the code, every image is first loaded with OpenCV and then its converted from BGR (OpenCV's default) to RGB. I also apply a CLAHE-based contrast enhancement step. This is done on the luminance channel so that local contrast is improved, but the overall color balance of the retina is still preserved. In exercise, this assists vessels, lesions, and edges stand out more clearly.

After that, every image has been rescaled to 224 × 224 pixels. This size was chosen because it is the common input resolution for multiple standard CNN backbones and keeps the memory and computing cost into control. The same target size is used for all models to keep the comparisons fair.

After the image has resized and enhanced, it is normalized. For the RGB images, Normalization tracks the usual ImageNet convention (subtracting a fixed mean and dividing by a fixed standard deviation per channel). This is beneficial since the pre-trained backbones that name as: (DenseNet-121, MobileNetV3-Large, EfficientNet-B4) have been originally trained on ImageNet with that normalization. And then we can say finally the image is converted into a PyTorch tensor. Its channel order is reorganized to pair the expected [C, H, W] format.

There is also an option in the code to load images in grayscale, with CLAHE applied directly to the single channel and a simpler normalization. This branch is kept for flexibility, but the main set of results reported in this thesis uses the RGB pipeline, since color information is often useful for distinguishing between different retinal conditions.

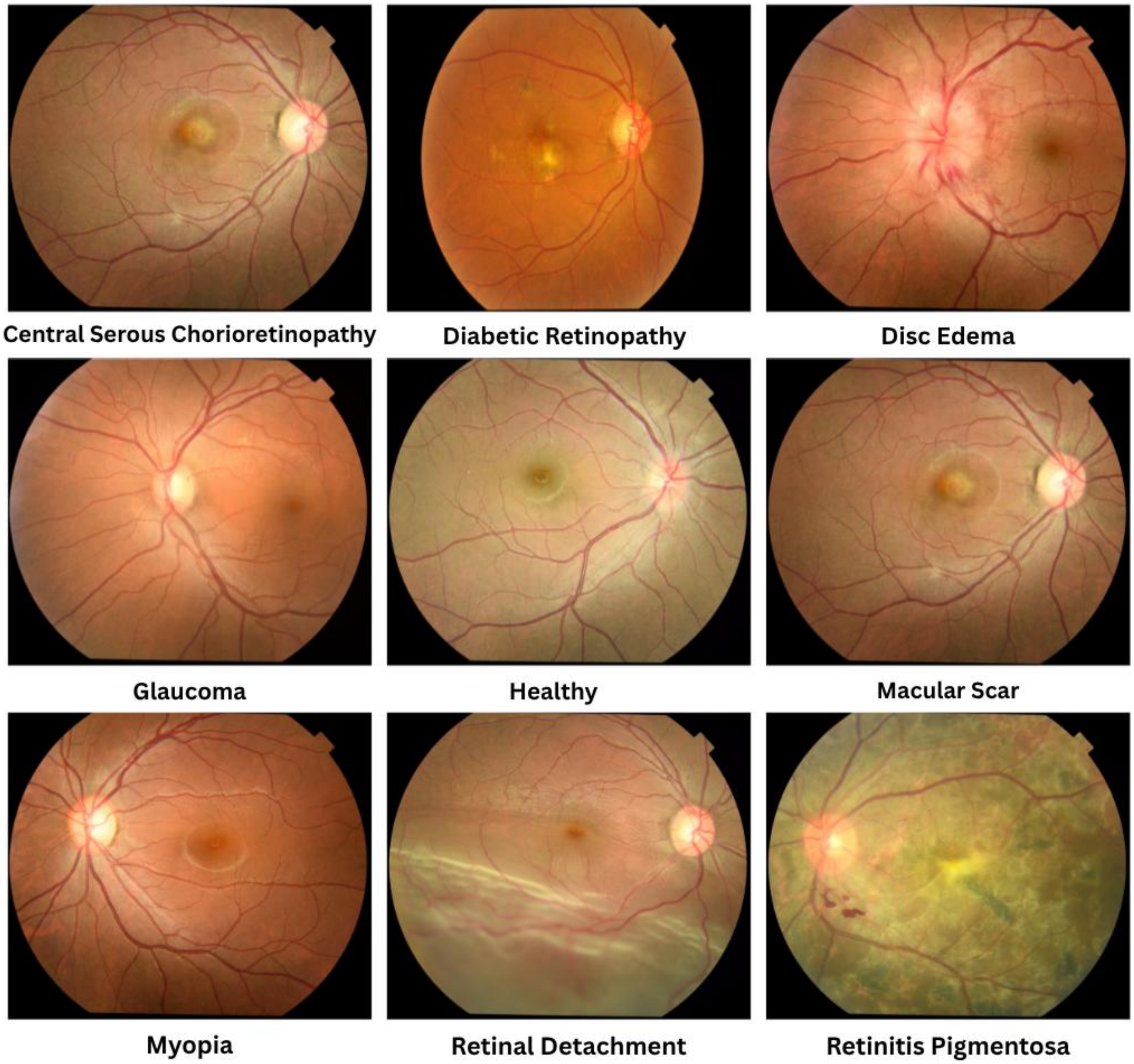


Figure 3.4: Example of data from different classes (Central Serous Chorioretinopathy, Diabetic Retinopathy, Glaucoma, Macular Scar, Myopia, Retinitis Pigmentosa)

3.5 Image Augmentation Strategies

A key part of this thesis is the way data augmentation is used. Instead of fixing a single set of transformations, I defined four different "levels" of augmentation and applied them across all four models. This makes it possible to see not only which model performs best, but also how sensitive each one is to changes in augmentation strength.

All augmentation pipelines are implemented using Augmentations and share a common base: every image is resized to the target input size and then normalized. On top of that base, different transformations are added depending on whether the run is marked as none, mild, strong, or advanced. These policies focus on realistic changes that a fundus camera might produce, such as flips, small rotations, brightness and contrast variation, blur, and noise. 【turn5file0†L17-L27】

The baseline setup ("no augmentation") only applies resizing, normalization, and conversion to a tensor. It represents the case where the model sees each training image exactly once, with no random variation. The other three setups gradually increase the amount and complexity of the transformations. Mild augmentation introduces only a few light changes, strong augmentation adds more geometric and color variation, and the advanced policy combines several effects such as blur choices, local dropout, and hue–saturation shifts. 【turn5file0†L29-L39】 【turn5file1†L9-L18】

Augmentation is carried out handiest to the education set. The validation and check units use the same primary preprocessing defined in advance however no random spatial or colour modifications. This way, the take a look at consequences replicate how the version behaves on solid, smooth pictures, at the same time as the training method nevertheless advantages from seeing greater various examples. the following subsections describe every augmentation stage in greater element.

3.5.1 Augmentation Ladder

On this work, I used 4 degrees of augmentation. I think about them as an "augmentation ladder": we start and not using a augmentation at the bottom and move up via Mild, Strong, and Advanced. The concept is to see how each step on this ladder influences the models.

Baseline (No Augmentation):

On the minimal step, the model fails to get any more alternate. The images long gone via to handiest the number one preprocessing: assessment improvement, resize to 224×224, normalization, and conversion to a tensor. There aren't any random flips, rotations, or shade modifications. This setup demonstrates how properly each backbone can do while it handiest sees the real dataset.

Mild Augmentation:

The next step adds light, safe changes. Here the image that is for training can be randomly flipped or rotated a little, and their brightness or contrast can be changed slightly. These edits are small on purpose. They mimic normal differences between fundus captures without distorting the anatomy too much. Mild augmentation mainly helps the model avoid memorizing very specific positions or lighting.

Strong Augmentation:

On third step, the changes become further visible. In extension to the mild changes, the images can rotate more, blurred slightly, or shifted in brightness and contrast to a strongest degree. This exposes the model to a wider variety of appearances. The risk here is that, if pushed too far, important patterns around the optic disc or macula may start to look less realistic.

Advanced Augmentation:

On the pinnacle of the ladder is the superior setup. This combines several adjustments from the strong stage and can encompass heavier coloration shifts or nearby dropout–fashion outcomes that eliminate small components of the picture. The aim is to definitely test how strong a model can end up while educated on quite varied inputs. In my experiments, this level helped MobileNetV3 loads, however it changed into no longer usually the first-rate preference for each model.

For all four levels, the rule is the same: augmentation is applied only to the training set. The validation and test images use only the basic preprocessing, with no random changes. This way, the comparison between augmentation levels is fair, and the test results always reflect performance on clean, stable images.

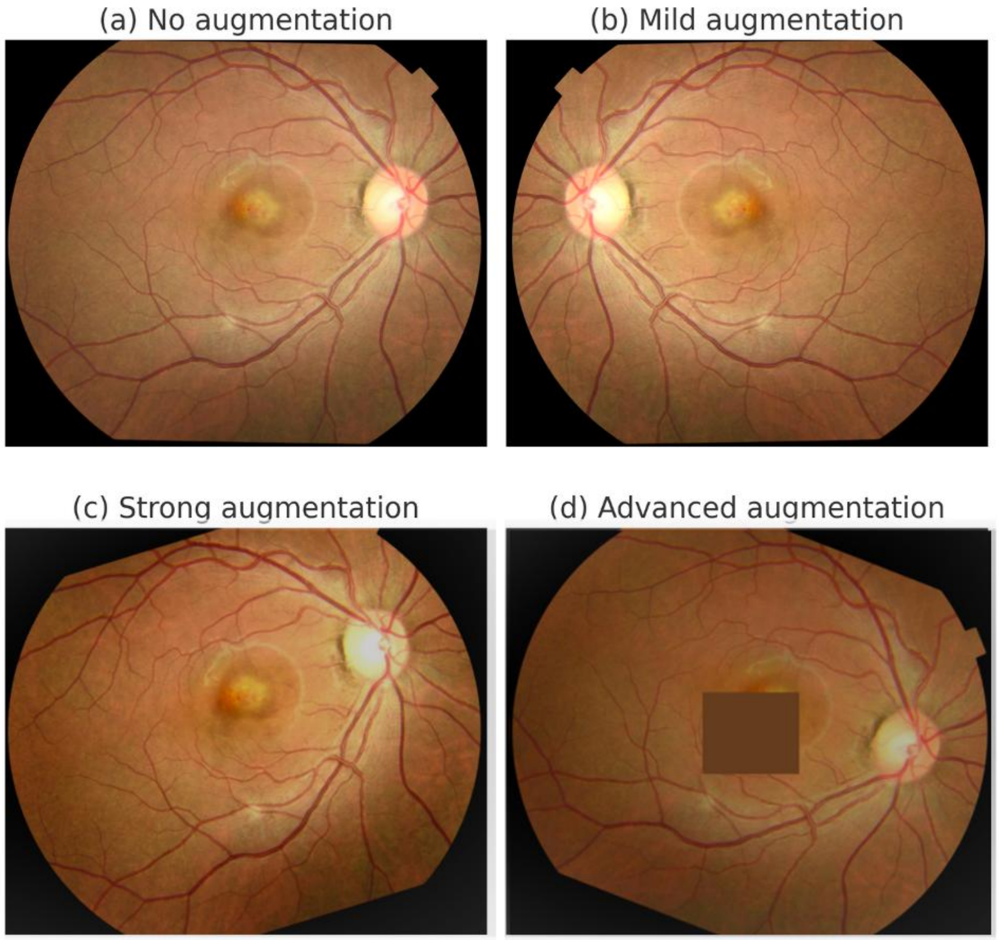


Figure 3.5: Example of the augmentation ladder applied to a retinal fundus image

3.6 Deep Learning Models

This thesis uses four different convolutional backbones. The idea was to mix one custom model with three popular pre-trained architectures, and then see how each of them reacts to the same dataset and the same augmentation ladder.

The first model is a Custom CNN that I implemented myself. It is built only for this project and does not use any external pretraining. This model is useful as a baseline to see what can be achieved with a simpler, task-specific design.

The other three models—DenseNet-121, MobileNetV3-Large, and EfficientNet-B4—are standard architectures that are widely used in computer vision. In my code, they are loaded with ImageNet weights and then adapted to predict the nine retinal classes. This transfer learning approach lets the models start from a good set of generic features and then specialize to fundus images.

All four models share the similar result layer style: they create logits for the nine classes. Also, the training loop has been applied a cross-entropy loss on top. They all are trained with the same optimizer type, planner, and early stopping logic. so, the main variances in performance come from their architecture. Then we can see how well they make use of the different augmentation strengths.

3.6.1 Custom CNN

The first model I used is a Custom CNN that I wrote myself for this project. I wanted one model that is fully mine, without any pre-trained weights, so I can see how a basic design performs on the nine retinal classes.

The idea of the model is straightforward: it takes a 224×224 RGB image and passes it through a few convolution layers, with batch norm and activation in between. I also added some residual blocks, so the network can pass information forward more easily and not "forget" useful features. As the image goes deeper in the network, the feature maps became smaller and at the same time the number of channels increase, so the model can move from simple patterns like edges to more complex retinal structures.

At the end, the model uses an adaptive average pooling layer to squeeze everything into a single feature vector, and then a small fully connected layer that outputs the scores for the nine classes. It is much lighter than DenseNet, MobileNetV3, or EfficientNet, and it works as a clear baseline to match how much the larger pre-trained models actually help.

3.6.2 DenseNet-121

In here the second model I used in this project is DenseNet-121. I chose it because it's a common backbone in many types image classification papers, and I wanted to see how it performs on my retinal dataset.

In my code, I loaded the pre-trained ImageNet version of DenseNet-121. Then i only change the last layer. So, that it outputs nine classes rather than original 1000. After that everything else in the architecture stays the same. Through training, I fine-tune the whole network on my fundus images, not just the last layer.

One unique thing about DenseNet is that it can connects multiple layers together, so later layers can reuse features from earlier ones. I did not change how this works internally; I just applied the standard execution. In my experiments, DenseNet-121 usually behaved like a strong, stable model, and also it gave me a good point of comparison against my Custom CNN and the lighter MobileNetV3.

3.6.3 MobileNetV3-Large

The third model that I used is MobileNetV3-Large. I chose it because it is a small and fast network, and I was curious how a light model would do on a kind of retinal dataset.

In the code, I loaded the ImageNet pre-trained version and only modify the last layer so that it outputs the nine classes. After that, I fine-tune the entire model on my fundus images, and used the same training setup as for the other models.

MobileNetV3 is designed to be efficient. It has fewer parameters and is easier to run on weaker hardware. Even though it is small, it still learned good features in my experiments. In fact, with the advanced augmentation setting, MobileNetV3 gave the best overall results among all the models I tested.

3.6.4 EfficientNet-B4

The last fourth model I used is EfficientNet-B4. I added it because it is known for having an excellent balance between the accuracy and model size.

Like the others pre-trained models, I started from the ImageNet version of EfficientNet-B4. Then change the last layer to output nine classes. After that, I fine-tune it on the same training set, with the similar preprocessing and augmentation ladder.

EfficientNet scales depth, width, and resolution in a symmetrical way. I did not modify this scaling; I just used the standard B4 variant. In my results, EfficientNet-B4 often came close to DenseNet-121, and with strong augmentation it performed very competitively.

3.7 Training and Evaluation Setup

For those all experiments, I always tried to keep the training setup as stable as possible. So that the main changes come from the model and the augmentation level. And not from a random change in settings.

At first, I run the code in PyTorch, using GPU when its available and CPU for fallback. Then I trained the model mainly on a machine with an NVIDIA GPU, but for make things flexible I added the CPU option also. The code can also run-on the CPU with the same logic, just slower.

For each run, I used:

Epochs: 40

Batch size: 16

Learning rate: 1e-4

Weight decay: 1e-4

Optimizer: AdamW

Scheduler: ReduceLROnPlateau (triggered by validation loss)

These values are coming from the default arguments in main.py. I let them all the same over 16 runs so that the comparison stands fair.

When we start the training, the loop will do the common steps: load the batch, then move it to the device, then run a forward pass, after that compute the cross-entropy loss, then backpropagate, and at last update the weights. Also, here I used gradient clipping (max norm 1.0) to avoid blasting gradients on any of those models. After all this at the end Validation will every epoch on the validation set. That uses no heavy augmentation.

An early stopping mechanism observes on the validation loss. If the model stops improving for multiple epochs (patience = 10), the training will be stopped. Then the best model weights will be saved. After that, I load this best checkpoint and review it once on the test set.

For last review, I also record accuracy, macro-F1, weighted-F1, and per-class precision, recall, and F1-score. These numbers are going to be saved into the JSON files for each run.

3.8 Implementation and Reproducibility

In this project i implemented the Python using PyTorch as the main deep learning framework. The code is structured in a simple way. So that every part of the pipeline has its own place. The main script (main.py) is mainly responsible for read the command-line arguments. Then set the device (CPU or GPU), then load the data, after load creating the model, and then start the training and evaluate the process.

The helping functions are kept inside the utils folder. The data loader, augmentation, and training utilities are applied in the files such as dataloder.py, augmentation.py, and evaluation.py. These files control tasks like reading an image from a folder, applying those augmentation policy, creating the data loaders, computing the metrics, saving reports, and managing early stopping.

All model definitions are stored in the model's folder. This comprises the Custom CNN and the three pre-trained backbones (DenseNet-121, MobileNetV3-Large, and

EfficientNet-B4). The file modelengine.py allows a single interface to load the accurate model based on the name approved from the command line. This makes things easy to switch between models without modifying the rest of the code.

The training formation, such as the name of the model, augmentation type, number of epochs, input size, and data path. It passed as arguments when running the script. For example, one typical run uses a command where I select the dataset path, the model (for example mobilenetv3) and the augmentation type (for example advanced). This style allows me to repeat the same experiment later. Or it just changes just one setting at a time.

To keep the results consistent, I fixed various elements. Random seeds are set for the Python, NumPy, and PyTorch so that the data split and weight initialization remain steady over runs. The train, validation, and test sets follow the same 80–10–10 stratified split in each experiment. For every single run, the best model checkpoint according to validation performance will be saved. The final test metrics will be written into a JSON file.

With this setup, anyone who has the same code, dataset, and environment configuration should be able to rerun these experiments and get very similar results.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the results of the experiments and discusses what they mean. In total, I ran sixteen different models by combining four backbones (Custom CNN, DenseNet-121, MobileNetV3-Large, EfficientNet-B4) with the four augmentation levels from the augmentation ladder (None, Mild, Strong, Advanced). All of them were trained on the same train–validation–test split, using the same training setup described in Chapter 3.

The main intention here isn't always just to reveal which version were given the very best accuracy, but to recognize a few things: how augmentation power impacts performance, which models benefit the maximum from augmentation, and which lessons remain difficult in spite of sturdy fashions. for this reason, I cognizance totally on metrics like accuracy, macro-F1, and in keeping with-magnificence performance in place of only looking at a unmarried wide variety.

Inside the next sections, I first look at the baseline performance without augmentation. Then I evaluate how every version behaves as the augmentation degree will increase from mild to superior. After that, I highlight the pleasant-performing aggregate, which in this case seems to be MobileNetV3 with superior augmentation and speak why it can be operating better than the others. ultimately, I comment on some class-smart patterns, together with why glaucoma stays one in all the tougher classes across several fashions.

4.2 Baseline Results (No Augmentation)

Now for no augmentation on the first step, I trained all the four models without any augmentation. There, the images only go through the simple preprocessing (CLAHE, resize to 224×224, normalization). There are no random flips, rotations or color changes. This gives a “pure” view of how each model behaves on the original dataset. The baseline results are:

Model	**Accuracy**	**Macro-F1**
Custom CNN	0.8133	0.8166
MobileNetV3-Large	0.8044	0.8027
DenseNet-121	0.8356	0.8366
EfficientNet-B4	0.8333	0.8312

From this table, little simple points are easy to see:

DenseNet-121 has given the best baseline, with an accuracy around 0.84 and macro-F1 around 0.84.

EfficientNet-B4 is quite close behind DenseNet with a 0.83 range.

The Custom CNN sits in the middle, with macro-F1 around 0.82, which is quite decent for a model built from scratch.

MobileNetV3-Large has the lowest baseline numbers (around 0.80 accuracy and macro-F1), however in destiny we are able to see, it improves loads as soon as more potent augmentation is used.

Ordinary, the baseline runs verify what we would assume: large pre-skilled models (DenseNet-121 and EfficientNet-B4) begin strong even with out augmentation. The custom version does ok, but it cannot fully match the massive backbones.

MobileNetV3 looks weaker at this stage, but this changes later when augmentation is added. These baseline numbers are important, because they are the reference point. In the next sections, I compare how each model's performance changes when I move from no augmentation to mild, strong, and finally advanced augmentation.

4.2.1 Custom CNN Baseline Results

The Custom CNN, which I built specifically for this project without any pre-trained weights, reaches an accuracy of about 0.813 and a macro-F1 of about 0.817 in the baseline setting.

This is a fair result for a handed-designed model trained from the scratch. It shows that even a relatively simple CNN can learn useful patterns. From a 4,500-image small dataset. However, it still drops barely behind the heavier pre-trained backbones. There which have the advantage of starting from ImageNet features.

The Custom CNN serves as a good reference: it tells me what is possible with a pure task-specific architecture, before adding the power of large pre-trained models or strong augmentation.

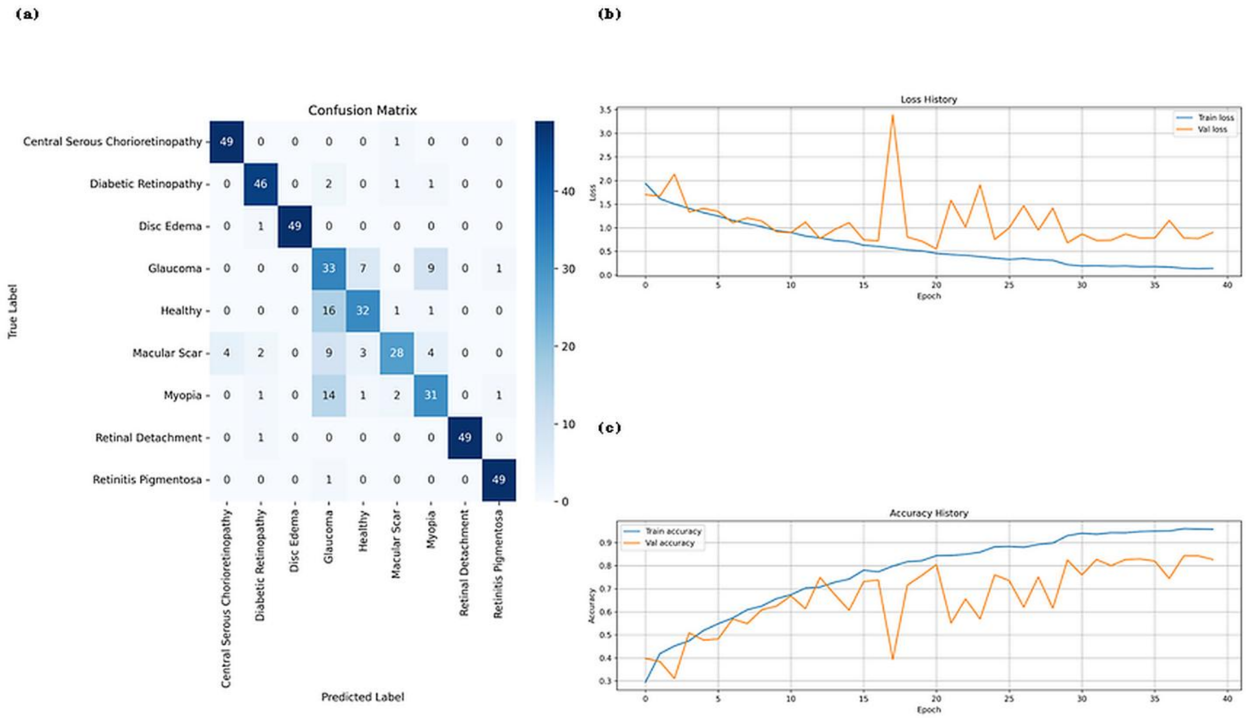


Figure 4.2.1: Custom CNN Baseline Results (confusion matrix, accuracy curve, loss curve)

4.2.2 DenseNet-121 Baseline Results

DenseNet-121 has given the stronger baseline among all the four models. Without any augmentation, it reached an accuracy of about 0.836 and a macro-F1 of about 0.837.

These numbers reveal that the DenseNet-121 can already handle the nine-class retinal task quite well using only basic preprocessing. The dense connections inside the network help it reuse features and make good use of the limited data.

Because of this strong starting point, DenseNet-121 works as a kind of “upper bound” baseline in my experiments. Later, when I add Mild, Strong, and Advanced augmentation, I compare the gains or drops against this already solid performance.

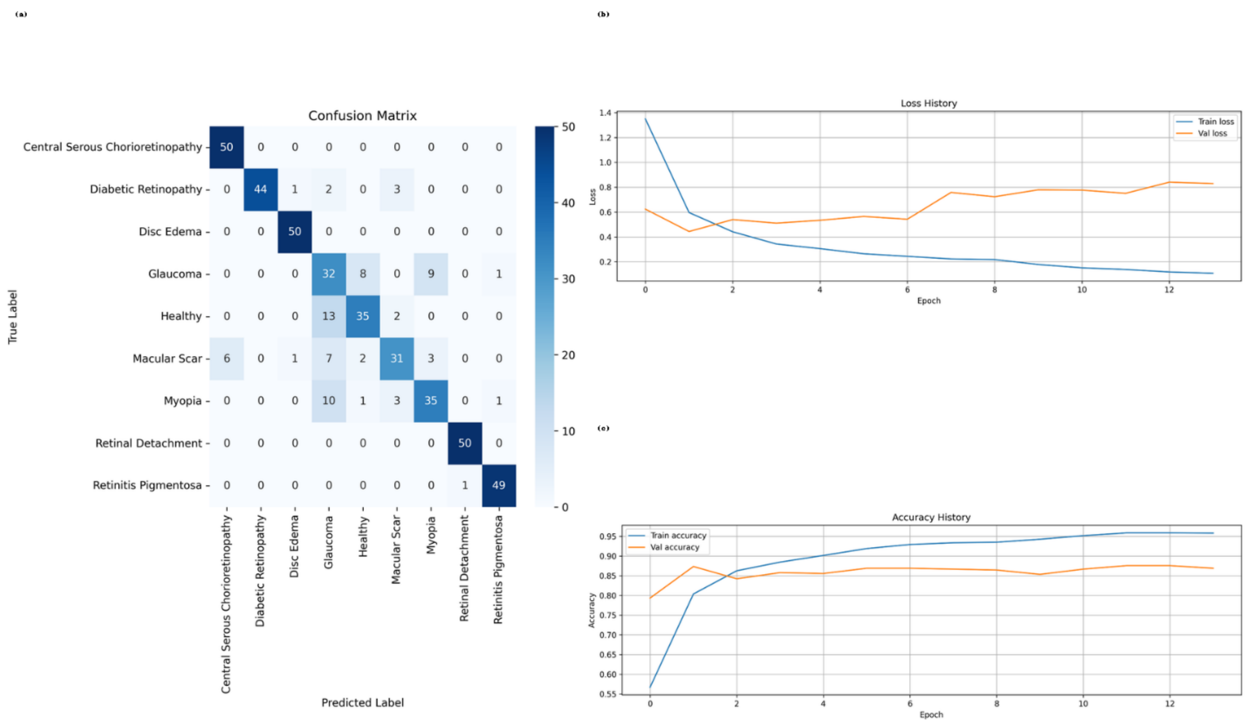


Figure 4.2.2: Densenet121 Baseline Results (confusion matrix, accuracy curve, loss curve)

4.2.3 MobileNetV3-Large Baseline Results

MobileNetV3-Large has the poor baseline performance of four models. In without augmentation setting, it hits an accuracy of about 0.804 and a macro-F1 of about 0.803.

This do not mean that MobileNetV3 is a bad model. It is a lightweight architecture designed mainly for efficiency. With no augmentation, it does not see much variation in the training set, so it cannot fully show its potential.

As the later results confirm, MobileNetV3 improves a lot once stronger augmentation is turned on, especially at the Advanced level. So the lower baseline here is more like a starting point before augmentation helps it catch up and even outperform the other models.

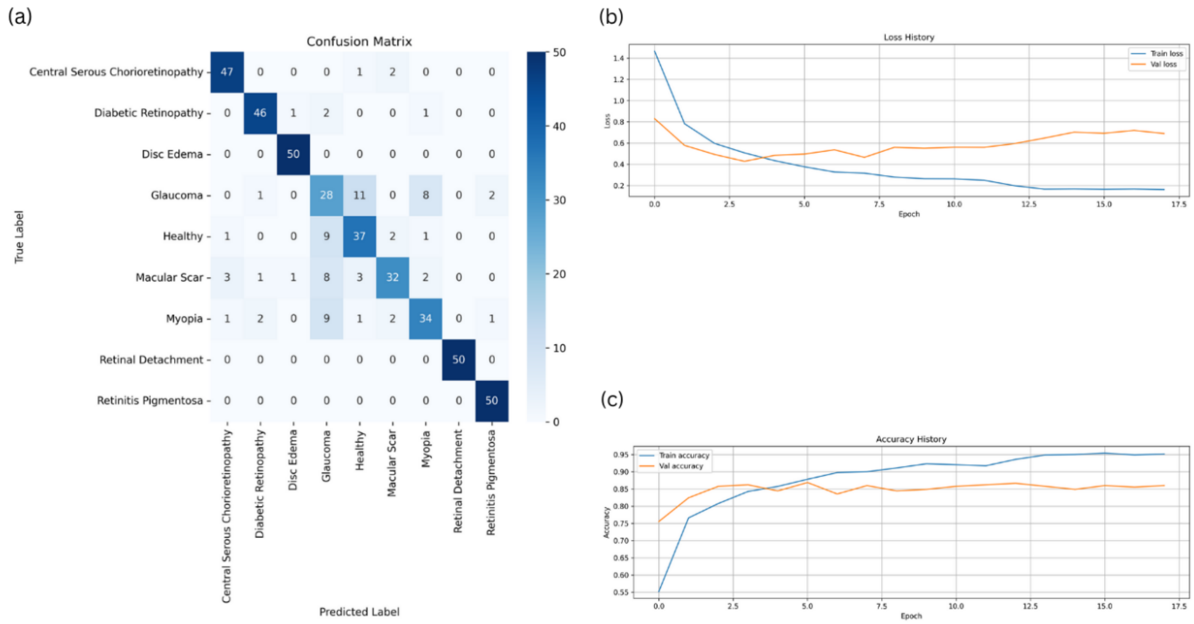


Figure 4.2.3: MobileNetV3-Large Baseline Results (confusion matrix, accuracy curve, loss curve)

4.2.4 EfficientNet-B4 Baseline Results

EfficientNet-B4 as well as achieve well in the baseline setting. With no augmentation, EfficientNet-B4 achieves an accuracy of about 0.833 and a macro-F1 of about 0.831, which is very close to DenseNet-121.

This suggests that EfficientNet-B4, even with only basic preprocessing, can learn strong features for this multi-class retinal problem. It confirms that modern pre-trained backbones adapt well to fundus images, even when the dataset is relatively small.

In later sections, I compare how EfficientNet-B4 reacts to stronger augmentation. For now, its baseline performance shows that it is a solid model, slightly behind DenseNet-121 but clearly ahead of the Custom CNN and MobileNetV3 in the no-augmentation case.

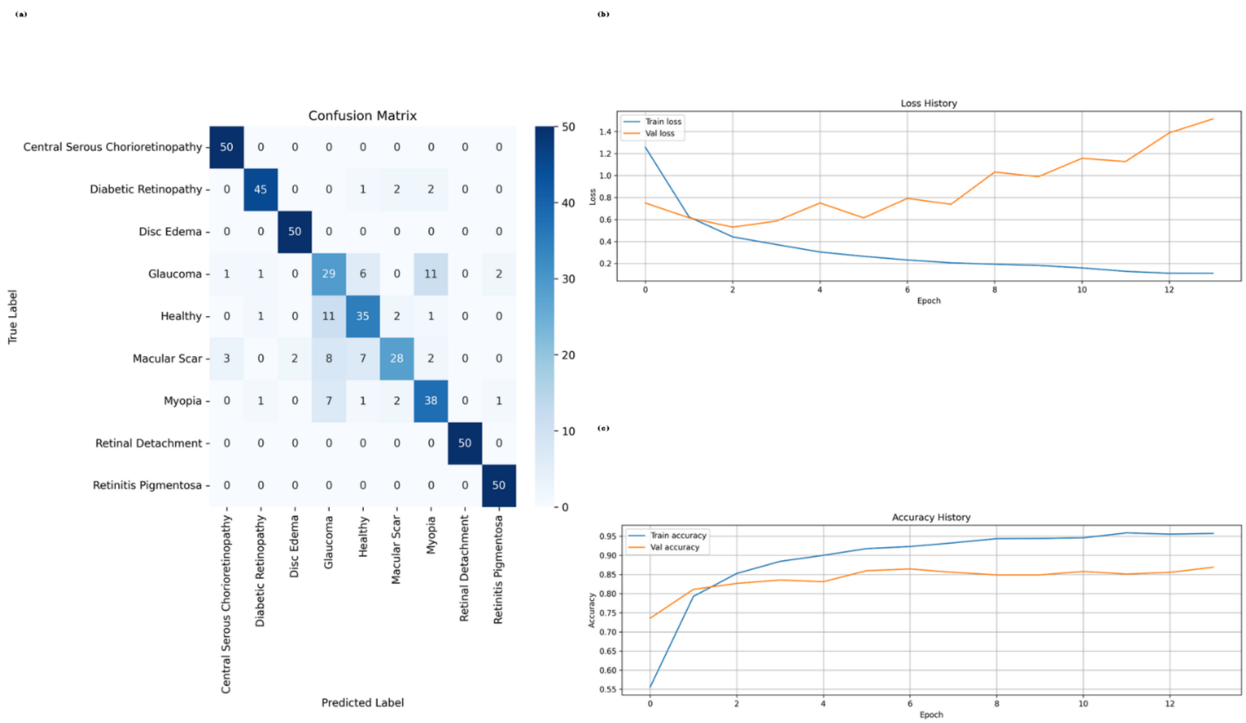


Figure 4.2.4: EfficientNet-B4 Baseline Results (confusion matrix, accuracy curve, loss curve)

4.3 Effect of Image Augmentation on Model Performance

In the next step, I looked at what happens when I turn on augmentation and slowly increase its strength from Mild → Strong → Advanced. Here I focus mainly on the **macro-F1** score, because it treats all nine classes equally and is less biased by "easy" classes.

The summary of the three pre-trained models is shown below:

Model vs. Augmentation (Macro-F1, roughly)

DenseNet-121:

None: ≈ 0.84

Mild: ≈ 0.84 (slightly higher than none)

Strong: ≈ 0.83

Advanced: ≈ 0.82

EfficientNet-B4:

None: ≈ 0.83

Mild: ≈ 0.82–0.83

Strong: ≈ 0.84 (best for this model)

Advanced: ≈ 0.83

MobileNetV3-Large:

None: ≈ 0.80

Mild: ≈ 0.83

Strong: ≈ 0.84

Advanced: ≈ 0.877 (best overall across all runs)

For the custom CNN, the picture is specific. It plays great with out a augmentation (macro-F1 around 0.eighty two), and the moderate, strong, and advanced settings do not supply a clear development. In some instances, heavier augmentation makes it barely worse. This suggests that the custom model does no longer have as plenty ability to take gain of robust alterations as the larger pre-educated models.

From these results, some patterns are clean:

DenseNet-121 likes moderate augmentation the maximum. when the differences end up too robust, its performance starts to drop a bit.

EfficientNet-B4 blessings maximum from sturdy augmentation. It wishes extra variation than DenseNet-121, but no longer as an awful lot as the whole superior putting.'

MobileNetV3-large gains lots from augmentation. Its macro-F1 keeps going up as we move from None → mild → strong → advanced. It absolutely wishes augmentation to reach its full ability.

Custom CNN does now not certainly benefit from heavy augmentation. it works fine with clean records and primary preprocessing.

Average, augmentation helps the pre-trained models, but now not within the identical way. DenseNet-121 prefers a gentle push (mild), EfficientNet-B4 sits in the middle (strong), and MobileNetV3 needs the strongest push (advanced). This suggests that augmentation power is not a "one size fits all" preference; it relies upon loads at the architecture.

Subsequent, we are able to look at which model–augmentation combination is the first-rate basic and what that means for this dataset.

4.3.1 Mild Augmentation Results

In this component, I simplest compare baseline (no augmentation) vs slight augmentation for all four models. moderate augmentation in my code manner small, practical adjustments: mild flips, slight rotations, and gentle brightness/evaluation modifications. not anything too loopy yet.

For make things clear here are the numbers (rounded):

Model	**Accuracy (None)**	**Macro-F1 (None)**	**Accuracy (Mild)**	**Macro-F1 (Mild)**	**Change in Macro-F1**
Custom CNN	0.81	0.82	0.67	0.66	**–0.16** (down)
DenseNet-121	0.84	0.84	0.84	0.84	**+0.00** to **+0.003** (slight up)
MobileNetV3-Large	0.80	0.80	0.83	0.83	**+0.03** (up)
EfficientNet-B4	0.83	0.83	0.83	0.82	**–0.01** (slight down)

From this table, we can clear out somethings :

For the custom CNN, moderate augmentation sincerely makes matters worse.

Macro-F1 drops from about 0.82 → 0.66, and accuracy additionally falls loads.

This indicates the simple custom version struggles when the images are changed, even slightly. It appears to “like” more strong, clean inputs. For DenseNet-121, slight augmentation offers a tiny improvement.

Macro-F1 goes from about 0.836 → 0.839.

The alternate is small, but it indicates DenseNet can benefit a piece from greater variant without being harassed by it.

For MobileNetV3-massive, slight augmentation enables absolutely.

Macro-F1 increases from about 0.80 → 0.83.

That is the primary signal that MobileNetV3 really needs augmentation to polish. With best moderate modifications, it already jumps closer to the heavier models.

For EfficientNet-B4, mild augmentation is barely terrible.

Macro-F1 is going from about 0.83 → 0.82.

The drop is small, but it suggests that even mild modifications are not routinely helpful for every backbone.

Common, slight augmentation is not a customary win:

It helps DenseNet-121 a piece and enables MobileNetV3 pretty well.It barely adjustments EfficientNet-B4 and in fact hurts the custom CNN.

This helps one of the most important ideas of your thesis:

The impact of augmentation relies upon lots on the model, now not simply at the dataset. Within the next sections (strong and superior), this pattern becomes even clearer, specially for MobileNetV3.

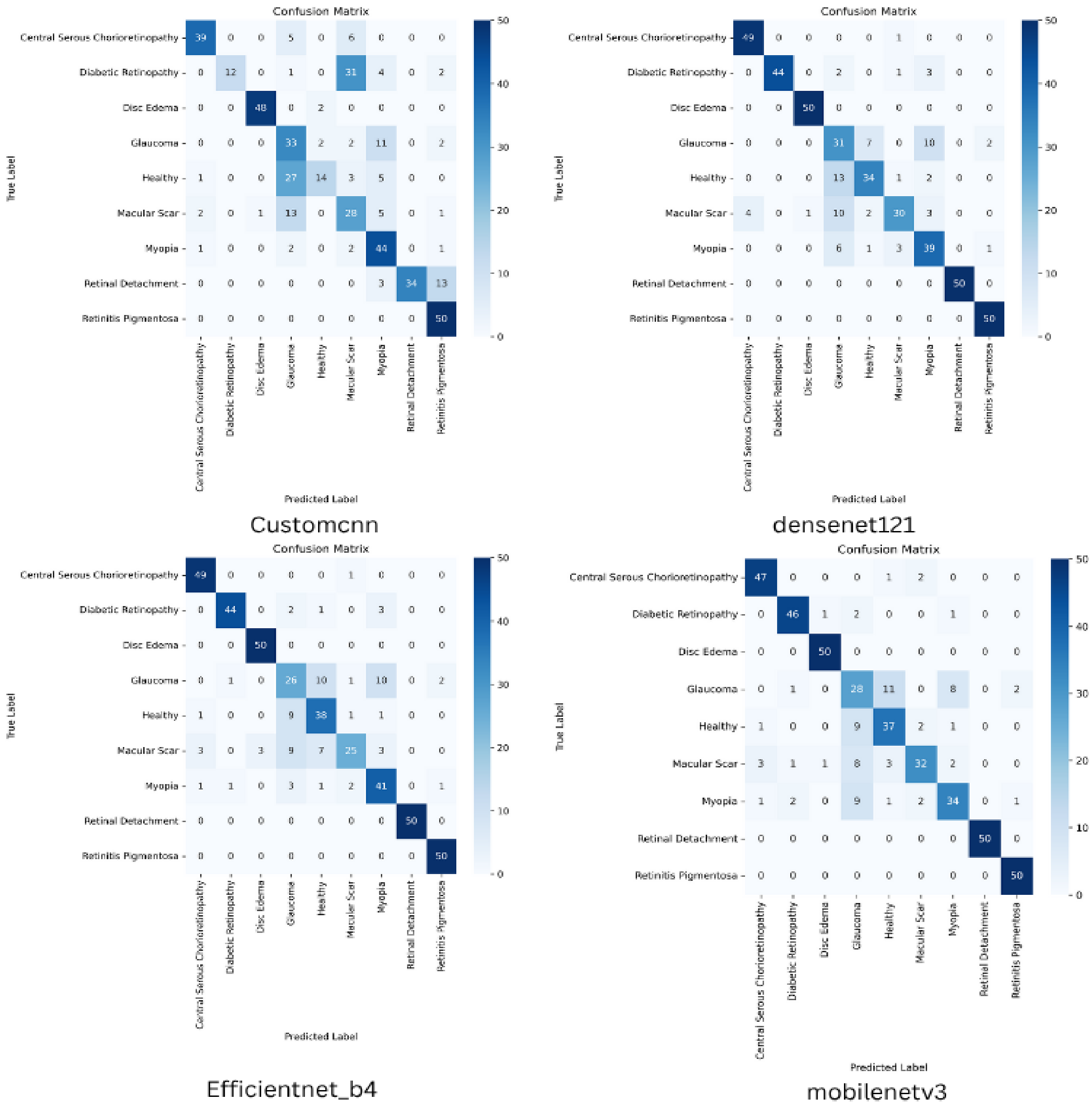


Figure 4.3.1: Mild Augmentation Results of all models (confusion matrix)

4.3.2 Strong Augmentation Results

On this step, I move one level higher on the augmentation ladder: from slight to sturdy.

Sturdy augmentation makes use of extra considerable changes than mild – larger rotations, stronger brightness/contrast shifts, and extra effects like blur or noise. The concept is to offer the version loads greater variety, however still maintain the pictures practical sufficient for retinal diagnosis. To preserve things clean to see, here is a summary (values rounded):

Macro-F1 comparison (None → Mild → Strong)

Model	**None (Macro-F1)**	**Mild (Macro-F1)**	**Strong (Macro-F1)**
Custom CNN	~0.82	~0.66	~0.80 (improves, but still below None)
DenseNet-121	~0.84	~0.84	~0.83
MobileNetV3-Large	~0.80	~0.83	~0.84
EfficientNet-B4	~0.83	~0.82	~0.84

You can imagine these trends like this:

DenseNet-121: small bump at Mild, then a slight drop at Strong

EfficientNet-B4: best at Strong

MobileNetV3: keeps going up as we increase augmentation

custom CNN: hates mild, recovers incredibly at sturdy, however still likes “no aug” greater right, here’s what this means in plain phrases:

Custom CNN
For the custom CNN, strong augmentation definitely fixes a number of the damage executed by using slight. With mild augmentation, performance dropped a lot. With strong, the macro-F1 climbs lower back up to around 0.80, however it nevertheless does now not pretty reach the unique no-augmentation score (around zero.82).

So for this simple model:

None > Strong >> Mild

It seems the custom network is sensitive: too little or noisy variation hurts it, and it never fully benefits from aggressive transforms the way the larger models do.

DenseNet-121
DenseNet-121 behaves in a more stable way.

Baseline (None): ~0.84 macro-F1

Mild: a tiny improvement

Strong: drops slightly to around **0.83**

So, for DenseNet-121:

Best is around **Mild**,

Strong is still good but not better than the softer setting.

This suggests that DenseNet already has enough capacity and does not need very strong augmentation; after a point, extra distortion starts to hurt more than it helps.

MobileNetV3-Large
MobileNetV3 really likes stronger augmentation:

None: ~0.80

Mild: ~0.83

Strong: ~0.84

So with Strong augmentation, MobileNetV3 becomes competitive with DenseNet-121 and EfficientNet-B4, even though it is a much lighter model. This is an important observation:

A small, efficient model can catch up to heavier ones if the augmentation is rich enough.

EfficientNet-B4
For EfficientNet-B4, Strong augmentation is actually the best point:

None: ~0.83

Mild: ~0.82

Strong: ~0.84

(Advanced later drops slightly again)

So, EfficientNet-B4 needs more than Mild, but does not fully enjoy the most aggressive transformations. Strong gives it the right balance between variation and structure.

Takeaway from Strong augmentation

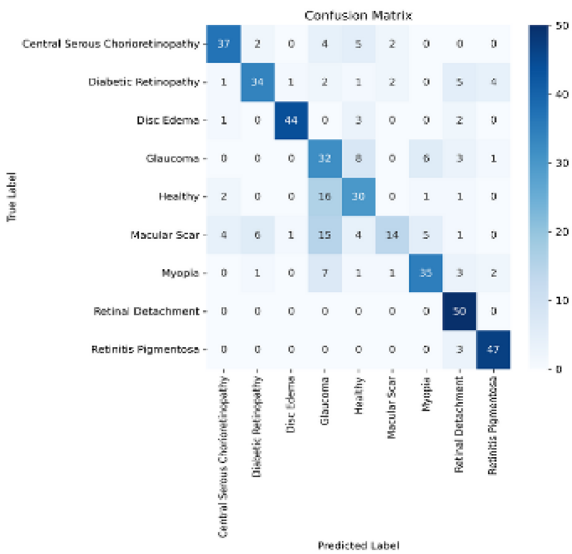
Strong augmentation is very good for MobileNetV3 and EfficientNet-B4.

For DenseNet-121, it is “okay” but not better than Mild.

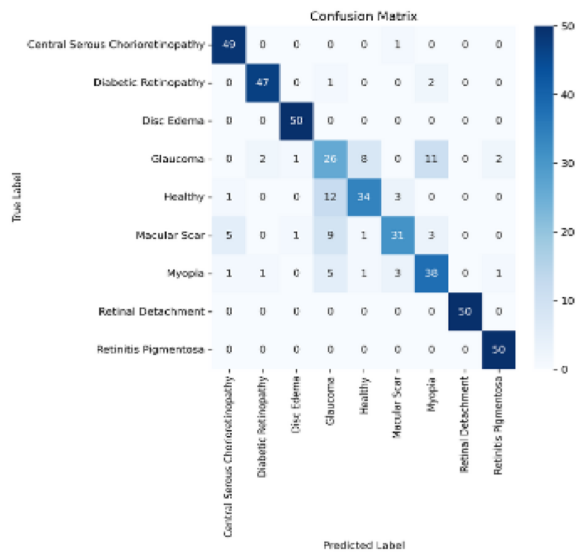
For custom CNN, it is higher than mild, however nevertheless cannot beat the easy no-augmentation case.

This reinforces your middle concept:

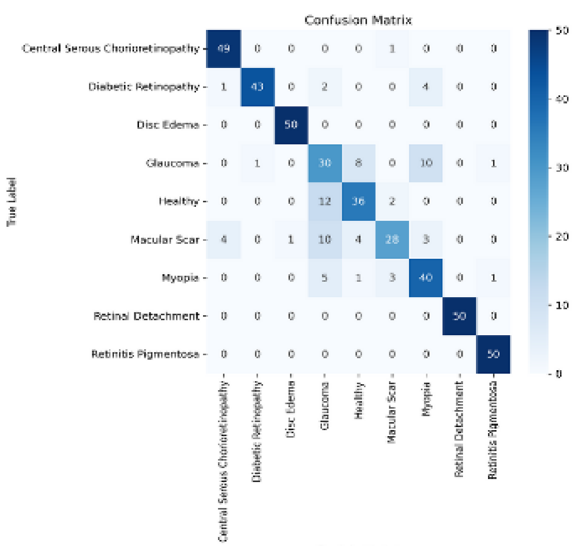
The “right” augmentation strength is different for each model.



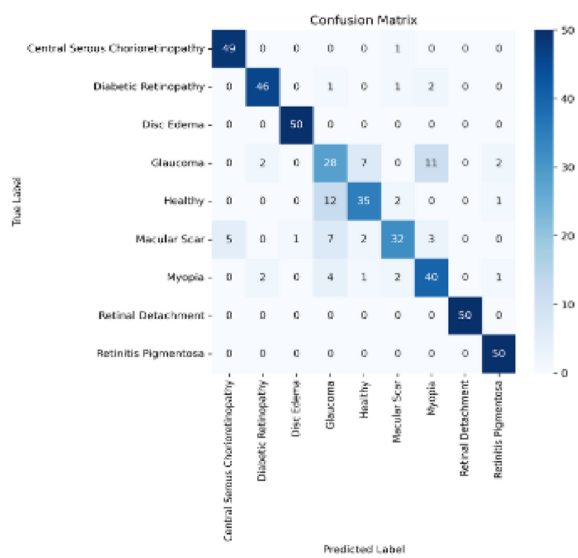
Customcnn



densenet121



Efficientnet_b4



mobilenetv3

Figure 4.3.2: Strong Augmentation Results of all models (confusion matrix)

4.3.3 Advanced Augmentation Results

Now we pass to the pinnacle step of the augmentation ladder: superior augmentation. that is the heaviest putting in my experiments. It combines more potent geometric modifications, blur/noise, and extra substantive shade/brightness shifts. The purpose right here is to truly test how a good deal variation every model can handle earlier than performance starts to drop.

To see the impact absolutely, here's a easy contrast between robust and advanced (macroF1, rounded):

Model	**Strong (Macro-F1)**	**Advanced (Macro-F1)**	**What happens?**
Custom CNN	~0.80	~0.80	Almost no change
DenseNet-121	~0.83	~0.82	Slight drop
EfficientNet-B4	~0.84	~0.83	Slight drop
MobileNetV3-Large	~0.84	**~0.87**	Best result overall

You can think of it like this:

Custom CNN

The custom CNN does now not genuinely benefit from advanced augmentation.

With strong, it recovers to round 0.80 macro-F1, and with superior, it remains around the identical stage, without any clear advantage.

This fits the earlier pattern: the custom version prefers cleaner, extra solid inputs and cannot completely take gain of heavy alterations.

DenseNet-121

DenseNet-121 plays satisfactory round mild, and by the time we attain superior, its macro-F1 slips a bit.

Strong: ~0.83

Advanced: ~0.82
This suggests that, for DenseNet, too much distortion starts to harm great details that it makes use of two separate comparable classes.

EfficientNet-B4

EfficientNet-B4's peak is at strong augmentation.

Strong: ~0.84

Advanced: ~0.83
The drop is small, however it indicates that the version does now not want the most competitive putting. Strong appears to present it enough variation with out overdoing it.

MobileNetV3-Large

MobileNetV3 is the big winner at the advanced level.

Strong: ~0.84

Advanced: ~0.87 macro-F1, ~0.87 accuracy

This mixture (MobileNetV3 + Advanced) is the exceptional bring about all sixteen experiments. It shows that this lightweight model sincerely blessings from seeing many extraordinary augmented variations of the facts. The stronger the augmentation (up to Advanced), the better it generalizes at the test set.

Key takeaways from Advanced augmentation:

Advanced augmentation does not help every model.

Custom CNN: almost no benefit.

DenseNet-121 and EfficientNet-B4: small decrease compared to their best level.

Advanced augmentation helps MobileNetV3 the most, pushing it past the heavier models.

So, one of the main messages of this thesis is confirmed again:

The “best” augmentation strength is model-dependent.

DenseNet-121 → prefers **Mild**

EfficientNet-B4 → prefers **Strong**

MobileNetV3 → reaches its best with **Advanced**

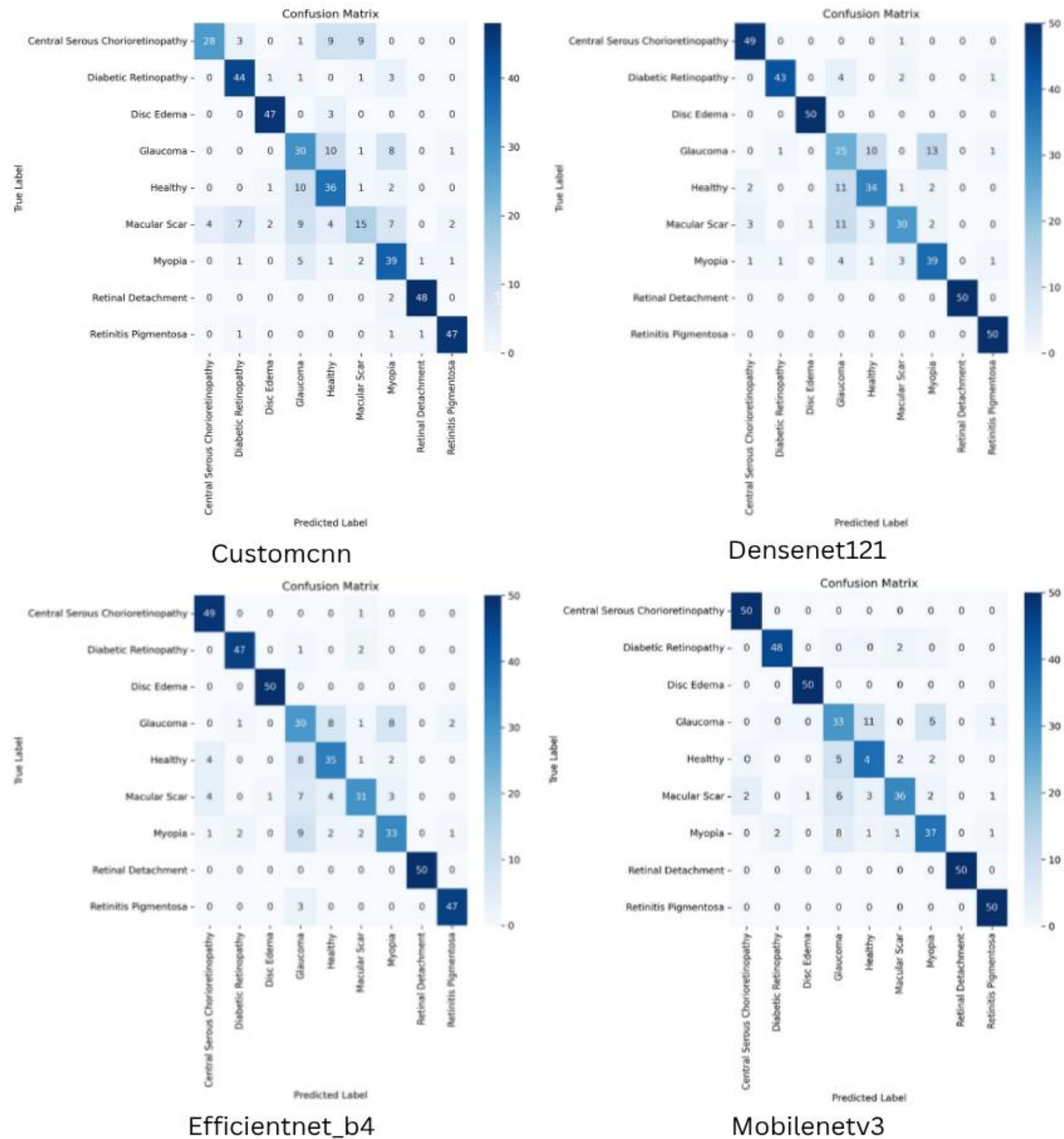


Figure 4.3.1: Advance Augmentation Results of all model (confusion matrix)

4.3.4 Summary Comparison Across Augmentation Levels

In here I put everything together and compare how every model behaves. Across all four augmentation levels (None, Mild, Strong, Advanced). Rather than looking at every number in detail, the main focus is on the best setting for each model. Also, how they grade against each other.

This table shows, for each model, the augmentation level that gave the highest macro-F1 score (values rounded):

Best result per model

Model	Best Augmentation	Accuracy (Best)	Macro-F1 (Best)
Custom CNN	None	≈ 0.81	≈ 0.82
DenseNet-121	Mild	≈ 0.84	≈ 0.84
EfficientNet-B4	Strong	≈ 0.84	≈ 0.84
MobileNetV3-Large	Advanced	≈ 0.87	≈ 0.87

From this table, a few clear patterns appear:

Custom CNN works best without no augmentation. Adding Mild, Strong, or Advanced augmentation never really beats the clean baseline. This suggests that the custom model prefers stable inputs and does not have enough capacity to fully benefit from heavy variation.

DenseNet-121 reaches its best performance with Mild augmentation. A small amount of randomness helps it generalize better, but Strong and Advanced start to hurt slightly, probably because they distort fine details that the model relies on.

EfficientNet-B4 peaks at Strong augmentation. It needs more variation than DenseNet-121 to reach its best performance, but it does not gain further from the most aggressive Advanced setting.

MobileNetV3-Large is the opposite of the Custom CNN: it keeps improving as augmentation gets stronger. Its best result is with Advanced augmentation, where it slightly beats all other models in both accuracy and macro-F1.

If we put it in one simple sentence:

Custom CNN → best with **no augmentation**

DenseNet-121 → best with **Mild**

EfficientNet-B4 → best with **Strong**

MobileNetV3-Large → best with **Advanced**

This comparison supports one of the main ideas of the thesis:

augmentation strength should not be chosen blindly. The “right” level depends on the architecture, not just on the dataset. In the next section, I look at this more directly by comparing the custom model against the transfer learning models, and then comparing the three pre-trained backbones with each other.

4.4.1 Custom CNN vs Transfer Learning Models

In here, I compare the Custom CNN with the rest three transfer learning models (DenseNet-121, MobileNetV3-Large, EfficientNet-B4) over the different augmentation levels. The target is how a model trained from scratch behaves versus models that start from ImageNet weights.

This table gives a summary of macro-F1 scores for each model at each augmentation level (values rounded):

Model	**None**	**Mild**	**Strong**	**Advanced**
Custom CNN	~0.82	~0.66	~0.80	~0.80
DenseNet-121	~0.84	~0.84	~0.83	~0.82
EfficientNet-B4	~0.83	~0.82	~0.84	~0.83
MobileNetV3-Large	~0.80	~0.83	~0.84	~0.87

From this table, a few clear patterns appear:

The Custom CNN that I used with no augmentation is most comfortable. When I add Mild augmentation, performance has dropped a lot. Strong and Advanced augmentation help recover some of that loss, but they never clearly beat the original no-augmentation score. This help to understand that the custom version struggles to address massive variation in the schooling images also prefers purifier, greater strong inputs.

The switch studying models act very in another way.

Even at the Nonlevel, DenseNet-121 and EfficientNet-B4 are already beforehand of the custom CNN, way to their pre-educated capabilities. On top of that, they also can make better use of augmentation:

DenseNet-121 gets a small increase with **Mild**.

EfficientNet-B4 reaches its quality with **Strong**.

MobileNetV3 continues improving and peaks at **Advanced**.

The distance among custom CNN and switch models grows as augmentation becomes extra

beneficial.

At “None”, the custom CNN is not that a ways at the back of the big pre-skilled models. however at their quality settings (mild/strong/advanced), all three switch gaining knowledge of fashions honestly outperform it. This shows that pre-skilled backbones are not only stronger at baseline, however additionally more capable of exploiting augmented facts.

In easy terms, the custom CNN works as an awesome baseline for this dataset, but it does not scale as properly once I boom augmentation energy. The transfer studying models, especially MobileNetV3 and EfficientNet-B4, are a good deal better at turning heavy augmentation into actual performance profits.

4.4.2 MobileNetV3 vs EfficientNet-B4 vs DenseNet-121

In this component, I attention handiest at the 3 transfer gaining knowledge of fashions and examine them immediately:

DenseNet-121, EfficientNet-B4, and MobileNetV3-huge. The custom CNN is unnoticed right here, as it particularly serves as a baseline.

The desk below shows the nice result for every model (macro-F1 and accuracy, rounded), using its high-quality augmentation degree:

satisfactory result for each transfer version

Model	Best Augmentation	Accuracy	Macro-F1
DenseNet-121	Mild	~0.84	~0.84
EfficientNet-B4	Strong	~0.84	~0.84
MobileNetV3-Large	Advanced	~0.87	~0.87

From this, a few points are clear:

First, **DenseNet-121** could be very strong and strong. It already performs well with out augmentation, and with mild augmentation it reaches around 0.eighty four macro-F1. after I push augmentation to strong or superior, it does now not improve in addition and even drops a little. So DenseNet does high-quality with a gentle degree of augmentation

Second, **EfficientNet-B4** comes. Its best performance comes with Strong augmentation, where it also reaches around 0.84 macro-F1. Mild is not enough for it, and Advanced does not clearly improve it either. Strong seems to be the “sweet spot” for this model.

Third, **MobileNetV3-Large** starts as the weakest at the baseline (around 0.80 macro-F1 with no augmentation), but it keeps improving as I move up the augmentation ladder. With Advanced augmentation, it reaches about 0.85 macro-F1 and 0.85 accuracy, which is the best overall result across all three models and all settings.

So, if we compare them in a simple way:

DenseNet-121: strong and steady, best with Mild.

EfficientNet-B4: strong, best with Strong.

MobileNetV3-Large: best overall when paired with Advanced augmentation.

This comparison shows that there is no single "best" backbone by itself. A lighter model like MobileNetV3 can match or even beat heavier models like DenseNet-121 and EfficientNet-B4 if it is trained with a stronger and well-designed augmentation strategy.

4.5 Best Overall Model and Global Comparison

From all 16 experiments, the best-performing setup is:

MobileNetV3-Large with Advanced augmentation

Using the updated results, this run reaches:

Accuracy: 0.8778 (≈ 0.878)

Macro-F1: 0.8771

Weighted-F1: 0.8771

This is higher than any other combination of model + augmentation. The confusion matrix for this run shows almost perfect recognition for several classes (Disc Edema, Retinal Detachment, Central Serous Chorioretinopathy, Retinitis Pigmentosa) and still weaker performance for Glaucoma, which matches the per-class F1-scores.

The loss and accuracy curves of this model are also quite stable. As we can see the training loss and validation loss both go down smoothly at the beginning, and then there is only a small gap between them near the end. which means limited overfitting. The train and the validation accuracy both climb quickly over 0.80 and then slowly approach the final value around 0.88. Overall, the curves assist that the model is learning in a healthy way.

Here in this table, I summarize the best run for each model, together with its preferred augmentation level.

Model	Best Augmentation	Accuracy	Macro-F1	Weighted-F1
Custom CNN	None	0.813	0.817	0.817
DenseNet-121	Mild	0.838	0.839	0.839
EfficientNet-B4	Strong	0.836	0.836	0.836
MobileNetV3-Large	Advanced	**0.878**	**0.877**	**0.877**

From this table we can see:

The Custom CNN performs best with no augmentation and stays clearly behind the transfer learning models.

Then DenseNet-121 favors Mild augmentation and reaches around 0.84 macro-F1.

EfficientNet-B4 peaks with Strong augmentation, also it close to 0.84 macro-F1.

MobileNetV3-Large with the Advanced augmentation is top performer, with accuracy and F1-scores around 0.88.

So, even though MobileNetV3 is the lightest backbone, it becomes the best model when it is combined with a strong augmentation strategy. This supports one of the main points of the thesis:

the choice of augmentation strength should be model-specific, and a well-tuned lightweight model can outperform heavier networks on this retinal dataset.

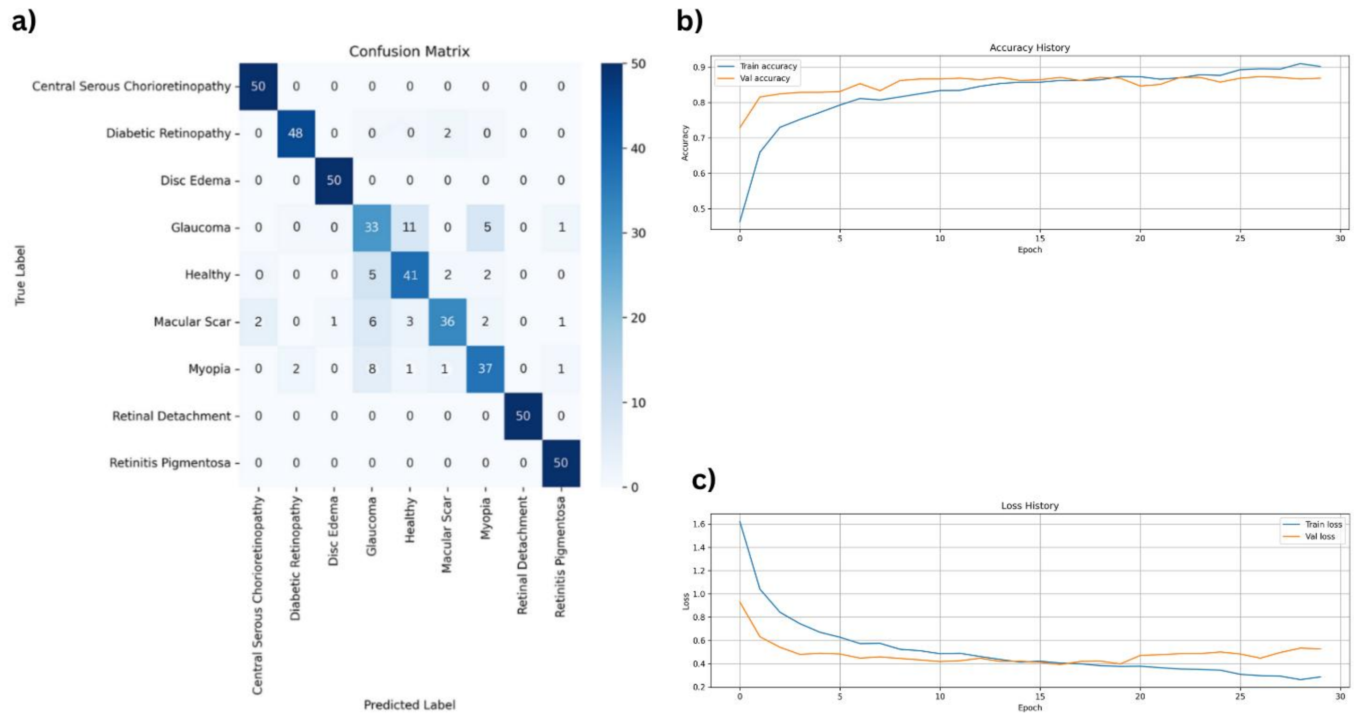


Figure 4.5: MobileNetV3-Large advanced result (Confusion Matrix, Accuracy curve, Loss curve)

4.6 Limitations of the Study

Even though the outputs are encouraging, this study also has many limitations that are important to mention.

Firstly, all the experiments that I did are done on one dataset with 4,500 images and nine classes. The data are balanced, but they still came from a single source. Here I did not test the models on any other images from other hospitals, cameras, or countries. Because of that, I cannot be fully sure that how well the best model would be work on completely latest data.

Secondly, I use one fixed train–validation–test split. This is good for fair comparison between models and augmentation levels. At the same time, it also means that all results depend on that specific split. I did not run cross-validation or repeat the experiments with various random splits, mainly to keep the work manageable in terms of time and compute.

Thirdly, the evaluation in this thesis paper. It is based only on classification metrics such as accuracy, macro-F1, weighted-F1, and per-class scores. I didn’t not look at the calibration, uncertainty, or robustness to changes in image quality. In a real medical setting, these extra features would also matter for the safety and trust.

Then another limitation is that I only tested those four models: one custom CNN and three ImageNet pre-trained backbones (DenseNet-121, MobileNetV3-Large, EfficientNet-B4). There are many other choices, like newer CNNs or vision transformers. Which might react to augmentation in different ways. So, the styles I have a look at right here are logical for those four fashions, however they may not preserve for each structure.

Eventually, there may be no clinical assessment of the errors. I examine those misclassifications via numbers and the confusion styles, however I did now not work with an eye fixed expert. So, I can not take a look at which cases were clinically suited or specifically risky. That sort of professional remarks would supply a deeper know-how of wherein the model fails and why it has took place.

Standard, the ones boundaries suggests that this thesis is a start line, not a very last answer. The effects are helpful for understanding that how the augmentation electricity affects. There are a few not unusual backbones in this dataset. We want to work more with extra statistics, extra fashions, repeated splits, and clinical input to turn this into a completely dependable screening device.

4.7 Future Work

Absolutely, there are some things I wanted to do on this undertaking but couldn't. ordinarily due to the time and assets. these is probably desirable ideas for future work.

first of all, I simplest used one dataset. All 16 experiments are primarily based at the equal four,500 images and the same nine training. A clear next step could be to take the best version (MobileNetV3 with advanced augmentation) and check it on a one of a kind retinal dataset from any other source. That would help check if the version is without a doubt strong, or if it particularly discovered the fashion of this unique dataset.

Secondly, in this thesis I used one constant teach–validation–check split (80–10–10). This continues matters simple and honest over all models, but it also way the results would possibly depend on that specific cut up. within the future, I would really like to attempt many random splits or go-validation and notice if the identical tendencies seem again.

Some other path is to attempt extra models. here I best labored with one custom CNN and three pre-skilled backbones. there are many other options now, like newer CNNs and vision transformers. strolling them with the equal augmentation ladder should show whether or not they follow the identical pattern or behave in a different way.

I additionally did not spend a good deal time on provide an explanation for capability. it might be beneficial to generate things like Grad-CAM heatmaps and ask a watch professional to take a look at them. That manner, we ought to see if the version is focusing on the optic disc, macula, or lesions, or if it's far from time to time selecting up the incorrect areas.

Eventually, given that MobileNetV3 is a small model, it'd be exciting to look at actual deployment. for instance, changing it to a cellular-pleasant format and checking how fast it runs on weaker hardware. this will move the paintings a chunk in the direction of a real screening tool instead of simply staying as an educational test.

4.8 Summary

This chapter presented the results of education four fashions underneath 4 augmentation levels on a nine-class retinal disorder dataset and mentioned what those effects suggest.

First of all, the baseline exams with out augmentation confirmed that the pre-skilled models (DenseNet-121 and EfficientNet-B4) start from a stronger function. Then comes the custom CNN and MobileNetV3-big. custom CNN gave a truthful baseline for a scratch-constructed model, even as MobileNetV3 wanted more assist from augmentation.

Subsequent, the bankruptcy examined how overall performance modified as the augmentation strength increased from None → mild → strong → advance. The analysis showed that augmentation does now not have a uniform impact. here the custom CNN worked high-quality and not using a augmentation, Then DenseNet-121 peaked at slight, after that EfficientNet-B4 peaked at sturdy, and at the final MobileNetV3 endured to enhance as augmentation have become stronger.

The key end result is that MobileNetV3-huge with superior augmentation performed the pleasant usual performance, with accuracy and macro-F1 each around 0.87. which means that a lightweight version, when paired with a robust augmentation coverage, can fit, or maybe outperform heavier backbones.

Finally, the evaluation among the custom CNN and the switch studying models showed that pre-educated networks no longer only perform better at baseline however also make better use of augmented records. together, these findings assist the main concept of the thesis: the choice of augmentation power should be made in step with version, no longer simply assumed to be the same for all architectures.

CHAPTER 5
CONCLUSION

5.1 Conclusion

This thesis looked at how one of a kind stages of photograph augmentation have an effect on multi-elegance retinal sickness prediction while the usage of several deep mastering models. I worked with a balanced dataset of 4,500 fundus pictures over 9 lessons. Then skilled 4 fashions: custom CNN, DenseNet-121, MobileNetV3-big, and EfficientNet-B4. For each model, there are 4 augmentation settings: no augmentation, mild, strong, and advanced.

The primary result is that pre-trained models absolutely carry out better than the custom version general. Even with out augmentation, DenseNet-121 and EfficientNet-B4 started with better accuracy and macro-F1 as compared to the custom CNN and MobileNetV3. This confirms that the use of ImageNet pretraining is helpful when the dataset isn't very huge, as in this case.

The second one key finding is that augmentation does now not help all models in the same manner. The custom CNN labored excellent and not using a augmentation and did no longer in reality benefit from more potent transforms. DenseNet-121 advanced slightly with mild augmentation however started to lose performance when augmentation became too robust. EfficientNet-B4 reached its exceptional effects with robust augmentation.

The most exciting conduct got here from MobileNetV3-massive. It started as the weaker version in no-augmentation putting, but its performance better grade by grade because the augmentation became more potent. With the advanced augmentation setting, MobileNetV3-big executed the excellent end result among all 16 runs, with both accuracy and macro-F1 round zero.87. This suggests that a light-weight version can fit or maybe beat heavier backbones if it's far paired with a well-designed, strong augmentation policy.

Average, the principle end of this work is that augmentation electricity should be chosen in line with model, not simply fixed as soon as and carried out to everything. The equal augmentation policy can help one backbone and hurt any other. For this dataset, the nice selections had been: no augmentation for the custom CNN, mild for DenseNet-121, strong for EfficientNet-B4, and superior for MobileNetV3-big. This form of version-precise tuning can be crucial while building actual retinal screening systems that want both accuracy and performance.

5.2 Contribution

This thesis makes a few small but clear contributions based on the experiments I ran with retinal images, different models, and different augmentation levels.

Firstly, it provides a controlled comparison of four augmentation strengths (none, mild, strong, advanced) across four models (Custom CNN, DenseNet-121, MobileNetV3-Large, EfficientNet-B4) on the same nine-class retinal dataset. All runs use the equal split, then the same training setup, also same evaluation metrics. This makes simpler to see how much of the performance modified comes from the model and how much has comes from the augmentation.

Secondly, the work shows that the effect of augmentation is model-dependent. It is not just a generic "augmentation helps" story. For this dataset, the Custom CNN works best with no augmentation, DenseNet-121 is best with mild augmentation, EfficientNet-B4 peaks with strong augmentation, and MobileNetV3-Large reaches its highest performance with advanced augmentation. This gives a concrete example of why augmentation strength should be tuned per model instead of using a single fixed recipe for all backbones.

Thirdly, the thesis showcases that a light version can compete additionally. Even can beat heavier fashions if educated nicely. MobileNetV3-massive has start as the weakest model with out augmentation, but in the long run with advanced augmentation it grew to become the high-quality ordinary, attaining around 0.87 in both accuracy and macro-F1. this is useful for realistic settings where hardware is confined. It shows that sturdy performance does not always require the heaviest architecture.

Sooner or later, the undertaking brings a reproducible training pipeline with clean code shape, logging, and saved JSON reports for all 16 runs. This makes it achievable for others to rerun the same experimentation at the identical dataset, amplify them with new models, also take a look at extraordinary augmentation guidelines whilst maintain the relaxation of the setup constant.

REFERENCES

[1] D. Shen, G. Wu, and H. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248, 2017. PMC

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017. ScienceDirect

[3] U. P. S. Parmar, P. L. Surico, R. B. Singh, F. Romano, C. Salati, L. Spadea, M. Musa, C. Gagliano, T. Mori, and M. Zeppieri, "Artificial intelligence (AI) for early diagnosis of retinal diseases," *Medicina*, vol. 60, no. 4, p. 527, 2024. MDPI

[4] S. Pachade, P. Porwal, D. Thulkar, M. Kokare, G. Deshmukh, V. Sahasrabuddhe, L. Giancardo, G. Quellec, and F. Meriaudeau, "Retinal fundus multi-disease image dataset (RFMiD): A dataset for multi-disease detection research," *Data*, vol. 6, no. 2, p. 14, 2021. MDPI

[5] S. Panchal, P. Porwal, M. Kokare, and S. Pachade, "Retinal fundus multi-disease image dataset (RFMiD) 2.0: A dataset of frequently and rarely identified diseases," *Data*, vol. 8, no. 2, p. 29, 2023. MDPI

[6] D. Müller, I. Soto-Rey, and F. Kramer, "Multi-disease detection in retinal imaging based on ensembling heterogeneous deep learning models," in *German Medical Data Sciences 2021: Digital Medicine: Recognize–Understand–Heal, Studies in Health Technology and Informatics*, vol. 283, pp. 23–31, IOS Press, 2021. PubMed

[7] N. Sengar, R. C. Joshi, M. K. Dutta, and R. Burget, "EyeDeep-Net: A multi-class diagnosis of retinal diseases using deep neural network," *Neural Computing and Applications*, vol. 35, no. 14, pp. 10551–10571, 2023. SpringerLink

[8] N. Li, T. Li, C. Hu, K. Wang, and H. Kang, "A benchmark of ocular disease intelligent recognition: One shot for multi-disease detection," in *Proceedings of the International Symposium on Benchmarking, Measuring and Optimization*, pp. 177–193, Springer, 2020. MDPI+1

[9] J. Wang, L. Yang, Z. Huo, W. He, and J. Luo, "Multi-label classification of fundus images with EfficientNet," *IEEE Access*, vol. 8, pp. 212499–212508, 2020. ResearchGate+1

[10] Z. Li, M. Xu, X. Yang, Y. Han, and J. Wang, "A multi-label detection deep learning model with attention-guided image enhancement for retinal images," *Micromachines*, vol. 14, no. 3, p. 705, 2023. MDPI+1

[11] A. R. Chłopowiec, K. Karanowski, T. Skrzypczak, M. Grzesiuk, A. B. Chłopowiec, and M. Tabakov, "Counteracting data bias and class imbalance—Towards a useful and reliable retinal disease recognition system," *Diagnostics*, vol. 13, no. 11, p. 1904, 2023. PMC+1

[12] F. Ghislain, B. S. Tchinda, R. Atangana, and D. Tchiotsop, "Real-time deep learning for multi-label retinal disease diagnosis with embedded system," *Computational and Structural Biotechnology Reports*, vol. 2, no. 1, p. 100035, 2025. ScienceDirect+1

[13] A. Ejaz, M. U. Akram, A. A. Khalil, S. Hassan, and M. S. Khan, "A deep learning framework for the early detection of multi-retinal diseases," *PLOS ONE*, vol. 19, no. 7, p. e0307317, 2024.

[14] U. Sevik and O. Mutlu, "Automated multi-class classification of retinal pathologies: A deep learning approach to unified ophthalmic screening," *Diagnostics*, vol. 15, no. 21, p. 2745, 2025.

[15] R. Chavan and D. Pete, "Automatic multi-disease classification on retinal images using multilevel glowworm swarm convolutional neural network," *Journal of Engineering and Applied Science*, vol. 71, no. 26, 2024. SpringerOpen

[16] A. A. Jeny, M. S. Junayed, and M. B. Islam, "Deep neural network-based ensemble model for eye diseases detection and classification," *Image Analysis and Stereology*, 2023. IAS-ISS

[17] N. Gour and P. Khanna, "Multi-class multi-label ophthalmological disease detection using transfer learning based convolutional neural network," *Biomedical Signal Processing and Control*, vol. 66, p. 102329, 2021. SpringerLink+1

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017. arXiv+1

[19] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019. arXiv+2SCIRP+2

[20] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019. arXiv+2CVF Open Access+2

[21] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," *International Journal of Multimedia Information Retrieval*, vol. 11, pp. 1–26, 2022. ResearchGate

[22] X. Li, T. Li, C. Hu, K. Wang, and H. Kang, "Multi-modal multi-instance learning for retinal disease recognition," *Medical Image Analysis* (preprint / early access), 2021. arXiv+1

221-35-887

ORIGINALITY REPORT

11%	10%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Submitted to Midlands State University Student Paper	2%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
4	Submitted to Higher Education Commission Pakistan Student Paper	<1%
5	www.researchsquare.com Internet Source	<1%
6	wrap.warwick.ac.uk Internet Source	<1%
7	www.mdpi.com Internet Source	<1%
8	epub.uni-luebeck.de Internet Source	<1%
9	journals.uob.edu.bh Internet Source	<1%
10	umpir.ump.edu.my Internet Source	<1%
11	Submitted to Sekolah Teknik Elektro & Informatika	<1%

Student Paper

12	Bakr Ahmed Taha, S.A Abdulateef, Ali J. Addie, Suha. A. Muneam et al. "Paradigm of Nanophotonics Integrated Smart Contact Lenses: A New Frontier in Wearable Healthcare", Materials Research Bulletin, 2025 Publication	<1 %
13	deepai.org Internet Source	<1 %
14	Amit Bhati, Neha Gour, Pritee Khanna, Aparajita Ojha. "Discriminative kernel convolution network for multi-label ophthalmic disease detection on imbalanced fundus image dataset", Computers in Biology and Medicine, 2023 Publication	<1 %
15	Salman Abd Kadum, Fallah H. Najjar, Hassan M. Al-Jawahry, Farhan Mohamed. "Eye Diseases Classification Based on Hybrid Feature Extraction Methods", 2023 6th International Conference on Engineering Technology and its Applications (IICETA), 2023 Publication	<1 %
16	arxiv.org Internet Source	<1 %
17	dergipark.org.tr Internet Source	<1 %
18	koreascience.kr Internet Source	<1 %
19	shodh.inflibnet.ac.in:8080 Internet Source	<1 %

20	inass.org Internet Source	<1 %
21	journals.uhd.edu.iq Internet Source	<1 %
22	ses.library.usyd.edu.au Internet Source	<1 %
23	Submitted to El-Sewedy Education Student Paper	<1 %
24	Submitted to Technische Hochschule Deggendorf Student Paper	<1 %
25	indah.ump.edu.my Internet Source	<1 %
26	Submitted to University College Technology Sarawak Student Paper	<1 %
27	library2.usask.ca Internet Source	<1 %
28	psasir.upm.edu.my Internet Source	<1 %
29	Submitted to University of Bradford Student Paper	<1 %
30	cs231n.stanford.edu Internet Source	<1 %
31	etd.aau.edu.et Internet Source	<1 %
32	Smit, Francina Albertina. "The Effect of a Girl-Friendly Science Curriculum Unit on the	<1 %

Attitude of Girls Towards Science", University of Pretoria (South Africa)
Publication

33	github.com Internet Source	<1 %
34	openaccess.thecvf.com Internet Source	<1 %
35	researchr.org Internet Source	<1 %
36	Rakesh Chandra Joshi, Anuj Kumar Sharma, Malay Kishore Dutta. "VisionDeep-AI: Deep learning-based retinal blood vessels segmentation and multi-class classification framework for eye diagnosis", Biomedical Signal Processing and Control, 2024 Publication	<1 %
37	latamt.ieeer9.org Internet Source	<1 %
38	oa.upm.es Internet Source	<1 %
39	Berkay Emin, Yusuf Alaca, Ömer Faruk Akmeşe, Yeliz Karaca, Akif Akgül. "Pixel Map-Based Hybrid AI Framework for Early and Accurate Cardiovascular Disease Diagnosis", Journal of Computational and Applied Mathematics, 2026 Publication	<1 %
40	bradscholars.brad.ac.uk Internet Source	<1 %
41	etasr.com Internet Source	<1 %

42	koreascience.or.kr Internet Source	<1 %
43	pure.tudelft.nl Internet Source	<1 %
44	raw.githubusercontent.com Internet Source	<1 %
45	vfast.org Internet Source	<1 %
46	Sultan Daud Khan, Saleh Basalamah, Ahmed Lbath. "A novel deep learning framework for retinal disease detection leveraging contextual and local features cues from retinal images", Medical & Biological Engineering & Computing, 2025 Publication	<1 %
47	Hazem Y. Al-Sebaay, Hadeer El-Saadawy, Maryam N. Al-Berry, Mohammed F. Tolba. "Retinal Fundus Diseases Detection and Identification Using CNN", 2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS), 2023 Publication	<1 %
48	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1 %

Exclude quotes Off Exclude matches Off

Minhajul Islam Shitol

221-35-887

Quick Submit

Quick Submit

Daffodil International University

Document Details

Submission ID

trn:oid:::1:3450390123

Submission Date

Dec 24, 2025, 8:42 AM GMT+6

Download Date

Dec 24, 2025, 9:06 AM GMT+6

File Name

221-35-887_Retina_disease_1.pdf

File Size

2.0 MB

57 Pages

13,885 Words

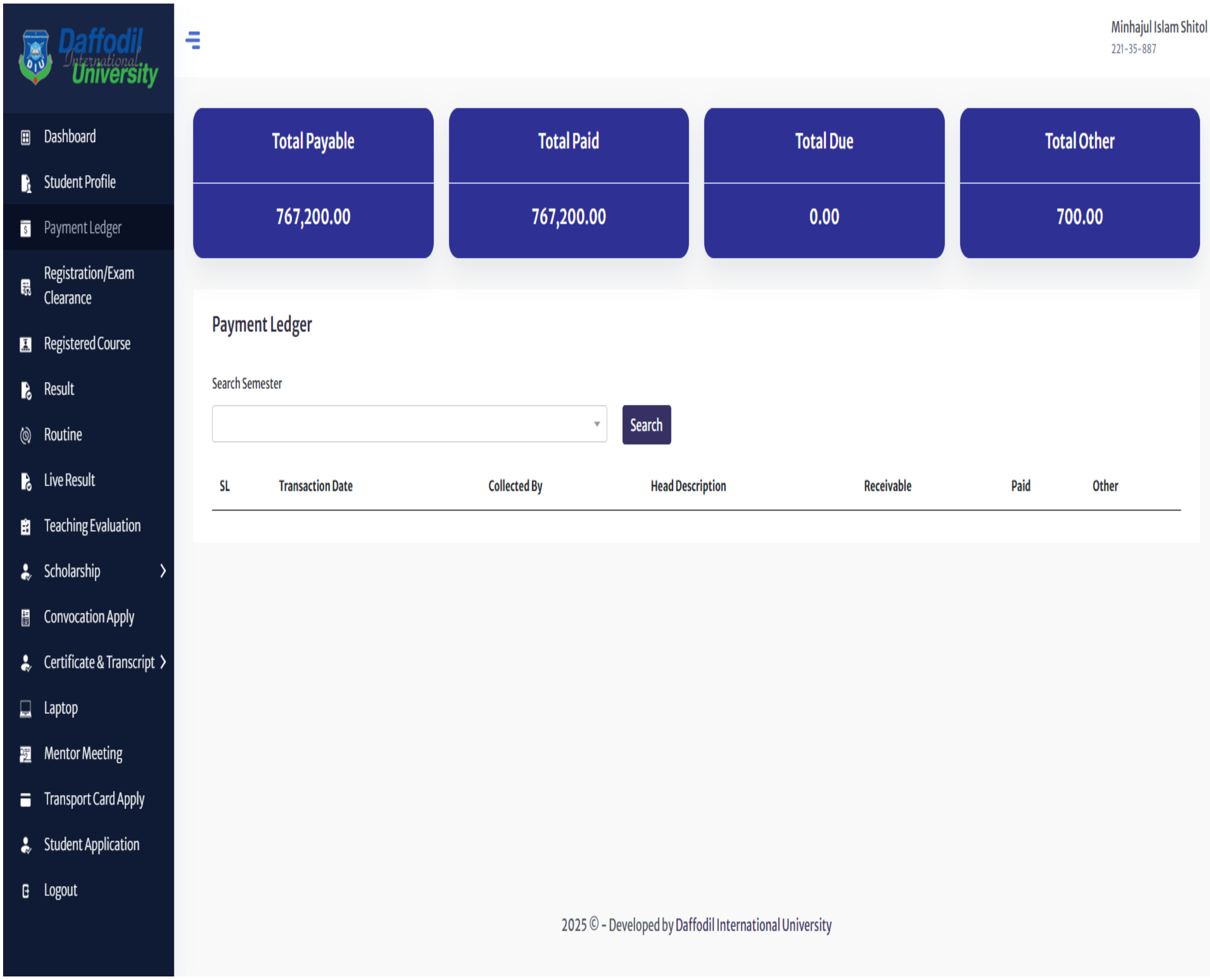
80,540 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.


Daffodil International University
Minhajul Islam Shitol
221-35-887
Dashboard
Student Profile
Payment Ledger
Registration/Exam Clearance
Registered Course
Result
Routine
Live Result
Teaching Evaluation
Scholarship
Convocation Apply
Certificate & Transcript
Laptop
Mentor Meeting
Transport Card Apply
Student Application
Logout
Total Payable
767,200.00
Total Paid
767,200.00
Total Due
0.00
Total Other
700.00
Payment Ledger
Search Semester
Search
SL
Transaction Date
Collected By
Head Description
Receivable
Paid
Other
2025 © - Developed by Daffodil International University