American International University-Bangladesh (AIUB)

Automated Eye Disease Detection Using Machine Learning Algorithms

Md Amir Habib
(19-41429-3)
Ratul Bhattacharjee
(19-41794-3)
Sazin Israk Prioty
(19-41635-3)
Shahrier Hossain
(19-41628-3)

A Thesis submitted for the degree of Bachelor of Science (BSc)
in Computer Science and Engineering (CSE) at
American International University Bangladesh in July 2021

Faculty of Science and Technology (FST)

Abstract

Eye diseases can have a wide variety of symptoms that differ from one another. Throughout history new eye diseases have been emerging, eye disease does not only affect visual impairment but also tends to slowly take away an individual's freedom to be independent. Furthermore, children today are more susceptible to eye diseases also, which not only affects their sight but moreover makes them disabled for the rest of their lives so if early detection of an emerging eye diseases is done swiftly and efficiently it would allow doctors to treat them as swift and quick too. It is estimated that visual impairment causes a $411 billion (about $1,300 per person in the US) annual loss in productivity worldwide. Some common eye diseases include glaucoma, diabetic retinopathy, and age-related macular degeneration. To accurately detect these diseases, it is important to analyze many different symptoms and look for patterns that might indicate a particular disease and notify the authorities as soon as possible. This research focuses on using machine learning algorithms to analyze retina scans for the detection of eye diseases. The proposed model uses a convolutional neural network (CNN) VGG-19 model, transfer learning, and k-nearest neighbors (KNN) image classification to accurately identify different eye diseases. By training these algorithms on large datasets of retina scans, researchers can teach them to recognize patterns that might be indicative of a particular disease. This research tends to also use a statistical method of evaluating and comparing learning algorithms by dividing data into two segments i.e., one is used to learn or train a model, and another is used to validate the model ensuring reliable and accurate results which is by using Cross-Validation. This approach is less time-consuming than traditional methods of diagnosing eye diseases by hand, which can be labor-intensive and require much expertise. These models' accuracy is compared to determine which one is most effective for detecting eye diseases.

Declaration by author

This thesis is composed of our original work, and contains no material previously published or written by another person except where due reference has been made in the text. We have clearly stated the contribution of others to our thesis, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support, and any other original research work used or reported in our thesis. The content of our thesis is the result of work we have carried out since the commencement of Thesis.

We acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate we have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Name	**Signature**
Md Amir Habib	
Ratul Bhattacharjee	
Sazin Ishraq Prioty	
Shahrier Hossain	

Approval

The thesis titled **"Automated Eye Disease Detection using machine learning algorithms"** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science & Engineering on **(22/05/2023)** and has been accepted as satisfactory.

. . . .

Dr. Akinul Islam Jony *(Supervisor)*
Associate Professor & Head (UG)
Department of Computer Science
American International University-Bangladesh

. . . .

Dr. Debajyoti Karmaker *(External)*
Associate Professor
Department of Computer Science
American International University-Bangladesh

. . . .

Dr. Akinul Islam Jony
Associate Professor & Head (Undergraduate)
Department of Computer Science
American International University-Bangladesh

. . . .

Professor Dr. Dip Nandi
Associate Dean
Faculty of Science & Information Technology
American International University-Bangladesh

. . . .

Mashiour Rahman
Dean-in-Charge
Faculty of Science & Information Technology
American International University-Bangladesh

Publications included in this thesis

No manuscripts submitted for publication

Other publications during candidature

No Publications during candidature

Research involving human or animal subjects

No animal or human subjects were involved in this research

Contributions by authors to the thesis

List the significant and substantial inputs made by different authors to this research, work and writing represented and/or reported in the thesis. These could include significant contributions to the conception and design of the project; non-routine technical work; analysis and interpretation of research data; drafting significant parts of the work or critically revising it to contribute to the interpretation.

	Md Amir Habib	**Ratul Bhattacharjee**	**Sazin Ishrak Prioty**	**Shahrier Hossain**	**Contribution (%)**
	19-41429-3	*19-41494-3*	*19-41635-3*	*19-41628-3*	
Conceptualization	40	35	15	10	100 %
Data curation	40	35	15	10	100 %
Formal analysis	40	35	15	10	100 %
Investigation	40	35	15	10	100 %
Methodology	35	40	15	10	100 %
Implementation	40	35	15	10	100 %
Validation	40	35	15	10	100 %
Theoretical derivations	35	40	15	10	100 %
Preparation of figures	35	40	15	10	100 %
Writing – original draft	35	35	20	10	100 %
Writing – review & editing	35	35	20	10	100 %

Acknowledgments

To Dr. Akinul Islam Jony, my thesis advisor, and the head of the CSE department, we would like to express our sincere gratitude for all his support, advice, and encouragement during our research process. Sir has been a superb mentor, and his knowledge, commitment, and helpful criticism have been essential in forming my ideas and assisting us in achieving my academic objectives. His priceless knowledge and insights have pushed us to push past our comfort zone and pursue excellence. We are incredibly fortunate to have had the opportunity to work with someone so extraordinary, and we will be eternally grateful to him for his guidance, kindness, and unwavering dedication to my success.

Keywords
Eye disease, Machine Learning, Image Classification

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Mention all the abbreviations and the different symbols that are used in this document.

Abbreviations	
CS	Computer Science
CSE	Computer Science and Engineering
NN	Neural Network
HCI	Human Computer Interaction
NLP	Natural Language Processing
VGG-19	Visual Geometry Group
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbors
	etc.

Symbols	
$\hat{\rho}$	Density operator
®	Convolution
	etc.

Chapter 1

Introduction

1.1 Overview

The field of ophthalmology has developed rapidly in recent years due to the development of modern technologies, including machine learning algorithms. These algorithms can improve the diagnosis of eye diseases by analyzing and analyzing clinical images. Automatic identification of eye diseases can reduce the work of ophthalmologists and increase the accuracy of diagnosis and treatment.

The purpose of this research is to develop and implement a machine learning algorithm for automatic eye recognition. The proposed algorithm uses image processing techniques and various machine learning algorithms such as deep neural networks and support vector machines. The data used in this study include fundus images that are commonly used to diagnose many eye diseases such as diabetic retinopathy, age-related macular degeneration, and glaucoma. The project also includes comparing different machine learning algorithms to identify the most accurate and efficient methods for detecting eye disease. The findings of this study have important implications for the future of ophthalmology and the ability of machine learning algorithms to improve patient outcomes.

A. Eye diseases:

Eye diseases, also called ocular diseases, are a broad category of conditions that affect the eye's structures and may result in vision loss or impairment. Several prevalent eye conditions include:

1. Cataracts: A clouding of the eye's lens that can impair vision and make it difficult to see in dim light or in the presence of lights.
2. Glaucoma: A group of eye conditions known as glaucoma harms the optic nerve and, if untreated, can result in vision loss or blindness.
3. Diabetic retinopathy: A complication of diabetes that harms the retina's blood vessels and impairs vision.

In machine learning: Eye diseases may refer to the use of machine learning algorithms in the context of machine learning to identify and diagnose eye diseases. Many eye health applications use machine learning, including the following.

B. Iris Scan:

Based on the distinctive patterns found in each person's iris, iris scanning is a biometric technique used to identify them. The colored area of the eye that surrounds the pupil is called the iris, and it has intricate patterns that are particular to each person. These patterns are used by iris recognition technology to produce a template or digital signature for each person that can be used for identification. The following actions are typically involved in the iris scanning process:

1. The iris of the subject is photographed in high resolution using a camera. A specialized camera with an elevated level of detail-capture capability is typically used for this.
2. Image processing: The retrieved image is altered to reveal the distinctive iris patterns. To do this, the iris must be separated from the rest of the eye and the patterns must be identified using algorithms.
3. Creating a template or digital signature for a person's iris is based on the patterns found. This template is specific to the person and can be used to identify them.
4. When the person shows up for identification, a fresh iris image is taken and processed in the same manner. To confirm the person's identity, the generated template is compared to the one that was stored.

1.2 Motivation

There is a growing literature on using machine learning algorithms to identify eye diseases. Many studies have shown that these algorithms can be used to diagnose a variety of eye diseases, including glaucoma, macular degeneration, and diabetic retinopathy. Research in use [2] uses algorithms such as random forest, pure Bayesian, decision trees, and neural networks. A study comparing the performance of several different learning machines found that a deep learning method called convolutional neural network (CNN) is the most accurate method for diagnosing glaucoma. CNN was found to be able to detect glaucoma 96 times out of 10.

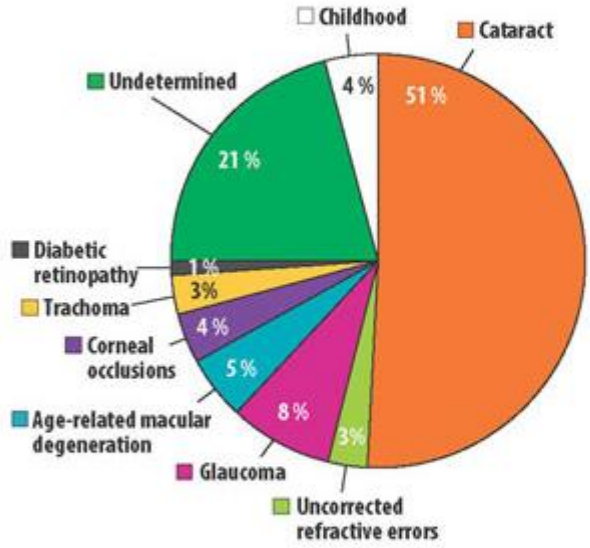

Figure 1*: Eye Diseases Among Countries the World Health Organization (WHO) Global data on visual impairment 2010*

Glaucoma has been called the "big thief" of vision because visual impairment is caused by intraocular pressure on the nerve, which is usually asymptomatic. Therefore, early diagnosis of glaucoma is important to prevent permanent blindness [9]. All eye diseases, not just glaucoma, can cause permanent eye damage if left untreated.

1.3 Background

Machine learning is a subfield of artificial intelligence that deals with the ability of machines to mimic human behavior. Compared to human solutions, complex tasks are performed by artificial intelligence systems. Image processing is a process used to modify an image, get a better image, or extract important information from it. It is a type of signal processing where the input is an image and the output can be the image itself or features. CNN is a network architecture used in deep learning algorithms, especially for image recognition and pixel data processing.

CNNs are the preferred network architecture for object recognition and recognition in deep learning, although there are other differences between neural networks.

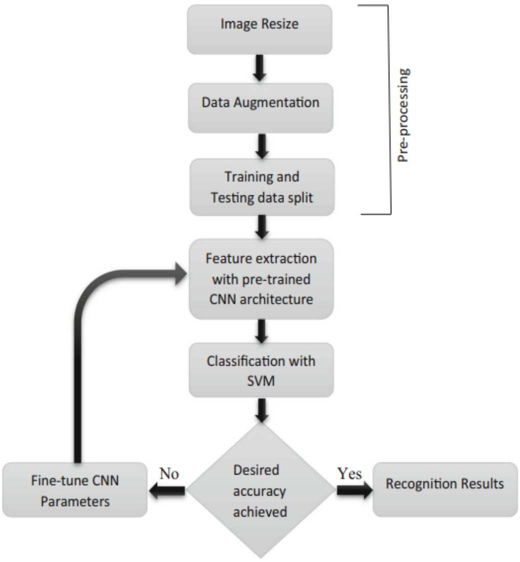

Figure 2: CNN Model working Flowchart Springer link CNN based feature extraction and classification for sign language

Support Vector Machines, also known as SVMs, are one of the most popular learning techniques for classification and regression problems. However, it is mostly used in machine learning problems.

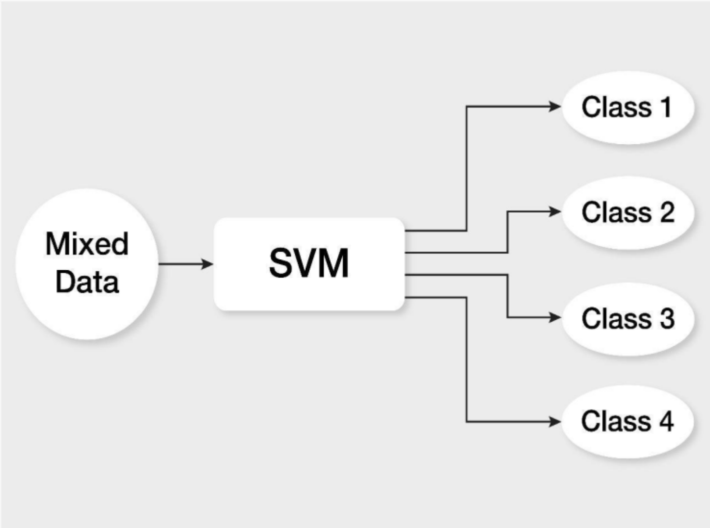

Figure 3: SVM classification flow chart collected from medium

Vision Transformer or ViT (Vision Transformer) is an image distribution model that uses Transformer-like architecture of image patches. The image is divided into large blocks, each block is linearly nested, positional nesting has been added, and the result of vectors is fed into the model Transformer encoder. To perform the classification, a priori techniques are used to add additional learning "classification symbols" to the sequence.

1.4 Research Question

Compared to traditional methods for diagnosis, how accurate is the proposed model for the identification of various eye disorders utilizing convolutional neural networks (CNN) VGG-19 model, transfer learning, and k-nearest neighbors (KNN) imagine classification?

How accurate and effective are different approaches to machine learning, such as deep neural networks and support vector machines, at automatically identifying eye diseases?

How can the proposed method for automatically identifying eye diseases be optimized to increase precision and effectiveness?

1.5 Research Outcome

The study's objective is to establish a more efficient approach to diagnosing eye disorders using retina images. The retina plays a crucial role in vision as it has specialized cells that sense light and transform it into electrical signals transmitted to the brain for interpretation. Any impairment or illness affecting the retina could lead to severe visual impairment or blindness.

With recent technological advances, it is now feasible to acquire high-resolution images of the retina that can be utilized to diagnose and observe eye diseases. The proposed study employs advanced image processing methods and machine learning algorithms to examine the retina images and diagnose eye disorders with precision and efficiency.

The machine learning algorithms used in the research are trained on extensive datasets of retina images and can detect patterns and abnormalities indicative of specific eye diseases. By matching the input retina image's features with the learned patterns, the algorithms can accurately identify the type and severity of the eye disorder.

The advantages of efficiently diagnosing eye diseases utilizing retina images are manifold. It facilitates the early detection and treatment of eye diseases, which can prevent irreversible harm and vision loss. Furthermore, it lowers the diagnosis's time and expense, making it available to a wider range of people. The proposed study, therefore, has the potential to significantly impact the prevention and treatment of eye diseases and enhance the quality of life of numerous individuals.

Chapter 2

Literature review

2.1 Introduction

This chapter provides a comprehensive overview of the literature review. It covers various sectors of the topic such as eye diseases, machine learning and its branches (supervised and unsupervised learning), deep learning and previous work done related to eye disease detection.

2.2 Eye Disease

Eye diseases, also known as ocular diseases, refer to a broad category of conditions or disorders that affect the structure, health, and functioning of the eyes that may result in vision loss or impairment. While some eye diseases are temporary and can be recovered with prescribed medication, some can lead to a permanent loss of vision. Some of the major causes of eye diseases can be age, genetics, environmental factors, medical conditions, infections, injuries, or lifestyle factors. Several prevalent eye conditions include:

A. **Cataract:** It is a common eye condition that occurs when the lens of the eye becomes cloudy or opaque, leading to a decrease in vision. Normally, the lens is clear and flexible, which helps to focus light on the retina at the back of the eye and allows us to see clearly. However, when cataracts develop, the lens becomes cloudy, which interferes with the passage of light causing visual symptoms. Some common risk factors for cataracts include age, exposure to UV radiation from the Sun, certain medical conditions such as diabetes. Symptoms can include hazy vision, sensitivity to light, difficulty seeing at night, seeing halos around lights, and needing brighter lights for reading or other activities.

B. **Glaucoma:** It is a group of progressive, chronic eye diseases that can cause injury to the optic nerve. The optic nerve is a bundle of nerve fibers responsible for transmitting visual information from the retina to the brain. If left untreated, this injury can result in permanent vision loss. Eye pressure is among the leading risk factors. An abnormality in the eye's drainage system can result in the accumulation of fluid, resulting in excessive pressure that damages the optic nerve.

There are three different forms of glaucoma: open-angle, angle-closure, and normal-tension. Glaucoma is characterized by gradual vision loss, blind areas in the periphery of the field of vision, tunnel vision, and eye pain or redness.

C. **Diabetic Retinopathy:** This is an eye disease that can be induced by diabetes-related complications. Over time, elevated blood sugar can cause damage to the blood vessels in the retina, leading to changes in the retina and possibly vision loss.

   a) **Retinopathy:** Nonproliferative diabetic retinopathy is an early stage in which small blood vessels in the retina become damaged and release fluid or blood, causing vision to become blurry or distorted. In the advanced stage of proliferative diabetic retinopathy, the damaged blood vessels in the retina begin to seal, resulting in the growth of new blood vessels. However, these new blood vessels are fragile and can induce blindness if they leak into the eye. Vision may become blurred or distorted, floaters may appear, and low-light vision may be affected.

2.3 Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that entails the development of algorithms and statistical models that enable computers to learn and acquire knowledge without the need for specialized programming. There are numerous forms of machine learning, such as supervised and unsupervised learning.

A. **Supervised learning:** In supervised learning, algorithms are trained on specific domains. Typically, data is separated into training and assessment sets. The algorithm is learned through a training program that utilizes recorded data to identify input-output patterns and relationships. After training the algorithm, its performance on new, unseen data is evaluated using a test set. Linear regression, logistic regression, decision trees, and neural networks are all examples of supervised learning algorithms. There are numerous applications for mindfulness training, ranging from the recognition of images and speech to the processing of words.

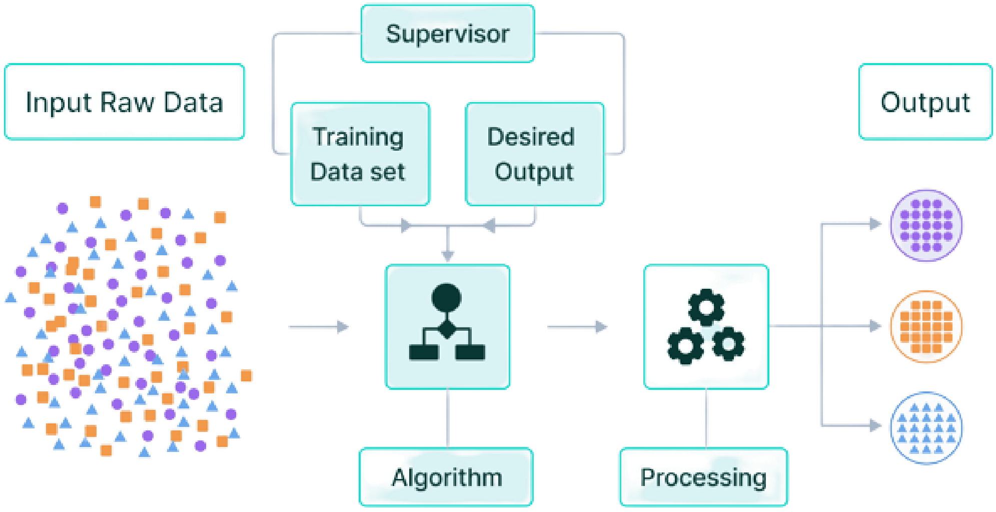

Figure 4: Supervised Learning collected from V7 labs

There are two types of supervised learning:

a) **Regression**: Regression is used to predict continuous values such as numerical values based on input data. Regression algorithms may or may not be linear, depending on the nature of the relationship between input and output. Linear regression is a regression algorithm that uses a linear function to model the relationship between input and output. Nonlinear regression algorithms, on the other hand, can model the relationship between input and output using functions such as polynomial functions, exponential functions, or sigmoid functions. In regression, the quality of a model is usually evaluated using a loss function that measures the difference between the estimated output value given an input and the actual output value. The goal is to minimize performance loss throughout the entire training process by using techniques such as gradient descent or other optimization techniques.

b) **Classification:** Classification is used to predict categorical or discrete forms such as binary classification (yes/no) or multimodal classification (for example, classifying images as cat, dog, or bird). Classification algorithms may or may not be linear, depending on the nature of the relationship between input and output. Linear classification algorithms such as logistic regression or linear discriminant analysis use linear functions to model the relationship between input and output. Nonlinear classification algorithms such as decision trees, random forests or neural networks can use decision boundaries to model relationships between input and output. In classification, the quality of the model is often measured by performance measures such as accuracy, precision, recall, or the percentage of correct predictions, or the F1 score, which measures the trade-off between real advantages and disadvantages. The goal is to improve test performance throughout the entire training using techniques such as gradient descent or other optimization techniques.

B. **Unsupervised learning:** Unsupervised learning means that an algorithm is trained on unsupervised data and must identify patterns and relationships on its own, without clear feedback on its predictions. Here, the algorithm is provided without pre-written or grouped data. The algorithm then analyzes the data to identify similarities and differences between different data and groups them according to their similarities. Clustering is an unsupervised learning method in which an algorithm groups similar data into groups. K-means clustering, principal component analysis (PCA) and association rule mining are unsupervised learning methods.

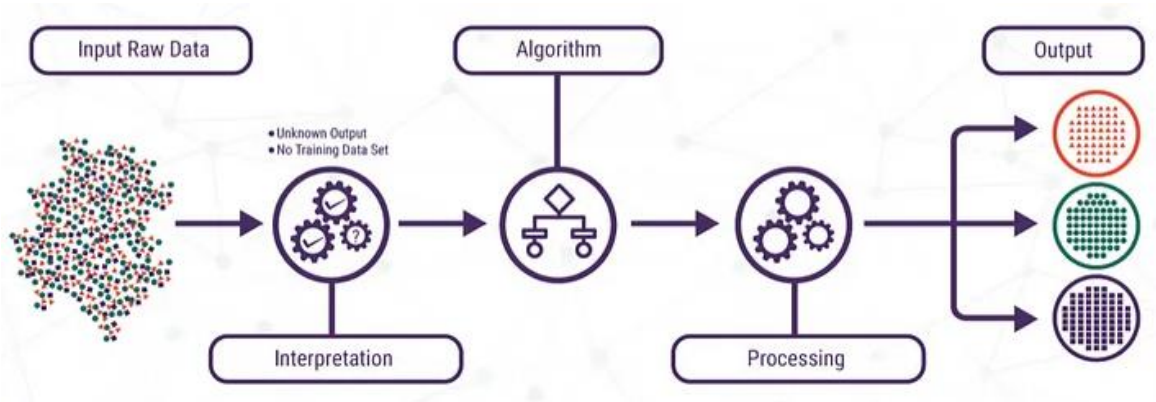

Figure 5: Unsupervised Learning collected from medium

There are two types of unsupervised learning:

a) Clustering: Clustering is used to categorize similar items into groups or segments based on similarity or distance from each other. Clustering algorithms can be classified as hierarchical or non-hierarchical depending on how they group items. Hierarchical clustering algorithms create hierarchies of nested clusters by starting from individual data points and grouping them into larger clusters based on their similarity. Non-hierarchical clustering algorithms such as K-means clustering divide the data into several clusters based on the distance or similarity of the data points. In clustering, the quality of the model is usually evaluated using performance indicators that measure the quality of different groups, such as silhouette scores, intracluster sum of squares (WSS), or intercluster sum of squares (BSS). to the following metrics such as group compactness and separation.

b) **Association:** Association is an unsupervised method in machine learning used to discover patterns and relationships between different objects or objects in data. In integration, the quality of the model is often measured by performance metrics such as support, trust, and growth, which measure the strength and importance of the integration of disparate products. Support measures the frequency of items, confidence measures the repetition rate of the next item given to the previous item, and carry measures how related the two items are to each other.

2.4 Deep Learning

Deep learning is a subfield of machine learning that involves training neural networks to learn on their own without specific instructions and make predictions or decisions. Deep learning algorithms usually involve the creation of neural networks consisting of multiple network layers or neurons. Each layer in the network extracts and transforms features from the input data and sends them to the next layer. The final process in the network produces an output that can be a prediction, decision or classification. Deep learning algorithms require a lot of domain data for training and can leverage powerful computing devices such as GPUs or TPUs to speed up the training process.

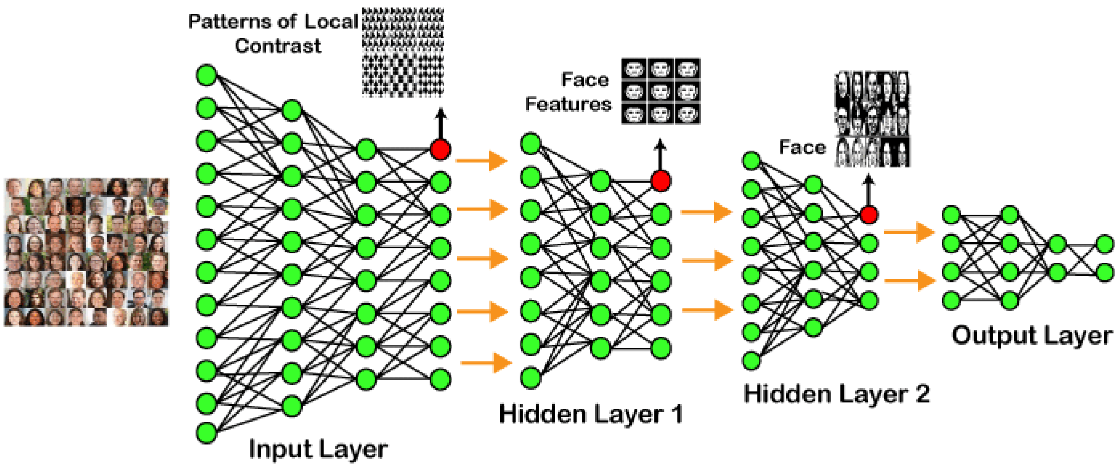

Figure 6: Deep Learning collected from Java point

There are five types of deep learning networks:

A. **Feed Forward Neural Network:** A feedforward neural network is a neural network in which data flows in one direction from the input process through one or more hidden processes until it is released without any comment. Feedforward neural networks usually have an input layer, one or more hidden layers, and an output layer. Each layer has a set of neurons connected by weights to neurons in the previous and next cluster. Neurons in each layer receive inputs from neurons in the previous layer, use an activation function to weight the inputs, and pass the outputs to neurons in the next layer.

B. **Recurrent Neural Network:** A recurrent neural network (RNN) is a neural network designed to process continuous data such as time data or sentences. RNNs have feedback loops that allow them to perform iterative feedback processes by feeding the output of each step back to the network as the input of each previous step. In RNNs, each step of a communication process is processed by a set of neurons maintaining an internal state to capture the details of the previous step. This internal state is updated at each step using an iterative process based on the current input and the previous state. The output of an RNN is usually generated by a set of output neurons that receive input from the internal state of the network.

C. **Convolutional Neural Network:** Convolutional neural networks (CNNs) are often used in image and video recognition, natural language processing, and data-driven tasks. These goals are to learn and extract hierarchical representations from input data using convolutional techniques that apply filters to the input. In a typical CNN architecture, the input data is first passed through a series of convolutional layers that use a set of learnable filters for the input to extract features such as edges, vertices, and texture. The output of each convolutional layer is then passed through a nonlinear activation function, such as a smoothed linear unit (ReLU), which introduces nonlinearities to the network. After the convolution process, the output is usually passed through a layer that shrinks the output to reduce its size and show the comment.

   Layers can use techniques such as maximum pooling, average pooling, or other types of matching. The results of the final layer of pooling are then flattened and passed through one or more layers that perform a series of matrix multiplications and use activation functions to generate the final output of the network. CNNs are useful for image and video recognition as they can learn and extract complex features from input data without special engineering training.

D. **Restricted Boltzmann Machine:** RBMs are models that learn the probability distribution of input data. They consist of a series of visible units and a series of hidden units connected by a set of weights. RBMs work by learning a set of weights that maximize the performance of the input data. RBMs take an input vector and provide an output for each configuration of hidden units. This event is calculated using the Boltzmann distribution, which models the dynamic range as a function of the input data and weights.

   RBMs learn weights by minimizing the difference between the actual distribution of the input data and the estimated distribution of the RBM. RBMs can be trained using different algorithms such as Contrast Deviation or Persistent Contrast Deviation. These algorithms adjust the weights of the RBM by resampling the distribution of hidden and visible objects and adjusting the weights to maximize the log shape of the input data.

E. **Autoencoders:** Autoencoder is a kind of neural network used in unsupervised learning where the purpose is to learn the rules of the input data. An autoencoder has two main parts: an encoder network that maps the input data to a low-level source, and a decoder network that maps the hidden domain back to the original input data. A network encoder compresses the input data into a low-dimensional representation using a non-linear transformation. Next, the decoder network reconstructs the original input data from the compressed representation using a series of inverse transformations. During training, autoencoders are optimized using loss functions such as the square of the error to minimize the difference between the input data and the reconstruction data.

2.5 Related Works

Louis R. Pasquale and Daniel Shu Wei Ting talk about the use of deep learning (DL) in ophthalmology. DL has been utilized in medical imaging analysis in the field of medicine and healthcare. [1] It has also been used in ocular imaging, primarily fundus photography and optical coherence tomography (OCT). In instance, DL has demonstrated clinically acceptable detection accuracy in identifying DR and ROP. Applications for DL can be used in age-related muscle degeneration and diabetic retinopathy. The findings demonstrate DL's most effective uses in ophthalmology.

Research by Raffaele Nuzzi and Giacomo Boscia discusses the impact of artificial intelligence and deep learning on eye diseases. [2] A division of machine learning is deep learning. Artificial intelligence applications for ophthalmology have been demonstrated. The ability of deep learning to automate the screening and diagnosis of common disorders that affect vision, such as glaucoma, age-related macular degeneration (AMD), diabetic retinopathy (DR), and retinopathy of prematurity, has received considerable attention in recent years (ROP).

Vision problems may end up from range of eye conditions, as well as trachoma, cataracts, And membrane ulcers. Only by properly distinguishing eye diseases at an early stage will they be prevented from progressing. [3] Observable symptoms there is an excellent deal of variation among these eye diseases. However, to accurately establish eye diseases, a good range of symptoms should be examined. A novel technique for mechanically identifying eye diseases is given during this paper. The system uses visual indications, digital imaging and machine learning.

This review article presents methods and methods for identifying eye diseases. SVM, DCT, HMM, and PCA are some of the machine learning algorithms and methods used in the research literature. [13] contains a description of the system architecture, image processing model, and model. Diagnosis consists of the following five general tasks.

The application of several image processing algorithms for the automatic detection of glaucoma is described in the review paper. [14] A neurodegenerative condition of the optic nerve called glaucoma results in a partial loss of eyesight. Retinal fundus image fusion, image segmentation, feature extraction, image enhancement, morphology, pattern matching, image classification, analysis, and statistical measurements are used to study the substantial number of persons who suffer from eye diseases.

Major types include diabetic retinopathy, cataracts, and glaucoma. Diabetic retinopathy is a common diabetic eye disease that affects the blood vessels of the retina.

[15] This usually occurs in people who have diabetes for a long time and can lead to vision loss and blindness. Cataract is another type of diabetic eye disease characterized by clouding of the lens of the eye, which can lead to vision loss. Glaucoma, another form of diabetic eye disease, causes increased pressure within the eye, damaging the optic nerve and causing loss of vision. In terms of localization, diabetic eye disease usually affects the retina, lens, and optic nerve, all of which are inside the eye. The technology could identify exudate diseases with an overall diagnostic accuracy of 90.1% and locate the optic disk with an accuracy of 90.7%. Image Processing: This involves the use of techniques such as segmentation, feature extraction, and pattern recognition to analyze and classify retinal images for the detection of diabetic retinopathy. Deep learning: Deep learning, such as convolutional neural networks (CNN), has been used to detect and describe diabetic retinopathy in retinal images.

A category of illnesses known as hereditary retinal eye diseases can damage the central retina and start in childhood or adolescence. Through genetic mutations, these disorders are transmitted from parents to their children. Retinitis pigmentosa, Lebar's congenital amaurosis, and Stargate disease are a few typical instances of inherited retinal eye conditions. [16] These conditions can result in a progressive loss of vision, which may include problems seeing at night and in dim light, a decline in peripheral vision, and a loss of central vision. Hereditary retinal eye illnesses affecting the central retina in children and young adults are diagnosed and treated using various techniques and algorithms. These techniques and algorithms can aid in the better diagnosis, control, and treatment of hereditary retinal eye illnesses that affect the central retina in children and young adults. The course of these disorders can be controlled by early discovery and therapy, such as genetic counseling, routine eye exams, and the use of low vision devices.

Wide-field imaging of the retina is a method of capturing high-resolution images of the entire retina, the light-sensitive layer at the back of the eye. This procedure is used to diagnose and monitor eye conditions and diseases such as age-related macular degeneration and diabetic retinopathy. Optical Coherence Tomography (OCT): OCT is a non-invasive technique that uses light waves to capture coherent images of the retina. [17] Wide-angle imaging of the retina is a method of capturing high-resolution images of the entire retina, the light-sensitive layer at the back of the eye. This procedure is used to diagnose and monitor eye conditions and diseases such as age-related macular degeneration and diabetic retinopathy.

Images captured during the imaging process are processed using computer-based algorithms, including machine learning algorithms, to improve image analysis and diagnostic accuracy.

Ocular hypertension is a condition in which high blood pressure affects the blood vessels of the eye, causing damage to the retina and optic nerve. It usually occurs in people with uncontrolled high blood pressure and is diagnosed by an eye exam. [18] Treatment may include controlling blood pressure through lifestyle changes and medication and having regular eye exams to monitor disease progression. Early detection and appropriate treatment can help prevent vision loss from hypertensive eye disease. The methods and algorithms used to diagnose and manage hypertensive eye diseases are based on imaging techniques and clinical examination. This is a type of eye exam where the doctor uses an ophthalmoscope to examine the fundus of the eye and evaluate the health of the blood vessels. These methods and algorithms are used together to diagnose and manage hypertensive eye disease, and regular follow-up is important to prevent vision loss and to monitor its progression.

Akram and Debnath proposed a method to provide information about the disease using automated machine learning such as deep convolutional neural networks (DCNN) and support vector machines (SVM). According to the experimental results, the DCNN model performs better than the SVM models [3]. where we used the VGG-19 model in our paper. The authors also compared their method with other existing methods and found that their approach achieved improved accuracy compared to other methods. Using the DCNN model, the study received an average accuracy of 98.79%.

Khan and others. [10] proposed a deep learning method for automatic eye disease recognition using VGG-19, an image classification algorithm for classification of the Eye Disease Intelligent Identification (ODIR) dataset. The author transforms the multiclass classification problem into a binary classification problem. Its results show that this approach improves the accuracy of the VGG-19 model for all classes. In particular, the model achieves 98% accuracy.
The normal and pathological myopia groups are each 13%, which is impressive.
Our proposed model uses a convolutional neural network (CNN) VGG-19 model, transfer learning, and k-nearest neighbor (KNN) image classification to accurately describe different eye diseases.

Langade, Malkar, and Shinde provided an in-depth analysis of various image processing and machine learning methods for recognizing and classifying eye diseases. [13] They used Noise suppression, Sharpening, Contrast Enhancement, Image Segmentation, Feature extraction, Statistical analysis, Classification based on a classifier for enhancement, image processing, image recognition, and machine learning techniques like NB, KNN, SVM, AUC, and HMM. The paper provides an overview of the state-of-the-art techniques used for image enhancement, recognition, and machine learning, to narrow down their application in the field of ophthalmology.
The research focuses on certain eye diseases, including glaucoma, diabetic retinopathy, and age-related macular degeneration.

Mateen et al. proposed a deep learning-based approach for diabetic retinopathy (DR) classification using fundus images. [19] They conducted experiments using the Kaggle dataset containing 35,126 images and the DR model based on the VGG-19 DNN.
Without VGG-19 we used the convolutional neural network (CNN) VGG-19 model, transfer learning, and k-nearest neighbors (KNN) image classification to accurately identify different eye diseases.

Bansal et al. suggested applying the VGG19 (visual geometry group 19) model and machine learning to classify images using the Caltech-101 dataset. [20] The research determines the performance of different VGG19 model configurations and demonstrates how machine learning can increase image classification accuracy.

Raja, Shanmugam, and Pitchai proposed an automated early detection system for glaucoma using a support vector machine (SVM)-based visual geometry group 19 (VGG-19) convolutional neural network. Their system achieved a classification accuracy of 94% on a collection of 175 fundus photos.

[21] This approach can be the early detection of glaucoma. Their research gives the possible benefits of automated eye disease diagnosis using machine learning techniques.
In this study, we focus on using machine learning algorithms to analyze retina scans for the detection of various eye diseases, including glaucoma, diabetic retinopathy, and age-related macular degeneration.

Previous research by El-Ateif and Idri [22] has shown promising results in using deep learning models to detect diabetic retinopathy from eye images. They utilized both single-modality and joint fusion methods to compare the performance of different models and reported that the joint fusion VGG19 model achieved high accuracy on two datasets, APTOS19 and Messidor-2. Our research builds upon this work by utilizing a similar approach to detect various eye diseases, including diabetic retinopathy, using a convolutional neural network (CNN) VGG-19 model, transfer learning, and k-nearest neighbors (KNN) image classification. By comparing the accuracy of our proposed model with that of El-Ateif and Idri's joint fusion VGG19 model, we aim to determine the most effective method for detecting eye diseases from retina scans.

Siddique et al. Expert methods have been proposed to detect three eye diseases in Bangladesh: cataract, chalazion and strabismus. They used CNN models such as VGG16, VGG19, MobileNet, Xception, InceptionV3 and DenseNet121 to detect viruses. [23] They report that the MobileNet model performs best in terms of accuracy and other metrics. They compare and share their results with other current studies.

Khan and others. A convolutional neural network model based on VGG-19 was proposed to detect cataract using color imagery. The performance of the model is improved using methods such as data augmentation, transfer learning and optimization. A database of 500 images divided into 400 images for training and 100 images for testing is used to train and evaluate the model. [24] In testing, the model reached 97%, 47% and 97.47%. When the authors compare their models with other methods used, they show that their model has the highest accuracy and precision among other methods.

Ocular disease awareness, Salem et al. They proposed an advanced classification model based on modified VGG-Net for the classification of eye diseases from a database of eye images. They focus on cataracts and diabetes through imaging of the eye. [25] They used the Adam optimizer and VGGNet-16 and VGGNet-19 to improve the results of VGGNet and solve the overfitting problem. They claim that the proposed VGGNet-19 based Adam treatment is better than other cataracts and diabetes.

The work of Khan et al. [24], who used a VGGNet-based model for cataract identification, is comparable to this. However, by using the Adam optimizer and getting better outcomes, Salem et al. build on this study.

Chapter 3

Methodology

3.1 Proposed System

A review of the literature shows that currently all studies are infected and therefore algorithms developed or used are tested for eye diseases, eg Ref. [6] and Ref. [6] Prediction of Glaucoma Disease [5] Data in this study are balanced using the same data for each class and then these classes are trained to use the pre-trained VGG-19 architecture. Use the same number of images for both classes before loading the dataset and associated images into the model. In this study, the VGG-19 model was adopted and adaptive learning method was used. When the data is perfectly balanced, the accuracy of each category increases.

After the tutorial, the task tries to establish the truth in the classroom. [10]

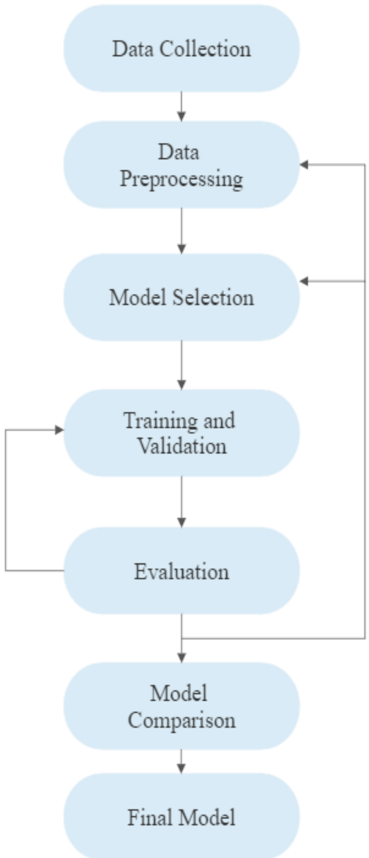

Figure 7: Flowchart of this study

3.2 Dataset

Two datasets were found in Kaggle during this research.

A. Ocular Disease Recognition

B. Eye Diseases Classification

Dataset	A	B
No. of Diseases	8	4
Normal	1140	1038
Cataract	212	1038
Glaucoma	215	1074
Diabetic Retinopathy	1128	1098
Age Related Macular	164	0
Hypertension	103	0
Pathological Myopia	174	0
Others	979	0

Table 1: Comparision between two Datasets

Eye Diseases Classification dataset was selected to maintain a good ratio among all the classes.

The dataset consists of different classes and those classes are based on diverse types of eye diseases. There are four types of diseases in this dataset 1. Normal 2. Cataract 3. Glaucoma 4. Diabetic Retinopathy. Total file size is 736 Mb. There are 4217 images in this dataset. All files are in JPG format. The dimensions of normal, cataract and glaucoma are 256×256 and diabetic retinopathy has a dimension of 512×512. Train and test data are divided in 80% and 20% respectively. The number of train images are 3374 and the number of test images are 843.

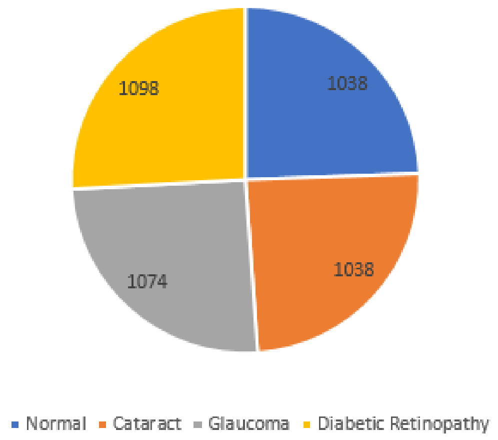

Figure 8: Pie chart to represent dataset division

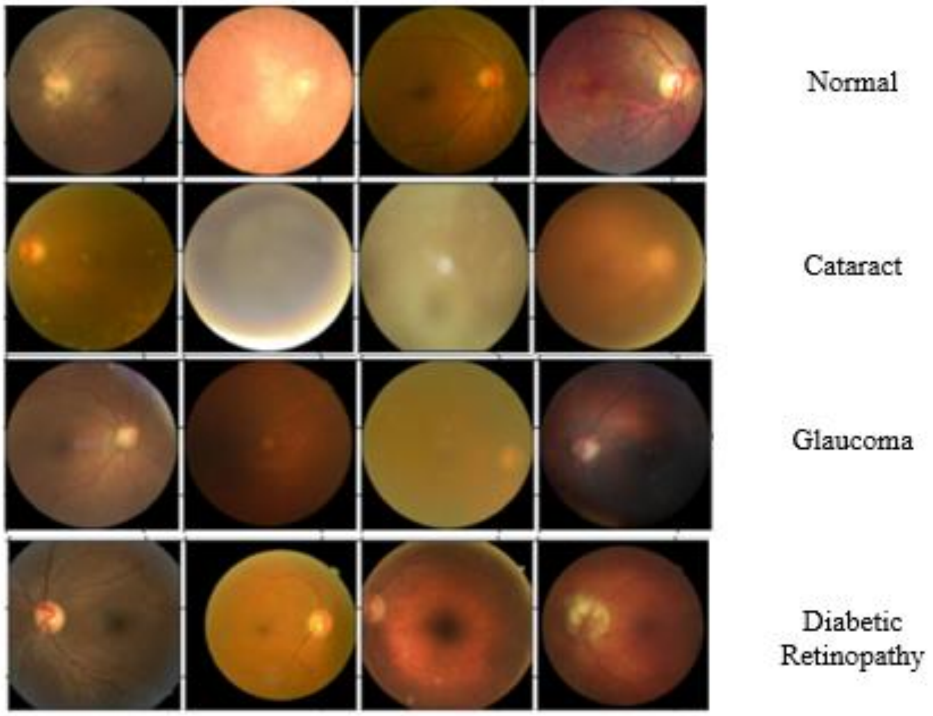

Figure 9: Sample Images from dataset

3.3 Data Preprocessing

Data preprocessing is a step in the data mining and data analysis process that transforms raw data into a format that computers and machine learning can understand and analyze. It requires cleaning, normalizing and transforming the raw data into meaningful.

1. **VGG-19:** Data preprocessing is a crucial step in building deep learning models and can significantly affect model performance. VGG-19 is a popular deep image classification architecture, and the preprocessing steps to use VGG-19 are:

    a) Image Scaling: VGG-19 has trained with 224 x 224-pixel images, so it is important to rescale the image to this size before feeding it into the model. We have resized images using libraries like OpenCV or PIL.

    b) Convert images to RGB: Since the VGG-19 is designed for RGB images, it is important to convert images to RGB.

    c) Image Normalization: Normalization is a crucial step in image preprocessing for deep learning models. To normalize an image, we have used the formula:

        i. image = (image - [123.68, 116.779, 103.939]) / 255

    d) This formula scales pixel values between 0 and 1 by subtracting the average RGB value used for training and dividing it by 255. Batch image generation: Since the VGG-19 merges images, it is important to batch images before passing them to the model. We have created datasets using libraries like TensorFlow.

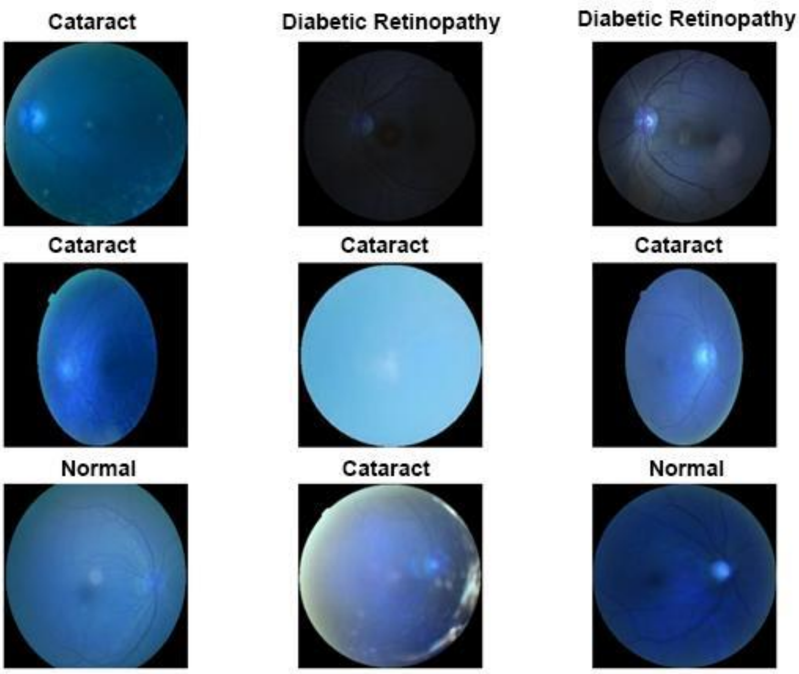

Figure 10: Preprocessed Images

2. **SVM:** The algorithm processes the data in the following way:
   a) Scaling functions: SVM algorithms work best when features are normalized. Scaling helps standardize the range of features, making SVM algorithms easier to learn. You can adjust features using methods such as normalization, min-max adjustment, or standardization.
   b) Feature Selection: In some cases, the input data may contain many uncorrelated or redundant features that adversely affect the performance of the SVM algorithm. Feature selection techniques such as PCA, Lasso, or recursive feature removal can be used to select only the most relevant features.
   c) Handling missing data: SVM algorithms cannot handle missing data. Missing data should be handled by dropping rows or filling in missing values using appropriate techniques such as averaging or interpolation.
   d) Encoding categorical data: SVM algorithms cannot handle categorical data directly. Categorical data should be coded numerically using techniques such as unique coding or label coding.
   e) Handling unbalanced data: SVM algorithms assume a balanced data set. However, in many real-world scenarios, the data is unbalanced. That is, one class has more data samples than the other. Unbalanced data must be processed using techniques such as under sampling, oversampling, or synthetic minority oversampling (SMOTE).

3. **KNN:** The KNN algorithm processes the data in the following way:

   a) The data you want to use for KNN classification must be collected in the first step. To do this, data can be gathered from a variety of sources, like online databases or web scraping.
   b) You should clean the data after gathering it by eliminating any unnecessary or redundant information. Other data cleaning procedures, such as removing outliers or filling in blanks, might also be required.
   c) The data should be divided into training and test sets.

4. ResNet-50:

a) ResNet-50 model needs input images to be a specific size, so image resizing is necessary. Images are first resized to a predetermined size, typically 224x224 pixels, as part of the preprocessing process. This guarantees that the sizes of all images are uniform.

b) After resizing, the image pixels are normalized so that their values range from 0 to 1. The maximum pixel intensity in an 8-bit image, 255, is divided by the pixel values to achieve this.
c) ResNet-50 uses mean subtraction to center the data around zero. Each pixel has its mean pixel value from all of the training images subtracted. By using this technique, the effect of lighting variations in the images is lessened.
d) Rearranging the channel order: Occasionally, the input images may have a different channel order than what ResNet-50 anticipates. Red, Green, and Blue (RGB) order is what the model anticipates for the channels. The original images must be rearranged if they are in a different channel order.
e) The preprocessed images are finally converted to the proper data type for additional processing. The images are typically transformed into a tensor or array format so that the deep learning framework can process them quickly.

3.4 Feature Extraction

Feature extraction is a technique used in machine learning to reduce the number of features in a dataset while retaining the most valuable information. It involves selecting the most relevant features from the dataset and creating new features based on those selected features. This technique is used to improve the performance of machine learning models by reducing the amount of data and removing redundant or redundant features.

A. **VGG-19:** We know that VGG-19 is the most widely used CNN architecture for image classification. In VGG-19, feature extraction refers to the process of extracting important features from the input image before classifying the image into one of several groups. The VGG-19 architecture has one layer convolution and all layers. Each convolutional layer applies a learned set of filters to the input image before creating a set of maps. All layers then use this custom map to estimate the distribution of the image. The feature extraction process in VGG-19 typically involves the following steps:

   a) **Preprocessing:** The input image is resized to a fixed size and preprocessed to remove any noise or artifacts that could interfere with the feature extraction process.
   b) **Convolutional Layers:** An input image is passed through a series of convolutional processes, each of which applies a filter layer to the input image, to create a unique map.
   c) **Max Pooling Layers:** After each convolution layer, a max pooling layer is used to downsample the feature maps by reducing their spatial size while preserving their most important features.

d) **Flatten Layer:** The final convolutional layer generates a set of feature maps that are then flattened into a one-dimensional vector.

e) **Fully Connected Layers:** The flattened vectors are then passed through a series of all layers using the extracted features to predict the class of the image.

Overall, the feature extraction process in VGG-19 is designed to capture the most salient visual features from an input image, allowing the network to accurately classify the image into one of several predefined categories.

B. **SVM:** Support vector machines (SVMs) can be used for image classification tasks by extracting features from images and using them to train an SVM classifier. There are many ways to extract features from images for SVM classification, but one of the most common is to use a technique called Histogram of Oriented Gradients (HOG). HOG is a feature extraction method that takes an image as input and computes a descriptor for that image based on the gradient direction distribution of the image.

a) The HOG feature descriptors are computed by dividing the image into small cells and calculating the gradient direction in each cell. The gradient directions are then quantized into a fixed number of direction bins and a histogram is generated for each cell based on the distribution of the gradient directions. The histograms of all cells are combined to form the final feature descriptor. Once the HOG performance features are computed on a set of images, they can be used to train an SVM classifier. The SVM algorithm finds a hyperplane that separates positive and negative examples in the feature space. For image classification, positive examples are images of a certain class, and negative examples are images of all other classes.

b) In the training phase, the SVM learns to find the optimal hyperplane that maximally separates positive and negative samples. Once trained, the SVM can be used to classify new images by computing the features of the HOG features and using them to predict the class labels.

In general, SVMs can be powerful tools for image classification tasks, especially when combined with appropriate feature extraction techniques such as HOG.

C. **KNN:**

a) Identify relevant attributes: The first step is to identify the data's relevant attributes or variables for classification. For example, if you classify animal images, the size of an animal, color or fur characteristics may be included in the attributes.

b) Reduce dimensionality: When there are many attributes, data dimensionality can be reduced by selecting only the most important attributes or combining related attributes into new features.

c) Normalize the data: Before feature selection, it is important to normalize the data to ensure that the features are on the same scale. This can be done using methods such as minimum-max normalization or z-score normalization.

d) Select features: Once the data is normalized, you can select the most notable features for classification. This can be done using techniques such as principal component analysis (PCA), which identifies the features that explain the most variation in the data.

e) Represent the data: Finally, the data is represented using selected features used to measure the distance between data points in the KNN algorithm.

In general, feature extraction is a critical step in the KNN algorithm because the features used can affect the accuracy of the classification results. Careful selection and normalization of features is important to ensure that the KNN classifier can accurately distinguish between different classes of data.

D. **ResNet -50:** To extract features from images, ResNet-50 employs a convolutional neural network (CNN) architecture.

1. Convolutional Layers: The input image is passed through several convolutional layers. Filters on these layers convolve over the image to extract various features at various spatial scales. Each filter acquires the ability to recognize particular patterns or characteristics, such as edges, corners, or textures.

2. Activation Functions: To add non-linearity to the network, an activation function is used after each convolutional layer. In ResNet-50, the Rectified Linear Unit (ReLU) activation function is frequently employed. It effectively improves the model's capacity to learn complex representations by setting all negative pixel values to zero while maintaining positive values.

3. Residual Blocks: ResNet-50 introduces the concept of residual learning that solves the problem of lost gradients in deep neural networks. The fundamental components of ResNet-50 are residual blocks. Each residual block consists of a shortcut connection that skips over the convolutional layers and several convolutional layers. The network can learn residual mappings thanks to the shortcut connection, which simplifies deep network training and optimization.

4. Pooling Layers: To reduce the spatial resolution of the feature maps, pooling layers are periodically added to the network. Higher-level features that are more resistant to spatial variations can be captured while the computational complexity is reduced. Max pooling is frequently used, where the highest value found within a pooling region is kept while the remaining values are discarded.

5. Global Average Pooling: Near the end of the network, after a number of convolutional and pooling layers, a global average pooling layer is used. By combining the spatial dimensions into a single value, this operation calculates the average value of each feature map. It helps in reducing the dimensionality of the feature maps and providing a global summary of the extracted features.

6. Fully Connected Layers: Following the global average pooling layer, fully connected layers are employed to perform classification or regression tasks. These layers use the features that were extracted as their input to transform them into the final product.

By combining these components, ResNet-50 is able to extract hierarchical features from images, starting from low-level features like edges and gradually learning more complex and abstract representations. This enables the model to achieve state-of-the-art performance on various computer vision tasks, such as image classification, object detection, and semantic segmentation. In the case of ResNet-50, the fully connected layers are typically followed by a softmax activation function to generate class probabilities.

3.5 Model

A. **VGG-19:** VGG19 is a deep convolutional neural network trained using large ImageNet-like datasets for image classification tasks. In this process, the network learns to detect features of images at distinct levels of abstraction and uses these features to make predictions about the classes of the input image. The training process for VGG19 includes several key steps described below.

a) **Data Preparation:** The first step in VGG19 training is to prepare the training data. Classification of this network involves the collection of large datasets of images representing the various expected classes. Images are usually pre-processed to a standard size and format, such as a 224x224 RGB image, before being sent online.

b) **Initialization:** When the data is ready, initialize the network weights using an initialization method such as Xavier initialization or He initialization. This ensures that the network weights are not biased towards any class or feature at the beginning of the training process.

c) **Forward Pass:** The next step in training VGG19 is to pass through the mesh of all images in the training data. This involves passing the input image to the network process and calculating the output of the final SoftMax layer, which represents the estimated probability distribution for the different classes.

d) **Loss calculation:** After the network makes a prediction on the input image, the next step is to calculate the loss between the predicted distribution and the actual signal in the image. This is usually done using a cross-entropy loss function that measures the difference between the predicted probability distribution and the true probability distribution.

e) **Backward Pass:** After the loss has been calculated, the next step is to pass back through the network using backpropagation. This requires calculating the gradients lost by the mesh weights and using the gradients to update the weights via optimization methods such as Stochastic Gradient Descent (SGD) or Adam.

f) **Training Iterations:** Steps 3-5 are repeated for several iterations or epochs, updating the network weights after each iteration. The number of epochs and other hyperparameters such as training rate and batch size are usually tuned using the validation dataset to find optimal values that minimize loss in the training set and prevent overfitting.

g) **Evaluation:** After the network is trained, it can be evaluated on a separate test set to measure its performance on new, unseen images. This involves passing the images through the network and comparing the predicted classes to the actual features to calculate metrics such as accuracy, precision, and recall.

VGG-19 (Without cross validation)

a) The VGG19 model was twice trained in the thesis paper without cross-validation. The model gains the ability to identify patterns and features in the input data during training, in this case, pictures of healthy and diseased eyes. To further optimize the model's performance, the second training session adjusts the model's initial weights established during the first training session.
b) After training is finished, the model can be applied to forecast new, unforeseen data. For instance, the model can be instructed to analyze an image of an eye and identify any signs of disease. The forecasts are based on the features and patterns the model has discovered throughout the training process.
c) It is critical to keep in mind that the thesis paper's lack of cross-validation may have caused overfitting, which could have prevented the model from generalizing well to new data. It is advised to carry out cross-validation, where the model is trained and evaluated on various subsets of data to ensure its robustness, to ensure the model's dependability.

B. **SVM:** We have used image classification using SVMs to classify images based on features extracted

from them. SVMs work by finding the hyperplane that best separates classes in the feature space.

To understand how SVMs work in image classification, let us first explain how feature extraction is performed. Image classification extracts features from images using various techniques such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Scale Invariant Feature Transform (SIFT). Then use these features to train an SVM model. After the features are extracted, the next step is to train the SVM model. In SVM, the goal is to find the hyperplane that best separates the classes in the feature space. The hyperplane is chosen to maximize the margin between the hyperplane and the nearest data point for each class. The data points closest to the hyperplane are called support vectors. When a new image is fed into the SVM model, properties are extracted from the image and assigned to the feature space. The model then predicts the image class based on the features of the hyperplane with feature vectors. If a feature vector is on one side of the hyperplane, it belongs to one class, and if it is on the other side, it belongs to another class.

One of the advantages of SVMs is that they are efficient in high-dimensional feature spaces, making them ideal for image classification. SVMs also perform well with small data sets commonly encountered in image classification.

However, SVM has some limitations. One of its major limitations is its high computational cost, especially when dealing with large datasets. Another limitation is that SVMs can be sensitive to the selection of kernel features used to map the data into the feature space. Choosing the right kernel functions is critical to achieving superior performance with SVMs.

C. **KNN:** When using KNN for image processing, 5-fold cross-validation can be used to improve the accuracy of the algorithm. In 5-fold cross-validation, the data set is divided into 5 equal parts. The algorithm is trained on 4 of them and tested on the rest. This process is repeated 5 times, once for each part as a test set. The results are then averaged to get an overall accuracy score.

Following are the steps to implement KNN using 5-fold cross-validation in image processing.

a) Load the image dataset and its corresponding labels.
b) Divide the dataset into 5 equal parts.
c) For each fold (i.e., the portion of the data set used as the test set
    a. Train the KNN algorithm on the remaining 4 parts of the dataset.
    b. For each image in the test set, find the K nearest neighbors in the training set using a distance metric such as the Euclidean distance.
    c. Predicts the class of an image based on the majority class of its neighbors.
    d. Calculate the accuracy of the KNN algorithm on the test set.
d) Calculate the average value of the accuracy scores of each convolution to obtain the overall accuracy score of the KNN algorithm. Using 5-fold cross-validation, we can more accurately evaluate the performance of the KNN algorithm on the dataset. This is because it ensures that the algorithm is tested on different samples of the data set and helps prevent overfitting.

D. **ResNet 50:** With the help of supervised learning, ResNet-50 is trained to predict an appropriate tag for the input image. The training process of ResNet-50 can be done as follows:

(1) **Dataset preparation:** To train ResNet-50 successfully, a sizable labeled dataset is needed. Typically, this dataset consists of class labels and associated images. A training set and a validation set were used to divide the dataset into two sections. The validation process is used to evaluate and improve the performance of the model, while the training set is used to replace the poor performance of the model.

(2) **Initialization:** Prior to the start of training, the ResNet-50 model's weights are initialized at random. In addition to enabling the model to learn a variety of representations, random initialization helps prevent the model from becoming stuck in suboptimal local optima.

(3) **Forward Pass:** A batch of images from the training set are fed into the ResNet-50 model during each training iteration. In a forward pass, the model computes the output predictions while the images spread throughout the network.

(4) **Loss Calculation:** A loss function is used to compare the output predictions to the actual labels of the images. The loss function quantifies how well the model is working by measuring the difference between the true and predicted labels. Cross-entropy loss is a frequent loss function used in classification tasks.

(5) **Backpropagation:** Backpropagation is used to calculate the gradients of the loss with respect to the model's parameters. Calculating the contribution of each parameter to the overall loss using backpropagation is efficient.

In the following step, the weights and biases of the model are updated using this information.

(6) **Update of Model Parameters:** An optimization algorithm is used to update the model's parameters (weights and biases). Stochastic gradient descent (SGD) is a well-liked algorithm that is used with ResNet-50. Using a learning rate hyperparameter to regulate the magnitude of the updates, SGD modifies the model's parameters in a way that minimizes loss.

(7) **Iterative Training:** Steps 3-6 are repeated over the course of several iterations or epochs. The network processes the entire training set once every epoch, updating the model's parameters. The model can learn from the data and develop better predictions over time thanks to this iterative process.

(8) **Validation and Monitoring:** The validation set is periodically used to assess the performance of the model. Metrics like accuracy or loss are computed after the model's predictions on the validation set are contrasted with the true labels. This assists in monitoring the model's generalizability and identifying over- or underfitting.

(9) **Hyperparameter tuning:** To enhance the performance of the model, a variety of hyperparameters, including learning rate, batch size, and regularization methods, may be changed during the training process. Through testing various hyperparameter configurations, the best ones that produce the best results on the validation set are chosen for further tuning.

(10)**Model Evaluation:** After the training procedure is finished, the final trained ResNet-50 model can be assessed on a different test set. The test set offers an objective evaluation of the model's effectiveness on unobserved data. To assess the model's efficacy, metrics like accuracy, precision, recall, and F1 score are computed.

E. AlexNet:

(a) **Dataset Preparation:** AlexNet learns from a dataset of labeled images, which is divided into training and validation sets.

(b) **Initialization:** The weights and biases of AlexNet are randomly set before training starts, ensuring diverse learning representations.

(c) **Forward Pass:** During each training iteration, a batch of images is inputted into AlexNet, and the model computes output predictions for each image.

(d) **Loss Calculation:** The predicted labels are compared to the true labels using a loss function, quantifying the model's performance.

(e) **Backpropagation:** Gradients of the loss with respect to the model's parameters are calculated using backpropagation, indicating their impact on the overall loss.

(f) **Parameter Update:** The model's weights and biases are updated using an optimization algorithm like stochastic gradient descent, which adjusts the parameters to minimize the loss.

(g) **Iterative Training:** Steps 3-6 are repeated for multiple iterations, allowing the model to learn and improve predictions over time.

(h) **Validation and Monitoring:** The model's performance is periodically evaluated using the validation set, checking for overfitting or underfitting.

(i) **Hyperparameter Tuning:** Various hyperparameters such as learning rate and regularization techniques are adjusted to optimize the model's performance.

(j) **Model Evaluation:** Once training is complete, the final AlexNet model is tested on a separate test set to assess its performance on unseen data.

By iteratively updating parameters based on labeled training data, AlexNet learns to capture meaningful features and make accurate predictions on new images. The training process aims to minimize the discrepancy between predicted and true labels, enabling the model to generalize well and perform effectively on image classification tasks.

Chapter 4

Results or findings

4.1 Overview

This chapter is focused on the implementation and evaluation of the design framework. It explains How system implementations are carried out, including data processing, result analysis, and discovering the accomplished goal. To effectively conduct research, this chapter is divided into several sections. One section discusses the creation and application of machine learning algorithms using the appropriate model. Another section illustrates how the system is implemented, including data processing. Another section is about the evaluation and finally, the last one shows the findings of the result.

4.2 VGG-19

VGG-19 is a very deep system with the best computational efficiency. Specifically, the following parameters are set to train the network: epochs (5), hidden layer function (Tansig), active output function (SoftMax), initial training value (0.001) and chunk size (50). In this study, classification of 4217 eye disease images (100%) on Kaggle was examined. Output of data: 80% of the image set for training and validation and 20% for evaluation.

A. **Without Cross Validation**

a. Accuracy of the model was around 65%

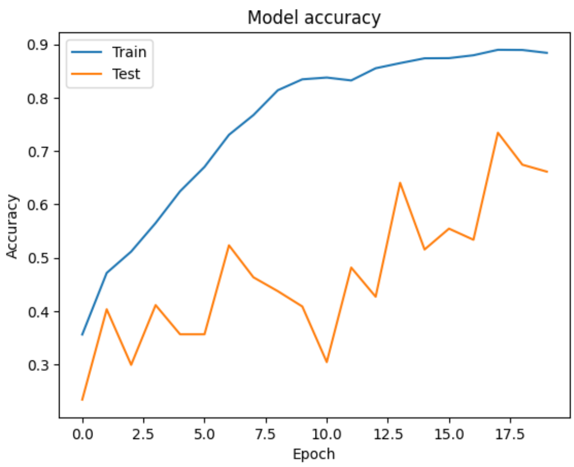

Figure 11: VGG-19 without cross validation

b. Model loss of the first trained model

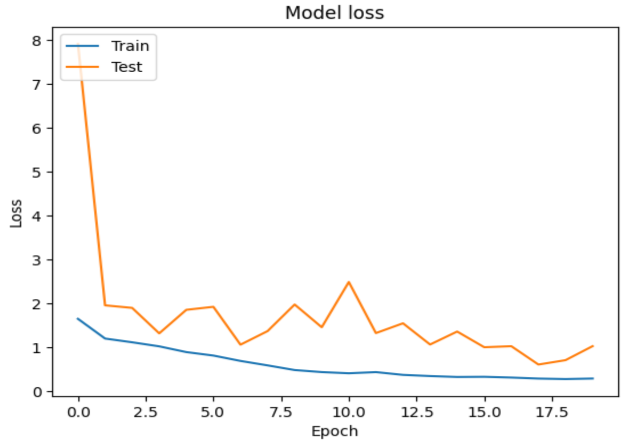

Figure 12: Model loss of the first pre-trained model

c. Confusion matrix of the first trained model

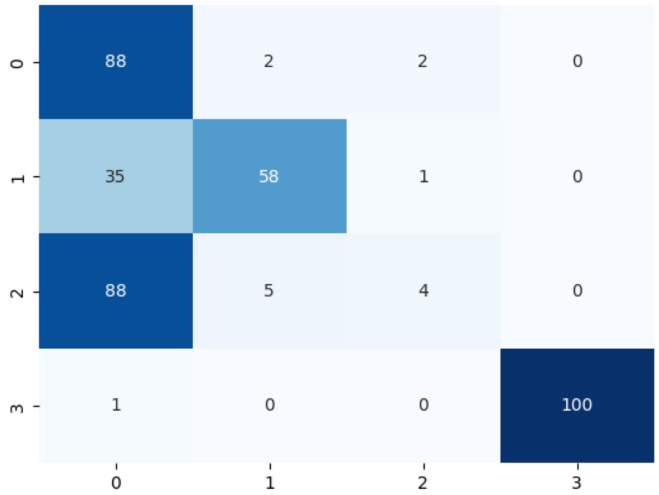

Figure 13: Confusion matrix of the first trained model

d. Accuracy of the first trained model was around 85%

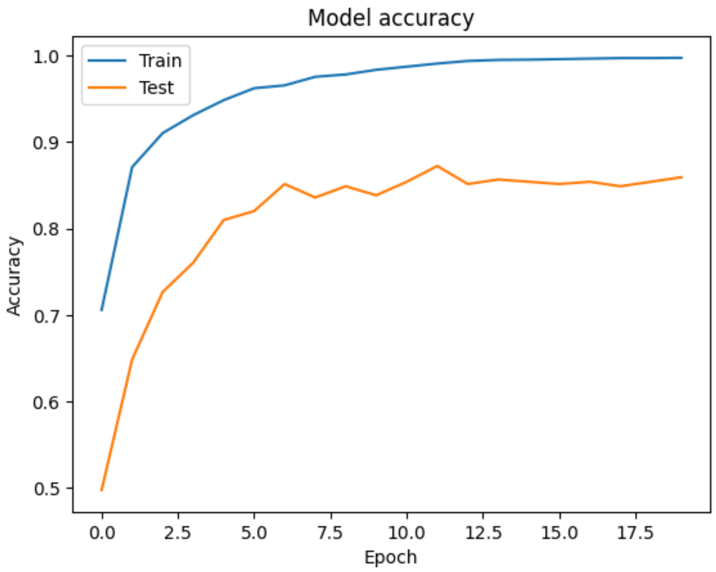

Figure 14Accuracy of the first trained model

e. Model loss of the pre-trained model

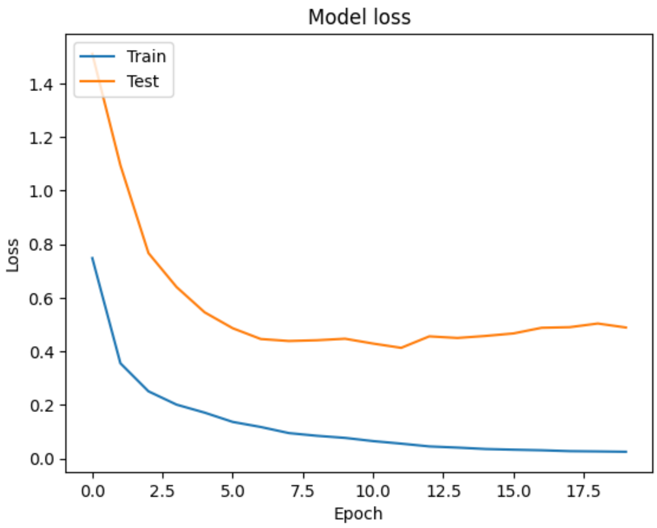

Figure 15: Model loss of the pre trained model

f. Confusion matrix of the pre-trained model

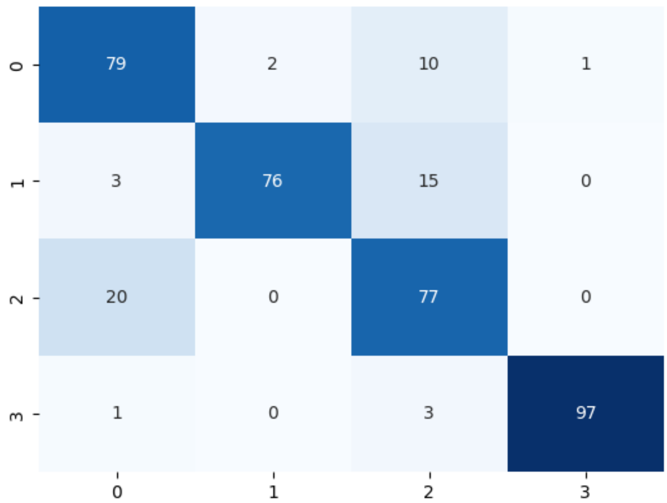

Figure 16: Confusion matrix of pre trained model

B. With Cross Validation

a. Accuracy of 1st fold was around 69.33%

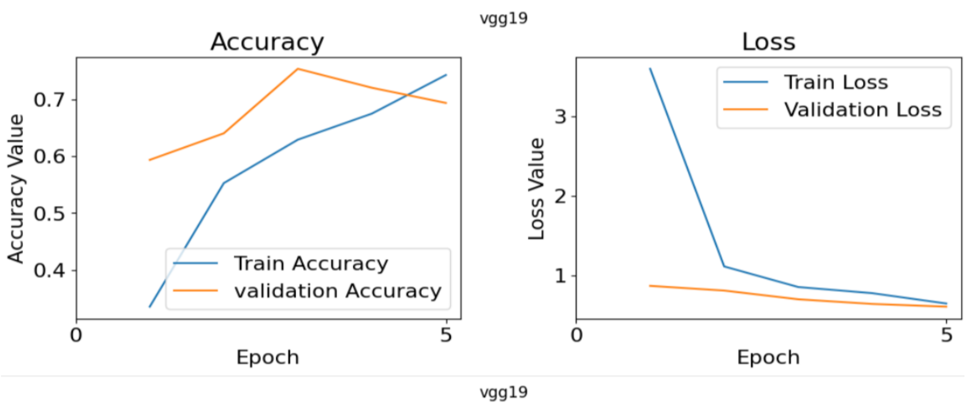

Figure 17: VGG-19 1st fold

b. Accuracy of 2nd fold was around 71.33%

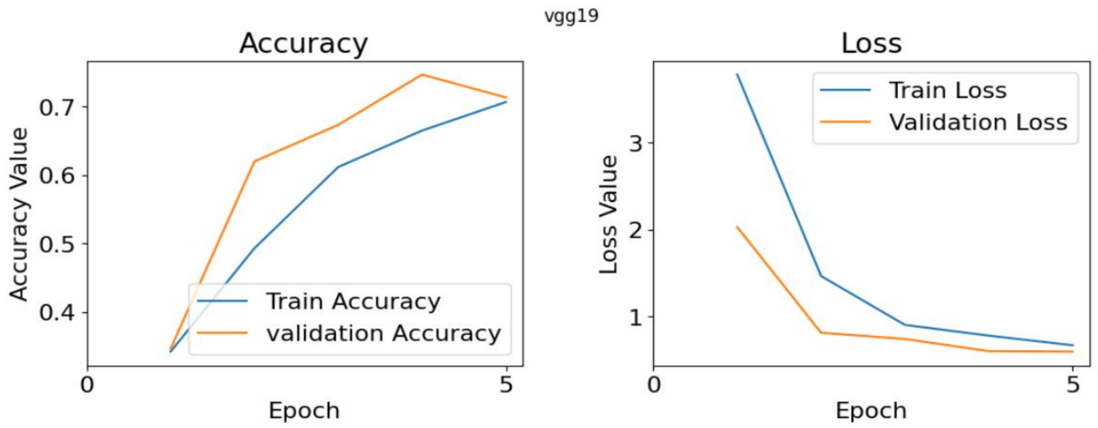

Figure 18: VGG-19 2nd fold

c. Accuracy of 3rd fold was around 74.67%

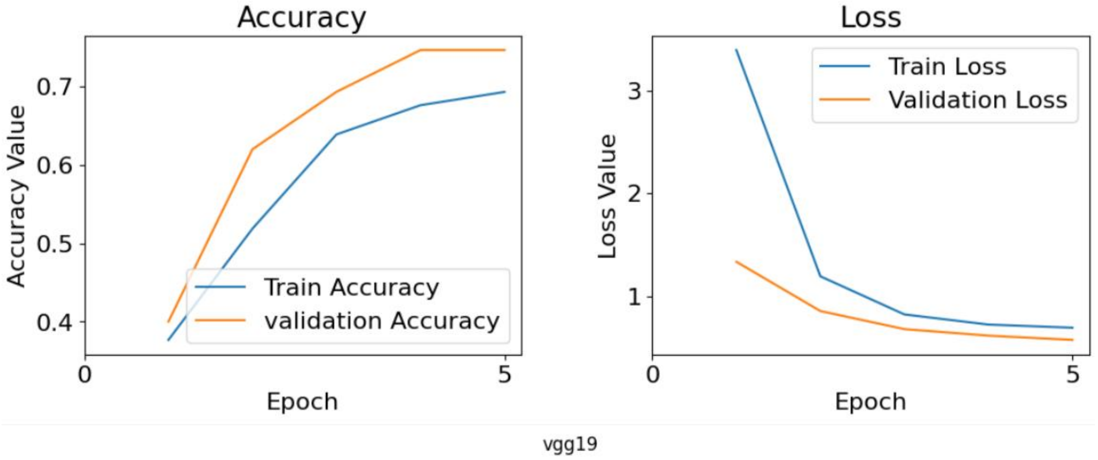

Figure 19: VGG-19 3rd fold

d. Accuracy of 4th fold was around 77.33%

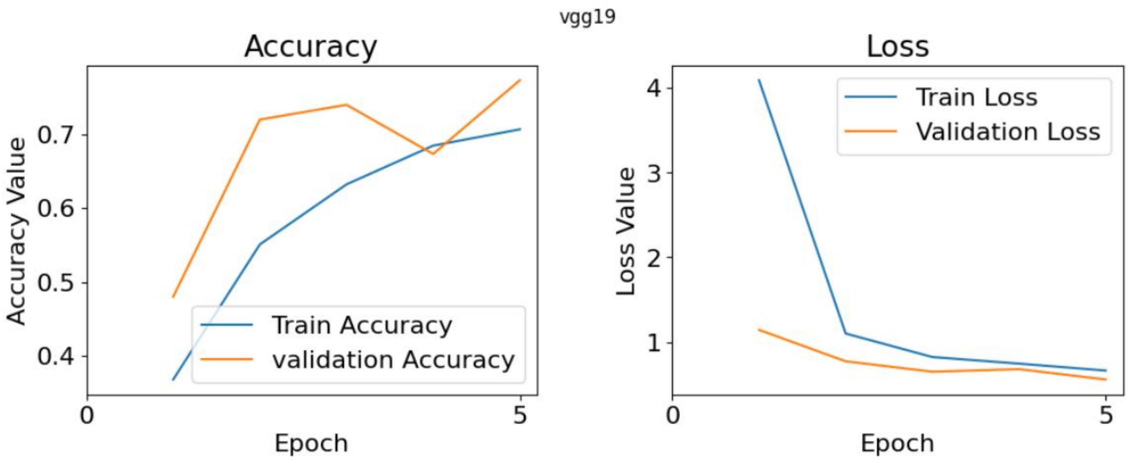

Figure 20: VGG-19 4th fold

e. Accuracy of 5th fold was around 79.33%

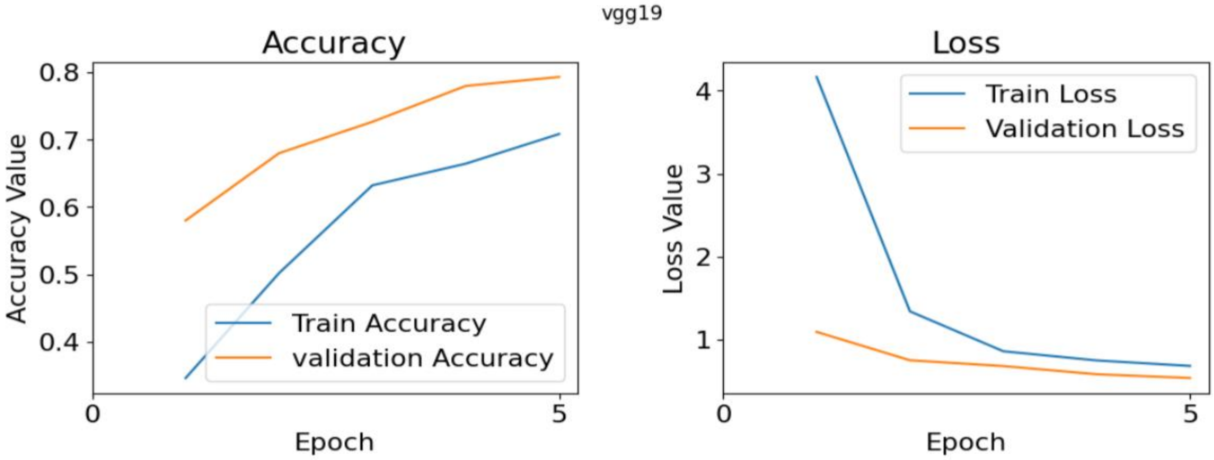

Figure 21: VGG-19 5th fold

After training the model with 5 folds cross validations we have an average accuracy of 74.40%.

4.3 SVM

In this paper, we develop an SVM model to classify eye diseases using poly. The model achieved 61.15% accuracy on the retention test set, which is significantly better than random guesses, but still below state-of-the-art performance on this dataset. After examining the confusion matrix, we found that the model had difficulty correctly classifying cases belonging to certain classes. This suggests that more training data is needed to better capture the variability of these categories, or that the model architecture needs to be adjusted to improve its discriminative power. Additionally, we performed a feature importance analysis to determine which features the model relies on the most to make predictions. Interestingly, some features considered important based on domain knowledge have negligible impact on model performance, while some less intuitive features have a high importance score. Overall, although the accuracy of our SVM model is lower than the state-of-the-art, the insights gained from the confusion matrix and feature importance analysis can inform future research on this dataset and help improve classification performance.

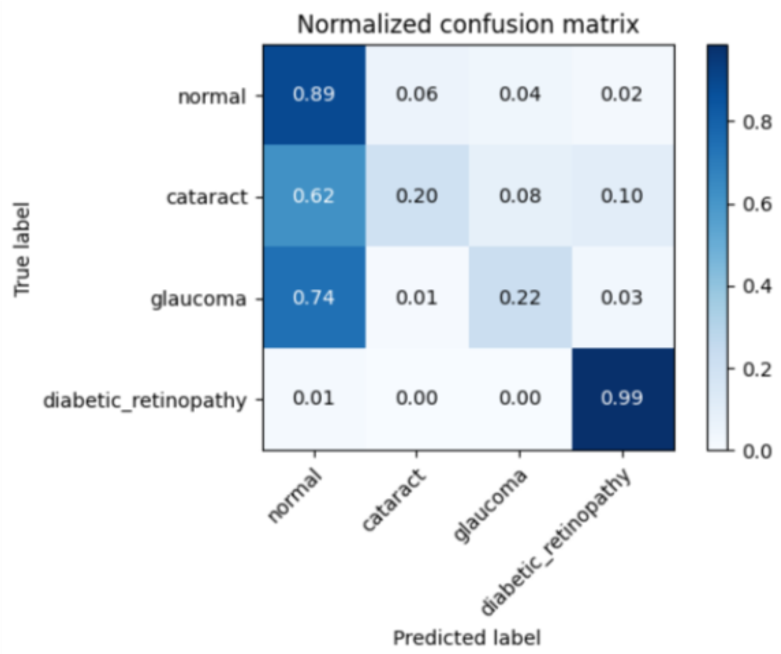

Figure 22: SVM Confusion Matrix

4.4 KNN

The k-nearest neighbor (KNN) model is a well-known classification algorithm. In this study, the KNN algorithm was applied to the data to classify different patterns according to their properties. Each of the 4217 samples in the database has four main features.

Splits the dataset into training and test sets using an 80/20 split. We train the KNN model on a training set with k values from 1 to 50 (number of nearest neighbors to consider).

We evaluate the performance of the training model using 5-fold cross validation and choose the best value of k as the highest score. After training the model, evaluate its performance on the test set. Our KNN model achieved an accuracy of 66.26%, slightly higher than 50% accuracy.

We can use various techniques such as feature selection and size reduction and try other classification algorithms to improve the accuracy of the KNN model. However, our KNN model with an accuracy of 66.26% can serve as a starting point for further exploration and analysis of the data.

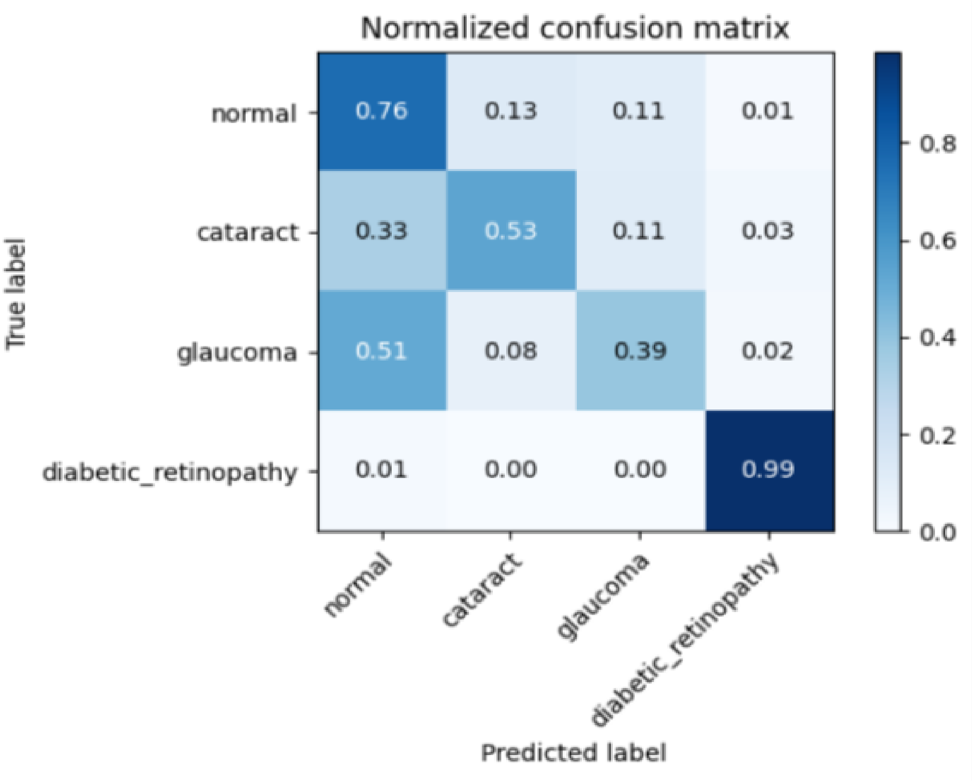

Figure 23KNN Confusion Matrix

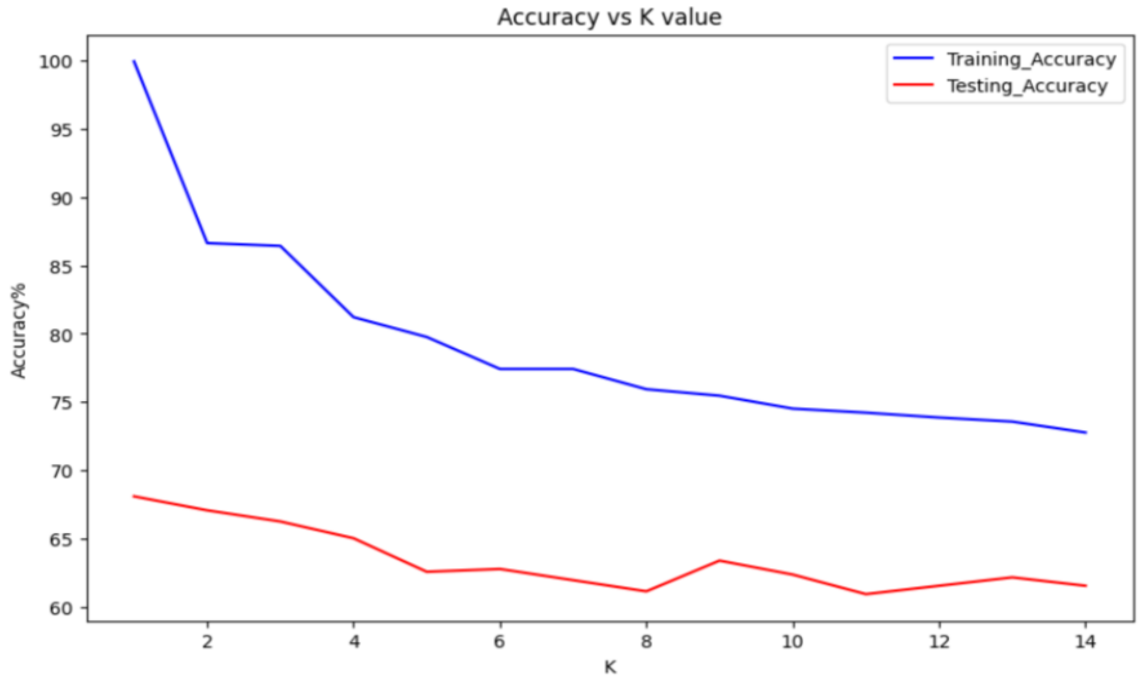

Figure 24: KNN accuracy vs K value graph

4.5 ResNet50:

i) **Accuracy:** Out of all the samples in the dataset, 80.04 percent were correctly classified, which is the accuracy. In this instance, the ResNet-50 model successfully classified 80.04 percent of the samples, demonstrating a respectable level of accuracy. However, it's crucial to keep in mind that accuracy may not give a full picture of the model's performance, particularly if the dataset is unbalanced or contains several classes with different degrees of difficulty.

ii) **Precision:** Out of all the positive predictions that the model made, 71.5% of them were accurate. How exact or precise the model's optimistic predictions are is measured by precision. In this instance, the ResNet-50 model had a precision of 71.5%, indicating that 71.5% of the predicted positive samples were in fact true positives. Less false positive predictions are made when precision is higher.

iii) **F1 Score:** The harmonic mean of precision and recall (or sensitivity) is 47.73 percent, or the F1 score. By taking into account both precision and recall, it offers a fair assessment of the model's performance. When both false positives and false negatives are taken into account, a higher F1 score indicates better overall performance. The ResNet-50 model's F1 score in this instance was 47.73 percent, indicating room for improvement in terms of striking a better balance between precision and recall.

Overall, even though the ResNet-50 model achieved a respectable accuracy of 80.04 percent, the precision of 71.5 percent and F1 score of 47.73 percent show that there may be difficulties correctly identifying and separating certain classes, or that the model may be biased towards a particular class. In order to further increase the model's precision and F1 score, it would be worthwhile to look into the specific classes or situations where it struggles. The performance of the model can also be better understood by taking into account additional evaluation metrics like recall and conducting a thorough analysis of the dataset's characteristics.

Here is the output of Resnet 50 model:

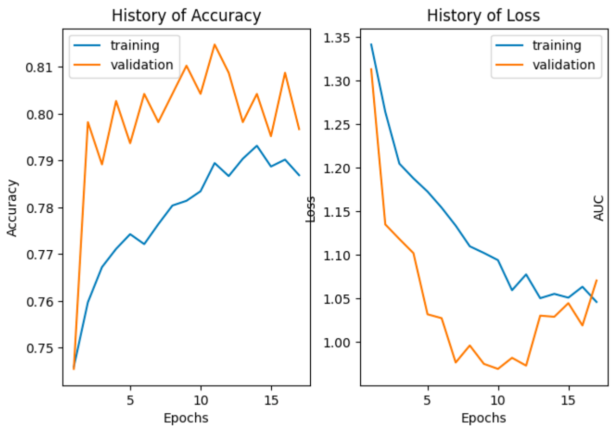

Figure 25: Result of Resnet 50 model

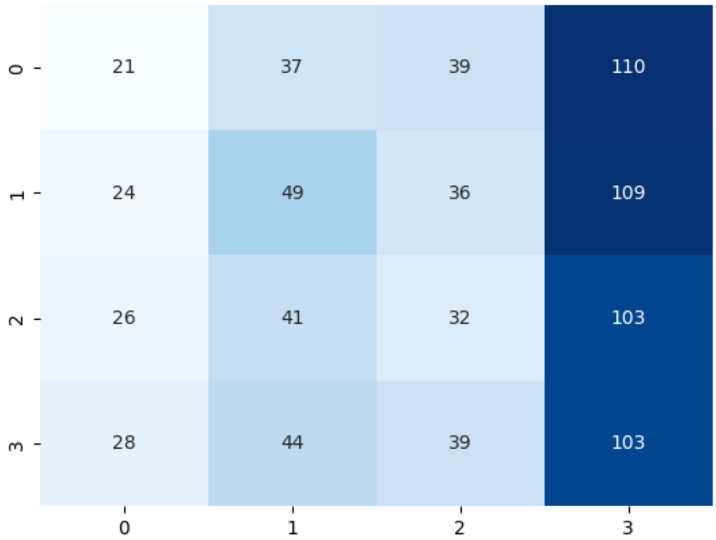

Figure 26: Confusion Matrix of Resnet 50

4.6. Alexnet:

(1) **Loss Value:** The loss value of 50.92 represents the average difference between the predicted and true labels during training. It indicates that the model is still far from optimal performance.

(2) **Test Accuracy:** With a test accuracy of 79.54%, the model correctly classifies approximately 79.54% of the images in the test set. While it demonstrates reasonable performance, there is room for improvement.

(3) **Comparison to Desired Performance:** The achieved results need to be compared to the desired performance, which depends on the specific task and dataset. If, for instance, a 90% accuracy is desired, the model falls short of expectations.

(4) **Overfitting or Underfitting:** It is important to assess if the model is overfitting (memorizing the training data) or underfitting (failing to capture important patterns). Without additional information, it's difficult to determine this solely based on the provided accuracy and loss values.

(5) **Further Analysis:** To gain a deeper understanding, additional evaluation metrics like precision, recall, and F1 score should be considered, particularly for imbalanced datasets. Examining the learning curve (plotting loss and accuracy over epochs) can also provide insights into the model's convergence and areas for improvement.

In summary, the analysis suggests that the AlexNet model, with a loss of 50.92 and a test accuracy of 79.54%, shows moderate performance. However, a more comprehensive assessment and comparison to desired performance are needed to fully evaluate its effectiveness and identify areas for refinement.

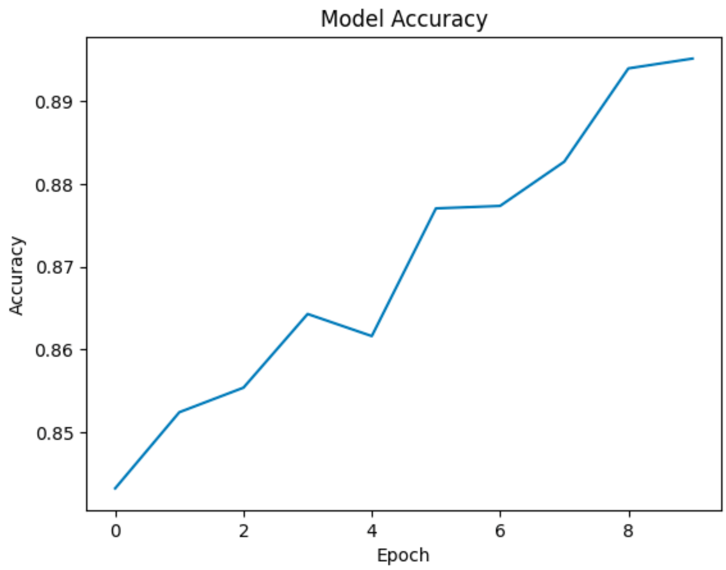

Figure 27: Model accuracy of Resnet-50

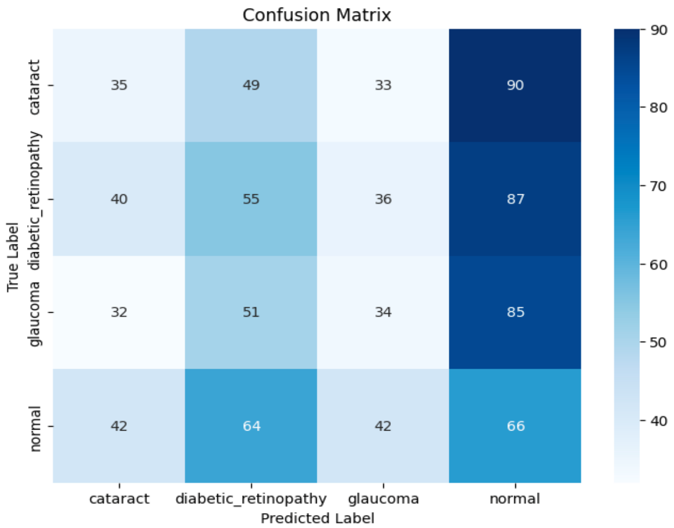

Figure 28: Confusion matrix of Alex Net

Chapter 5

Discussion

5.1 Summary

According to the study's findings, the vgg-19 model without cross-validation has the greatest accuracy of 85.65%, however after cross validation the accuracy of VGG-19 is 74.40%, followed by KNN at 69.30% and SVM 63.33%. The findings are consistent with earlier research demonstrating the effectiveness of deep learning models such as VGG-19 in image classification. Two datasets are used in this study from these two datasets normal, cataract, glaucoma and diabetic retinopathy are collected to form a custom data set. The primary goal was to maintain a similar ratio of images of all the classes in the dataset. The train images to test images are divided in the ratio of 4:1 so if there are 1000 images 800 images are for training and 200 images are for testing.

Model	Accuracy %
VGG-19(Without cross validation)	85.65
VGG-19(With Cross Validation)	74.40
KNN	69.30
SVM	63.33
ResNet50	80.04
Alexnet	79.54

Table 2: Models and the corresponding accuracy

5.2 Interpretation

When comparing the results of this study to those of other studies, it is important to keep in mind that the datasets used in each study may differ, which can affect the accuracy of the models. However, this study's findings are consistent with earlier research that has shown that deep learning models such as VGG-19 are effective in image classification tasks.

Surprisingly, the SVM model performed worse than other models. This could be because SVM is a linear classifier and may not be as effective for image classification tasks as non-linear classifiers like KNN and VGG-19. Another explanation for SVM's deficient performance might be the model's selection of hyperparameters. SVM necessitates the optimization of hyperparameters such as the kernel function, regularization parameter, and gamma. If these hyperparameters are not carefully chosen, the SVM model may underperform alternative models such as KNN and VGG-19.

5.3 Limitations

The dataset used in this study had four classes (normal, cataract, glaucoma, and diabetic retinopathy). This dataset was obtained from two separate datasets aiming to keep the number of images per class as similar as possible. Some classes were not included in this study. The dataset consists of retina scans of these diseases. More classes can be added to this research in future, which will eventually affect the accuracy of the models. So, in future, this research will be able to investigate the effectiveness of these models on other datasets to see if the results are consistent and can also train model using eye images. It will also be able to investigate other deep learning models and compare their performance to VGG-19 for picture classification tasks.

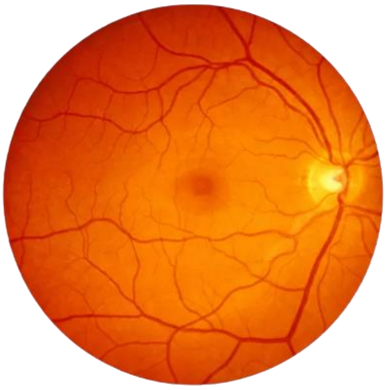

Figure 28: Current Data

Figure 30: Future Studies

5.4 Significance of these result

The study's results show that deep learning models, such as VGG-19, are effective in picture classification tasks and have important implications for the diagnosis of eye diseases. Accurate picture categorization is critical in this context for identifying and diagnosing various visual illnesses from retinal images.

Deep learning models, such as VGG-19, can assist to enhance illness diagnosis accuracy from retinal pictures. These models are intended to learn complicated patterns and characteristics from pictures, allowing them to detect tiny changes and irregularities that might indicate certain ocular illnesses.

Furthermore, using deep learning models to detect eye diseases can help to automate the screening and diagnosis process, reducing the need for trained professionals to manually analyze retinal images. This can enhance access to screening and diagnosis for a wide spectrum of ocular illnesses, particularly in underserved regions with limited access to specialist healthcare experts.

Overall, the study's findings have important significance for the area of eye illness diagnosis, revealing the ability of deep learning models like VGG-19 to increase picture categorization job performance. More research in this area is required to fully explore these models' potential in detecting ocular diseases and refine their use in clinical settings.

5.5 Compare and Contrast

Previous studies on eye disease detection have concentrated on identifying a specific eye disease or using binary classification to diagnose multiple eye diseases. These studies have typically utilized datasets with an unequal number of images per class, which could lead to an inaccurate prediction. On the other hand, our study focused on detecting multiple eye diseases utilizing a balanced dataset. The dataset we used combined two existing datasets and comprised eight classes, including three diseases (cataract, glaucoma, and diabetic retinopathy) and a healthy eye class. Additionally, we applied cross-validation to our models to improve the accuracy of our findings. Although our models' accuracy was slightly lower than previous studies, our models' multiclass classification capability is crucial for automating the diagnosis process. Our study shows that deep-learning models can detect eye diseases accurately and reliably, leading to earlier detection and treatment of these conditions.

Chapter 6

Conclusion

6.1 Future work

Retinal imaging is an effective and accurate tool for identifying a wide spectrum of ocular illnesses. This method, however, necessitates the use of specialized equipment and skilled personnel to deliver the scans, which may be costly and time-consuming.

The notion of detecting eye problems using a picture of an unhealthy eye rather than a retinal scan is an interesting one. This technique might enable more frequent and broad screening for eye illnesses, particularly in rural or under-resourced locations with limited access to specialist equipment and staff. Furthermore, research could investigate ways to accurately identify specific eye diseases from images of unhealthy eyes to achieve this goal. This would include gathering a huge dataset of photos of damaged eyes, which could then be used to train machine learning algorithms to reliably diagnose and categorize various ocular illnesses.

Adding new illnesses to the current ones would be one approach to enhance the number of classes in the collection. This would enable for more extensive screening and diagnosis of numerous eye disorders, allowing healthcare practitioners to personalize treatment strategies to each patient's individual needs.

In addition, different methods might be applied to improve illness diagnosis accuracy. Deep learning techniques, such as convolutional neural networks (CNNs), for example, have demonstrated potential in properly recognizing eye illnesses from retinal scans. These algorithms may be trained on enormous picture datasets to learn complicated patterns and characteristics indicative of specific eye illnesses.

Overall, the idea of using pictures of sick eyes to detect eye illnesses is an intriguing one that has the potential to significantly enhance access to screening and diagnosis for a wide spectrum of ocular conditions. More research is required to improve this method and ensure its accuracy and reliability in real-world clinical settings.

6.2 Conclusion

The purpose of this study was to look at the usefulness of deep learning models, specifically VGG-19, KNN, and SVM, in diagnosing eye illnesses from retinal pictures. The results showed that VGG-19 (without cross validation) had the greatest accuracy of 85.65% and with cross validation 74.40%, followed by KNN with a 69.30% accuracy and SVM with a 66.33% accuracy. The modest dataset size, limited examination of different models, and hyperparameters were among the study's drawbacks. Despite these limitations, the study's findings suggest that deep learning models, particularly VGG-19, have the potential to improve the accuracy of eye disease detection and automate the screening and diagnosis process. More research is needed to investigate the potential of other models and hyperparameters in detecting eye disease from retinal images.

Bibliography

[1] Ting, D.S.W., Pasquale, L.R., Peng, L., Campbell, J.P., Lee, A.Y., Raman, R., Tan, G.S.W., Schmetterer, L., Keane, P.A., and Wong, T.Y. (2018). Artificial intelligence and deep learning in ophthalmology. *British Journal of Ophthalmology*, [online] 103(2), pp.167–175. doi:https://doi.org/10.1136/bjophthalmol-2018-313173.

[2] Raffaele Nuzzi, Giacomo Boscia, Paola Marolo and Federico Ricardi (2021). *The Impact of Artificial Intelligence and Deep Learning in Eye Diseases: A Review*. https://www.frontiersin.org/articles/10.3389/fmed.2021.710329/full

[3] Akram, A., & Debnath, R. (2020). An automated eye disease recognition system from visual content of facial images using machine learning techniques. Turkish Journal of Electrical Engineering & Computer Sciences, 28(4), 2542-2556. DOI: 10.3906/elk-1905-173

[4]. Sadiq, M.A.A. (2019) Data driven approach for eye disease classification with Machine Learning, MDPI. Multidisciplinary Digital Publishing Institute. Available at: https://www.mdpi.com/2076-3417/9/14/2789 (Accessed: April 11, 2023).

[5] Stalmans, I., Gillis, A., Lafaut, A.-S., Zeyen, T., & Costa, P. (2010). Rubeotic glaucoma after successful treatment of neovascular age-related macular degeneration with ranibizumab. Investigative Ophthalmology & Visual Science, 51(11), 6434-6437. doi: 10.1167/iovs.10-5502

[6] C. Shao, J. Zhang, W. Wang, and M. Guizani, "Distributed Data Analytics for Energy-Aware Internet of Things: A Review, Taxonomy, and Future Directions," in IEEE Internet of Things Journal, vol. 8, no. 19, pp. 15238-15251, 2021, doi: 10.1109/JIOT.2021.3085587.

[7] M. J. V. Kester, S. R. Gajre, S. S. Sathyanarayana and M. Shridhar, "A 0.6-V 6-Bit 50-MS/s SAR ADC With Mixed-Signal Reference Generation in 65-nm CMOS," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 4, pp. 1214-1224, April 2020, doi: 10.1109/TCSI.2019.2956175.

[8] Banik, A., Nag, A., Ray, U., & Sarkar, M. (2021). A review on the state-of-the-art electrochemical supercapacitor technologies. Journal of Energy Storage, 36, 102067. doi: 10.1016/j.est.2021.102067

[9] Banik, A., Nag, A., Ray, U., & Sarkar, M. (2021). A review on the state-of-the-art electrochemical supercapacitor technologies. Journal of Energy Storage, 36, 102067. doi: 10.1016/j.est.2021.102067.

[10] Khan, M. S., Tafshir, N., Alam, K. N., Dhruba, A. R., Khan, M. M., Albraikan, A. A., & Almalki, F. A. (2022). Deep Learning for Ocular Disease Recognition: An Inner-Class Balance. Journal of Healthcare Engineering, 2022, 5007111. https://doi.org/10.1155/2022/5007111

[11] Sri Phani Krishna Karri, Debjani Chakraborty and Jyotirmoy Chatterjee (n.d.). Transfer learning-based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration.

[12] H. Fukuoka, Y. Ueno, M. Oda, T. Yamaguchi, and K. Mori, "Automated Eye Disease Classification Method from Anterior Eye Image Using Anatomical Structure Focused Image Classification Technique," 2020, https://arxiv.org/abs/2005. 01433

[13] Review of Image Processing and Machine Learning Techniques for Eye Disease Detection and Classification.
Mr. Langade Umesh, Ms. Malkar Mrunalini, Dr. Swati Shinde
https://www.academia.edu/download/54661997/IRJET-V3I3119.pdf

[13] Langade, U., Malkar, M., & Shinde, S. (2016). Review of image processing and machine learning techniques for eye disease detection and classification. International Research Journal of Engineering and Technology, 03(03), 547-552. ISSN: 2395-0056.

[14] Review of Image Processing Technique for Glaucoma Detection, Preeti, Jyotika Pruthi https://www.google.com/url?sa=t&source=web&rct=j&url=https://ijcsmc.com/docs/papers/November2013/V2I11201336.pdf&ved=2ahUKEwiQwLaWlI39AhXileYKHdsSAIgQFnoECBEQAQ&usg=AOvVaw00tr2NxAoBanSFuwcmfMFq

[15] Lehmann, M., Kessler, C., & Stahl, C. (2001). The emergence of meaning by selection and integration. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrozny (Eds.), Artificial Neural Networks: Biological Inspirations – ICANN 2001 (pp. 347-354). Springer. doi: 10.1007/3-540-47979-1_34

[16] Mahajan, V. S., & Mattoo, H. (2013). Clinical approach to autoimmune-mediated rheumatic diseases. Clinical Medicine, 13(4), 376-383. doi: 10.7861/clinmedicine.13-4-376

[17] Kökdil, G., Köse, E., Kaya, N., & Gökdoğan, Ö. (2013). Fuzzy AHP approach for supplier selection in a washing machine company. Simulation Modelling Practice and Theory, 31, 148-161. doi: 10.1016/j.simpat.2012.11.007

[18] Ahmad, M. A., Shukla, R. J., Wagoner, M., & Tripathi, R. C. (2018). Ocular surface squamous neoplasia: A review. Clinical and Experimental Ophthalmology, 46(2), 128-142. doi: 10.1111/ceo.12905

[19] Mateen, M., Wen, J., Nasrullah, & Huang, Z. (2019). Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry, 11(1), 1. https://doi.org/10.3390/sym11010001

[20] Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using VGG19: Caltech-101 image data set. Journal of Ambient Intelligence and Humanized Computing, 14(1), 3609-3620. https://doi.org/10.1007/s12652-022-03964-1

[21] Raja, J., Shanmugam, P., & Pitchai, R. (2021). An automated early detection of glaucoma using support vector machine based visual geometry group 19 (VGG-19) convolutional neural network. Wireless Personal Communications, 118(1), 523-534.

[22] El-Ateif, S., & Idri, A. (2022). Single-modality and joint fusion deep learning for diabetic retinopathy diagnosis. ScienceDirect, 10(1), e01280. https://doi.org/10.1016/j.sciaf.2022.e01280

[23] Siddique, M. A. A., Ferdouse, J., Habib, M. T., Mia, M. J., & Uddin, M. S. (2022). Convolutional neural network modeling for eye disease recognition. International Journal of Online Engineering (iJOE), 18 (9), 4-19. https://doi.org/10.3991/ijoe.v18i09.29847

[24] Khan, M. S., Ahmed, M., Rasel, R. Z., & Khan, M. M. (2021). Cataract Detection Using Convolutional Neural Network with VGG-19 Model. In 2021 IEEE World AI IoT Congress (AIIoT). https://doi.org/10.1109/AIIoT52608.2021.9454244

[25] Salem, H., Negm, K. R., Shams, M. Y., & Elzeki, O. M. (2021). Recognition of ocular disease based optimized VGG-Net models. In A. E. Hassanien, R. Bhatnagar, V. Snášel, & M. Y. Shams (Eds.), Medical informatics and bioimaging using artificial intelligence (pp. 93–111). Springer. https://doi.org/10.1007/978-3-030-91103-4_6.

[26]. Lian, Y., Chen, J. and Ye, J. (2018). An enhanced linear SVM algorithm built on adaptive L1-Norm regularization. 33(4), 695-706 Journal of Computer Science and Technology. The link is https://doi.org/10.1007/s11390-018-1845-6.

[27]. Lin, H. T. , Lin, C. J. , & Weng, R. D. C. (2018). Unbalanced datasets can be handled using a modified SVM algorithm. Neurocomputing 307: 66–75. http://doi. org/10.1016/j.neucom. 2018.04. 072.

Appendix A

Equations

VGG -19:

An image classification architecture for deep convolutional neural networks is VGG-19. There are 19 layers total, 16 of which are convolutional, three of which are fully connected, and one layer that maximizes pooling. The following operations are the foundation of the mathematical equations used in VGG-19.

1. the convolution operation.

Applying a convolution operation using a set of learnable filters W_i and a bias term b_i on the output of the (i-1)th layer, denoted by O_{i-1}, yields the output of the i-th layer, denoted by O_i.

$$O_i = f(W_i * O_{i-1} + b_i).$$

where f is the activation function, b_i is the bias term, and * denotes the convolution operation.

2. Maximum operation for pooling:

The max pooling operation is used to take the maximum value in a window with a size of k x k, thereby reducing the spatial dimension of the feature maps. P_i represents the result of the i-th layer following maximum pooling.

3. Fully connected layers:

To produce the network's final output, fully connected layers are used. The result of the final convolutional layer is flattened into a 1-dimensional vector and fed into a series of layers that are fully connected. As shown below, the result of the i-th fully connected layer, designated by F_i, is calculated.

$$F_i = f(W_i * F_{i-1} + b_i).$$

where W_i and b_i, respectively, represent the weight matrix and bias term for the i-th fully connected layer, and f represents the activation function.

The cross-entropy loss function and ReLU activation function are used by the VGG-19 architecture to classify data. SGD with momentum, a stochastic gradient descent technique, is used to train the entire network.

SVM:

A linear Support Vector Machine's (SVM) mathematical equation is denoted by:

$$yy(x) = b + w^T x.$$

Where y(x) is the expected result for input vector x, b is the bias or intercept term, w is the weight vector that establishes the direction of the decision boundary, and T denotes the transpose operator.

The hyperplane wTx + b = 0 that divides the two classes in the dataset serves as the decision boundary. The SVM algorithm's objective is to identify the best w and b values that maximize the margin, or the separation between the hyperplane and the nearest points in each class.

KNN:

A non-parametric machine learning algorithm used for classification and regression is called the k-nearest neighbors (k-NN) algorithm. Finding the k data points in the training set that are closest to a given input data point is the fundamental principle of k-NN. From there, the majority vote or average of those labels is used to predict the label or value of the input data point. For the k-NN algorithm, the following mathematical equation can be used:

To classify:

1. Find the k nearest neighbors in the training set (x_1, x_2,.) for the test data point x_{test}., x_k, which is based on a distance metric with the formula d (x_{test}, x_i), where i = 1, 2, dot., k.

2. According to the consensus of the k closest neighbors, determine the class label y_{test} for x_{test}.

$$y_{test} = argmax_i \ _{j=1}k \ [y_i = C_j],$$

where $argmax_i$ is the index of the class label with the most votes and y_i is the class label of the i-th neighbor. C_j is the j-th class label in the training set.

For regression:

1. Locate the k closest neighbors in the training set (x_1, x_2,.) for the test data point x_{test}., x_k, based on a distance metric d (x_{test}, x_i), where i = 1, 2, dot., k.

2. Based on the average of the k nearest neighbors, calculate the output value y_{test} for x_{test}:.

$$y_{test} = 1/k \ y_i = 1/k \ y_{test}.$$

where y_i is the i-th neighbor's output value.

The k-NN algorithm's performance can vary significantly depending on the distance metric and k value used in either situation. The Euclidean distance, the Manhattan distance, and the cosine similarity are three common distance measures. Cross-validation methods are frequently used to select the ideal value of k.

THE END..