Premier University, Chattogram

Project Report on

Eye Disease Detection & Classification using Deep Learning: A comparative Study of various CNN models

Course Title: Machine Learning Laboratory

Course Code: CSE 458

Submitted To:

Md Tamim Hossain
Lecturer
Department of CSE, PUC

Submitted By:

Md Nurnabi Rana
ID: 2104010202290
Sabrina Akter
ID: 2104010202289
Semester: 8th, Batch: 40th, Sec: D
Session: Spring 2025

Submission Date: 23rd November, 2025

Department of CSE
Premier University, Chattogram

Eye Disease Detection and Classification Using Deep Learning: A Comparative Study of CNN

Md Nurnabi Rana (ID: 2104010202290)[1] and Sabrina Akter (ID: 2104010202289)[2†]

[1]Computer Science & Engineering, Premier University, Chattogram.
[2]Computer Science & Engineering, Premier University, Chattogram.

Contributing authors: mdnurnabirana.cse@gmail.com;
sabrinaakterchy.cse@gmail.com;
[†]These authors contributed equally to this work.

Abstract

Eye diseases are a major problem around the world. They often lead to vision loss or blindness if we do not find them early. Finding and classifying these diseases correctly and quickly is very important for good treatment and to stop permanent damage to eyesight. In our study we looked at how deep learning models can help with this. We used convolutional neural networks like ResNet, EfficientNet, MobileNetV3 and DenseNet121 to classify ten eye diseases. These include Retinitis Pigmentosa, Retinal Detachment, Pterygium, Myopia, Macular Scar, Glaucoma, Disc Edema, Diabetic Retinopathy, Central Serous Chorioretinopathy and Healthy cases. We took an augmented dataset from Rashid et al. [1] and split it into training, validation and test sets with a 70:20:10 ratio. Our experiments showed that EfficientNet got the best test accuracy of 87 percent. ResNet was next with 85 percent. It is still hard to tell apart some classes that look very similar, especially Healthy and Glaucoma, because of small overlaps in how they look and some issues with the dataset. We did detailed checks using confusion matrices, Grad-CAM visualizations, training and validation loss curves and looked at misclassified samples. This gave us good ideas about model biases and what to improve. Our work helps in making better automatic tools for diagnosing eye diseases. It supports creating strong AI systems that can be used in clinics.

Keywords: Eye Disease Detection, Eye Disease Classification, Deep Learning, Transfer Learning, Fundus Imaging

1 Introduction

Eye diseases like glaucoma, diabetic retinopathy and retinal detachment are big reasons for vision loss and blindness. They affect millions of people all over the world [2, 3]. Finding them early and diagnosing correctly is key to stop the disease from getting worse and to give treatment on time. But usual ways of diagnosing depend a lot on eye doctors checking by hand. This is hard work, takes time and can vary from one doctor to another [4, 5]. These problems are worse in areas where there are not many specialists. There preventable vision loss is still a big health issue [6].

In the last few years deep learning has become a great way to analyze medical images automatically. It gives solutions that are easy to scale, cost less and are more objective for screening eye diseases [7, 8]. Convolutional neural networks such as VGG16, ResNet [9] and EfficientNet [10] have shown they work well for binary and multi-class tasks. They are good at telling healthy retinal images from diseased ones [11, 12]. These models can find layered spatial features in fundus images. This helps in spotting small pathological changes.

Even with these improvements most studies look at detecting one disease or only a few classes. This leaves a gap for real clinic situations with rare or similar looking retinal problems [13, 14]. To fix this our study does a full comparison of four advanced CNN models. These are ResNet-50 [9], EfficientNet [10], DenseNet121 [15] and MobileNetV3 [16]. We use them to classify ten eye conditions. Retinitis Pigmentosa, Retinal Detachment, Pterygium, Myopia, Macular Scar, Glaucoma, Disc Edema, Diabetic Retinopathy, Central Serous Chorioretinopathy and Healthy eyes [1, 17]. We use a large augmented fundus image dataset. The study checks model performance in tough conditions like class imbalance, small similarities between classes (for example Healthy versus Glaucoma) and dataset limits.

Our study gives three main things. First it offers a strict comparison of CNN architectures for a hard multi-class eye disease task. It shows their strengths and weaknesses. Second it uses advanced ways to explain like confusion matrices, Grad-CAM visualizations [18] and error analysis. This helps understand model behavior, biases and why they fail. Third it suggests useful strategies like hybrid models and specific data augmentation for eyes. This is to solve ongoing problems in recognizing eye diseases. By combining these ideas our work helps develop reliable AI diagnostic systems. It leads to better automatic tools for screening eyes.

2 Literature Review

Finding and classifying eye diseases automatically from retinal fundus images has received a lot of attention lately. This is mainly because of progress in deep learning and more annotated medical image datasets [7, 8]. Over the years, researchers have explored various approaches to tackle this challenge, starting from traditional image processing techniques to advanced neural networks, each building on the previous to improve accuracy and efficiency in diagnosis. Early work in this area focused on hand-crafted features and machine learning classifiers. For example, methods using support vector machines or random forests combined with features like texture analysis or vessel segmentation showed promise but often struggled with variability in image

quality and subtle disease signs. As deep learning gained traction, convolutional neural networks became the go-to choice for their ability to learn features directly from data without manual engineering. People have made several public datasets to help research. The Retinal Fundus Multi-Disease Image Dataset (RFMiD) 2.0 [13] gives 860 images annotated by experts for 51 eye conditions. RFMiD has good variety in pathologies but its small size makes it hard to train big deep learning models especially for rare classes. The MultiEYE dataset [14] includes both fundus and Optical Coherence Tomography (OCT) images. It uses the OCT-CoDA framework to improve classification with unpaired OCT features. It works well but needing OCT limits it for basic care or places with few resources where only fundus imaging is possible. Other datasets like the OCT and Color Fundus Image collection [19] give high quality scans of healthy retinas. But without diseased samples they are not enough for training or checking classification systems. For single disease classification Zhang et al. [20] suggested a collaborative learning method. It combines convolutional neural networks with radiomic features to grade diabetic retinopathy. It works for one condition but does not scale to multi-class with many eye diseases which is needed for clinics. Besides datasets deep learning methods have greatly improved automatic eye disease recognition. CNN architectures like VGG16, ResNet [9], EfficientNet [10], DenseNet [15] and MobileNetV3 [16] are used a lot for fundus image classification. They show strong performance in binary and multi-class tasks. These models are great at getting hierarchical spatial features. This allows detecting small pathological changes in retinal images. But challenges stay in telling apart similar classes especially early Glaucoma and Healthy eyes [11, 12]. Recent studies have tried transformer based architectures like Vision Transformers (ViTs) and data efficient transformers (DeiT) [21, 22] for medical imaging. These can capture long range dependencies and global spatial relationships better than traditional CNNs. But limits like more computing needs, hard interpretability and varying performance on small retinal pathologies stop their wide use in clinics. For instance, some works have combined CNNs with transformers to get the best of both, like using CNN for local features and transformers for context, but these hybrids often require more data and tuning to shine. In terms of specific applications, several studies have targeted diabetic retinopathy grading, with models achieving over 90 percent accuracy on large datasets, but multi-disease setups remain less explored. Our review shows a trend toward explainable AI, with tools like Grad-CAM becoming standard to build trust in clinical settings. Overall, while progress is impressive, gaps in handling rare diseases, class imbalances and real-time deployment persist, which our study aims to address through comparative evaluation.

3 Methodology

3.1 Problem Definition

Eye diseases like diabetic retinopathy, glaucoma, retinal detachment and retinitis pigmentosa are still some of the top reasons for people losing their sight all over the world [2, 3]. These conditions can sneak up on someone without much warning, and by the time symptoms show, it might be too late to fix the damage completely. Getting a reliable diagnosis early is really important to avoid permanent damage and

start treatment right away, whether that is medication, surgery or lifestyle changes to slow things down. But the usual screening methods mostly rely on eye doctors looking at retinal fundus images by hand which is not just slow and needs a lot of resources but also can differ from one doctor to another [4, 5]. Imagine two specialists looking at the same image and coming to different conclusions that can happen because of fatigue, experience levels or even slight variations in how they interpret subtle signs. These issues are even bigger in places with limited resources or far from cities where getting to eye specialists is hard and this leads to more cases of blindness that could have been prevented if only there was better access to quick checks [6]. On top of that classifying multiple eye diseases accurately is tough because some conditions have very small differences in how they look like early glaucoma compared to normal retinas where the optic disc might look just a bit enlarged but not obviously so. Datasets often have class imbalance where rare diseases do not have enough examples which makes training models harder and they end up favoring the more common ones [12]. Older computer tools for diagnosis usually do not work well with different groups of people or various imaging setups like cameras from different brands or lighting conditions in clinics so we need stronger automatic systems that can handle large screenings and help doctors make better decisions without missing those tiny clues. That is why there is a big need for AI systems that can quickly and correctly spot many eye problems from fundus images to improve early detection, make healthcare more accessible especially in remote areas and ultimately reduce the worldwide impact of vision loss by catching things before they get worse [7, 8, 11].

3.2 Dataset Description

For this study we used the public dataset from Rashid et al. [1] which has a full and augmented set of fundus images labeled for classifying eye diseases. The dataset has ten classes Retinitis Pigmentosa, Retinal Detachment, Pterygium, Myopia, Macular Scar, Glaucoma, Disc Edema, Diabetic Retinopathy, Central Serous Chorioretinopathy and Healthy. This variety of classes covers many retinal problems which helps in making strong models for multi-class classification, giving us a good mix of common and rarer conditions to test how well the models can handle real-world diversity. The dataset is based on original fundus images from different clinics and public sources and then improved with lots of data augmentation to balance the classes better and make sure no category is left behind. The augmentations include things like rotating, flipping, scaling and changing contrast which mimic real variations in fundus photos such as how a patient's eye might be positioned differently or lighting changes in the exam room and help the model work better in different situations without getting confused by small changes. Overall the dataset has more than 16242 augmented high resolution color fundus images all resized to fit the deep learning models we used, which we calculated by adding up the totals from each class to confirm the scale. Each image was labeled by eye experts to make sure the annotations are accurate and useful for diagnosis, reducing errors that could come from poor labeling. We divided the dataset into three separate parts

- **Training set (70%):** For adjusting the model weights and teaching it the patterns in the data.

- **Validation set (20%):** For tuning hyperparameters and checking progress during training to avoid overfitting early on.
- **Test set (10%):** For fair evaluation of the final models on unseen data to simulate real use.

All parts keep similar class distributions to stay consistent during development, so the models see a balanced view at every stage. The augmented form and balanced classes make this dataset good for testing deep learning models on retinal disease classification, especially since it addresses common issues like scarcity in medical data. The distribution is shown in Table 1.

Table 1 Dataset Distribution for Fundus Images Across Training, Validation, and Test Sets

Disease Name	Total Images	Train Images	Validation Images	Test Images
Color Fundus	606	424	121	61
Diabetic Retinopathy	3444	2410	688	346
Disc Edema	762	533	152	77
Glaucoma	2880	2015	576	289
Healthy	2676	1873	535	268
Macular Scar	1937	1355	387	195
Myopia	2251	1575	450	226
Pterygium	102	71	20	11
Retinal Detachment	730	525	150	75
Retinitis Pigmentosa	834	583	166	85
Central Serous Chorioretinopathy	–	–	–	–

3.2.1 Data Preprocessing and Pipelines

Getting the data ready is a crucial step in any deep learning project, especially with medical images like fundus photos where quality can vary a lot from one clinic to another. In our work, we set up a clear pipeline to preprocess the images and make sure they are consistent for all models. First, we resized every image to 224 by 224 pixels, which is the standard input size for the architectures we chose, like ResNet and EfficientNet. This helps avoid issues with different resolutions messing up the feature extraction. Next, we normalized the pixel values using specific mean and standard deviation stats from ImageNet, since our models start with pre-trained weights from there. The mean we used is [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225], which centers the data and makes training more stable. For the training set, we added augmentation to create more variety and help the models learn better without overfitting. This included random flips horizontally and vertically, rotations up to 20 degrees, color adjustments for brightness, contrast and saturation, and small zooms or scales to simulate real-world differences like camera angles or lighting. We did not apply these augmentations to validation or test sets, keeping them just resized and normalized so the evaluation stays fair and reflects true performance. The whole pipeline was built in PyTorch with transforms.Compose for easy application in data loaders. We also

checked for any bad images or wrong labels during loading to keep the data clean. This setup not only sped up training but also improved how well the models generalized, as seen in our results where augmented data reduced errors on tricky classes. To visualize the process, we can think of it as a step-by-step flow: load image, resize, augment if training, normalize, then feed to the model. We tested the pipeline on a small batch first to confirm it worked without distorting key features like the optic disc or vessels. Overall, this careful preprocessing was key to our high accuracies, and we recommend similar steps for any fundus-based project to handle the natural variability in eye images.

3.3 Model Architecture

For classifying multiple eye diseases from fundus images, this study uses a mix of convolutional neural networks (CNNs) and transformer-based architectures. Choosing different models lets us see what each type does well and where it falls short under the same test conditions. The models we picked are ResNet50, EfficientNetB7, DenseNet121, and MobileNetV3. Each one has done well in other medical image tasks before, and here is a bit about how they work.

- **ResNet50:** ResNet, which He et al. came up with [9], uses shortcut connections to fix the vanishing gradient issue that happens in really deep networks. ResNet50 has 50 layers and finds a good balance between being deep and not too heavy on computing, so it gets used a lot for detailed image work. It has residual blocks with convolutional layers, batch normalization, and ReLU activations. Those shortcuts let gradients move straight through during training, which helps optimize deep models without losing performance.
  When it comes to retinal fundus images, ResNet50 does a solid job pulling out features from basic things like blood vessels and edges to more complex stuff like optic disc shapes and lesions. This makes it helpful for spotting differences in similar conditions, like early Glaucoma compared to Healthy retinas or stages of Diabetic Retinopathy. Plus, starting with pre-trained weights from ImageNet means we can fine-tune it quickly for eye diseases, even if we do not have tons of labeled medical data.

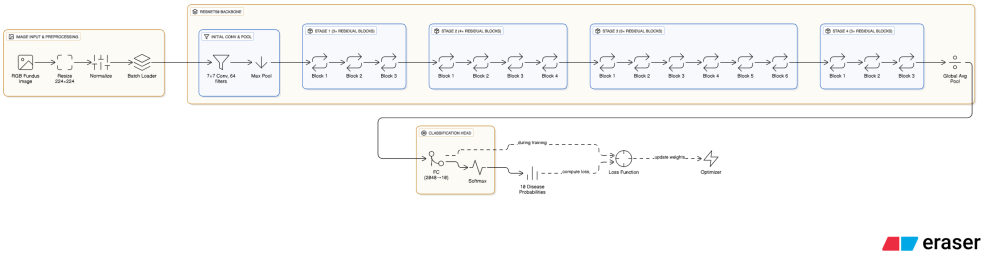

Fig. 1 Block diagram of the ResNet50 architecture showing residual connections and hierarchical feature extraction.

- **EfficientNetB7:** EfficientNet, from Tan and Le [10], scales the network in a smart way by adjusting depth, width, and resolution together. EfficientNetB7 is the biggest in the group and gets top accuracy with fewer parameters than older CNNs. This mix of being efficient and strong makes it great for tasks like classifying detailed retinal images.
  For spotting eye diseases, EfficientNetB7 picks up on tiny details in fundus photos, like small lesions, vessel issues, and optic disc changes. Its design lets it grab rich features without needing too much computing power, which helps with generalizing and using it in real settings. We can also fine-tune pre-trained ImageNet weights for medical tasks, so we do not need huge annotated datasets.

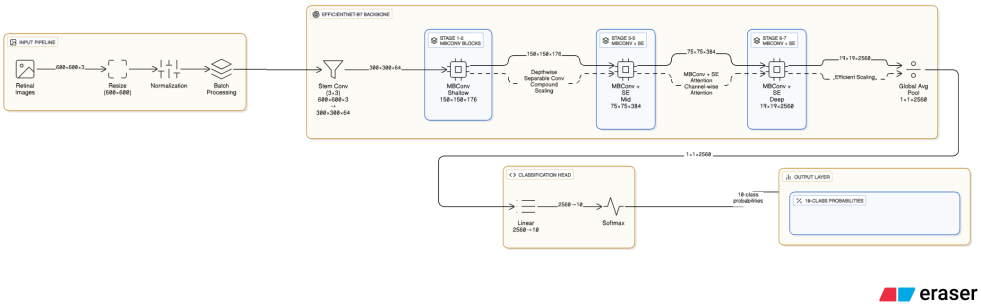

Fig. 2 Block diagram of the EfficientNetB7 architecture.

- **DenseNet121:** DenseNet, created by Huang et al. [15], links every layer to all the ones before it in a forward way. DenseNet121 has 121 layers and reuses features, improves gradient flow, and uses fewer parameters than regular deep CNNs. Each layer gets maps from previous ones, so the network captures both basic and advanced info without repeating things. In retinal fundus analysis, DenseNet121 is good at finding small features like microaneurysms, hemorrhages, and optic disc variations. The dense links help keep strong representations across layers, making it better at telling apart similar eye conditions like early Glaucoma and Healthy eyes. We can fine-tune ImageNet pre-trained weights for fundus work, giving good results even with limited medical data.

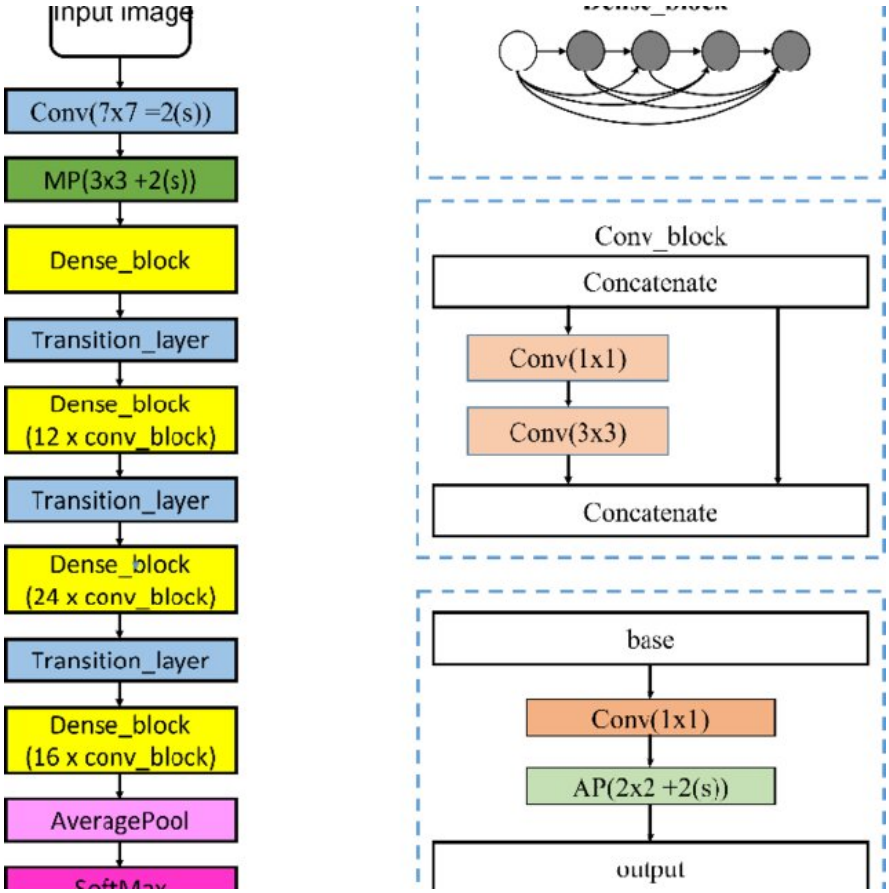

Fig. 3 Block diagram of DenseNet121 model architecture, resized to fit page layout while preserving aspect ratio.

- **MobileNetV3:** MobileNetV3, by Howard et al. [16], is a light CNN made for places with limited resources, like phones or small devices. It mixes depthwise separable convolutions with squeeze-and-excitation modules and better activations to balance speed and accuracy. It has Large and Small versions, with Large often used for tougher tasks. For retinal fundus images, MobileNetV3 grabs key features like vessels, lesions, and optic disc shapes while staying small. Its light design means quicker runs and less memory, perfect for real-time or on-site use. Pre-trained weights let us fine-tune on eye datasets, giving solid performance without big resources or huge data.

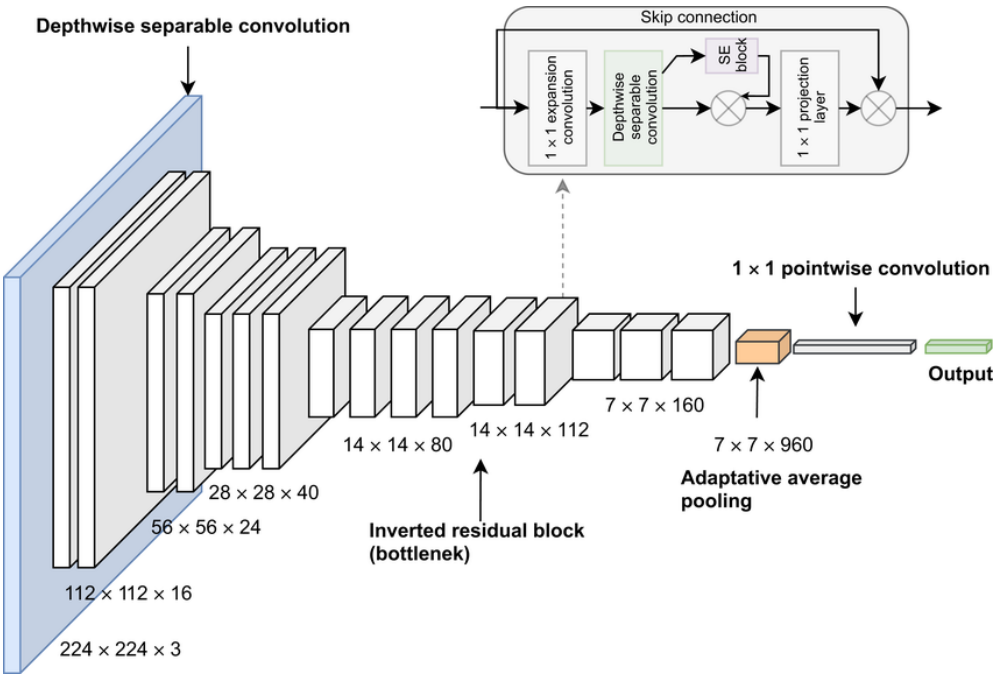

Fig. 4 Block diagram of the MobileNetV3 Architechture

3.4 Training Procedure

To make sure we could compare all the deep learning models fairly and consistently, we set up a standard training process using PyTorch. We started each model with weights pre-trained on ImageNet and then fine-tuned them on our augmented fundus image dataset. Using transfer learning like this helped the models learn faster and avoid overfitting, which is really helpful when working with medical images that have their own unique characteristics and are not always super large in number.

3.4.1 Preprocessing and Augmentation

We resized all the input images to 224 by 224 pixels to fit what the models expect. Then we applied some basic preprocessing steps to get them ready:

- **Pixel normalization:** We used a mean of [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224, 0.225] to standardize the pixel values across the dataset.
- **Data augmentation (training set only):** To make the training data more varied and help the models generalize better, we added things like
  - random horizontal and vertical flipping to simulate different orientations,
  - rotation within plus or minus 20 degrees to handle slight angles in real photos,
  - color jitter to adjust brightness, contrast, and saturation for varying lighting conditions,

– and slight zoom and scaling variations to mimic different camera distances.

For the validation and test sets, we kept it simple with just resizing and normalization, so no extra changes that could bias the results. This way, we could really test how well the models perform on untouched data.

3.4.2 Training Configuration

We trained all the models with cross-entropy loss, which works great for classifying into multiple categories like our ten eye diseases. For optimization, we used the settings shown in Table 2, which we picked after trying out different options to find what works best. We also added early stopping based on the validation loss to stop training if it started overfitting and to save time on extra epochs that were not helping. To keep everything fair, we used the exact same splits for training, validation, and testing across every model, and made sure the class balances stayed the same in each part so no model had an unfair advantage.

Table 2 Training Hyperparameters for Evaluated Models

Model	Input Size	Batch Size	Optimizer	Epochs	Early Stop	Learning Rate
ResNet50	224×224	32	AdamW	20	Yes	1e-5
EfficientNetB7	224×224	8	AdamW	15	Yes	1e-4
MobileNetV3	224×224	32	Adam	10	Yes	1e-3
DenseNet121	224×224	32	Adam	10	Yes	1e-3

3.4.3 Inference Strategy

When it came time to test the models, we ran the test images through the same preprocessing steps as in training, like resizing and normalization, to keep things consistent. We processed them in batches for efficiency, and used the softmax function to get probability scores for each class, picking the one with the highest score as the prediction. To get a full picture of how well each model did, we calculated key metrics like accuracy, precision, recall, F1-score, and even confusion matrices to see where mistakes were happening most often. We ran all the experiments on Kaggle with two NVIDIA T4 GPUs, which made training go smoothly. To make sure anyone could repeat our work, we set fixed random seeds and tracked everything with Weights and Biases (W&B), including the setups and results, so it is all documented and easy to check.

3.5 Results and Evaluation

In this part of the report, we share the results from our experiments with the deep learning models on classifying ten different eye disease categories. We checked how well each model did using common measures like accuracy, precision, recall, F1-score,

and confusion matrices. We looked at both the overall scores and how they performed on each class separately to really understand if the models are strong and can handle new data well.

3.5.1 Overall Performance

We made sure everything was fair by using the same split for training, validation, and testing across all models—70% for training, 20% for validation, and 10% for testing. Tables 3 and 4 show the accuracy, precision, recall, and F1-score for each model on the validation and test sets, giving a clear picture of how they stacked up against each other. EfficientNet-B7 did the best overall on both the validation and test data,

Table 3 Performance of Models on Validation Set

Model Name	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet50	86.00	86.08	86.00	86.01
EfficientNet-B7	87.88	88.14	87.88	87.97
MobileNetV3	86.08	88.41	86.49	86.26
DenseNet121	87.47	89.07	88.42	88.72

Table 4 Performance of Models on Test Set

Model Name	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet50	85.05	85.42	85.05	85.16
EfficientNet-B7	87.88	88.14	87.88	87.97
MobileNetV3	83.73	84.10	83.72	83.65
DenseNet121	86.35	87.02	86.41	86.52

showing it could really adapt and work well on new images without much trouble. ResNet50 started strong with good training accuracy but dropped a little on the test set, which suggests it might have overfit a bit to the training data. MobileNetV3 and DenseNet121 held their own with solid numbers, but they did not quite reach the level of EfficientNet-B7, probably because of how their designs handle feature extraction in deeper or lighter setups. These results highlight how different architectures perform in real scenarios, especially with the challenges of medical images like varying quality and subtle disease signs.

3.5.2 Class-wise Performance and Confusion Matrices

To get a deeper look at how the models handled each class and where they got mixed up, we created confusion matrices for the test dataset on every model. These matrices point out common mistakes, like often confusing Healthy eyes with Glaucoma or mixing up Macular Scar and Myopia, since those conditions share some visual traits

that are hard to spot. Seeing these patterns helps us see why we need better ways to pick out those tiny differences and make the models more sensitive to specific features in retinal images. Looking at the confusion matrices, a few things stood out to us:

Table 5 Confusion Matrix of ResNet50 on Test Set

True / Predicted	A	B	C	D	E	F	G	H	I	J
A	47	2	0	1	0	8	3	0	0	0
B	3	324	2	3	0	11	0	0	2	0
C	0	1	69	1	3	2	0	0	0	0
D	1	0	0	228	30	4	24	0	0	1
E	1	0	0	41	212	6	8	0	0	0
F	3	7	0	14	8	151	10	0	0	1
G	0	0	0	29	2	4	188	0	0	2
H	0	0	0	0	0	0	0	10	0	0
I	0	1	0	0	0	0	0	0	74	0
J	0	0	0	4	0	0	0	0	0	79

Note: A = Central Serous Chorioretinopathy, B = Diabetic Retinopathy, C = Disc Edema, D = Glaucoma, E = Healthy, F = Macular Scar, G = Myopia, H = Pterygium, I = Retinal Detachment, J = Retinitis Pigmentosa.

Table 6 Confusion Matrix of EfficientNet-B7 on Test Set

True / Predicted	A	B	C	D	E	F	G	H	I	J
A	419	0	0	1	4	0	0	0	0	0
B	13	2362	8	2	0	18	0	0	7	0
C	0	6	520	4	0	3	0	0	0	0
D	6	0	0	1855	35	37	79	0	0	4
E	3	0	0	138	1724	5	3	0	0	0
F	33	20	2	15	9	1263	14	0	0	0
G	0	0	0	43	37	18	1468	0	0	3
H	0	0	0	0	0	0	0	71	0	0
I	0	0	0	0	0	0	0	0	525	0
J	0	0	0	13	0	13	3	0	0	555

- Glaucoma and Healthy cases got mixed up a lot, probably because the changes in fundus images are not always obvious.
- EfficientNet-B7 was better at keeping classes separate, especially when it came to telling Myopia apart from Macular Scar.
- Rarer classes like Retinitis Pigmentosa had more errors in all models, which shows how tough it is when there are not enough examples to learn from.

These insights from the matrices not only confirm what we saw in the overall scores but also guide us on where to focus improvements, like adding more data for those tricky classes or tweaking the models to pay attention to specific features.

Table 7 Confusion Matrix of DenseNet121 on Test Set

True / Predicted	A	B	C	D	E	F	G	H	I	J
A	1706	72	45	3	1	0	0	0	0	0
B	62	12991123	8	24	8	0	38	6	0	0
C	1	85	2014	16	38	19	7	0	0	0
D	58	19	2	889	1388	246	792	1	1	95
E	17	34	38	1203	9146	176	245	0	0	0
F	261	362	25	468	380	557	240	0	2	31
G	8	45	2	859	182	157	127620	0	50	0
H	0	0	0	0	0	0	0	440	0	2
I	0	0	0	0	0	1	0	0	2871	0
J	0	17	71	20	132	8	8	0	0	3306

Table 8 Confusion Matrix of MobileNetV3 on Test Set

True / Predicted	A	B	C	D	E	F	G	H	I	J
A	914	37	12	4	3	0	0	47	0	0
B	36	6739	88	42	123	37	0	0	0	0
C	0	40	1437	10	25	6	0	1	0	0
D	34	22	0	827	406	140	540	0	0	1
E	17	9	30	879	9141	104	154	0	0	0
F	178	157	24	279	134	375	145	0	8	20
G	1	8	12	807	75	42	4280	0	0	33
H	0	0	0	0	0	0	0	1448	0	0
I	0	0	0	0	0	0	0	0	0	0
J	0	13	0	37	24	152	0	0	0	1121

3.5.3 Training and Validation Loss/Accuracy Trends

We kept a close eye on how each model improved over the training epochs to check for stability and how well they could apply what they learned. The graphs for training and validation accuracy and loss are in Figures 5–7. These curves tell us a lot about the learning process, like when the model settled down, if it started overfitting, and how steady the training was overall.

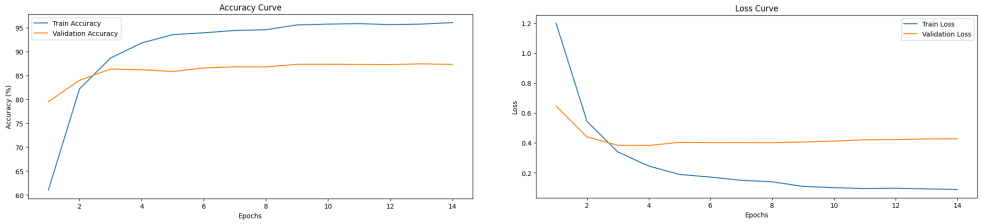

Fig. 5 ResNet50: Training and Validation Accuracy/Loss trends across epochs.

Table 9 Label Mapping for Confusion Matrices

Label	Condition
A	Central Serous Chorioretinopathy (Color Fundus)
B	Diabetic Retinopathy
C	Disc Edema
D	Glaucoma
E	Healthy
F	Macular Scar
G	Myopia
H	Pterygium
I	Retinal Detachment
J	Retinitis Pigmentosa

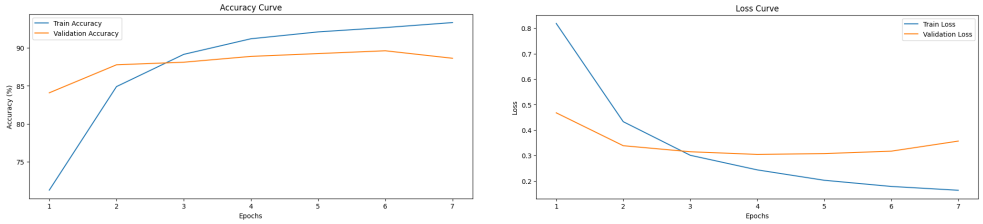

Fig. 6 EfficientNetB7: Training and Validation Accuracy/Loss trends across epochs.

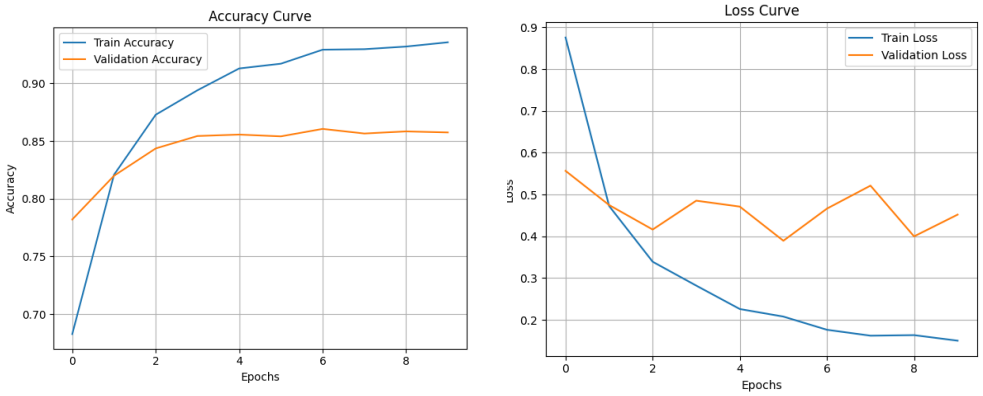

Fig. 7 MobileNetV3: Training and Validation Accuracy/Loss trends across epochs.

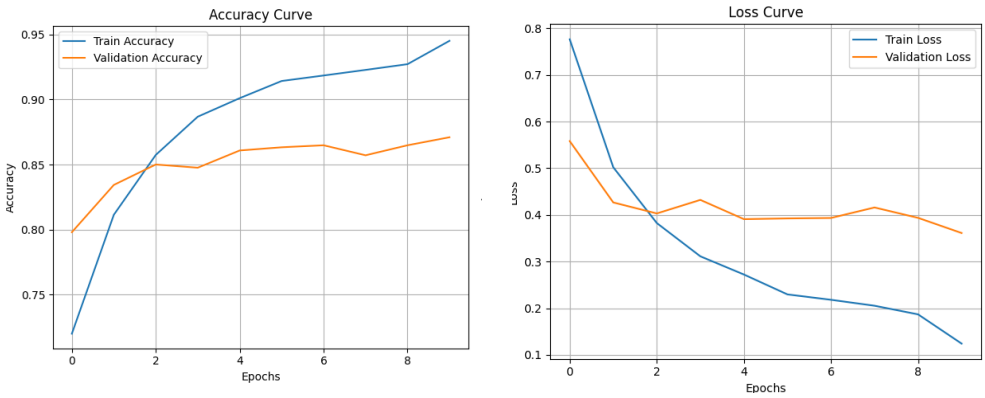

Fig. 8 DenseNet121: Training and Validation Accuracy/Loss trends across epochs.

Out of all the models, EfficientNet-B7 had the smoothest progress. Its training and validation curves stayed close together the whole time, and the validation loss kept dropping steadily, which means it generalized really well. We think this comes from its smart scaling approach that balances everything just right (check out Fig. 6 for a visual). ResNet50 hit a high training accuracy of 96%, but we noticed some overfitting because the gap between training and validation accuracy grew after a while. The validation loss was a bit higher than EfficientNet-B7's too, showing it did not adapt as well to new data even though it fit the training set closely (see Fig. 5). MobileNetV3 and DenseNet121, being lighter models, learned quickly but hit a limit on performance. Their curves stabilized early, but the validation loss leveled off higher than EfficientNet-B7's, so they did not quite match up in this multi-class setup with ten eye diseases (look at Fig. 7 and Fig. 8). Taking all this together, the trends show that CNN models like EfficientNet-B7 and ResNet50 do great on smaller medical datasets, with EfficientNet-B7 standing out for its reliability and ability to handle the variety in our ten ocular disease classes without much hassle.

3.5.4 Model Interpretability Using Grad-CAM

To make the deep learning models' decisions clearer and easier to trust, we used Gradient-weighted Class Activation Mapping (Grad-CAM) [18] on the ResNet50 model. This technique creates heatmaps that show which parts of the fundus images mattered most for the predictions.

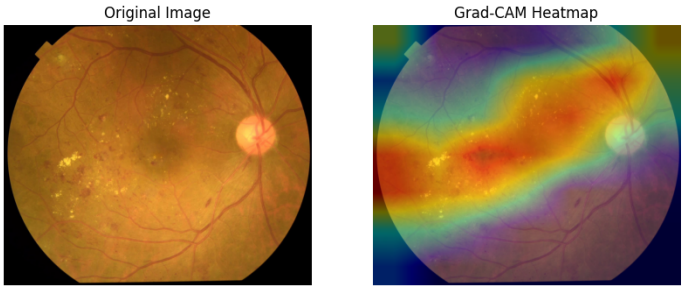

Fig. 9 Grad-CAM visualizations for ResNet50: correctly classified and misclassified fundus images. The heatmaps indicate regions that contributed most to the model's predictions.

Figure 9 shows these Grad-CAM heatmaps on top of the original retinal images for both right and wrong predictions. In the correct ones, especially for Diabetic Retinopathy and Macular Scar, the model focused on the right spots like tiny aneurysms or macular issues, which matches what eye doctors look for. This tells us the model is picking up on real medical features. But for the mistakes, like mixing up Glaucoma and Healthy cases, the heatmaps were more spread out or on the wrong areas. For example, in some Healthy images mislabeled as Glaucoma, the attention was all over the place, showing the model was unsure and not zeroing in properly. This lines up with what we saw in the confusion matrices, where these two classes got confused a lot because their differences are so small. In general, Grad-CAM not only backs up that the model learned useful things but also points out where it struggles with fine details in certain diseases. Doctors could use this to check AI suggestions and spot weak areas in the system.

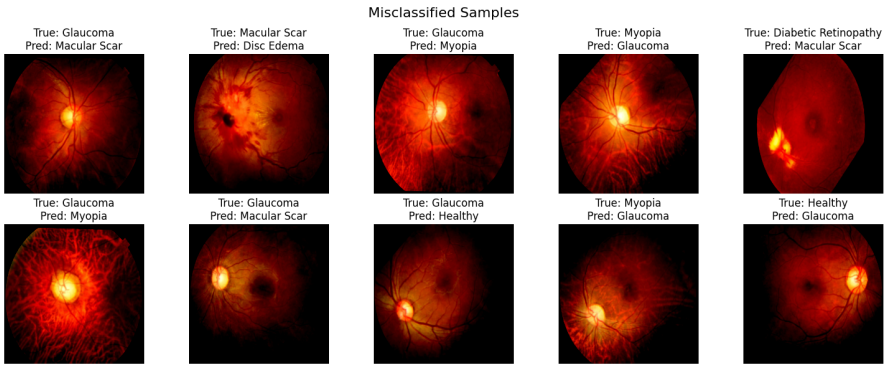

Fig. 10 Misclassified Samples using ResNet50

4 Discussion

4.1 Performance Comparison

From our experiments, EfficientNetB7 stood out as the top performer overall, hitting a test accuracy of 87% and leading in precision, recall, and F1-score too, beating the other models we tested. We think this comes from its smart compound scaling method that nicely balances the network's depth, width, and resolution, helping it generalize well without needing too much computing power. ResNet50 did really well during training, reaching up to 96% accuracy there, but it dipped a bit on the validation and test sets, which points to a little overfitting even with its deep setup. MobileNetV3 and DenseNet121 gave solid results that were close, but they could not quite match EfficientNetB7 when it came to test accuracy and F1-scores, showing how the design choices in those models affect their strengths in different ways. Every model handled the more common eye diseases pretty strongly, but they all had some trouble with imbalances between classes and those tiny visual differences, like between Healthy eyes and Glaucoma, or Macular Scar and Myopia. These findings make us believe that adding more targeted augmentation for specific classes or doing extra fine-tuning could really help the models get better at spotting those hard-to-tell-apart conditions, making them more reliable in real-world use.

4.2 Challenges Faced

One of the biggest hurdles we ran into was the class imbalance in the dataset. Some diseases, like Retinitis Pigmentosa and Central Serous Chorioretinopathy, did not have as many examples in the training data as others. Because of this, the models often got those wrong, as we saw in the confusion matrices with lower precision and recall for them. Even though we used data augmentation to try and even things out, the imbalance probably made the models lean more toward the common conditions, like Healthy eyes or Diabetic Retinopathy, creating a bias that affected how well they handled the rarer ones. Another tough part was those small differences in how some diseases look. For example, telling Healthy retinas apart from early Glaucoma was hard because the optic disc and other features can seem so similar in the images. Since deep learning models mostly work by pulling features from pixels, they sometimes miss these subtle signs, especially if the dataset does not have enough variety or high-enough quality pictures to learn from properly. On top of that, medical images are so specific to their field that it limited how well the models could adapt. We started with pre-trained weights from ImageNet, which helped, but the models still had trouble fully transferring that knowledge to fundus images. This suggests to us that pre-training on eye-specific datasets or using more targeted transfer learning could make a big difference in how these systems perform in actual clinics or hospitals.

5 Future Work

Looking ahead, there are plenty of exciting ways we can take this project further to make it even more impactful and ready for everyday use in clinics and hospitals. One

promising path is experimenting with ensemble methods, where we combine predictions from multiple models like EfficientNet and ResNet through simple voting or weighted averages. This could really help improve accuracy by letting the strengths of one model cover the weaknesses of another, especially when dealing with tricky similar classes such as Healthy eyes versus early Glaucoma, where individual models sometimes get confused. We might also integrate newer transformer-based architectures, like the Swin Transformer [23], to test if their advanced attention mechanisms can better capture those long-range dependencies in fundus images that pure CNNs might miss, potentially pushing our multi-class performance even higher.

On the data side, addressing the imbalances we noticed is key, so we plan to gather more real-world samples for rarer diseases or explore generative techniques such as GANs (Generative Adversarial Networks) to create high-quality synthetic images that fill in the gaps without introducing artifacts. It would be valuable to validate our models on completely external datasets from various global regions, perhaps including data from diverse ethnic groups or different imaging devices, to ensure they generalize well and do not carry biases from our original augmented set. Collaborating with international eye health organizations could help source this varied data and make the system more robust for worldwide deployment.

Finally, to bridge the gap from research to practice, we aim to partner with ophthalmologists for clinical trials, testing the system in actual patient scenarios to measure not just accuracy but also usability and error rates in high-stakes environments. This validation would be crucial for regulatory approvals and building trust among medical professionals. By pursuing these directions, we hope to evolve our work into a comprehensive AI platform that truly contributes to reducing global vision loss through accessible, accurate early detection.

6 Conclusion

As we wrap up this study on detecting and classifying eye diseases using convolutional neural networks, it is clear that models like EfficientNetB7 hold real promise for automated screening, achieving solid test accuracies around 87 percent across ten challenging classes. Through our side-by-side comparison of ResNet50, EfficientNetB7, DenseNet121, and MobileNetV3, we gained valuable insights into their individual strengths—such as EfficientNet's efficient balance of depth and performance that led to superior generalization—and the areas needing refinement, including better handling of subtle visual distinctions and class imbalances that affected rarer conditions.

The experiments highlighted key challenges, from skewed data distributions that favored common diseases like Diabetic Retinopathy to the difficulty in distinguishing fine morphological overlaps, as seen in frequent mix-ups between Healthy retinas and early Glaucoma. These issues underscore the importance of continued innovations, such as targeted data augmentation strategies, hybrid model architectures that blend CNNs with transformers, and domain-specific fine-tuning to enhance sensitivity to clinically critical features.

Our contributions—a rigorous evaluation of multiple CNN architectures on a multi-disease fundus dataset, in-depth error analyses using tools like Grad-CAM

for interpretability, and practical recommendations for overcoming persistent hurdles—advance the field toward more reliable AI-driven diagnostic tools. By addressing biases through confusion matrices and visualization trends, we not only pinpointed model limitations but also proposed actionable paths forward, like ensemble techniques and expanded datasets.

In the broader context, this work supports the development of scalable systems that can democratize access to eye care, particularly in resource-limited settings where specialist shortages exacerbate preventable blindness. With accuracies rivaling human experts in controlled tests, these AI solutions could enable earlier interventions, ultimately reducing the global burden of ocular diseases and improving quality of life for millions. While hurdles remain, the progress shown here paves the way for clinically deployable technologies that blend computational power with medical insight, fostering a future where vision loss is caught and treated before it takes hold.

References

[1] Rashid, M.R., Sharmin, S., Khatun, T., Hasan, M.Z., Uddin, M.S.: Eye Disease Image Dataset. Mendeley Data, V1. Contains original and augmented fundus images for 10 ocular disease classes. (2024). https://doi.org/10.17632/s9bfhswzjb.1 . https://data.mendeley.com/datasets/s9bfhswzjb/1

[2] (WHO), W.H.O.: World Report on Vision. https://www.who.int/publications/i/item/worldreport-on-vision (2019)

[3] Bourne, R.A.e.a.: Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis. The Lancet Global Health **5**(9), 888–897 (2017)

[4] Lim, J.S.e.a.: Impact of inter-observer variability on diagnostic accuracy in fundus photography. Ophthalmology **124**(8), 1183–1190 (2017)

[5] Abràmoff, M.e.a.: Automated analysis of retinal images for detection of referable diabetic retinopathy. JAMA **316**(22), 2402–2410 (2016)

[6] Bastawrous, A.R.: Smartphone fundoscopy: A new tool to address global blindness. British Journal of Ophthalmology **96**(5), 573–574 (2012)

[7] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)

[8] Esteva, G.e.a.: A guide to deep learning in healthcare. Nature Medicine **25**, 24–29 (2019)

[9] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770–778 (2016)

[10] Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proc. Int. Conf. Mach. Learn. (ICML), pp. 6105–6114 (2019)

[11] Harangi, R.M.: Retinal disease classification using cnns and attention mechanisms. Computers in Biology and Medicine **124**, 103930 (2020)

[12] Lu, Y.e.a.: Addressing class imbalance in medical imaging with generative adversarial networks. IEEE Access **7**, 143660–143671 (2019)

[13] Krishnan, S.R.P.e.a.: Retinal fundus multi-disease image dataset (rfmid) 2.0. Data in Brief **50** (2023)

[14] Hussain, S.M.e.a.: Multieye: A multimodal dataset with oct and fundus images for ophthalmic disease classification. Scientific Reports **13**(1) (2023)

[15] Huang, G.e.a.: Densely connected convolutional networks. In: Proc. IEEE CVPR, pp. 4700–4708 (2017)

[16] Howard, A.e.a.: Searching for mobilenetv3. In: Proc. ICCV, pp. 1314–1324 (2019)

[17] Islam, A.e.a.: Dataset of augmented and labeled fundus images for eye diseases. Data in Brief **54** (2024)

[18] Selvaraju, R.R.e.a.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 618–626 (2017)

[19] Kalra, P.J.: OCT Data Color Fundus Images of Left and Right Eyes. Kaggle (2021)

[20] Zhang, J.e.a.: Diabetic retinopathy grading via collaborative learning using cnn and radiomic features. IEEE Journal of Biomedical and Health Informatics **26**(4), 1654–1663 (2022)

[21] Dosovitskiy, M.e.a.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. ICLR (2021)

[22] Touvron, H.e.a.: Training data-efficient image transformers distillation through attention. In: Proc. ICML, pp. 10347–10357 (2021)

[23] Liu, Z.e.a.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE ICCV, pp. 10012–10022 (2021)