Automatic Classification of White Blood Cell Images using Convolutional Neural Network

Author

Rabia Asghar

NUST CEME 00000321011

Supervisor

Dr. Arslan Shaukat

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

December, 2021

Automatic Classification of White Blood Cell Images using Convolutional Neural Network

Author

RABIA ASGHAR

NUST CEME 00000321011

A thesis submitted in partial fulfillment of the requirements for the degree of MS Computer Engineering

Thesis Supervisor:

Dr. Arslan Shaukat

Thesis Supervisor's Signature:_____

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERINGCOLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

December, 2021

Declaration

I certify that this research work titled "*Automatic Classification of White Blood Cell Images using Convolutional Neural Network"* is my own work. The workhas not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student
Rabia Asghar

Registration Number
0000321011

Signature of Supervisor

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammaticaland spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

Rabia Asghar

Registration Number
0000321011

Signature of Supervisor

Copyright Statement

Acknowledgements

I am especially thankful to Allah Subhana-Watala to have giving me guidance, patience and potential to carry this work and for each new idea which you build up in my brain to make better. Indeed without his valued guidance and help I could not have done my thesis. Indeed Allah Subhana-Watala is the most sympathetic and worthwhile of honor.

I would like to express special thankful to my supervisor Dr. Arslan Shaukat for his guidance, help and precious suggestions and virtuous support in the whole duration. It's a dedication to have such appreciable people as adviser for research work.

I am also grateful to the whole thesis committee: Dr. Usman Akram and Dr. Farhan Hussain for their enormous cooperation help and support.

I am extensively thankful to my precious parents who support me throughout in my life. I would also gratitude to my sisters who have inspired and supported me during my thesis work.

Finally, I would also like to express special thankful to my friends and any other individuals who have inspired and supported me throughout the whole duration.

Dedicated to my remarkable parents and siblings whose enormous cooperation, guidance and support who motivated me to this marvelous *accomplishment*

Abstract

Human immune system contains white blood cells (WBC) that are good indicator of many diseases like bacterial infections, AIDS, cancer, spleen, etc. White blood cells have been sub classified into four types: monocytes, lymphocytes, eosinophils and neutrophils on the base of their nucleus, shape and cytoplasm. Traditionally in laboratories, pathologists and hematologists analyze these blood cells through microscope and then classify them manually. This manual process takes more time and increases the chance of human error. Hence, there is a need to automate this process. We have first applied different CNN models, InceptionV3, VGG16, MobileNetV2, LeNet and ResNet50 to automatically classify the white blood cells. These CNN models are applied on Kaggle dataset of microscopic images. Although we achieved reasonable accuracy ranging between 92 to 95%, still there is need to enhance the performance. Hence, inspired by these architectures a framework has been proposed to automatically classify the four types of white blood cells with increased accuracy. The aim is to develop a convolution neural network (CNN) based classification system with decent generalization ability. The proposed CNN model has been tested on white blood cells images from Kaggle and LISC datasets. Accuracy achieved is 99.57% and 98.67% for both datasets respectively. Our proposed convolutional neural network-based model provides competitive performance as compared to previous results reported in literature.

Keywords: Subtypes of WBCs, White Blood Cells, convolution neural network, classification, feature extraction

Table of Contents

List of Figures

List of Tables

List of Acronyms Used in the Document

Acronym Used	Definition
AIDS	Acquired Immune Deficiency Syndrome
CNN	Convolution Neural Network
FPR	False Positive Rate
GBPS	Genetic-Based Parameter Selector
GLCM	Grey Level Co-Occurrence Matrix
HSV	Hue, Saturation, Value
JPEG	Joint Photographic Experts Group
LBP	Local Binary Pattern
LNE	Leukocyte Nucleus Enhancer
NN	Neural Network
PCA	Principal Component Analysis
RGB	Red, Green, Blue
SVM	Support Vector Machine
TPR	True Positive Rate
WBC	White Blood Cells
TPR	True Positive Rate

CHAPTER 1: INTRODUCTION

Technology has reorganized the world with automatic numerous procedures which were human being dependent preliminarily. Starting from basic device like calculator to large machines infactories, technology is all over [1]. The machines in factories are constructed to succeed the instructions in sequence to perform the job without any human leadership. With the growth of image processing and artificial intelligence came to inspire the world more than human invention. By applying artificial intelligence, the machines are learned to copy the human mind by training from experience and practice. Any other actual life fields using artificial intelligence in medical fields have become field of interest, and the achievement to make Computer Aided Design (CAD) was built in mid 1960s. In medicine field, artificial intelligence approach has been expanded to automatic the diagnosis of various diseases such as cancer, hepatitis, HIV, allergic infections nephrotic syndrome, etc [2]. Conventionally, hematology specialists perform categorization and numbering of these WBCs manually with the assistance of a microscope. However, this process is time-consuming, error sensitive, and complicated to operate. White Blood Cells classification has become an essential topic and various researchers have shared in this medical field by introducing various Computer Aided Design systems. With the growth of these systems in the field of medicine has released physicians from the burden of manual categorization. So, this research introduces a Computer Aided Design based technique to automatic classification of white blood cells.

1.1 Background, Scope and Motivation

White blood cells (WBC's) play vital role in human body immune process to recognize different diseases and also used to analysis of healthiness. Classification of white blood

cells (WBC's) subtypes has great influence as examined briefly in the above parts. White blood cells (WBC's) produce itself distinct from other blood groups by having cytoplasm and nucleus. There are five important sub type of white blood cell (WBC's): Lymphocyte, Monocytes, Neutrophils, Basophil and Eosinophil. These white blood cells (WBC's) sub-types categorized on the base of color, size, texture, cytoplasm and their morphology of nucleus [2].

Conventionally, pathologist and hematologist used microscope to examine the human blood. Manual investigation method of human blood is slower, very time consuming and is suggestible to human errors. There is demand to a significantly and rapid system which bring out results immediately. Last some decades, a progressive development comes in field of medical through medical imaging technologies. Most important computational techniques of digital image processing in most all the fields but in field of medical its result is imperious. Pattern recognition, image processing and segmentation of imagining technologies become the main part of all computerized recognition system. This modernization of image processing took high innovation in pathology sides. Still this era numerous automated techniques suggested by using artificial intelligence, machine learning and fuzzy logics but research gaps have there to improve the results, efficiency and diminish the complexity. This thesis aims to develop a convolution neural network (CNN) with decent generalization ability and great execution speed for the classification of various types of white blood cells: lymphocytes, monocytes, eosinophils, and neutrophils. The datasets used for this research are Kaggle and LISC. The Kaggle dataset contains 12500 images of size 320 x 240.The LISC dataset contains 10,000 images of size 720 x 576., and the results indicate that our proposed model gives high accuracy in terms of white blood cells classification.

1.2 **Problem Statement**

White Blood Cells (WBCs) are a very essential factor of our body immune system defending our body against germs, viruses, and bacteria by ingesting them, destroying infectious agents, or producing antibodies.

The percentage value parallel to each subtype is the standard range of WBCs present in the blood of a normal healthy person. An absence or imbalance in the number of any white blood cells type can be caused by different diseases. These diseases are identified by doing blood tests. Each subtype of white blood cells helps in tackling certain types of disorders such as cancer, hepatitis, HIV, allergic infections, nephrotic syndrome, etc. It is noted that specific disorders end up misbalancing or attacking only certain types of WBCs. It is necessary to count and recognize the persons WBCs to recognize these diseases. So, the objective of this research is to analyze different Artificial Intelligence techniques in order to introduce an efficient model for classification of white blood cells that can easily be implemented on traditional modern personal computer systems.

1.3 **Aims and Objectives**

Main objectives of this research are:

- Review and analyze the existing research work for the classification of various types of white blood cells: lymphocytes, monocytes, eosinophils, and neutrophils.
- Proposing a convolution neural network (CNN) model with decent generalization ability and great execution speed for the classification of four different types of white blood cells: lymphocytes, monocytes, eosinophils, and neutrophils.

1.4 **Structure of Thesis**

This thesis report is arranged as follows:

Chapter 2 Briefly represents the basic human immune system and introduction of pathology. Furthermore, the specifications of WBC's are also considered.

Chapter 3 covers the analysis of the existing research work done for the sub classification of white blood cells by the researchers did in past years.

Chapter 4 describes the proposed research work in details. This section also defines the method undertaken to classify white blood cells based on convolution neural network.

Chapter 5 Gives the review of datasets and the execution measures used for interpretation of the proposed research work. All results are explained in detail along with tables and figures needed.

Chapter6 concludes this framework and introduce the future scope of this research work.

CHAPTER 2: Pathology and Human Immune System

This chapter gives a concise view of few of the prior background, which comprise

- Human body Immune Scheme
- Introduction of Clinical Pathology

2.1 Human Body Immune Structure

Human immune structure is commonly defense structure across diseased and infecting organisms. It is built up of white blood cells (WBC's), proteins, antibodies and tissues which fight across those diseased, invading, viruses and bacteria's that are inimical for human body health [3]. Usually, human body immune structure shows important part to preserve the human body health. Any disorganization of human body immune structure leads to invading, infection and diseased. Human body immune system is shown in Figure 2.1.

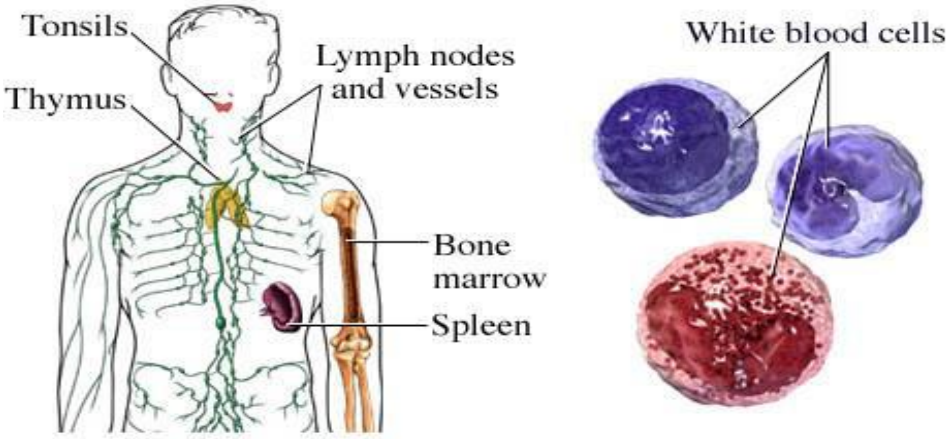


Figure 2.1 Human Body Immune System [3]

Human body Immune structure is similar to a interconnection of organs and different cells which defend the human body [4]. White blood cells (WBC's) show vital part in human body immune structure which well-known just as leukocytes. Leukocytes consist at numerous sections of the human body like that bone marrow, thymus and spleen. In view of large percentage, it is known as lymphoid organs. It spreads all across the human body throughout blood veins and lymphatic veins. Leukocytes mainly construct in the red bone marrow of bones in human body and small amount at another particular organs. Usually, a healthy human body system consists of 4 to 11 thousand units only in one cubic level blood. Different types of white blood cells are shown in Figure 2.2. White blood cells (WBC's) [5] have different four main types:

- Neutrophils
- Lymphocyte
- Monocytes
- Eosinophils
- Basophils

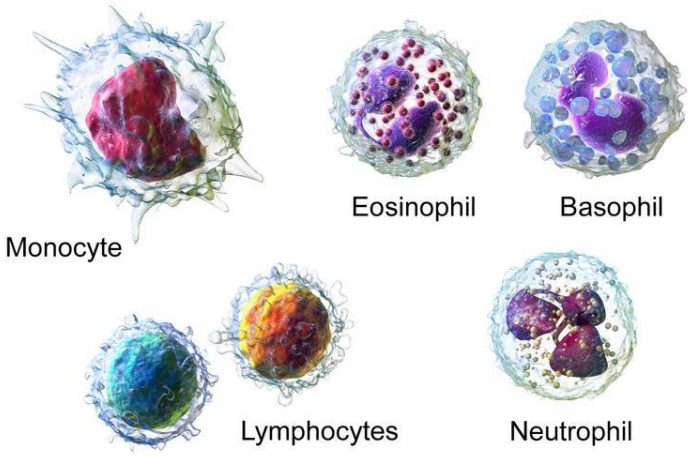


Figure 2.2 White Blood Cells [5]

2.1.1 Neutrophils

Neutrophils are existent in first highest percentage in white blood cells (WBC's), which are 50 to 70 percent, neutrophils of WBC's in a healthy person. Neutrophils has multi lobed nucleus that has usually two to five lobes. Granules also in it which are in several and in very small size and the color of cytoplasm is light pink. Neutrophils are shown in Figure 2.3. It is front defender when virus strikes. It protects against bacterial, virus infection and fungal. When neutrophils fight across the infections its granules become dark. Neutrophils granules become dark when they fight across the viruses. Life span of neutrophils is approximately 5.4 days.

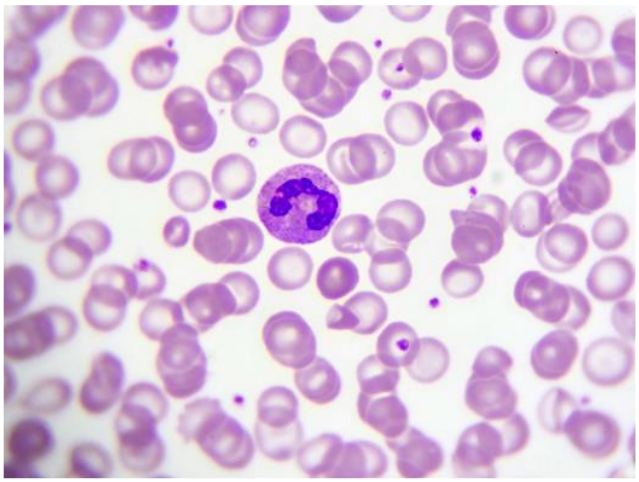


Figure 2.3 Neutrophils

2.1.2 Lymphocytes

Lymphocytes are sub type of WBC's, which is existent in second majority percentage and is shown in Figure 2.4. Adult human has 20 to 40 percent lymphocytes of the white blood cells. It has mono lobed nucleus. Lymphocytes shape can differ but its size is very small and in usual they have round appearance which consists of small number of bluish cytoplasm. Lymphocytes are shown in Figure 2.4. A unique attributes of this white blood cell has an ability to distinguish the foreign attacker which invaded in past and it include

a memory. Next time when intrusion the same attacker, it works across very fast as consider to prior. Lymphocytes size will be very large and numerous cytoplasm inside it when they fight across the attacker. Usually, it protects against diseases, germs, viruses and bacteria.

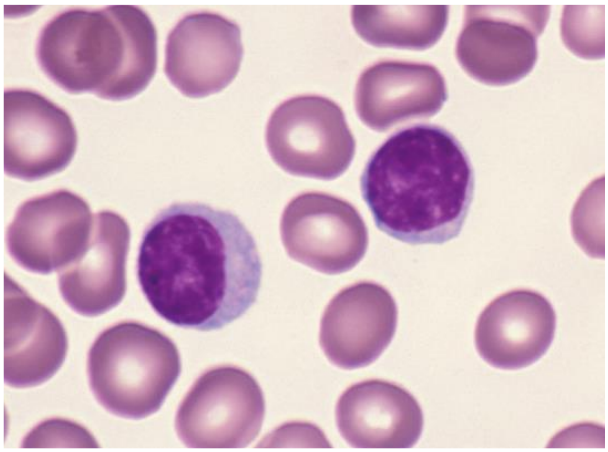


Figure 2.4 Lymphocytes

2.1.3 Monocytes

Monocytes are not existent in majority percentage in white blood cells, which are 5.3 percent monocytes of the white blood cells in a healthy person. It looks like lymphocytes but it has very large in size and numerous cytoplasm's. Monocytes are shown in Figure 2.5. Granules also inside it which are faintly stain and in usual they have not smoothly appearance which consists of small number of dull bluish gray cytoplasm. Monocytes not invaded across the attackers such as to other WBC's, but it produces protection by digesting the different particles. The life span of monocytes cell is among hours to days.

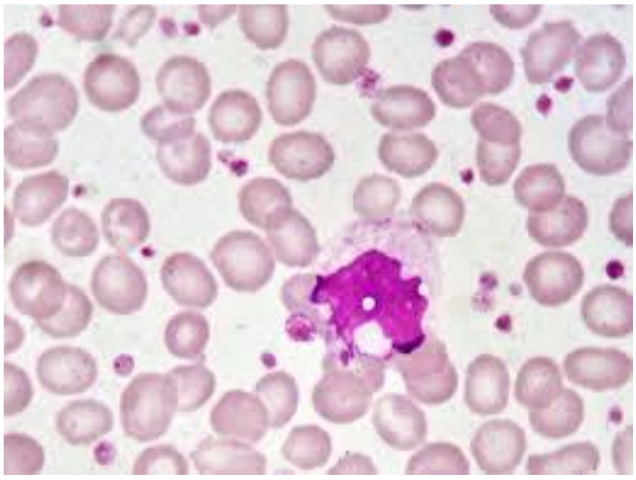


Figure 2.5 Monocytes

2.1.4 Eosinophils:

Eosinophils percentage in human body is 2 to 4 percentage which differs in a day or occasionally. Eosinophil has bi-lobed nucleus which connected to another small strand. Its cytoplasm consists full of granules and the color of cytoplasm is pink orange. It responded across parasitical infections, diseases, viruses, allergies, bacteria's, central nervous system and collagen infections. Eosinophils are shown in Figure 2.6. It also contributes defending from infections and allergies such as high fever, hives and asthma. The life span of eosinophils cell is 8 to 12 days.

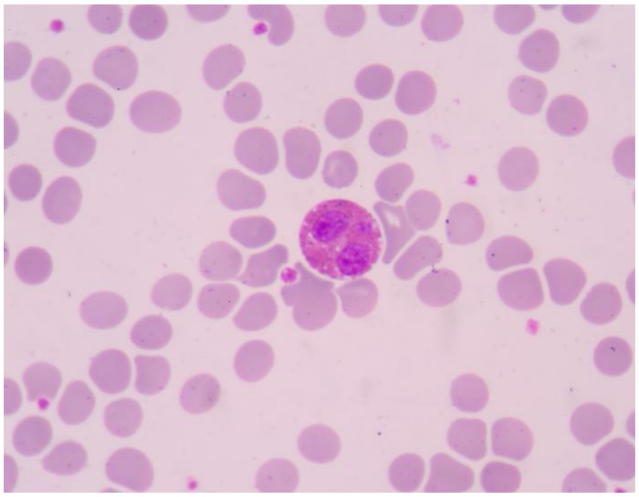


Figure 2.6 Eosinophils

2.1.5 Basophils

Basophils are the smallest percent of white blood cells (WBC's) in body which consist of less than 0.5 percentages. Basophil has bi-lobed or tri-lobed nucleus and it is hard to visualize due to unrefined granules in the cytoplasm's. Its cytoplasm consists of granules and the color of cytoplasm is deep blue purple color. Basophils are shown in Figure 2.7. These smallest white blood cells produce alarm to human body immune system when attackers to attack the blood. It produces a chemical like histamine which intimate allergic diseases. The life span of basophils cell is only few hours.

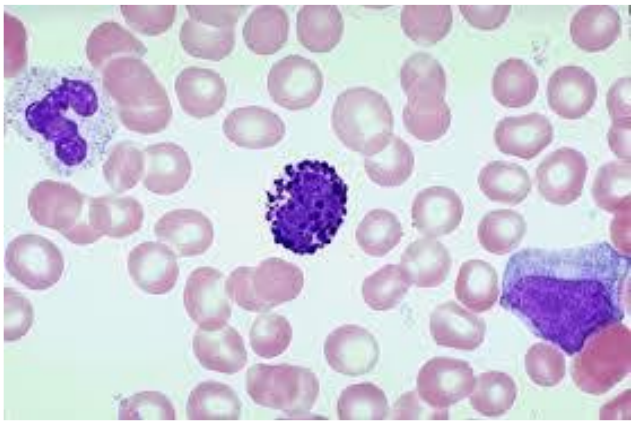


Figure 2.7 Basophils

Table 2.1 shows summary details of white blood cells. These percentage, diameter nucleolus, granules and Life time are different parameters which help in white blood cells classification. An imbalance in the number of any white blood cells type can cause different diseases.

Table 2.1 Summary Details of WBC's

Sr#	Type	Percentage (%)	Diameter (μm)	Nucleolus	Granules	Life Time
1	Neutrophils	50 - 70 %	10 -12	Multilobed	Fine, Faintly Pink	5.4 days
2	Lymphocytes	20 - 40 %	7- 8	Single	None	Number of years
3	Monocytes	5.3 %	15 - 30	Single	None	Hours to Days
4	Eosinophils	2.3 %	10 – 12	Bi-lobed	Full of pink-orange	8 – 12 Days
5	Basophils	0.4 %	12 - 15	Bi or Tri lobed	Large blue	Few Hours

2.2 Pathology:

Pathology is a field of medical in which analyzing of infections and abnormalities. Pathology word derives from Greek word 'pathos' which express 'suffering'. History of pathology derives from archaic time. In archaic times, Egyptian initiated culture to protect the documents of infections. At that era's papyrus on which begins the information's about bones, cancer injuries, hepatitis, HIV, parasites and other infections [6]. He determined to create record of different diseases and hereafter many inspirited people listed the information of different diseases such that tumors, hepatitis, HIV, parasites, tuberculosis, etc. In 19th centenary, after development of microscope, a biggest innovation had developed in clinical pathology. It was initial time, when individually cell of organs of the body intensely studied which guidance to distinguish the diseases.

Microscope is shown in Figure 2.8. The evolution of microscope and their possibility led to technical advancement in field of medical. Pathology is a vast field, which classifies into three main sub types in such a way anatomical pathology, molecular pathology, and clinical pathology. These categories more classified conforming to study and distinction form of research of pharmaceutical.

Figure 2.8 Microscope [6]

2.2.1 Clinical Pathology:

In this medical field of clinical pathology, diseases analyze through research laboratory study of body's tissues, organs and fluid. For example, examine the chemical part of blood, tissues and cells to analyze any bacteria and viruses are existent in body either not. Clinical Pathology is a vast field, which classifies into three main sub types in such a way chemical, hematology and immunology. Now we have considered only hematology, which are investigation of blood to recognize diseases respectively.

For the goal of white blood cells classification, hematologist took blood stain and applied preprocessing step. Hereafter through microscope, he manually categorized the white blood cells according to knowledge. This manual process is very time consuming and the apparent of human error appear. To overcome the current issue, a modern and fast technique proposes through modern and fastest technology of digital image processing.

CHAPTER 3: LITERATURE REVIEW

The immune system is a multifaceted network made up of organs, cells, and tissues working together to protect our bodies from infections and diseases [7]. White Blood Cells (WBCs) are a very essential factor of our body immune system defending our body against germs, viruses, and bacteria by ingesting them, destroying infectious agents, or producing antibodies. White blood cells are grouped into five different types:

- Monocytes (3-9%)
- Lymphocytes (20 - 40%)
- Eosinophils (2-4%)
- Basophils (0-1%)
- Neutrophils (50-70%)

The percentage value parallel to each subtype is the standard range of WBCs present in the blood of a normal healthy person. An absence or imbalance in the number of any white blood cells type can be caused by different diseases. These diseases are identified by doing blood tests. Each subtype of WBC helps in tackling certain types of disorders such as cancer, hepatitis, HIV, allergic infections, nephrotic syndrome, etc. It is noted that specific disorders end up misbalancing or attacking only certain types of WBCs. It is necessary to count and recognize the persons WBCs to recognize these diseases. Conventionally, hematology specialists perform categorization and numbering of these WBCs manually with the assistance of a microscope. However, this process is time-consuming, error sensitive, and complicated to operate [8].

With the growth of image processing and artificial intelligence in the field of medicine has released physicians from the burden of manual categorization. Many researchers and engineers have provided methods and different classification techniques to classify WBCs efficiently. Some of them used texture and geometrical feature classification

methods, which includes preprocessing stage where medical images are denoised and region of interest is segmented out. Next stage is feature extraction, where different texture and statistical features are obtained from a certain image and passed to some machine learning classifier such as KNN, multilayer perception, Bayes classifier etc. for classification. Second classification method is using image as features and passing them through a convolution neural network (CNN), which consists of convolutions layers, fully connected layers and a output layer. There are many different types of CNN models that are used by researcher for WBCs classification.

This chapter reviews the relevant work done in the domain of sub type of white blood cells (WBC's) classification. This chapter categorized as:

- Existing Techniques
- Comparative Evaluation

3.1 Existing Techniques:

Many researchers and engineers have provided methods and different classification techniques to classify WBCs efficiently. Some of them used texture and geometrical feature classification methods, which includes preprocessing stage where medical images are denoised and region of interest is segmented out. Next stage is feature extraction, where different texture and statistical features are obtained from a certain image and passed to some machine learning classifier such as KNN, multilayer perception, Bayes classifier etc. for classification. Second classification method is using image as features and passing them through a convolution neural network (CNN), which consists of convolutions layers, fully connected layers and a output layer. Some of them explained in this literature review chapter and classified in the group on the base of inputs.

- Geometrical And texture features
- Images as features

3.1.1 Texture and Geometrical Features

Rosyadi et al [9] produced a research, which introduced a method to segment and recognize white blood cells. They used optical microscopic to produced images of blood samples just as dataset. In this research white blood cell sub types were lymphocyte, neutrophils, monocytes, eosinophils used to classify. They divided their research into four phases such that image preprocessing, features extraction, segmentation and classification. First stage, image pre-processing consists of reconstruction of RGB images of white blood cells to gray scale and then binary images. After this in second stage cropping, edge detection, and resizing enforced. In feature extraction stage, five different geometrical features were examined in this analysis. Those geometrical features were normalized, circularity, area, solidity, eccentricity was used. Most commonly used K-Means algorithm for classification. It worked depend on Euclidean distance. In Euclidean distance we calculate distance from cluster centroid to specific substance. In K-mean algorithm number of centroids is recognized and then it assigns every data point to nearest cluster. Their research focused on the determined and significant of each feature related to accuracy and influence on accuracy. After those experiments, they initiate circularity features was more important than other features because it acquire approximately 67 percent accuracy 43 which was the lowest and eccentricity features had highest accuracy which was reached up to 20 percent. So, their research concluded that selection of features more significant rather than more features.

Haung et al [10] did their research, on the same problem of sub classification of white blood cells which introduced the method of segmentation and recognize sub types of white blood cells. Their research divided into four parts: nucleus segmentation and detection, features extraction and classification. In segmentation of nucleus used leukocyte nucleus enhancer in which increase distinction of nucleus colors and suppressed other color channels. After leukocyte nucleus enhancer process, multiple levels of Otsu's thresholding enforced to overcome the other than leukocyte cells. In features extraction phase, used gray level co-occurrence matrix in which eighty texture features selected through each image. Shape based feature thus roughness and compactness are added. PCA used to lower the dimensions of large dimension features set. Genetic –based parameter selector used for classification in which fifty times cross validations applied. Though their research concluded that genetic based parameter algorithm more appropriate than K-means clustering. After experiment, result of genetic algorithm got 91 percent accuracy.

P.Yampri et al [13] also produced their research on the same problem of white blood cells classification. In their research, they used smear images of blood as input and then segmented out WBC's through color base method. Automatic thresholding process applied to separate cytoplasm of white blood cells and nucleus and that part of cells which further handled to extract out different features. For features extraction they used Eigen cells. Different steps were applied such that cell images to vectors, calculated mean and covariance and then computed Eigen values and Eigen vectors. Principle of component analysis (PCA) used to reduce high dimensions to lower dimensions of Eigen space. Through their research, they got 92 percent accuracy.

Gautham et al [14] produced a research on the same problem of leukocyte classification. Their research divided into four steps such as pre-processing, features extraction, segmentation and classification. First stage, image pre-processing consists of

reconstruction of RGB images of white blood cells to gray scale and then binary images. Histogram and contrast stretching also applied. After that Otsu's thresholding process used for segmentation, segmented out regions treated with morphological operations. Closing operations executed in which dilation process come after by erosion. They used nucleus to extracting different geometrical features such that area, circularity, eccentricity and perimeter. After experiment, result of Naïve Bayes Classifier got 80.88 percent accuracy.

S.S.Savkare did their research which introduced a method to segment and recognize white blood cells. Process initiated with preprocessing step in which used Laplacian and median filter to increase the appropriate information's. After pre-processing stage, it change the input images to Red, green, blue (RGB) to HSV and then HSV to L*a*b where 'a' is chromaticity layer which specify color fall through red green axis , 'L' is luminosity layer, and 'b' is chromaticity layer which specify color fall from blue yellow axis. Most commonly used K-Means algorithm for segmentation of white blood cells. Moreover, refine to results, it used mathematical morphological operations and also used watershed algorithm to filter out each cell. After experiment, their research of segmentation by K-Mean produced good results.

Mazin Othman et al [15] did their research on the same scenario of sub classification of white blood cells. They divided their research into three phases such that segmentation, labeling that stores the location and number of each white blood cells and features extraction. The initial step is image segmentation, next step is labeling that stores the location and number of each WBC's and the final step is take out expressive features. After segmentation process the most important sixteen features of these white blood cells passed as input to the network. Most commonly used multilayer perceptron back propagation (MLP-BP) algorithm for classification of WBC's. Half of the dataset were

used to train the network and the remaining half used for test. After experiment, they achieved 96% classification accuracy.

So, we completed existing literature survey corresponding texture features and geometrical features used toward classify which significant features are list out below in the Table 3.1.

Table 3.1 Features Tables

SR No.	Feature Types	Features
1	Geometrical	Perimeter, Area, Eccentricity, Circularity, Solidity, Compactness, Roughness, Eigen Vectors
2	Texture	Gray Level Co-Occurrence Matrix (GLCM), Local Binary Patterns (LBP)

3.1.2 Images as Features

Macawile et al [11] did their research on white blood cells classification. Their research proposed a process to classify white blood cells sub types: eosinophils, monocytes, lymphocytes, and neutrophils. Segment out white blood cells sub types were lymphocyte, neutrophils, monocytes and eosinophils through blob analysis of Hue, Value and Saturation (HSV) saturation elements. Transfer learning method used for classification purpose. Transfer learning method used different pre-trained CNN which has capability to classify all type of objects. They used different pre-trained convolution neural network such as ResNet-101, GoogleNet and AlexNet. A sample of CNN architecture is presented in Figure 3.1.

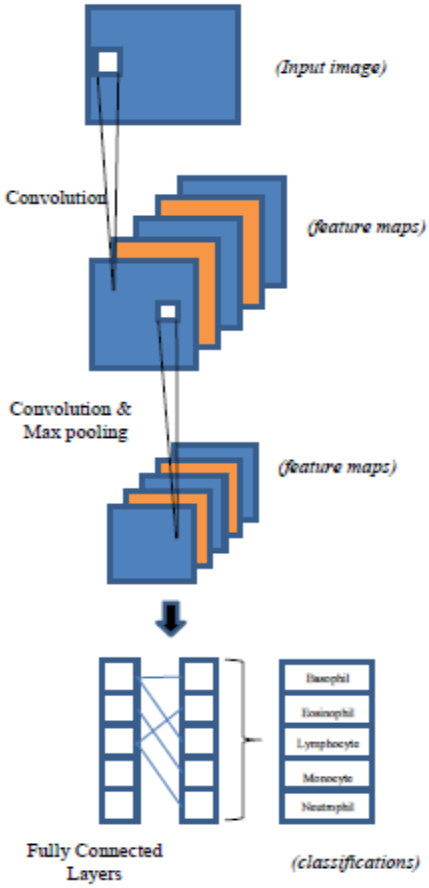


Figure 3.1 Architecture of Convolution Neural Network (CNN) [11]

CNN presented that huge computations and more time needed to train. Thus, they used different pre-trained Convolution neural network which basically used for classification of white blood cells. They were compared the results of those three different pre-trained CNN networks and Alex-Net gives highly correct results was 96%.

Wei Yu et al [16] did their research on the same scenario of white blood cells classification. In their research, deep learning method used for classification purpose. In this method, they used CNN which was considered above. They also used different pretrained networks such as ResNet50, VGG 16, Xception, VGG19, and Inception V3. These different networks pretrained by ImageNet Large Scale Visual Recognition

(thousands different classes classified from more than millions of images). Sample of this architecture contained 3 fully connected layer is shown in Figure 3.2.

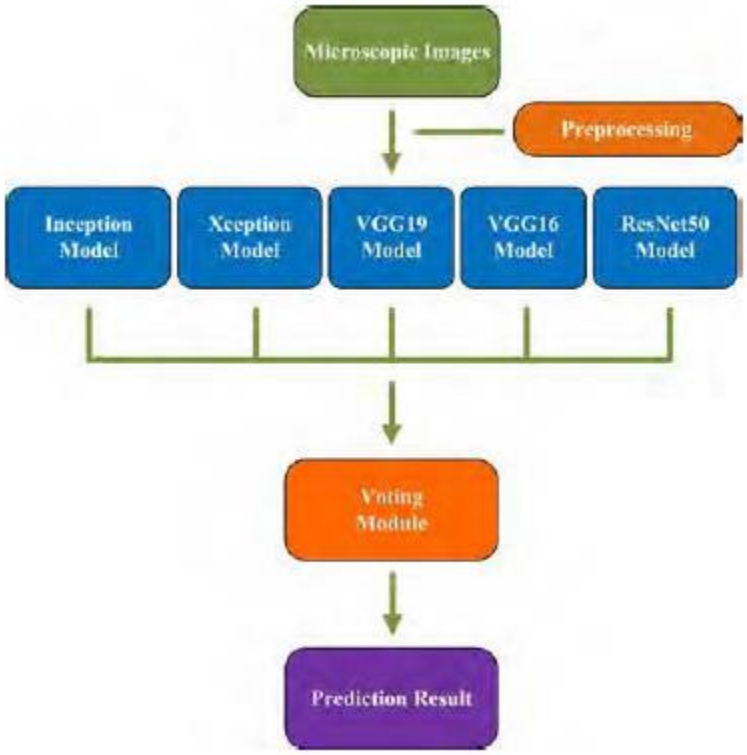


Figure 3.2 Voting Model of CNN's [16]

These different pre-trained models are tuned about requirements and compared their results through voting mechanism to achieve highly accurate final results. After experiment, results of convolution neural network (CNN) achieved 88.5% accuracy.

O Liang et al [19] did their research on white blood cells classification by digital image processing through combining deep neural networks. In their research, shown an interest about usage of CNN which was ineffective to preserve long term relationships between the essential features of images. They used BCCD dataset of 12444 microscopic images for the classification of four major subtypes of white blood cells. They proposed a procedure which produced good results. The introduced method was combination of

CNN with recursive neural network (RNN). They passed their input images to both models separately and then merged the features extracted from both models. Resultant of this merger module is sent to output layer predicting results using SoftMax. They used transform learning process to different pre-train ImageNet datasets for CNN and custom loss functions also used for more accurate and fast results. After experiment, they achieved 91% accuracy.

H. Fan et al also produced their research on the same problem of white blood cells classification by deep neural networks. They introduced method of leukocytes localization and segmentation that named 'Leukocyte Mask'. In this process, they used different pixel level information's to train deep convolution neural networks. The convolution neural networks occupied to find the area of WBC's and separated area of white blood cells forward propagate by deep neural networks to acquire the Leukocyte-mask. They confirm their final results and compare with current techniques. They concluded their research results were also accurate, fast and robust.

Ashish Khanna et al [20] did their research on the same scenario of leukocyte classification. They used publicly available dataset of 13000 microscopic images for classification of four major subtypes of white blood cells such that Eosinophils, Lymphocytes, Monocytes and Neutrophils. Their CNN architecture consists of two convolutional layers, two pooling layers; one fully connected hidden layer and an output layer. 88% of classification accuracy was achieved. A sample of CNN architecture is shown in Figure 3.3.

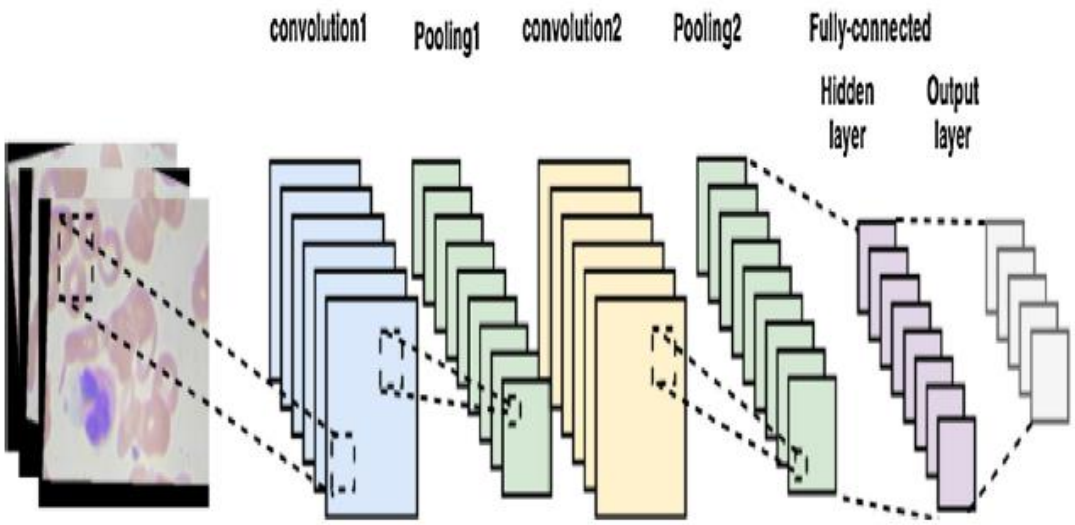


Figure 3.3 Architecture of the CNN Model [20]

Table 3.2 shows the work done of different authors on white blood cells classification.

Table 3.2 Comparison Analysis Chart

Author	**Year**	**Dataset**	**No. of Images used Training/Testing**	**Method**	**Accuracy**
S.S.Savkae [17]	2015	Local Dataset	70	Water Shed	88.77%
Rosyadi [9]	2016	Local Dataset	N/A	K-Mean	67%
Gautam [25]	2016	Local Dataset	20 / 68	Naïve Bayes Classifier	80.88%
Wei Yu [23]	2017	Drump tower hospital of Nanjig university	N/A	CNN	88.50%
Mazin Z.Othmn [15]	2017	Local Dataset	50	MLP-BP	96%
Macawie [11]	2018	ALL-IDB dataset	N/A	AlexNet	96%
O.liang [19]	2018	Local Dataset	12000	CNN + RRN	91%

Author	Year	Dataset	No. of Images used Training/Testing	Method	Accuracy
Haoyi Fan [21]	2019	jiangxi tecom science cooperation, BCISC,LISC	90 (54/18)	ANN	95.90%
M. Sharma [37]	2019	BCCD	9957/2487	CNN	87%
M. Togacar [38]	2019	Local Dataset	N/A	CNN	97.78%
E.H. Mohamd [40]	2020	BCCD	2500/620	Pre-trained deep learning models	97.03%
B. Ergen [42]	2020	Local Dataset	N/A	CNN & feature selection	97.95%
A.Cinar [36]	2021	Kaggle,LISC	N/A	Alexnet-Google Net-SVM	99.73%, 98.23%

CHAPTER 4: EXPERIMENTAL METHODOLOGY

In this research, a classification convolution neural network (CNN) model is proposed for the classification of white blood cells. White blood cell sub types were lymphocyte, neutrophils, monocytes and eosinophils used to classify. We have used different convolution Neural Network (CNN) models for the classification of 4 major WBCs subtypes, Eosinophils, Lymphocytes, Monocytes and Neutrophils. The different CNN models used in this research are ResNet-50, Inception V3, VGG16, LeNet and MobileNetV2. The complete architecture of these CNN models and proposed CNN model is described in this chapter.

4.1 Convolutional Neural Network

Convolutional neural networks (CNN) are very prominent in the domain of image processing and artificial intelligence. For their remarkable capability to produce un-supervised perceptive decisions, deep learning techniques have come to be one of the hot fields while it becomes to application of Artificial Intelligence in actual life issues. The initial of convolutional neural networks scientist is Yann-Lecon from New-York who further effort just as organizer in Facebook artificial intelligence groups. The design of convolutional neural network has comparison with human neural structure. Just as in human and vertebrate's neural structure, deep learning models study information's by a convolutional neural network layered framework. This idea,artificial convolutional neural architectures were described in form to produce right decisions at appropriate time without except human guidance [46]. The working structure of convolutional neural network (CNN) determines its sources from the performance principle of ANN. Artificial neural network is constituted of multiple perceptron (also called neurons) along different layers. In artificial neural network all input weights out of one hidden layer are transferred to the next layer but

these weights not propagated back. Therefore, it is called Feed Forward Neural Networks. Furthermore, an artificial neural network might or might not have different hidden layers but input layers and output layers are a necessary. At the same time convolutional neural networks have various hidden layers across with input layers and output layers consist of many neurons. In convolution neural network, the input layers and all hidden layers are really convolution layers. These different input and hidden layers can direct connect with one another rather pooling layers. The different convolution layers determine for extracting effective features of input images which is separated into different quadrilateral and sent these input images for processing. The architecture of a convolution neural network is shown in Figure 4.1.

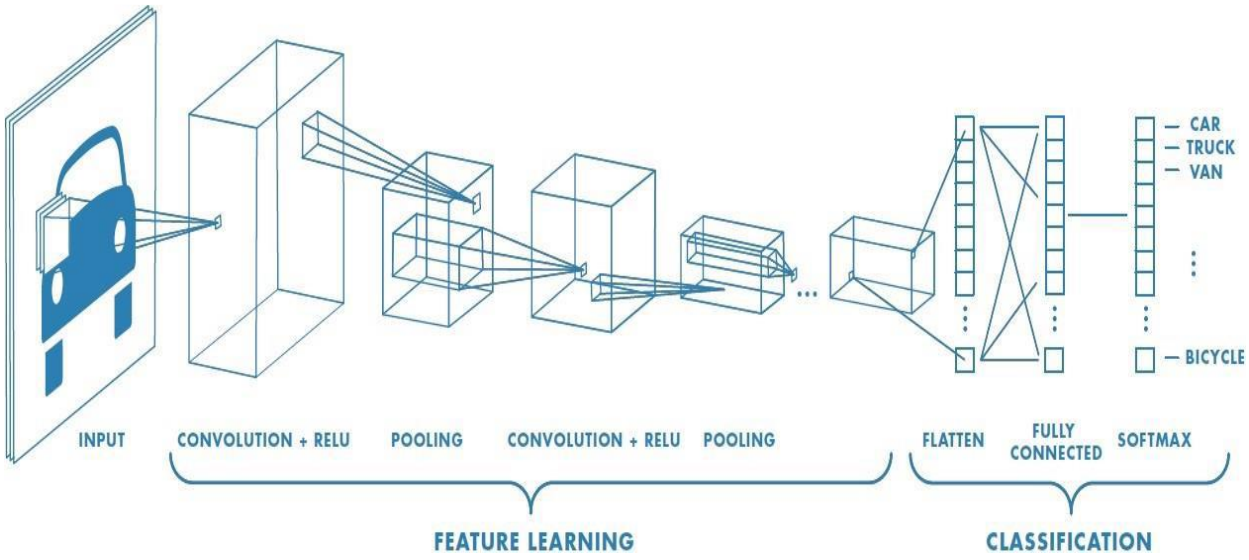


Figure 4.1: Typical CNN architecture. [46]

The specific architecture of CNN and applications of convolution neural network are defined in this chapter.

4.2 Architecture of Convolutional Neural Network

A CNN is built up of input layers, different hidden layers and output layer. The hidden layers contain many convolution neural layers, activation functions, different pooling layers and many fully-connected (FC) layers. The convolutional neural network comprises of many convolutional layers, activation functions and different pooling layers determined by the CNN architecture. The architecture till the many hidden layer is appropriate for features extraction during the last FC layer are given for classification purpose [47]. The major building sections of a common convolution neural network are detailed below:

4.2.1 Convolution layers:

The CNN layer comprises various independent filters or kernels and those kernels are convoluted individually with input images. The CNN system is complete in such away that the filter slides above the input image and dot product within each interrelated image pixels and the filter is taking. Arbitrary initialization of kernels is complete and these kernel values are restored depend on the consequent study by the CNN network. Originally the opening layers review for primary patterns like corner and lines. As the system maintain to train, the kernels are literally returning dot product of neuron of the earlier layers with the relevant weights.

4.2.2 Rectified Linear Units (ReLU) layers:

In convolution neural network (CNN) it is demonstration to cover an activation layers and nonlinear layers instantly after the convolution layers. The objective is to propose non linearity to the structure that was carrying out only linear applications all over the

different convolution layers. Rectified Linear Units mostly facilitate the training procedure of the neural network without induce a notable outcome in the model achievement [47]. It alleviates the disappearance gradient problem which appears just as the training procedure is gradual in lower hidden layers of the convolution neural network as the gradient loss exponentially over the hidden layers. The different activation functions used by Rectified Linear Units is f(x) =max (0, x)that is practiced to total values of present layer of output is shown in Figure 4.2. In short, Rectified Linear Units change the different negative value to zero. This method increases the number of non-linearity in the framework without depressing the interested fields of convolution neural layers.

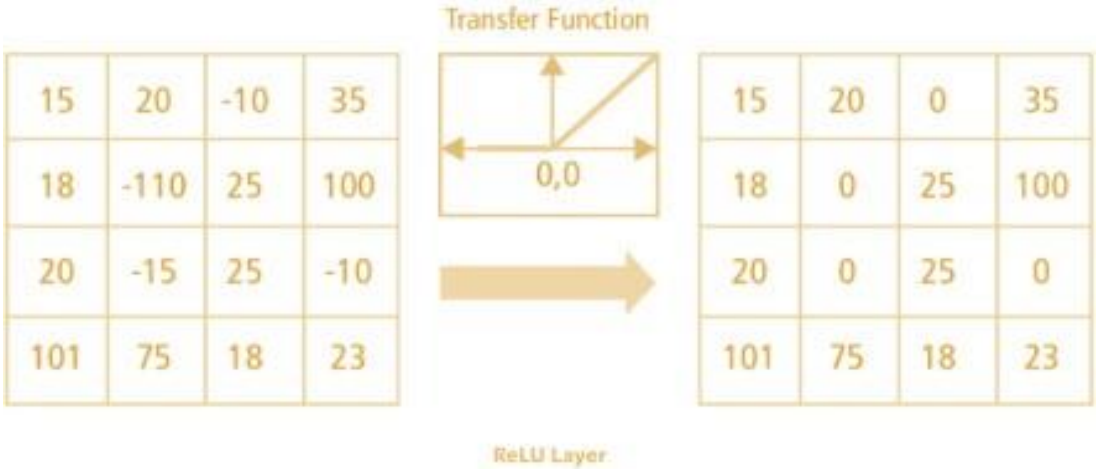


Figure 4.2: ReLU Operation [47].

4.2.3 **Pooling Layers**

The pooling layer is familiar while down sampling layer and it objective is to lower the spatial resolutions of design and basically minimize the neural network calculations by diminishing the total number of parameter. There are different non-linear functions applicable for performing pooling layer like max pooling function, average pooling function and L2 pooling. This means spatial size of input is minimized [48]. The idea after the pooling functions is this once the specific feature for a location is found that,

then the exactlocation is not necessarily required. Thus, there are two advantages of pooling. First one is this it minimizes the calculations by minimizing the total parameters. Secondly, it reduces over-fitting. Average and max pooling functions are shown in Figure 4.3.

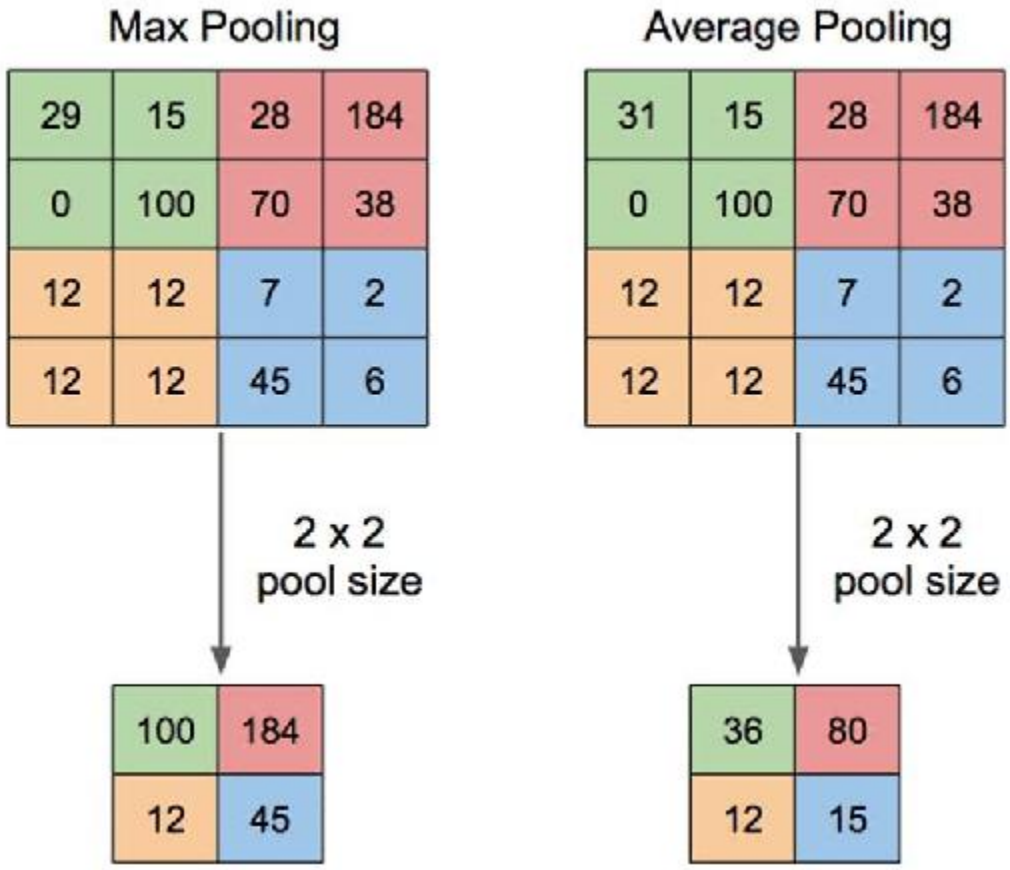


Figure 4.3: Max and Average pooling example [48]

In a typical CNN, after several convolutional, ReLU and pooling layers, Fully Connected (FC) layers are used for high level reasoning. The high-level features in an image are extracted by the convolutional layers and those features can then be combined in a non- linear fashion using FC layers. The features from convolutional layers perform better whencombined together by FC layer. The neurons in FC work on

the mechanism same as normal artificial neural network (ANN) or multi-layer Perceptron (MLP).

4.2.4 Fully Connected layers

In a common convolution neural network, after numerous convolutional layers, rectified linear units and various pooling layers and Fully Connected layers are used as high-level analysis. Convolutional layers are used for extracted high level features of an image. The high-level features through convolution layers achieve improved when connected together through fully connected layers. Neurons in fully connected layers work on same mechanism just as usual artificial neural network and multi-layer Perceptron.

4.3 Convolutional Neural Network based methods

When using Convolutional neural network for each classification problems, it may be completed by two approaches as shown below in Figure 4.4. The first way is well known as "Learning from the Scratch". In the first approach weights of all neurons are arrange with different arbitrary values and network is completely depending on the preferred dataset for training purpose. The second approach is well known as Transfer Learning method in which different trained parameter in details of all the weights learned from another dataset and these are used as initialization of training purpose for the preferred dataset. Transfer learning approach may be applied in two different methods. The first method is Fixed Feature extraction [49] [50] and the second method is Fine tuning.

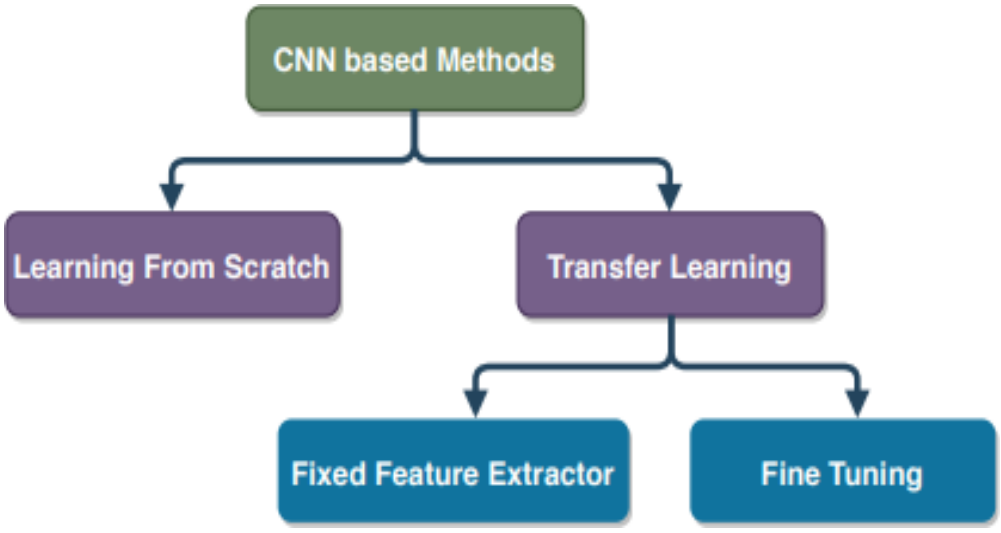


Figure 4.4: Convolutional Neural Network based different Methods [50]

4.3.1 Fixed Feature Extractor Method

Fixed feature extractor is the first method of transfer learning approach. In this method all biased and weights are learned from another large dataset are precisely used for classification issue of selected dataset and here is no any need to re-train the convolutional neural network about the specified dataset [50]. Usually it denotes that the fixed Feature extractor chunk of the convolution neural network architecture is current used and fixed features extracted through this system and it can be classified through using several classifiers such as linear support vector machine (SVM) or soft-max classifier.

4.3.2 Fine tuning Method

Fine tuning method is implemented through re-train the convolutional neural network on the specified dataset. This means we freeze all the weights of any of the first layers of network and train just some initial layers about the specific task of dataset. Usually

for doing that, first all of the neurons in last output layer are set about overall number of classes in the specified dataset. After that the bias and weight about freeze layers of CNN network are arranged based on pre-trained CNN architecture. These pre-trained parameters along with total number of epochs, learning rate and batch size of the network and different optimizers are explained and further training of un-freeze layers of the network is complete in form to adjust all the weights and then bias values of these network layers based on the task specified dataset [50]. In our research we used that type of transfer learning approach.

4.4 **Convolutional Neural Network (CNN) Models**

As discussed, previous, Different CNN models used in this research are ResNet-50, Inception V3, VGG16, LeNet and MobileNetV2.The complete framework of these CNN models are described in this chapter.

4.4.1 **VGG-16**

VGG-16 [51] is a pre-train CNN network that was considered after in 2014 through A.Zisserman and K.Simonyan in paper of “Very Deep Convolutional Neural Networks as Large Scale Image pattern Recognition". The highest test accuracy accomplished through VGG-16 on Image-Net dataset was 92.7%.Image-Net is a very large dataset consists of about 14 million different images which associate to 1000 different classes. This architecture is advancement in AlexNet pre-train architecture and it changes the wide size offilter (5x5 and 11x11) in first two convolution layers through 3x3 multiple filters.

The default size of the input image is 224x224x3 where 3 show number of depth (channels) of the image. The input image is given to the different

convolutional chunks consist of convolutional layers followed through max pooling layers. The maximum filter size in convolution layers is 3x3. All of these filters slide above the given image and extract corresponding high-level features. In one convolution layer, the kernel size is 1x1 in which linear transformations followed through non linearity transformations. In this case the stride of convolutional layer is fixed in order to 1. The padding system is adjusted in this way that following all convolutions the spatial resolutions is maintained, thus padding values in this case is adjust to 1. In this case the kernel window size of maximum pooling is 2x2 by stride of 2.

Followed through stacks of convolutional chunks (convolution layers + maximum pooling layers), there exist three FC layers. In case that ILSVRC competition such as specified dataset (Image-Net) has about 1000 classes, thus the total channels in VGG-16 last fully connected layer was kept remain equal to 1000. In this case, all hidden layers (convolutional chunks) are implemented with rectified linear unit, in which nonlinear activation functions.

The specified architecture of VGG 16 is shown below in Figure 4.5.

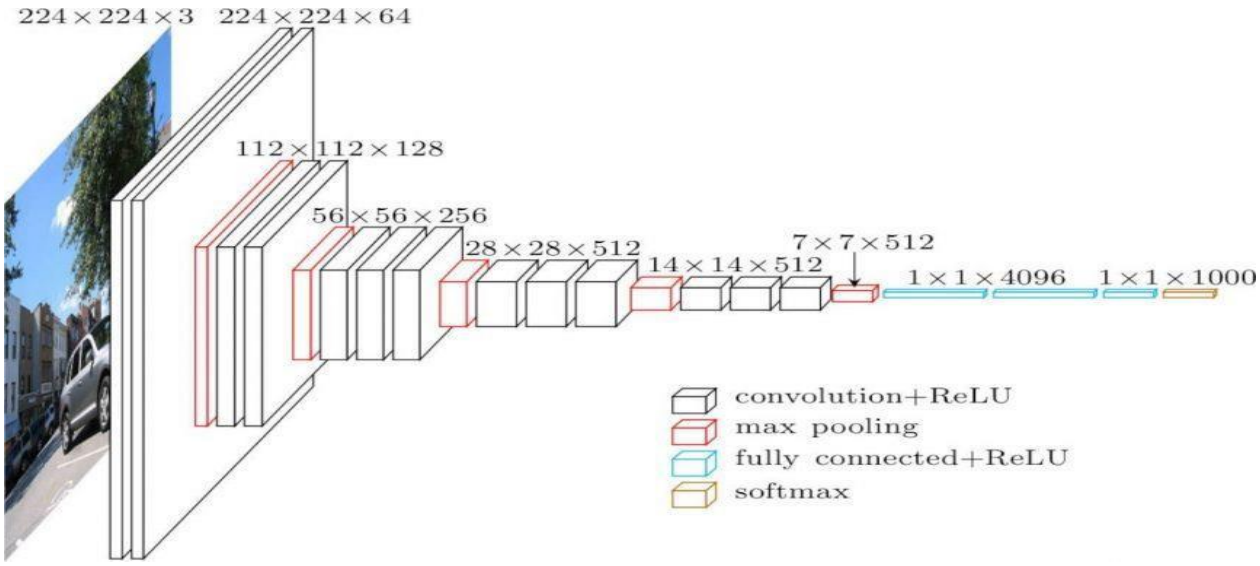


Figure 4.5: Detailed Architecture of VGG-16 [51]

4.4.2 **Res-Net 50**

Res-Net 50 (Residual Neural Network) is an architecture given in 2014 by Shaoqing Ren [52]. In prior research, recursive convolutional neural network (R-CNN) and fast recursive convolutional neural network (R-CNN) are proposed and it indicated a deeper convolutional neural network (CNN) produce a low error rate. ResNet win ILSVRC in 2015 classification function. A lot of success of convolutional Neural Networks has recognized to all of additional neural layers. The perception beyond their functionality is this all these layers continuously get high level complex features. First layer get determines edges of an image, second layer reviews shapes, third layer determines different objects and the last layer (fourth) reviews eyes and so forth. We want to get deeper convolution neural network, Kaiming He et al [52] experimentally represent that highest threshold as depth with conventional convolutional neural network.

Kaiming He et al [53] plot testing and training errors of 20 layers convolutional neural network (CNN) versus 56 layers convolution neural network (CNN) as shown below in Figure 4.6. These plots describe that including more CNN layers must produce more complex functions and failure must be associated to overfitting. Although, plot represents that training errors of 56 layers is more than 20 layers CNN network and is shown in Figure 4.6.

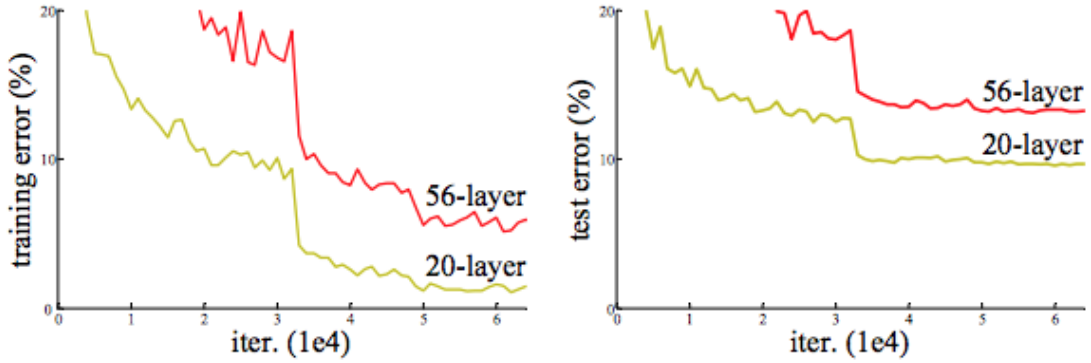


Figure 4.6: training and testing error along twenty layers and fifty-six layers plain networks. The deeper convolution neural network has highest training error and test error [53].

The failure of 56 layers convolutional neural network CNN can be criticism on optimization functions, activations of CNN network, and the vanishing gradient problems. Exploding gradients are mostly easy to criticize for that; although use of batch normalization confirms that gradient have strong norm. The problems of training of very deep convolutional neural networks (CNN) have been reduced with institution of new CNN layer. Deep residual convolutional neural network method is initiated to minimize the previous issues. By delivering a similar mapping from previous layers it is very easy to develop it. Basic residua blocks are shown in Figure 4.7.

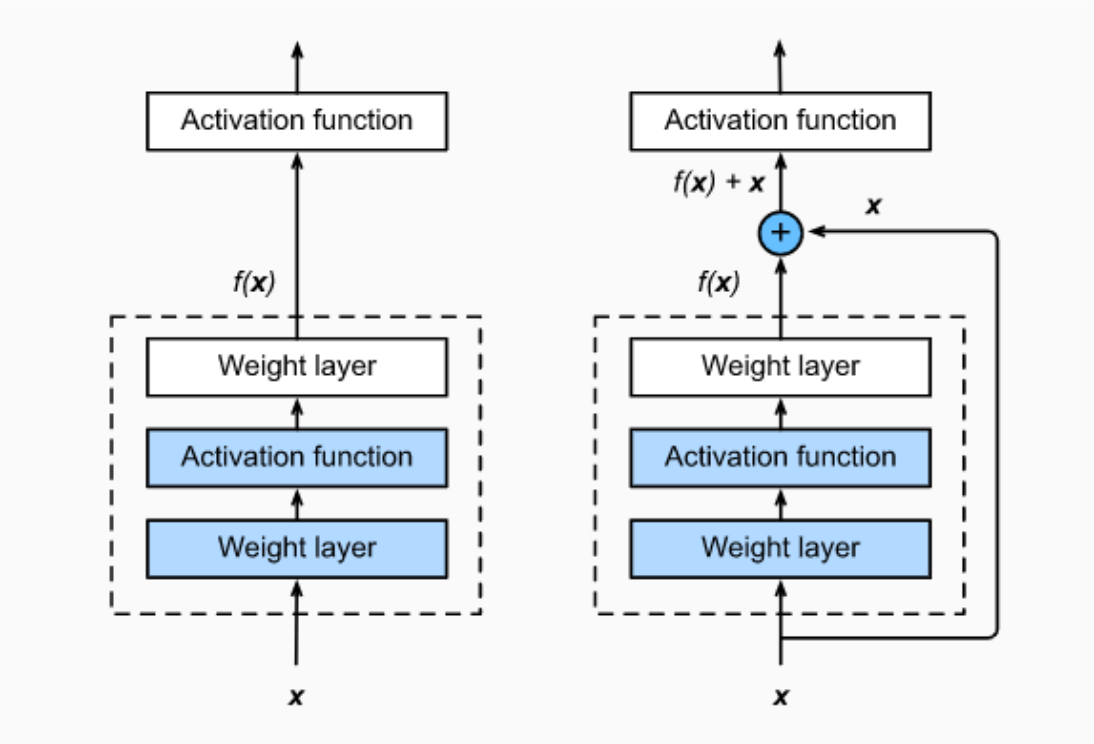


Figure 4.7: Residual Blocks (He et al., 2014)

Residual networks might be used when inputs and outputs have same dimensions. We used the following formulas to determine output. Residual block calculation is shown below in equation 4.1.

In equation 4.1 y represents the outputs, x shows the inputs, F is the activation functions and W_i are the updating weights**.**

$$y = F(x, \{W_i\}) + x \quad (4.1)$$

Following some properties and theories relevant to identity functions as described below:

- The insertions of identity connections did not initiate extra parameters. Although, the computations complexity for deep neural networks and residual deep neural networks are almost same.
- The dimensions of parameters must be identical for executing addition operations. All the dimensions might be matched through using following methods:
    I. Extra zero should be add for increasing the dimensions.

II. Projection could be used toward equivalent the dimensions i.e. (1 x1 convolutional)

Deep convolutional network of VGG16 is shown in Figure 4.8. Plain of 34-layer CNN network and 34-layer residual deep neural network. Concurrently in residual deep neural network, it represents less kernels and low complexity during training method about VGG network [53].

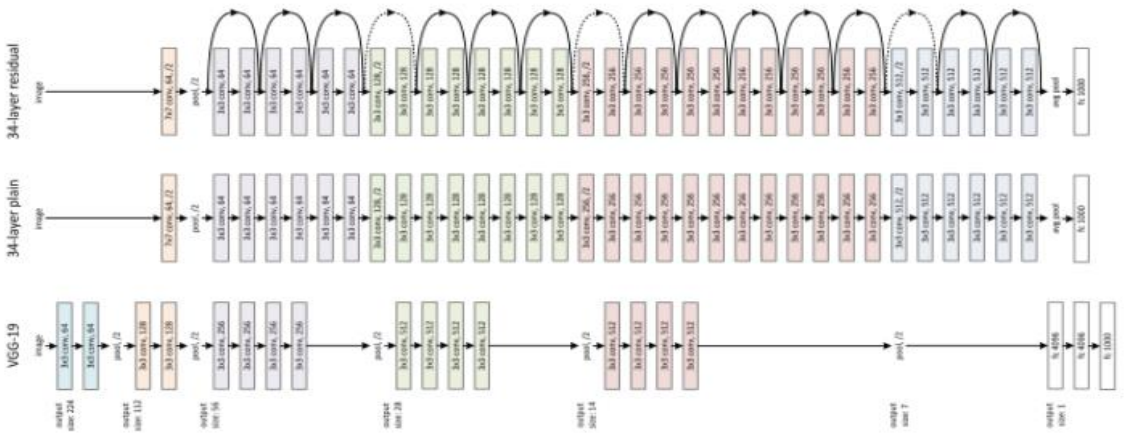


Figure 4.8 Deep Neural architecture of Image-Net.VGG19, plain neural network along 34 parameters and residual deep neural network along 34 layers [53].

A convolutional network with a filter/kernel sizes of 7*7; 64 all the kernel/filters with stride of 2 offering us layer1. Next layer we saw maximum pooling with stride of 2. In next convolutional 1*1; 128 after this 3*3; 64 different filters and last 1*1; 256 different filters. These 3 layers done over in three times thus giving nine layers. Kernel size of 1*1; 128 after this a filter of 3*3; 128 and last filter of 1*1; 512 this is replicated for four times thus give us total 12 layers. After this a filter of 1*1; 256 and also 2 more filters the size of 3*3; 256 and 1*1; 1024 that is copied total six times thus it gives us total eighteen layers. Then a 1*1; 512 filters along two filters of 3*3; 512 and 1*1; 2048 and that is replicated total three times thus it gives nine layers. After this we apply maximum pooling and at last fully connected layer (FC) consists 1000 neurons and a soft- max activation

function is used thus it gives us one layer [52]. In this we did not count different activation functions and maximum or average pooling layers. Thus, this gives us total 50 layers of deep convolutional layers. Architecture of ResNet50 is shown below in Figure 4.9.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 4.9 Architecture of ResNet50 [52]

4.4.3 **Inception-v3**

Inception-V3 [54] is deep neural network architecture extensively used as classification task. Inception-V3 architecture model was instituted by Szegedy in GoogleNet architecture wherever Inception-V3 was designed through updating inception modules [46]. Inception-V3 deep convolutional neural network has various symmetric and a-symmetric building chunks, wherever every chunk has many blocks of convolution layers, maximum and average pooling layers, dropout layers, concatenated and fully connected (FC) layers.Inception-V3 architecture has total 42 layers and acquire approximately 29.3 million training parameters, which measures the computations cost and it 2.5 highest than Google-Net architecture. Definitely, the combinations of lowest parameters were count and added regularization along batch normalization of different classifiers labels allow training of highest quality neural network on comparatively

modest sized of training set and is shown below in Figure 4.10. Inception-V1 [54] in which salient chunks in the images could have very highest variations in image size. Due to these huge variations in the locations of the information's, selecting the appropriate filter kernel size for convolutional task it comes tough. The larger filter size is selected for information's that is assigned globally, and smaller filter size is selected for information's that is assigned locally. It also difficult to delivers gradient updates through entire architecture. Large convolutional tasks are computationally very expensive. To overcome that problem the filters with various sizes work on the same level. The inception architecture is shown below in Figure 4.10. It implements convolution operations on input, with three different sizes of kernels (1 x1; 3 x3; 5 x5). Moreover, maximum pooling operations are performed. The outputs of networks are connected and then it sends to next inception modules [55].

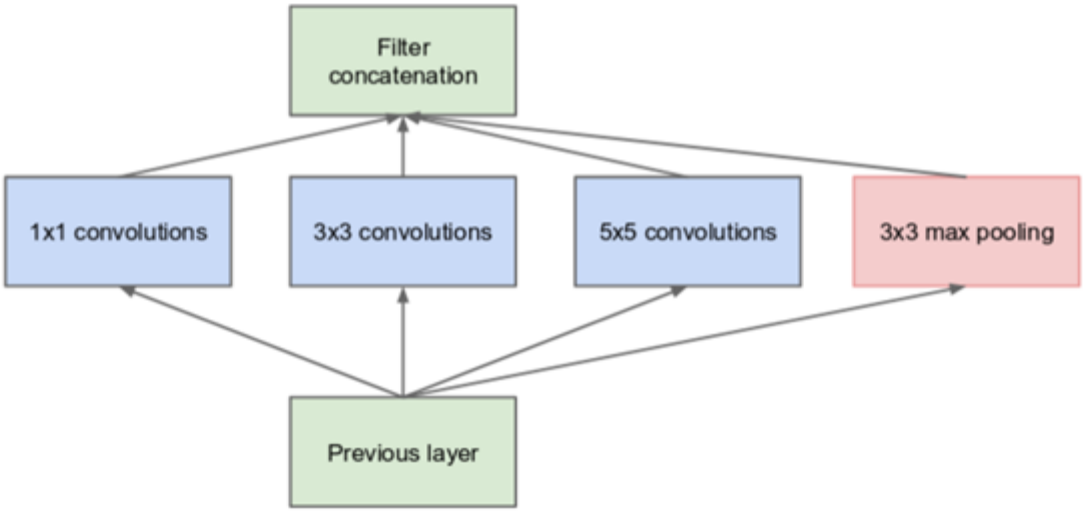


Figure 4.10 Inception Architecture [54]

Deep neural networks are computationally very expensive. To make deep neural network cheaper, restrict the different input channels through adding 1x1 extra convolutional layer before the 5x5 and 3x3 convolution layers. The 1x 1 convolutional layer is initiated after

the maximum pooling operation, instead than before the max pooling layer [55]. The inception architecture with dimensions reduction is shown in Figure 4.11.

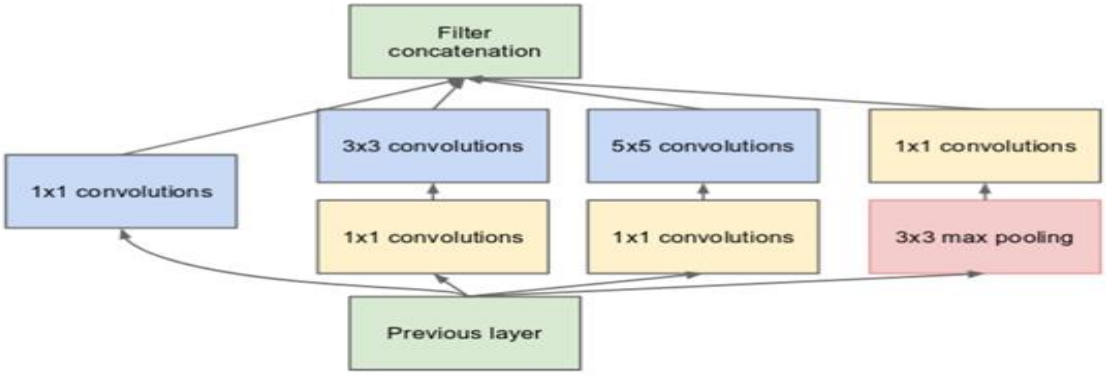


Figure 4.11 Inception architecture with dimensions reduction [54]

Using the architecture of inception module with dimensions reduction, a deep neural network was made. The architecture of Inception-v1 module is shown below in Figure 4.12.

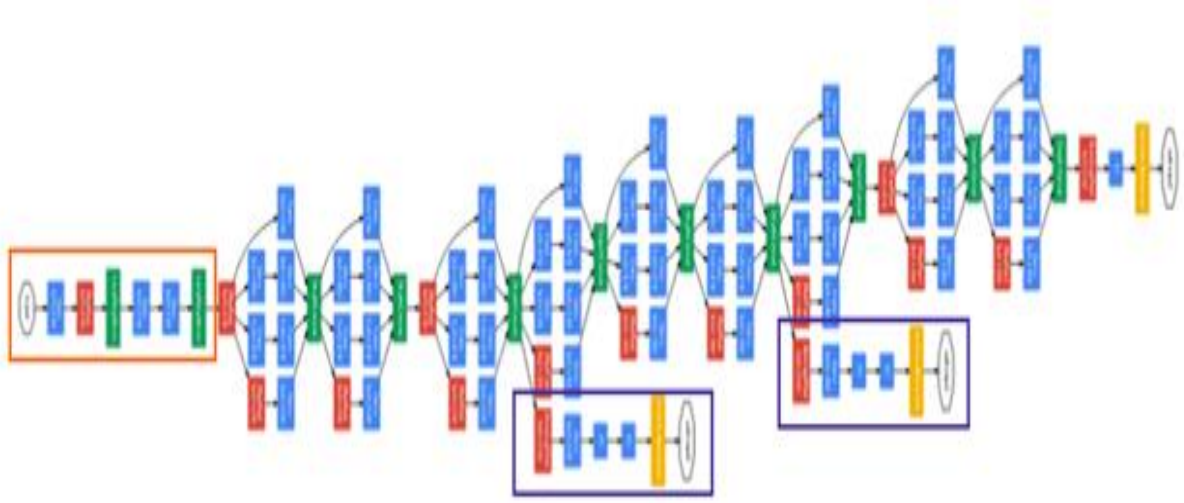


Figure 4.12 Inception-v1 Architecture [55]

Google-Net has nine different inception architectures. It is twenty-two different layers and 27 along with different pooling layers. It is very deep classifier. This architecture also

minimizes the gradient issues.

Inception–v2 [55] module consists of 5x5 convolutional layer into two 3x3 convolutional tasks to increase the computational speed. A 5x5 convolutional is approximately 2.78 time expensive than convolutional layer of 3x3. So, this process increases the performance of architecture and reduced the computations. This architecture is shown below in Figure 4.13.

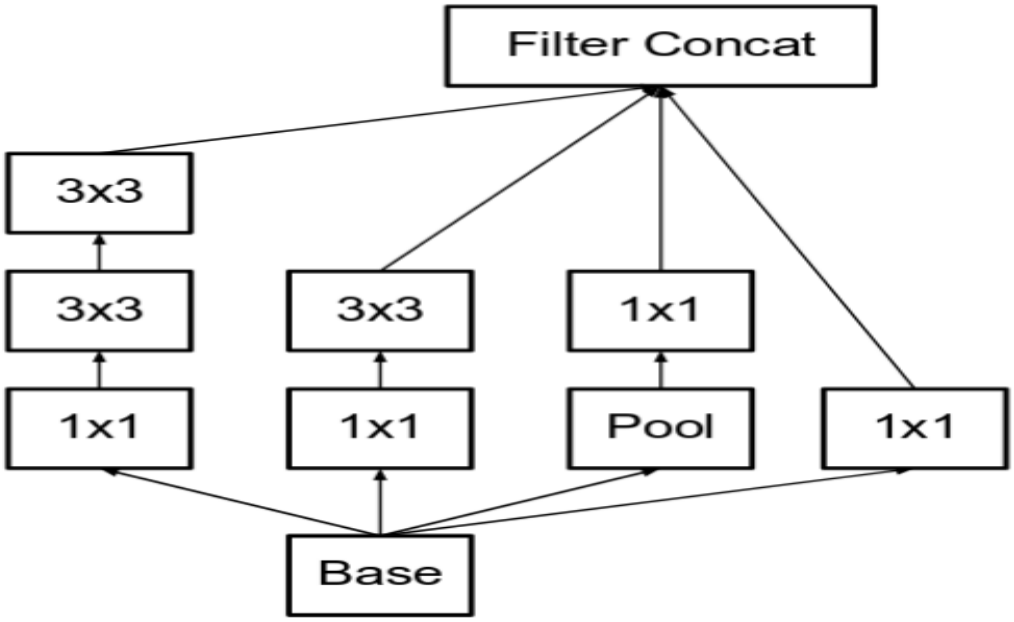


Figure 4.13 5x5 convolutional layer is illustrated as 3x3 convolutional layers [55]

Furthermore, convolutional of kernel size nxn is factorized into a sequence of nx1 and 1xn convolutions. A 3x3 convolutional layer is equal to performing filter of 1x3 convolutional and then filter of 3x1 convolutional and is shown below in Figure 4.14. And this method is cheapest then performing single 3x3 convolutional.

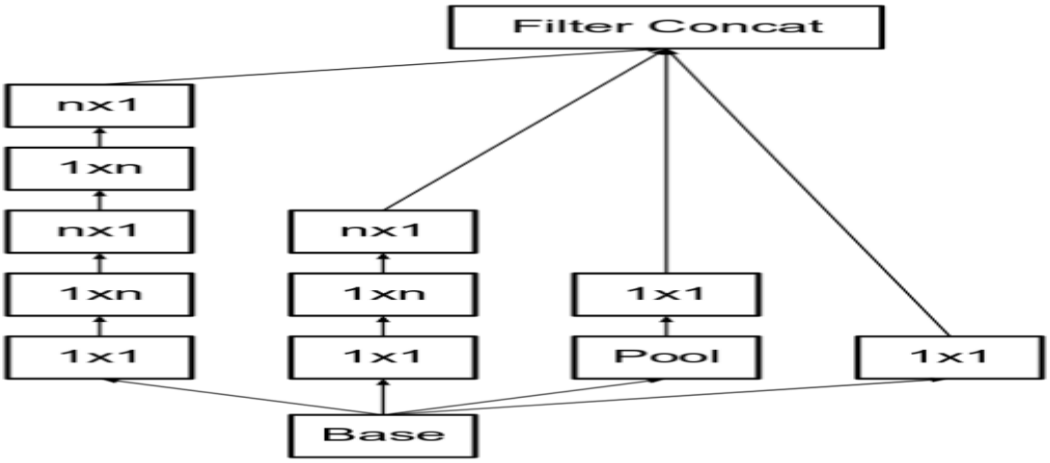


Figure 4.14. 5x5 convolutional layer is illustrated as 2 3x3 convolutional layers, which are described as 3x1 and 1x3.

The kernel channels in the architecture were enlarged to minimize the bottleneck and are shown below in Figure 4.15.

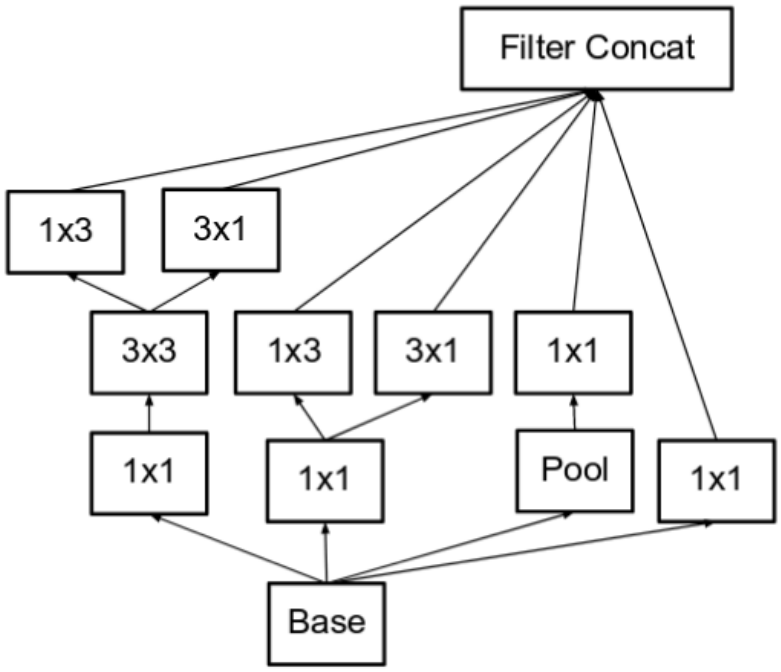


Figure 4.15 Wider Architecture of Inception-v2 [54]

.

The Inception-v3 architecture is progressively made, as described below.

- **Factorized convolutional layers**

This step helps to minimize the computations just as. It minimizes the parameters engaged in a deep network. It also checks the efficiency of network.

- **Small convolutions**

This step replaced the highest convolution layers along smallest convolutions and it leads to fastest training. Because a 5 × 5 kernel has twenty-five different training parameters, two same 3 × 3 kernels is replacing into 5 × 5convolutions and it has total 18 parameters.

- **Asymmetric convolutional layers**

In asymmetric convolutional layers a 3 × 3 convolutional layers can be recovered through 1 × 3 convolutional layer and followed as a 3 × 1 convolutional. Moreover, a 3 × 3 convolutional layers is replaced through a 2 × 2 convolutional [56]. Asymmetric convolution architecture is shown below in Figure 4.16.

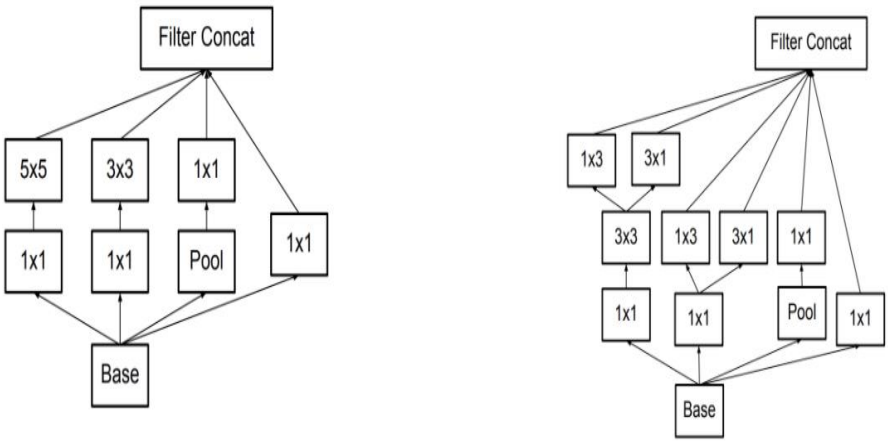


Figure 4.16 Asymmetric convolution architecture of Inception-v3 module [56]

- **Auxiliary classifiers:**

The auxiliary classifiers [57] are very small convolution neural network included between convolution layers throughout training process. Although, the loss sustained is also added to main convolutional network loss. In inception-v3 auxiliary classifiers behaves as a regularize [58]. Auxiliary classifier is shown below in Figure 4.17.

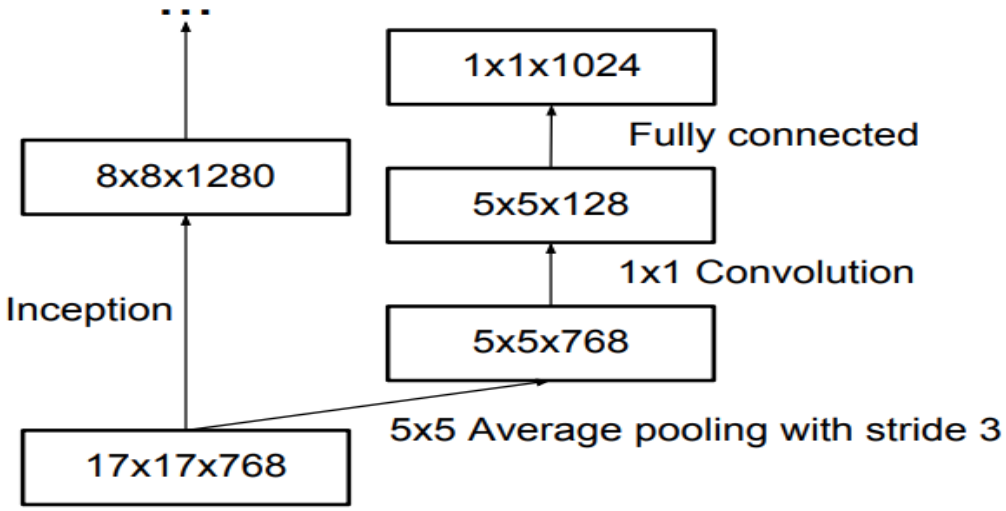


Figure 4.17. Auxiliary Classifier Architecture [57]

- Grid size minimization

Grid size minimization [59] is normally done through maximum/average pooling operations [60]. Moreover, to overcome the bottleneck of computations cost, more powerful architecture is shown below in Figure 4.18.

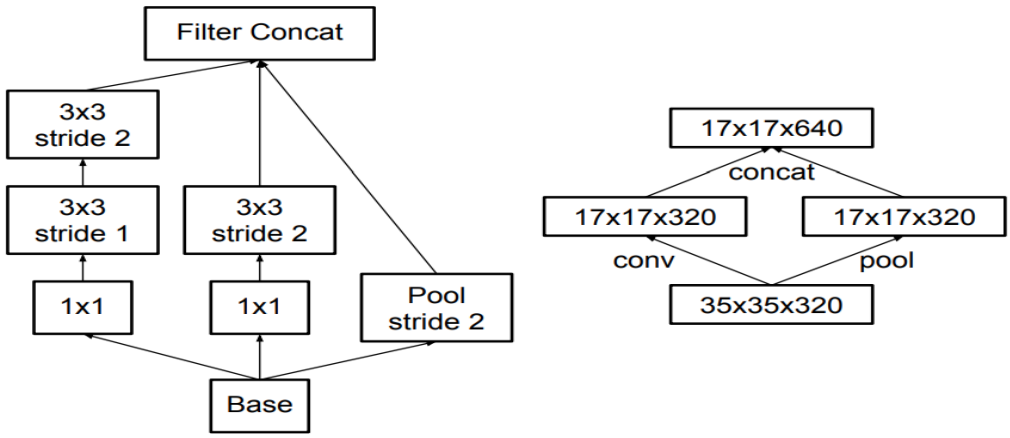


Figure 4.18 Grid size minimization architecture [59]

All the steps described earlier are combined into final architecture of inception-v3 architecture is shown below in Figure 4.19.

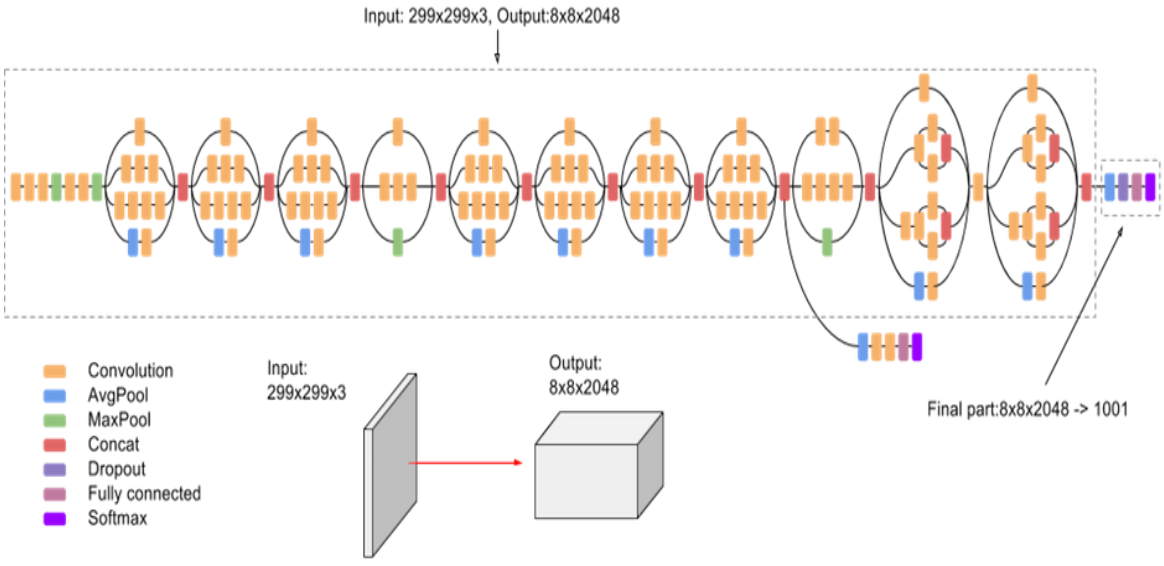


Figure 4.19 Inception-v3 architecture [60]

4.4.4 MobileNet-V2

MobileNetV2 [61] is a deep neural network architecture presented in 2018 by Andrew Howard. MobileNetV2 a preferable architecture is initiated along inverted residual framework. Non linearities in small layers are minimized through this architecture. MobileNetV2 just as a backbone for high level feature extractions, high performance is also achieved for segmentation and object detection operations. In MobileNetV2 architecture, there are different 2 types of chunks. One block of MobileNetV2 is residual chunk with stride just of one. The second block with stride just of two and it used for downsizing. Both types of blocks contain three layers. Moreover, first convolution layer is 1×1 with rectified linear unit (ReLU6). The second convolutional neural layer is a depth wised convolution layer [62]. And the third layer is 1×1 convolutional layer still without some non-linearity. Residual convolutional network block is shown in Table 4.1.

Table 4.1 Bottleneck residual convolutional neural network block[62]

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

In this table “t” is the expansion factor. Moreover, the value of t=6 for all main experiments. MobileNetV2 architecture consists of initially fully convolutional layers with 32 kernels, followed through different 19 residual convolutional layers. When input got sixty-four different channels, then the central output could get 384 different channels. MobileNetV2 is shown below in Table 4.2.

Table 4.2 MobileNetV2[62]

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Here “t” expresses the expansion factors, “c” denotes the different output channels, “n” represents the repeating numbers and “s” is the stride of 3×3 filters/kernels were used for spatial convolutional operations. Every line represents the sequence of one or many same layers, repeated “n” time. In primary network the computational cost is 300 million multiply add operations and it uses more than 3.4 million different parameters [62]. Mobile Net V2 architecture is shown in Figure 4.20.

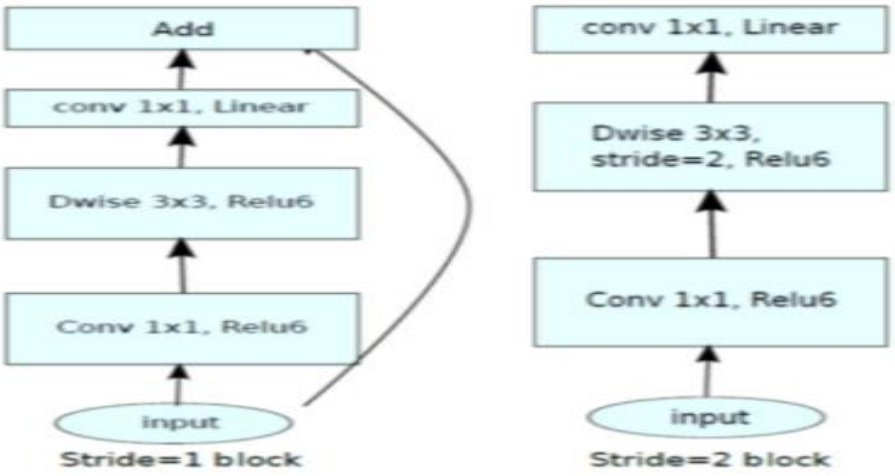


Figure 4.20. MobileNetV2 Architecture [62]

4.4.5 LeNet Architecture

LeNet [63] is a CNN architecture proposed by Mohammed Kayed and Ahmed Anter. This CNN model contains convolution layers, ReLU, pooling, fully connected hidden and output layer and is shown in Figure 4.21. This architecture has total 7 layers. All convolution layers operation is performed with 5 x 5 size kernels. All pooling layers operation is performed with 2 x 2 size kernels. In the end it has fully connected layers: the first fully connected hidden layer has 84 nodes, the second fully connected output layer contained 10 channels. The final output layer is the soft-max activation layer.

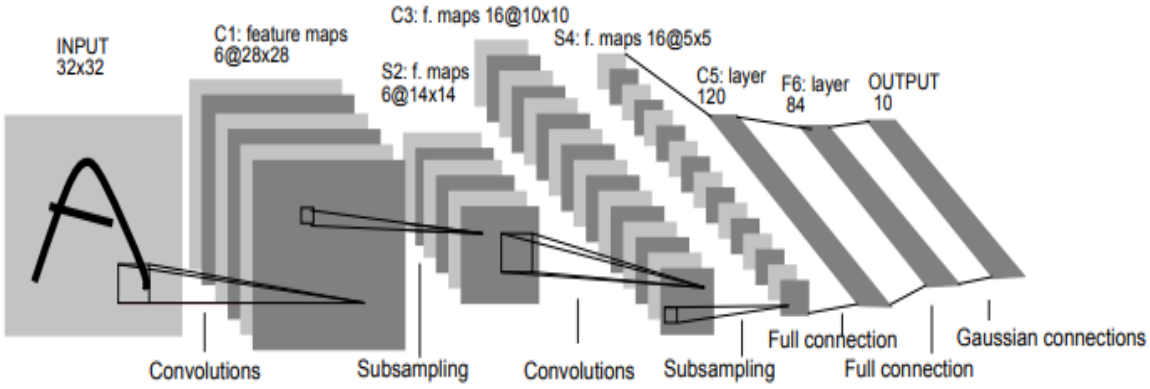


Figure 4.21 LeNet Architecture [63]

4.5 Proposed CNN Model

This section gives a thorough review of the proposed model and it is organized through the following chunks.

- Datasets
- Developed Model

4.5.1 White Blood Cell Datasets

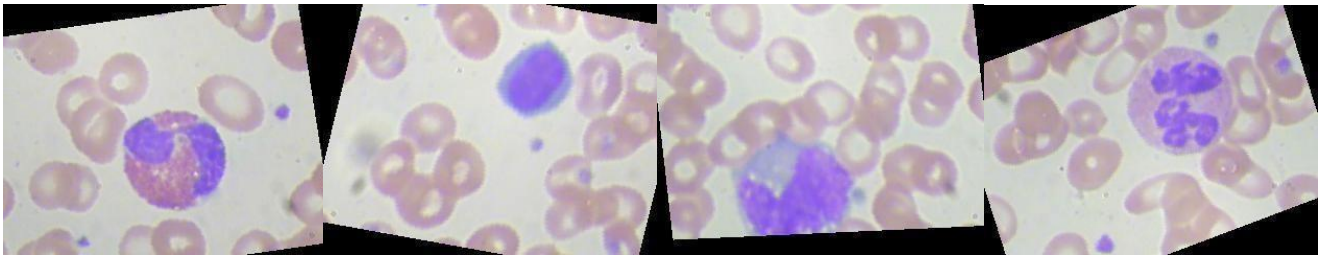
The datasets used for this research are obtained from Kaggle and LISC [64, 65].

4.5.1.1 Kaggle Dataset

The Kaggle dataset contains 12,500 JPEG images of size 320 x 240. Each class has approximately 3000 images. The dataset has four different categories of WBC's; Eosinophils, Lymphocytes, Monocytes and Neutrophils, and images of each class are shown in Figure 4.22.

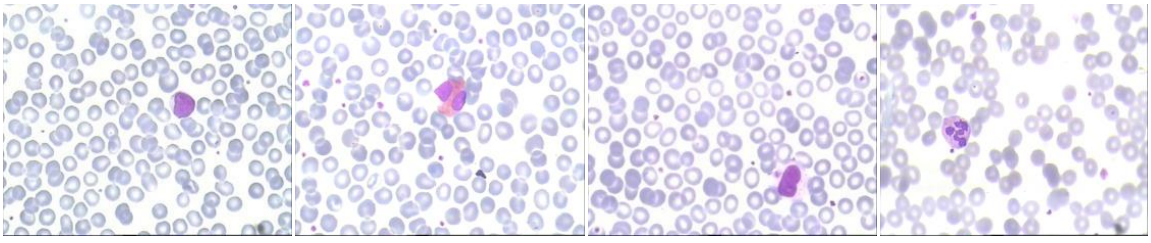
4.5.1.2 LISC Dataset

In LISC dataset, images have been taken from peripheral blood of 8 normal subjects and 400 samples have been obtained from 100 microscope slides. These samples have been saved in the BMP format. We have applied rotation augmentation method to these images with degree of rotation set to 90,180,270 degrees. After augmentation, the LISC dataset contains 10,000 images of 720 x576 sizes. In this dataset, there are 4 types of white blood cells namely Eosinophils, Lymphocytes, Monocytes and Neutrophils, sample images of which are shown in Figure 4.23.



(a)Eosinophil (b) Lymphocyte (c)Monocyte (d) Neutrophil

Fig.4.22. Sample images in Kaggle dataset



(a)Eosinophil (b) Lymphocyte (c)Monocyte (d) Neutrophil

Fig.4.23. Sample images in LISC dataset

4.5.2 Developed Model

This section illustrates the method undertaken to classify white blood cells. The suggested methodology is based on Convolutional neural network (CNN). CNN fall into the category of Artificial Neural Networks and is proven to be valuable in image classification. It takes a raw image as an input and generates a result signifying the likelihoods that input belongs to certain class. The proposed architecture is shown in Figure 4.24. It consists of three convolutional layers, three pooling layers, two fully connected hidden layers and an output layer.

4.5.2.1 Convolutional Layer

The key layer in CNN is convolutional layer, giving the network its name. This layer performs the operation called "convolution" of an input image to a filter also known as kernel. This filter extracts a features-map from an image. The important parameters for the layer include number of filters, filter size, input image size, number of channels in input and output image, padding, and activation function. As highlighted before, our architecture contains three convolutional layers listed below:

1) First Layer: Kernel size: 3 x 3, number of filters: 32, Activation function: Relu, Stride: 1 and input size: 100 x 100 (3 channels).

2) Second Layer: Kernel size: 3 x 3, number of filters: 64, Activation function: Relu, Stride: 2.

3) Third Layer: Kernal size: 3 x 3, number of filters: 64, Activation function: Relu, Stride: 1

4.5.2.2 Pooling Layer

Pooling is another layer of CNN. Its purpose is to gradually reduce the resolution and size of the input image. It helps to avoid overfitting and significantly decreases the computation cost of network. We used max pooling with same parameters for all three pooling layers in our architecture with details given:

Pooling type: Maximum, Pooling Size: 2 x 2, Stride: 1, Dropout 0.20

4.5.2.3 Fully Connected Layer

The last layers of CNN model are fully connected layers. It connects all inputs to every activated location. This layer drives the final output of classification problem. In our proposed methodology, we have two fully connected hidden layers and one fully connected output layer listed below:

1) Fully Connected Hidden Layer: Total nodes: 64, Activation: Relu

2) Fully Connected Output Layer: Total nodes: equal to number of classes, Activation: Softmax

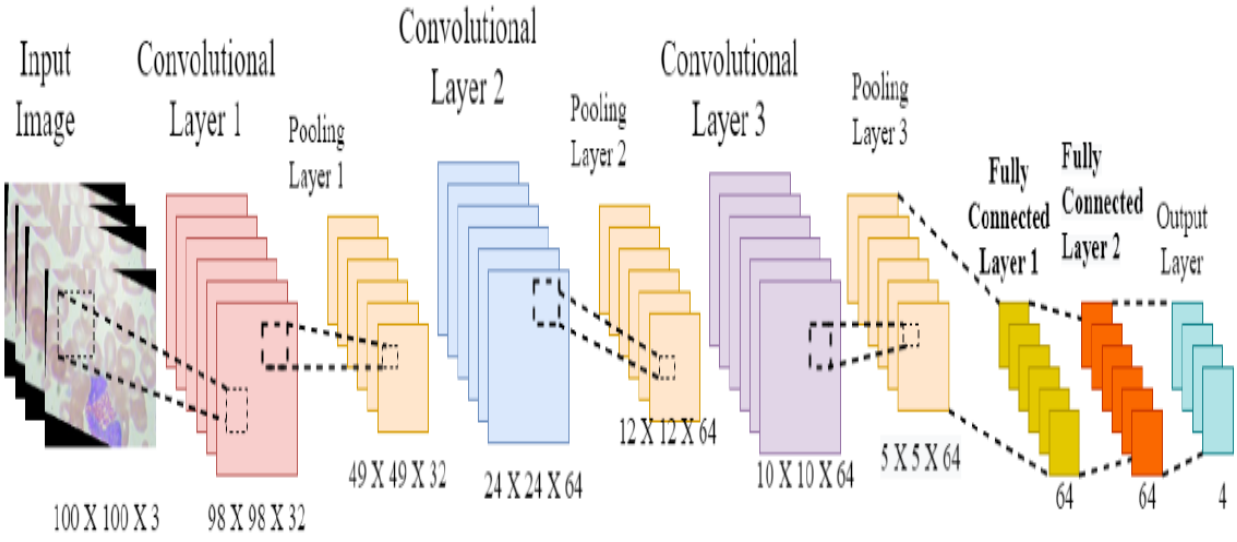


Figure 4.24 Architecture to our proposed CNN Model

CHAPTER 5: Experimental Setup and Results

In this chapter we have explained implementation details of the proposed model and results.

5.1 Experimental System

Before initiate of experimental system, small details of machine which is used to do experiment. We had experienced 7th generations machine that has 4 CPUs "Intel Core m3", the rate is 1.6 GHz, RAM is 8GB, graphic memory 4GB and the window 8 is installed. Python is used as main tool to computations and performs different experiments. Machines Description is shown in Table 5.1.

Table 5.1 Machine's Description

Generation	7th
Processor	Intel core m3
No. of Processor	4
Speed	1 GHz ~ 1.6 GHz
Memory	8 GB
Graphic Memory	4 GB
Operating System	Windows 08 (64-bit)
Tool	Python

After the experiment setup, we have approached the datasets. We have used the standard datasets to do the experiments. The Kaggle dataset contains 12500 images of size 320 x 240. The dataset contains four different categories of WBC's Eosinophils, Neutrophils, Monocytes and Lymphocytes. In LISC dataset, Images have been taken from peripheral

blood of 8 normal subjects and 400 samples have been obtained from 100 microscope slides. These samples have been saved in the BMP format. We applied rotation augmentation method to these images and degree of rotation is 90,180 and 270. After augmentation method the LISC dataset contains 10,000 images of 720 x 576 sizes.

For training our model, all images in both datasets are resized to 100 x 100. The standard Kaggle and LISC dataset descriptions are defined in Table 5.2 and Table 5.3.

Table 5.2 Standard Kaggle Dataset Descriptions

Format	JPEG
No. of Images	12500
Dimension	320 * 240
No. of Classes	4
No. of Images in each Class	3000

Table 5.3 Standard LISC Dataset Descriptions

Format	BMP
No. of Images	10,000
Dimension	720 *576
No. of Classes	4
No. of Images in each Class	2500

5.2 *Model Evaluation*

We used the method of model evaluation in which data is separated into three uneven parts.

1. Training Data: Subset of the original dataset used to build model.
2. Validation Data: This data subsection is used to verify the performance of the model developed in training phase. It delivers a test program for the fine-tuning of model's parameters and choosing the finest performing model.
3. Testing Data: Subset of the dataset that is unseen to the model. The best performing model carefully chosen is tested on this data to examine its expected future performance.

5.3 Evaluation Parameters Metrics

There are certain parameters used in machine learning to determine the performance of the model. This work has used four generally utilized parameters, accuracy, recall, precision, and F-measure to evaluate the model fitness. These parameters can be calculated using confusion matrix. Equations for these parameters and confusion matrix are defined in Table 5.4.

Table 5.4: Equations for the Evaluation parameters

Confusion Matrix

	Positive	**Negative**
Positive	TP: True Positive	FP: False Positive
Negative	FN: False Negative	TN: True Negative

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TP} \quad (5.1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5.2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5.3)$$

$$\text{F\_Measure} = \frac{2(\text{precision}.\ \text{recall})}{\text{Precision}+\ \text{recall}} \quad (5.4)$$

5.4 Results

Firstly, CNN architectures LeNet, VGG16, ResNet50, Inception V3 and MobileNetV2 are used. With these architectures, we have used loss function of sparse categorical cross entropy with Adam gradient-based optimizer. We have trained these models for 150 epochs. We have used learning rate of 0.0001 and dropout is 0.5. These CNN models are applied on Kaggle dataset. The input size of 4 pre-trained CNN models is 224x224. The input size of LeNet CNN model is 32x32.

- MobileNetV2
- VGG-16
- ResNet50
- Inception-V3
- LeNet
- Proposed CNN Model

5.4.1 MobileNetV2

Graph for model loss of MobileNetV2 is shown in Figure 5.1 and accuracy graph is shown in Figure 5.2. In our instance, the minimum value of cross entropy loss function is 0.0523 after 145 epochs. The maximum value of validation accuracy comes out to be 0.9253 and the maximum value of training accuracy is 0.9323.

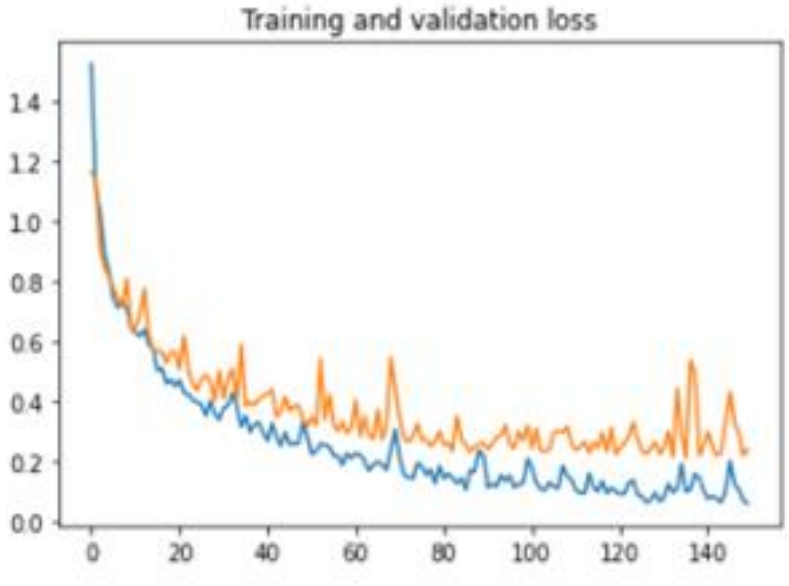


Figure 5.1 Graph representing model loss of MobileNetV2

Figure 5.2 Graph for accuracy against each epoch of MobileNetV2

In the WBC's classification using MobileNetV2, 29 images from eosinophils, 7 from lymphocytes, 11 images from monocytes and 17 images from neutrophils are misclassified. 93.6% accuracy is achieved. Results in the form of confusion matrix are presented in Table 5.5. The results for evaluation metrics are shown in Table 5.6.

Table 5.5 Confusion matrix of MobileNetV2

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	219	10	10	9
LYMPHOCYTE	4	241	2	1
MONOCYTE	5	4	237	2
NEUTROPHIL	10	4	3	231

Table 5.6 Results of MobileNetV2 on testing dataset.

Class	**Accuracy**	**Precision**	**Recall**	**F_Measure**
Eosinophils	0.890	0.89	0.92	0.90
Lymphocytes	0.971	0.97	0.97	0.97
Monocytes	0.952	0.95	0.98	0.97
Neutrophils	0.932	0.93	0.88	0.90
Average	0.936	0.935	0.937	0.935

5.4.2 VGG-16

Graph for model loss of VGG16 is shown in Figure 5.3 and accuracy graph is shown in Figure 5.4. In our instance, the minimum value of cross entropy loss function is 0.0598 after 148 epochs. The maximum value of validation accuracy comes out to be 0.9164 and the maximum value of training accuracy is 0.9204.

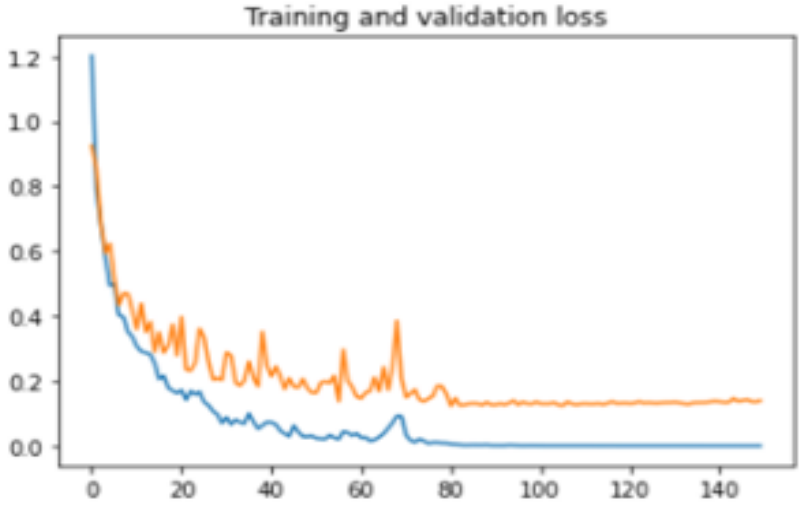


Figure 5.3 Graph representing model loss of VGG16

Figure 5.4 Graph for accuracy against each epoch of VGG16

With VGG-16, 15 images from eosinophils, 11 from lymphocytes, 14 images from monocytes and 40 from neutrophils, 80 images in total of 992 have been misclassified. Overall accuracy of 92% is achieved with VGG16. Results in the form of confusion matrix are presented in Table 5.7. The results for evaluation metrics are shown in Table 5.8.

Table 5.7 Confusion matrix of VGG16

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	233	10	4	1
LYMPHOCYTE	5	237	4	2
MONOCYTE	3	9	234	2
NEUTROPHIL	25	5	10	208

Table 5.8 Results of VGG16 on testing dataset.

Class	Accuracy	Precision	Recall	F_Measure
Eosinophils	0.930	0.93	0.82	0.87
Lymphocytes	0.961	0.96	0.97	0.96
Monocytes	0.95	0.95	0.98	0.96
Neutrophils	0.84	0.84	0.91	0.88
Average	0.920	0.920	0.920	0.917

5.4.3 ResNet50

Graph for model loss of ResNet50 is shown in Figure 5.5 and accuracy graph is shown in Figure 5.6. In our instance, the minimum value of cross entropy loss function is 0.0547 after 146 epochs. The maximum value of validation accuracy comes out to be 0.9192 and the maximum value of training accuracy is 0.9251.

Figure 5.5 Graph representing model loss of ResNet50

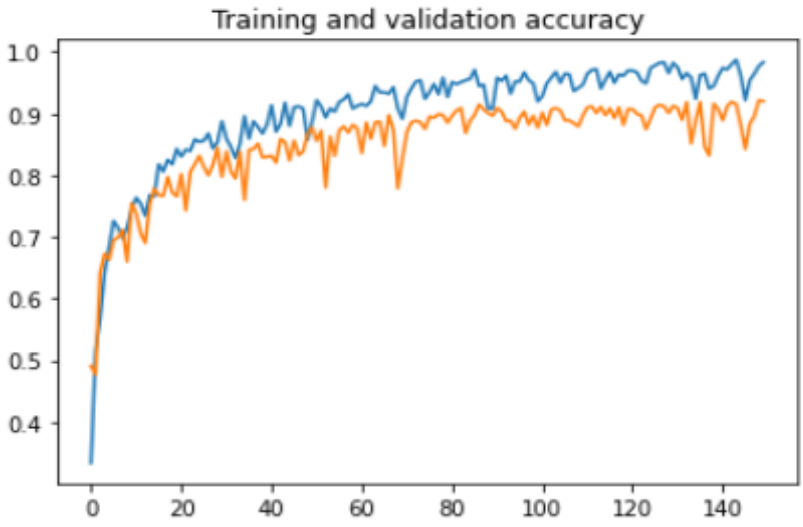


Figure 5.6 Graph for accuracy against each epoch of ResNet50

A total of 16 images from eosinophils, 17 from lymphocytes, 8 images from monocytes and 38 from neutrophils, 79 images are misclassified by using ResNet50. Overall accuracy of 92.3% is obtained. Results in the form of confusion matrix are presented in Table 5.9. The results for evaluation metrics are shown in Table 5.10.

Table 5.9 Confusion matrix of ResNet50

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	232	9	4	3
LYMPHOCYTE	8	231	5	4
MONOCYTE	4	0	240	4
NEUTROPHIL	20	10	8	210

Table 5.10 Results of ResNet50 on testing dataset

Class	**Accuracy**	**Precision**	**Recall**	**F_Measure**
Eosinophils	0.94	0.94	0.99	0.96
Lymphocytes	0.92	0.92	0.80	0.86
Monocytes	0.97	0.97	0.99	0.98
Neutrophils	0.86	0.86	0.90	0.88
Average	0.923	0.922	0.92	0.92

5.4. 4 Inception-V3

Graph for model loss of Inception V3 is shown in Figure 5.7 and accuracy graph is shown in Figure 5.8. In our instance, the minimum value of cross entropy loss function is 0.0478 after 147 epochs. The maximum value of validation accuracy comes out to be 0.9425 and the maximum value of training accuracy is 0.9581.

Figure 5.7 Graph representing model loss of InceptionV3

Figure 5.8 Graph for accuracy against each epoch of InceptionV3

Overall, 95.67% accuracy is achieved with InceptionV3 model, where comparatively better results are obtained. With this pre-trained architecture, 15 images from eosinophils, 4 from lymphocytes, 7 from monocytes and 18 images from neutrophils are classified incorrectly. Results in the form of confusion matrix are presented in Table 5.11. The results for evaluation metrics are shown in Table 5.12.

Table 5.11 Confusion matrix of InceptionV3

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	233	9	3	3
LYMPHOCYTE	2	244	1	1
MONOCYTE	4	2	241	1
NEUTROPHIL	10	7	1	230

Table 5.12 Results of Inceptionv3 on testing dataset

Class	Accuracy	Precision	Recall	F_Measure
Eosinophils	0.942	0.94	0.92	0.93
Lymphocytes	0.982	0.98	0.99	0.98
Monocytes	0.972	0.97	0.99	0.98
Neutrophils	0.932	0.93	0.93	0.93
Average	0.957	0.955	0.9575	0.955

5.4.5 LeNet Architecture

Graph for model loss of LeNet is shown in Figure 5.9 and accuracy graph is shown in Figure 5.10.

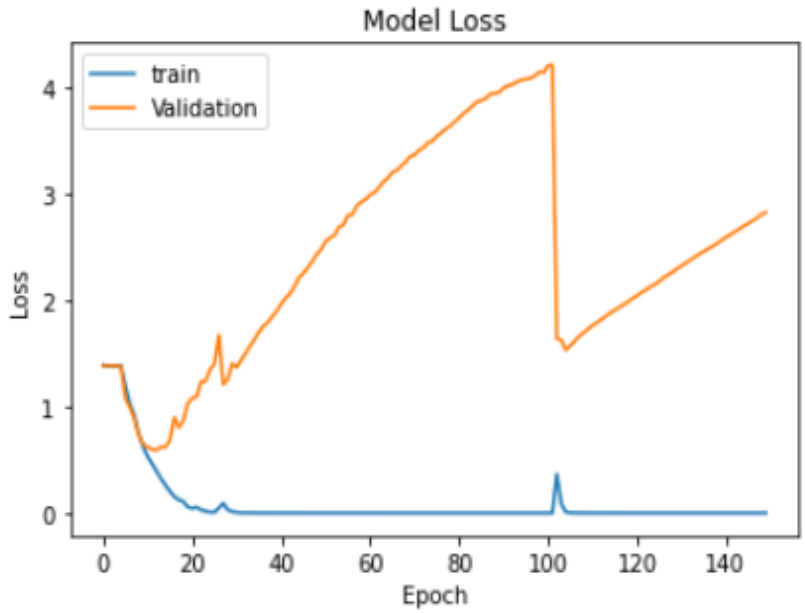


Figure 5.9 Graph representing model loss of LeNet

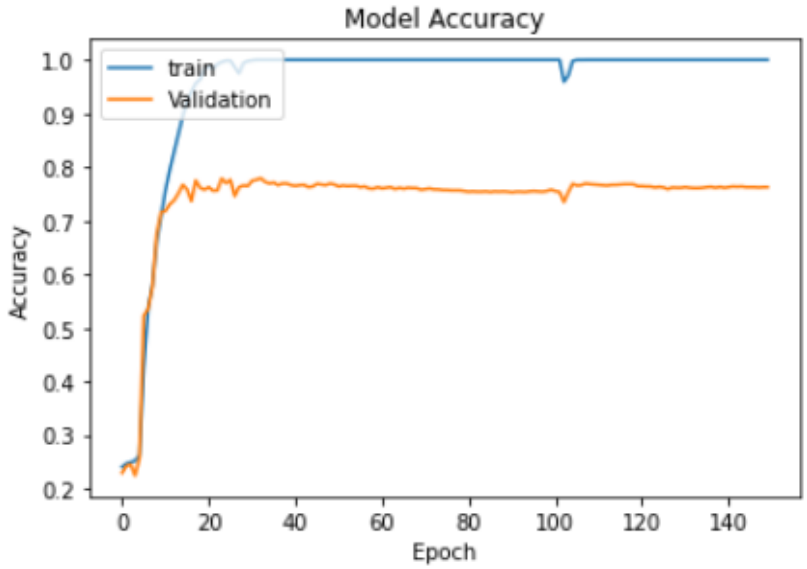


Figure 5.10 Graph for accuracy against each epoch of LeNet

In confusion matrix, ‘output class’ is a class which classifier predict and ‘target class’ is actual class which it belongs. As show in the table, we got maximum accuracy is 74.67%. Results in the form of confusion matrix are presented in Table 5.13. The results for evaluation metrics are shown in Table 5.14.

Table 5.13 Confusion matrix of LeNet

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	233	9	4	2
LYMPHOCYTE	10	211	15	12
MONOCYTE	9	11	225	3
NEUTROPHIL	5	10	15	218

Table 5.14 Results of LeNet on testing dataset.

Class	**Accuracy**	**Precision**	**Recall**	**F_Measure**
Eosinophils	0.77	0.77	0.78	0.77
Lymphocytes	0.70	0.70	0.76	0.73
Monocytes	0.76	0.76	0.79	0.77
Neutrophils	0.74	0.74	0.63	0.68
Average	0.74	0.74	0.74	0.737

The results of CNN architectures LeNet MobileNetV2, VGG16, ResNet50 and Inception V3 separately are reported for Kaggle dataset. Performance parameter values obtained with these CNN architectures are presented in Table 5.15.

Table 5.15 Performance parameter results of CNN models

	ACC %	**REC %**	**PREC %**	**F Measure %**
LeNet	74.67	74.2	74	73.7
MobileNetV2	93.6	93.7	93.5	93.5
VGG16	92	92	92	91.75
ResNet50	92.3	92	92.2	92
Inceptionv3	95.7	95.5	95.75	95.5

5.4.5 Proposed CNN Model Results

In quest for better performance and inspired by these architectures, we have proposed and trained our own CNN model. For training, three major parameters are decided, loss function, optimizer, and metrics of evaluation. Our CNN model uses loss function of sparse_categorical_crossentropy with default Adam gradient-based optimizer. The training dataset is fed to our model and is trained for 150 epochs, saving best weights depending on loss function. The proposed convolutional neural network model has been tested with WBC's images from both Kaggle and LISC datasets.

5.4.5.1 Results on Kaggle Dataset

To train a model, three major parameters are decided i.e., Loss function, optimizer, and metrics of evaluation. Our CNN model uses loss function of sparse_categorical_crossentropy with default Adam gradient-based optimizer. The training dataset is fed to our model and is trained for 150 epochs, saving best weights depending on loss function. Graph for model loss is shown in Figure 5.11 and accuracy achieved along each epoch is shown in Figure 5.12.

The performance of network is measured through cross entropy loss function, which is widely used to evaluate the performance of convolution neural networks. Cross entropy function value increases if predicated value and actual value are not same. In ideal case, cross entropy value is zero.

The cross entropy of the convolution neural network is shown in Figure 5.11 with respect to training and validation. In this case, for each epoch the network is trained, and then training and validation loss are calculated.

This process stops when it reaches at minimum level and the value remains constant for each increasing epoch. In our instance, the minimum value of cross entropy is 0.0276 after 145 epochs. It seems to be nearby zero. In our case maximum error comes out to be 0.0562, which is combined form of training and validation**.**

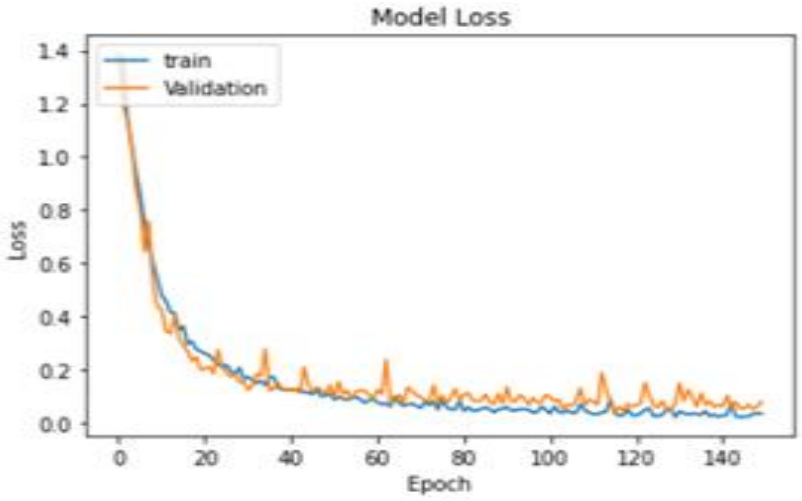


Figure 5.11 Graph representing model loss of proposed CNN model on Kaggle dataset

The training and validation accuracy of our convolution neural network is shown in Figure 5.12. In this case, for each epoch the network is trained, and then training and validation accuracy are calculated. This process stops when it reaches at maximum level and the value becomes constant on further increasing the epoch. In our case, the maximum value of training accuracy comes out to be 0.9905 after 145 epochs and the maximum value of validation accuracy is 0.9822.

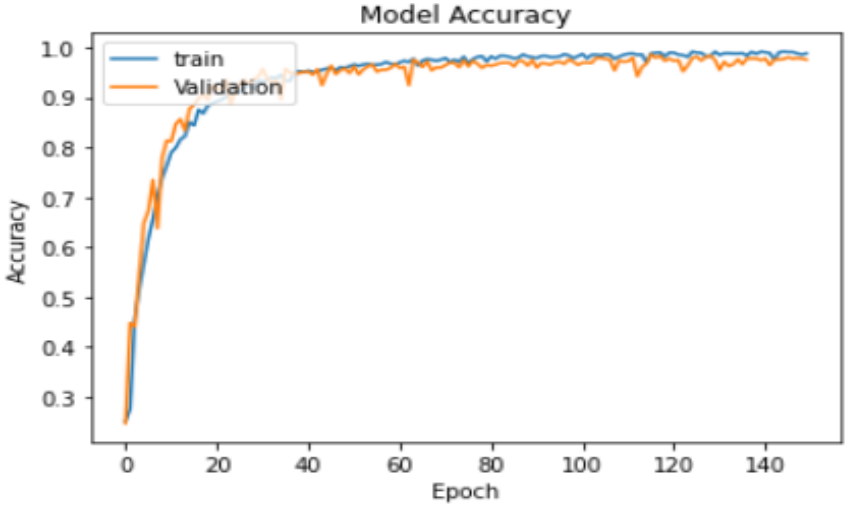


Figure 5.12 Graph for accuracy against each epoch of CNN model on Kaggle dataset .

The model weights for minimum loss are saved and labels for testing dataset are predicted. The results for evaluation metrics for Kaggle dataset are shown in Table 5.16.

From Table 5.16, it can be seen that for the Eosinophils, Neutrophils, Lymphocytes, Monocytes class, the precision is 99%, 99.3%, 100% and 100% respectively. The F-measure rates of the four classes separately obtained are 99.3% in Eosinophils, 98%, in Neutrophils, 100% in Lymphocytes and 100% in Monocytes. For the Eosinophils, Neutrophils, Lymphocytes, Monocytes class, the recall is 99.4%, 98.5%, 100% and 100% respectively. It mixed up the Eosinophils and Neutrophils because they are near identical and have same sizes and shapes. All images of Lymphocytes and Monocytes are 100% correctly classified. Average accuracy achieved for all classes is 99.57%.

Table 5.16 Results of CNN on testing dataset

Class	Accuracy	Precision	Recall	F_Measure
Eosinophils	0.990	0.990	0.994	0.993
Lymphocytes	100	100	100	100
Monocytes	100	100	100	100
Neutrophils	0.993	0.993	0.985	0.98
Average	0.9957	0.995	0.99	0.9932

Results in the form of confusion matrix are presented in Table 5.17. For Kaggle dataset, 2 from the eosinophils class and 1 image from the neutrophils class were classified incorrectly. It misclassifies the Eosinophils and Neutrophils because as explained earlier

the two type of cells are similar in shape and size. All images of other two classes are 100% correctly classified.

Table 5.17 Confusion matrix of Kaggle dataset

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	246	0	0	2
LYMPHOCYTE	0	248	0	0
MONOCYTE	0	0	248	0
NEUTROPHIL	1	0	0	247

5.4.5.2 Results on LISC Dataset

Graph for model loss of LISC dataset is shown in Figure 5.13 and accuracy graph is shown in Figure 5.14. In our instance, the minimum value of cross entropy loss function is 0.0354 after 147 epochs. The maximum value of validation accuracy comes out to be 0.9712 and the maximum value of training accuracy is 0.983.

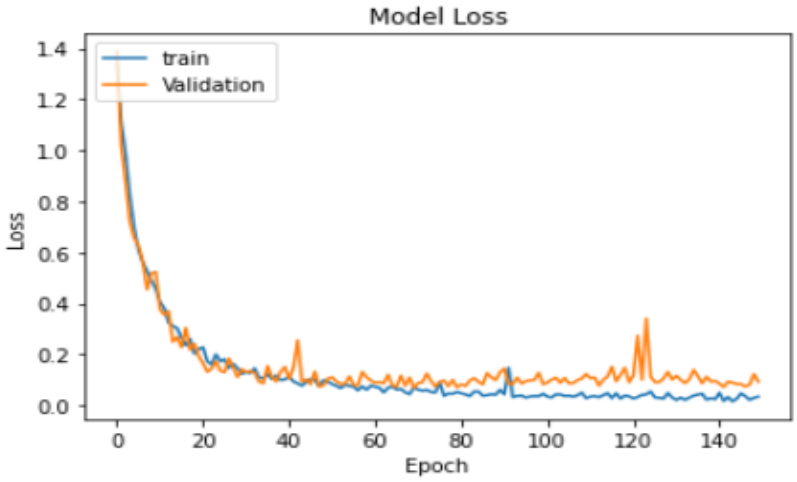


Figure 5.13 Graph representing model loss of proposed CNN model on LISC dataset

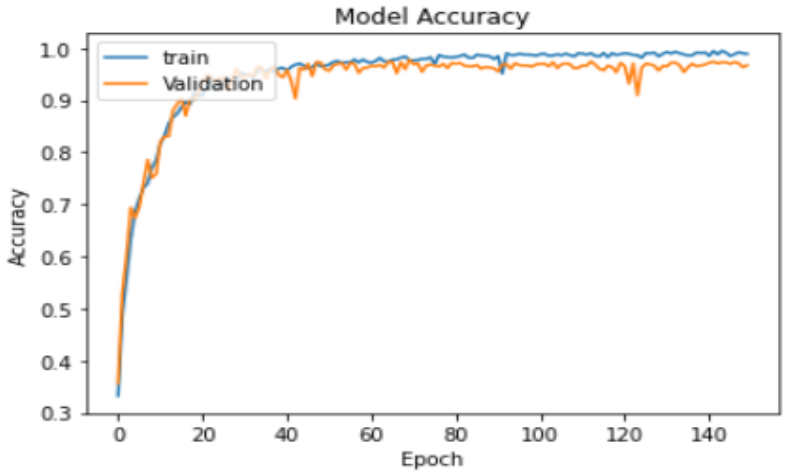


Figure 5.14 Graph for accuracy against each epoch of CNN model on LISC dataset

Confusion matrix is shown in Table 5.18 for LISC dataset. 2 images of eosinophils, 3 of lymphocytes and 1 of neutrophils, hence 6 images in total of 396 are misclassified. Average accuracy of 98.67% is achieved. Performance parameters of white blood cells classification obtained with proposed CNN model for LISC dataset is shown in Table 5.19.

Table 5.18 Confusion matrix of LISC Dataset

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	97	0	0	2
LYMPHOCYTE	1	96	0	2
MONOCYTE	0	0	99	0
NEUTROPHIL	1	0	0	98

Table 5.19 Results of CNN on testing dataset

Class	Accuracy	Precision	Recall	F_Measure
Eosinophils	0.985	0.985	0.998	0.98
Lymphocytes	0.972	0.972	0.952	0.977
Monocytes	100	100	100	100
Neutrophils	0.99	0.99	0.99	0.98
Average	0.986	0.98	0.98	0.984

5.4.6. Model Evaluation

We have trained and tested the Kaggle dataset using different dropout rate, optimizer and loss function. Model Evaluation results on Kaggle dataset is shown in Table 5.20.

- We achieved 99.29 percent accuracy by using dropout rate is '0.15', optimizer is 'SGD' and loss function is 'Sparse Categorical Cross entropy'.

- We achieved 98.68 percent accuracy by using dropout rate is '0.10', optimizer is 'SGD' and loss function is 'Sparse Categorical Cross entropy'.

- We achieved 98.64 percent accuracy by using dropout rate is '0.7', optimizer is 'RMSprop' and loss function is 'Sparse Categorical Cross entropy'.

- We achieved 98.58 percent accuracy by using dropout rate is ‘0.5’, optimizer is ‘RMSprop’ and loss function is ‘Sparse Categorical Cross entropy’.

Table 5.20 Model Evaluation Results on Kaggle dataset

Dropout rate	***Optimizer***	***Loss function***	***Epochs***	***Loss Rate (%)***	***Accuracy (%)***
0.15	SGD	Sparse Categorical Cross entropy	250	0.0341	99.29
0.10	SGD	Sparse Categorical Cross entropy	250	0.0507	98.68
0.7	RMSprop	Sparse Categorical Cross entropy	250	0.0664	98.64
0.5	RMSprop	Sparse Categorical Cross entropy	250	0.0567	98.58

5.4.7 Transfer Learning (Kaggle, LISC dataset)

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. In short, transfer learning is a machine learning method where model trained on one dataset is reused on a second related dataset. We trained complete LISC dataset and then tested the entire Kaggle dataset. After that trained complete Kaggle dataset and then tested the entire LISC dataset.

- Trained on complete LISC dataset
- Epoch’s size: 150
- Tested the entire Kaggle dataset
    - Achieved 97.87% maximum accuracy.

- Trained on complete Kaggle dataset
- Epoch's Size :150
- Tested the entire LISC dataset
    - Achieved 95.28% maximum accuracy.

5.4.7.1 Modification in proposed CNN architecture

To improve the accuracy the architecture of our proposed model used for transfer learning contains following layers:

- Four convolutional layers
- Four pooling layers
- Three fully-connected hidden layers
- One fully-connected output layer

The CNN architecture contains following layers:

1. **Four Convolution Layers**
    - **First layer:** Kernel size: 3x3, number of output filters:32, Activation: ReLu, Stride:1, input size: 200x200x3
    - **Second layer:** Kernel size:3x3, number of output filter:64, Activation: ReLu, strides= (2,2)
    - **Third layer:** Kernel size:3x3, number of output filter:64, Activation: ReLu, Stride=1
    - **Four layer:** Kernel size:5x5, number of output filter:128, Activation: ReLu, Strides= (2,2)
2. **Four Pooling Layers**

    All four pooling layers contains same parameters

    - Pooling type: Maximize, Pool size: 2x2 Stride :1 and Dropout = 0.20
    -
3. **Fully-connected Layer: Three Hidden Layers**

Three layers contain same parameters

- Total nodes: 128, Activation: Relu

4. **Fully-connected Layer: Output Layer**
    - Total nodes: number of classes, Activation: Softmax

5.4.7.2 Transfer Learning Results with Modified CNN

- Kaggle and LISC
- Trained on complete LISC dataset
- Epoch's Size: 200
- Tested the entire Kaggle dataset
    - Achieved 98.86% maximum accuracy.
- Trained on complete Kaggle dataset
- Epoch's Size: 200
- Tested on entire LISC dataset
    - Achieved 97.32% maximum accuracy.

5.4.7.3 Transfer Learning Results with Modified CNN (Kaggle, LISC)

Figure 5.15 and 5.16 representing model loss and accuracy graphs of LISC dataset. Figure 5.17 and 5.18 representing model loss and accuracy graphs of Kaggle dataset.

Trained complete LISC dataset and then tested the entire Kaggle dataset. After that trained complete Kaggle dataset and then tested the entire LISC dataset.

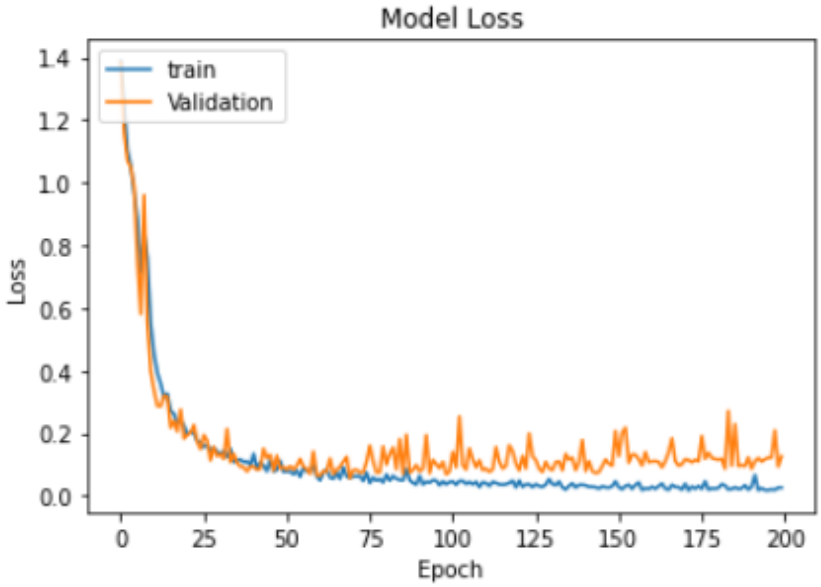


Figure 5.15 Graph representing model loss of LISC dataset

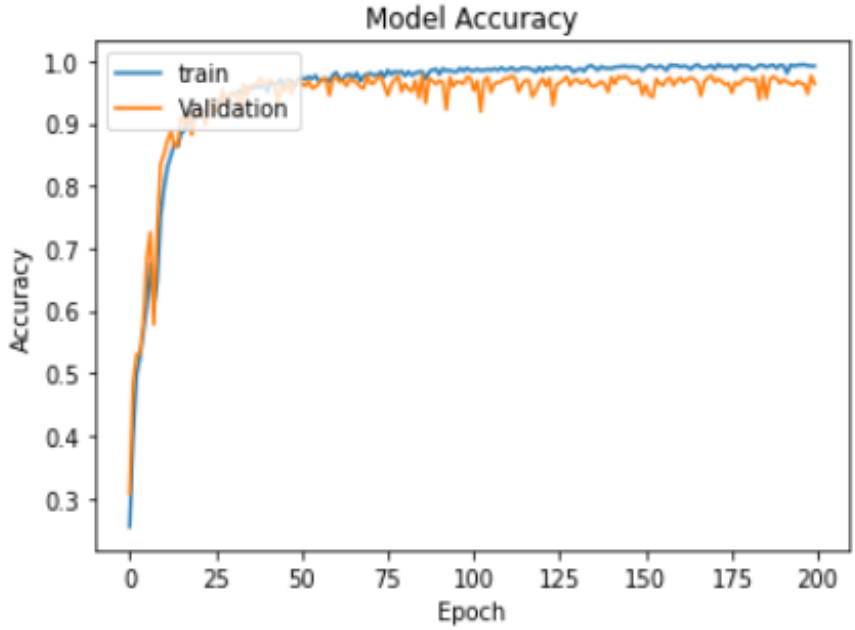


Figure 5.16 Graph for accuracy against each epoch of LISC dataset

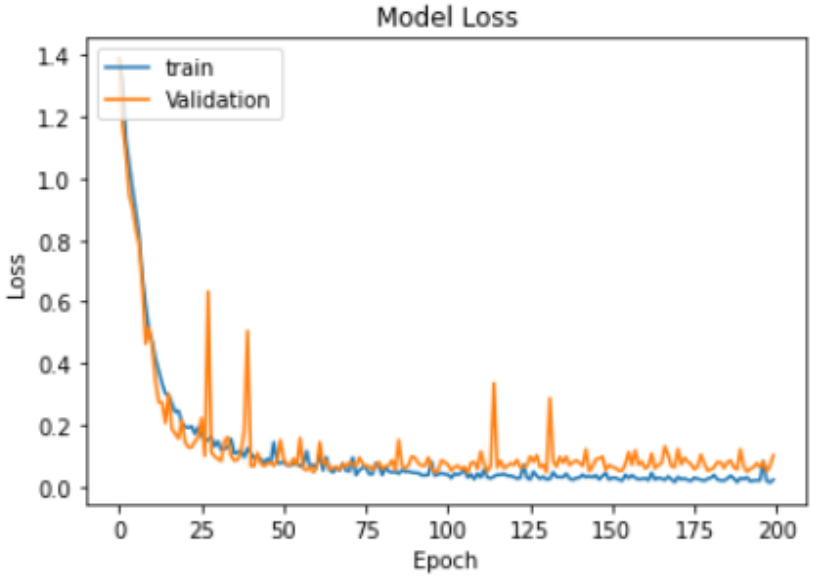


Figure 5.17 Graph representing model loss of Kaggle dataset

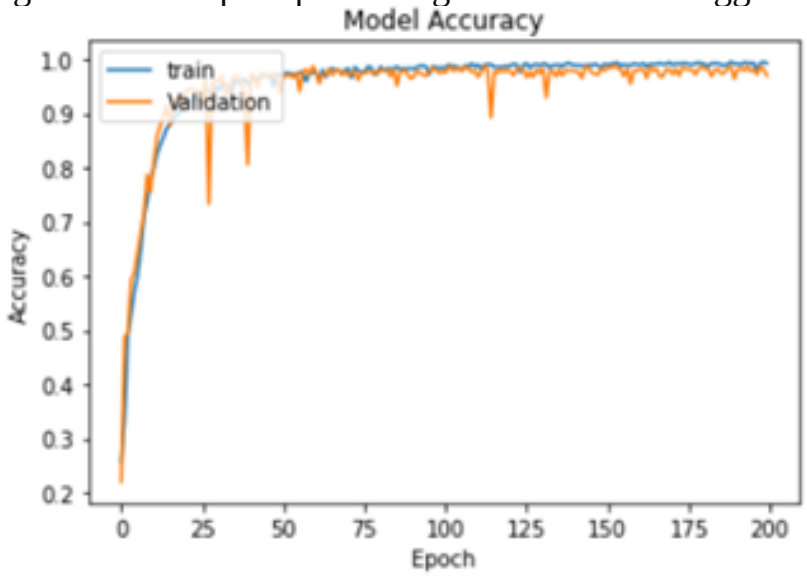


Figure 5.18. Graph for accuracy against each epoch of Kaggle dataset

For Kaggle dataset, 5 images from the eosinophils, 1 from lymphocytes, 4 from monocytes and 2 from neutrophils, 12 images in total of 12500 were misclassified.

For LISC dataset 5 images from the eosinophils, 4 from the lymphocytes, 3 from monocytes and 6 from neutrophils were misclassifies. Performance parameters of white blood cells classification obtained with modified CNN model for both datasets is shown in Table 5.21. Confusion Matrix of Kaggle and LISC datasets are shown in Table 5.22 and Table 5.23.

Table 5.21. Test results of transfer learning (Kaggle, LISC)

Type	Truth	Classified	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Kaggle Dataset						
E	3120	3125	0.98	0.98	0.97	0.97
L	3124	3125	0.99	0.99	0.98	0.98
M	3121	3125	0.98	0.98	0.99	0.98
N	3123	3125	0.99	0.99	0.98	0.99
LISC Dataset						
E	2495	2500	0.97	0.97	0.95	0.96
L	2496	2500	0.98	0.98	0.99	0.98
M	2497	2500	0.99	0.99	0.98	0.99
N	2494	2500	0.97	0.97	0.97	0.97

Table 5.22. Confusion Matrix of Kaggle dataset

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	3120	0	0	5
LYMPHOCYTE	0	3124	0	1
MONOCYTE	0	2	3121	2
NEUTROPHIL	2	0	0	3123

Table 5.23. Confusion Matrix of LISC dataset

	Predicted Class			
Actual Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	2495	2	0	3
LYMPHOCYTE	0	2496	4	0
MONOCYTE	0	0	2497	3
NEUTROPHIL	2	4	0	2494

5.5 Comparison

We have also compared the results of our proposed method with other related works from literature. The comparison is given in Table 5.24. To our knowledge, results achieved in literature using Artificial Neural Network (ANN) by Fan et al. [11] are 95.90%. Accuracy of 97.95% has been achieved by Ergen et. al. [53] using CNN. Using Pre-trained Deep Learning Models, Mohamed et. al. [56] reported an accuracy of 97.03% and96.63% has been achieved by Macawile et al. [3] using AlexNet network. Our results on the two datasets are competitive as compared to results reported by Cinar et al [28] as they have also employed the same two datasets. Using Alexnet-GoogleNet-SVM, Cinar et. al. [28] achieved an accuracy of 99.73% for Kaggle dataset and 98.23% for LISC dataset. Our Convolutional Neural Network model has performed very well for the classification of white blood cells giving an accuracy of 99.57% for Kaggle dataset and 98.67% for LISC dataset. This shows that our proposed CNN model is effective in correctly classifying the 4 types of WBCs and competes with results reported in literature on the two datasets.

Table 5.24 Comparison with Related Work

Author	***Year***	***Dataset***	***No. of Images used Training/Testing***	***Method***	***Accuracy***
S.S.Savae [17]	2015	Local Dataset	70	Water Shed	88.77%
Rosyadi [9]	2016	Local Dataset	N/A	K-Mean	67%
Gautam [25]	2016	Local Dataset	20 / 68	Naïve Bayes Classifier	80.88%
Wei Yu [23]	2017	Drump tower hospital of Nanjig university	N/A	CNN	88.50%

Author	***Year***	***Dataset***	***No. of Images used Training/Testing***	***Method***	***Accuracy***
Mazin Z.Othmn [15]	2017	Local Dataset	50	MLP-BP	96%
Ashish Khanna [20]	2018	Local Dataset	13000	CNN	88%
Machawie [11]	2018	ALL-IDB dataset	N/A	AlexNet	96.63%
O.liang [19]	2018	Local Dataset	12000	CNN + RRN	91%
H.Fan [21]	2019	jiangxi tecom science cooperation, BCISC,LISC	90 (54/18)	ANN	95.90%
M. Sharma [37]	2019	BCCD	9957/2487	CNN	87%
M. Togacar [38]	2019	Local Dataset	N/A	CNN	97.78%
E.H. Mohamed [40]	2020	BCCD	2500/620	Pre-trained deep learning models	97.03%
B. Ergen[42]	2020	Local Dataset	N/A	CNN & feature selection	97.95%
C. Zhao [43]	2021	Local Dataset	N/A	CNN	96%
A.Cinar [36]	2021	Kaggle, LISC	N/A	Alexnet-GoogleNet-SVM	99.73%, 98.23%
Proposed Method	**2021**	**Kaggle Dataset, LISC Dataset**	**12500, 10,000**	**CNN**	**99.57%, 98.67%**

Chapter 6: Conclusion

6.1 Conclusion

In this research a novel approach is used to classify the sub types of white blood cells (WBCs) i.e. monocytes, lymphocytes, eosinophils and neutrophils using machine learning. This approach uses convolutional neural network for classification of four types of white blood cells. We have first applied different CNN models, InceptionV3, VGG16, MobileNetV2, LeNet and ResNet50. These CNN models are applied on Kaggle dataset of microscopic images. The achieved accuracy through Inception V3 is 95.7 %, VGG16 is 92%, MobileNetV2 is 93.6%, LeNet is 74.67% and ResNet50 is 92.3%. Although we achieved reasonable accuracy ranging between 92 to 95%. Motivated by these architectures and a quest for better performance, a CNN based model has been proposed to classify the sub types of white blood cells (WBCs). The proposed architecture consists of three convolutional layers, three pooling layers, two fully connected hidden layers and output layer. The classification is performed using microscopic blood cell images obtained from Kaggle and LISC dataset. During testing, the proposed algorithm has shown optimal performance in terms of classification with 99.57% for Kaggle dataset and 98.67% accuracy for LISC dataset.

It also observed that proposed method is fully automatic and edge over on some previous methods which are more complex and high resource required to perform computation to produce a certain level of accuracy. But this method is simpler than those and given more optimum accuracy.

6.2 Future Work

In future, the proposed model can be applied to the classification of other cells and tissues of body that can help the pathologists in effective diagnosis. The same convolutional neural network model with few modifications can be tested for some other blood cells such that platelets, red blood cells and tissues.

REFERENCES

1. N. M. Salem, “Segmentation of white blood cells from microscopic images using K-means clustering,” in *2014 31st National Radio Science Conference (NRSC)*, Cairo, Egypt, Apr. 2014, pp. 371–376. doi: 10.1109/NRSC.2014.6835098.

2. M. Z. Alom, C. Yakopcic, T. M. Taha, and V. K. Asari, “Microscopic Blood Cell Classification Using Inception Recurrent Residual Convolutional Neural Networks,” in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, Dayton, OH, Jul. 2018, pp. 222–227. doi: 10.1109/NAECON.2018.8556737.

3. P. S. Hiremath, P. Bannigidad, and S. Geeta, “Automated Identification and Classification of White Blood Cells (Leukocytes) in Digital Microscopic Images,” *in Recent Trends in Image Processing and Pattern Recognition (RTIPPR)*, 2010, pp. 59-63.

4. R. Ahasan, A. U. Ratul, and A. S. M. Bakibillah, “White Blood Cells Nucleus Segmentation from Microscopic Images of strained peripheral blood film during Leukemia and Normal Condition,” in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, Dhaka, Bangladesh, May. 2016, pp. 361-366. doi: 10.1109/ICIEV.2016.7760026.

5. Puttamadegowda J. and Prasannakumar S. C., “White Blood cell sementation using Fuzzy C means and snake,” in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Bengaluru, India, Oct. 2016, pp. 47–52. doi: 10.1109/CSITSS.2016.7779438.

6. J. Laosai and K. Chamnongthai, “Acute leukemia classification by using SVM and K-Means clustering,” in *2014 International Electrical Engineering Congress (iEECON)*, Chonburi, Thailand, Mar. 2014, pp. 1–4. doi: 10.1109/iEECON.2014.6925840.

7. S. M. Mazalan, N. H. Mahmood, and M. A. A. Razak, “Automated Red Blood Cells Counting in Peripheral Blood Smear Image Using Circular Hough Transform,” in *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*, Kota Kinabalu, Dec. 2013, pp. 320–324. doi: 10.1109/AIMS.2013.59.

8. S. Manik, L. M. Saini, and N. Vadera, “Counting and classification of white blood cell using Artificial Neural Network (ANN),” in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, Jul. 2016, pp. 1–5. doi: 10.1109/ICPEICES.2016.7853644.

9. T. Rosyadi, A. Arif, B. Achmad and others, "Classification of leukocyte images using k-means clustering based on geometry features," in *2016 6th International Annual Engineering Seminar (InAES)*, Yogyakarta, Indonesia ,pp. 245–249. doi: 10.1109/INAES.2016.7821942.

10. H. Kutlu, E. Avci, and F. Özyurt, “White blood cells detection and classification based on regional convolutional neural networks,” *Med. Hypotheses*, 2019. doi: 10.1016/j.mehy.2019.109472

11. M. J. Macawile, V. V. Quinones, A. Ballado, J. D. Cruz, and M. V. Caya, “White blood cell classification and counting using convolutional neural network,” in *2018 3rd International Conference on Control and Robotics Engineering (ICCRE)*, Nagoya, Apr. pp. 259–263. doi: 10.1109/ICCRE.2018.8376476.

12. D.-C. Huang and K.-D. Hung, “Leukocyte nucleus segmentation and recognition in color blood-smear images,” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Graz, Austria, pp. 171–176. doi: 10.1109/I2MTC.2012.6229443.

13. P. Yampri, C. Pintavirooj, S. Daochai, and S. Teartulakarn “White blood nucleus extraction using K-Mean clustering and mathematical morphing,” 2014. doi: 10.1109/CONFLUENCE.2014.6949220.

14. A. Gautam, P. Singh, B. Raman, and H. Bhadauria, “Automatic classification of leukocytes using morphological features and Naïve Bayes classifier,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 1023–1027, 2017. doi: 10.1109/TENCON.2017.7848161.

15. M. Z., T. S., and A. B., “Neural Network Classification of White Blood Cell using Microscopic Images,” Int. J. Adv. Comput. Sci. Appl, vol. 8, no. 5, 2017, pp. 99-104. doi: 10.14569/IJACSA.2017.080513.

16. Yu, Wei, J. Chang, Cheng Yang, L. Zhang, Han Shen, “Automatic classification of leukocytes using deep neural network,” in 2017 *IEEE 12th International*

Conference on ASIC. doi: 10.1109/ASICON.2017.8252657.

17. S. Savkare, and S. P. Narote, “Blood Cell Segmentation from Microscopic Blood Images,” in *2015 International conference on Information Processing (ICIP)*, Puna, India, pp. 502-505. doi: 10.1109/INFOP.2015.7489435.

18. Y. X. and J. S. “Automatic classification of leukocytes using deep neural network, 2017, 12th I. C. on A. (ASICON): 1041-1044, pp. 259–263. doi: 10.1109/ICCRE.2017.8376476.

19. G. Liang, H. Hong, W. Xie, and L. Zheng, “Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification,” *IEEE Access*, 2018. doi: 10.1109/ACCESS.2018.2846685.

20. P. Tiwari, J. Qian, Q. Li, B. Wang, D. Gupta, A. Khanna, and J.J.P.C. Rodrigues, “Detection of subtype blood cells using deep learning,” *Cogn. Syst. Res.*, vol. 52, pp. 1036–1044, 2018. doi: 10.1016/j.cogsys.2018.08.022.

21. H. Fan, F. Zhang, L. Xi, Z. Li, G. Liu and Y. Xu, "LeukocyteMask: An automated localization and segmentation method for leukocyte in blood smear images using deep neural networks," *J. Biophotonics*, vol. 12, no. 7, Jul. 2019. doi: 10.1002/jbio.201900488.

22. M. Z. Alom, C. Yakopcic, T. M. Taha, and V. K. Asari, “Microscopic Blood Cell Classification Using Inception Recurrent Residual Convolutional Neural Networks,” in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, Dayton, OH, pp. 222–227. doi: 10.1109/NAECON.2018.8556737.

23. W. Yu, J. Chang, C. Yang, L. Zhang, H. Shen, Y. Xia and J. Sha, "Automatic classification of leukocytes using deep neural network," in *2017 IEEE 12th International Conference on ASIC (ASICON)*,pp.1041–1044, doi: 10.1109/ASICON.2017.8252657.

24. R. Ahasan, A. U. Ratul, and A. S. M. Bakibillah, “White Blood Cells Nucleus Segmentation from Microscopic Images of strained peripheral blood film during Leukemia and Normal Condition,” in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, Dhaka, Bangladesh, pp. 361-366. doi: 10.1109/ICIEV.2016.7760026.

25. Puttamadegowda J. and Prasannakumar S. C., “White Blood cell sementation using Fuzzy C means and snake,” in *2016 International Conference on*

Computation System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, pp. 47–52.doi: 10.1109/CSITSS.2016.7779438.

26. O. Ryabchykov, A. Ramoji, T. Bocklitz, M. Foerster, S. Hagel, C. Kroegel, M. Bauer, U. Neugebauer and J. Popp, "Leukocyte subtypes classification by means of image processing," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, doi: 10.1106TENCON.2016.787861.

27. S. Manik, L. M. Saini, and N. Vadera, "Counting and classification of white blood cell using Artificial Neural Network (ANN)," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, pp. 1–5. doi: 10.1109/ICPEICES.2016.7853644.

28. J. Lou, M. Zhou, Q. Li, C. Yuan and H. Liu, "An automatic red blood cell counting method based on spectral images," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. doi. 250-256, 2016.

29. N. M. Salem, "Segmentation of white blood cells from microscopic images using K-means clustering," in *2014 31st National Radio Science Conference (NRSC)*, Cairo, Egypt, pp. 371–37.doi: 10.1109/NRSC.2014.6835098.

30. J. Laosai and K. Chamnongthai, "Acute leukemia classification by using SVM and K-Means clustering," in *2014 International Electrical Engineering Congress (iEECON)*,Chonburi, Thailand, pp.1–4 .doi: 10.1109/iEECON.2014.6925840.

31. S. M. Mazalan, N. H. Mahmood, and M. A. A. Razak, "Automated Red Blood Cells Counting in Peripheral Blood Smear Image Using Circular Hough Transform," in *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*, Kota Kinabalu, pp. 320–324. doi: 10.1109/AIMS.2013.59.

32. L. B. Dorini, R. Minetto and N. J. Leite, "Semiautomatic white blood cell segmentation based on multiscale analysis," in *IEEE journal of biomedical and health informatics, 2012*, vol. 17, pp. 250-256, 2012.

33. S. Mohapatra, D. Patra and K. Kumar, "Blood microscopic image segmentation using rough sets," in *2011 International Conference on Image Information Processing*, pp. 1–4. doi: 10.1109/iEECON.2011.6925840.

34. S. H. Rezatofighi and H. Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," *Computerized Medical Imaging and Graphics*, vol. 35, pp. 333-343, 2011.

35. P. R. Tabrizi, S. H. Rezatofighi, and M. J. Yazdanpanah, "Using PCA and LVQ neural network for automatic recognition of five types of white blood cells," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, Buenos Aires, pp. 5593–5596. doi: 10.1109/IEMBS.2010.5626788.

36. Çınar and S. A. Tuncer, "Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid Alexnet-GoogleNet-SVM,", *SN Appl. Sci.*, vol. 3, no. 4, pp. 503-508, Apr. 2021, doi: 10.1007/s42452-021-04485-9.

37. M. Sharma, A. Bhave, and R. R. Janghel, "White Blood Cells Classification Using Convolutional Neural Network," in *Soft Computing and Signa Processing, vol.98, 2019.doi:* 2019.04.032.

38. M. Toğaçar, B. Ergen, and Z. Cömert, "Classification of white blood cells using deep features obtained from Convolutional Neural Network models," *Appl. Soft Comput. J.*, vol. 97, pp. 106-112, 2020.doi: 10.1016/j.asoc.2020.106810.

39.E. H. Mohamed, W. H. El-behaidy, G. Khoriba, and J. Li, "Improved White Blood Cells Classification Based on Pre-trained Deep Learning Models," vol. 16, no.1, pp. 37–45, 2020. doi: 10.1080/21665432.2020.1654323.

40. N. Theera-Umpon and S. Dhompongsa, "Morphological granulometric features of nucleus in automatic bone marrow white blood cell classification," *IEEE Transactions on Information Technology in Biomedicine,* vol. 11, pp. 353-359, 2007.

41. C. Di Ruberto, A. Dempster, S. Khan and B. Jarra, "Morphological image processing for evaluating malaria disease," in *International Workshop on Visual Form*, 2011, vol. 35, pp. 333-343.

42. M. Togacar, B. Ergen, and M. E. Sertkaya, "Subclass Separation of White Blood Cell Images Using Convolutional Neural Network 'Models," pp. 63–68, 2019.

43. S.Bunrit, N.Kerdprasop, A. Kerdprasop, “Evaluating on the Transfer Learning of CNN Architectures to a Construction Material Image Classification Task”, Int. J. Mach. Learn. Computing, 2019. doi: 10.18178/ijmlc.2019.9.2.787.

44. R. Rashid, S. Khan and B. Jarra, “Deep Convolutional Neural Networks for Lungs Segmentation and Diagnosis of Tuberculosis from Chest X-Ray (Master)”, National University of Science and Technology, *SN Appl. Science.*, 2019, vol. 4, pp. 502-512. doi: 10.1005/s42478-021-04485-8, 2019.

45. M. Yani, D. Patra and K. Kumar, “Application of transfer learning using convolutional neural network method for early detection of terry’s nail”, in Journal of Physics: Conference Series, 2019, vol. 1201, pp.115-123.

46. S.Prajapati, A.Nagaraj, and R. Mitra, “Application of transfer learning using convolutional neural network methods”, in *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 70-76. IEEE.

47. K.Simonyan, and A. Zisserman, “A comparative study of fine-tuning deep learning models for plant disease identi fi cation”, 2020, vol. 11, pp. 353-359. doi: 1409.1556.

48. K. Simonyan and A. Mitra, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

49. S. Bunrit, N. Kerdprasop, & K. Kerdprasop, "Evaluating on the Transfer Learning of CNN Architectures to a Construction Material Image Classification Task," in International *journal of Machine learning and Computing, 2019,* pp. 316–322. doi: 10.1301/AIMS.2019.59.

50. M. Yani, "Application of transfer learning using convolutional neural network method for early detection of terry’s nail", in Journal of Physics, vol. 11, pp. 353-359, 2019.

51. S. A. Prajapati, R. Nagaraj, & S. Mitra, " Very deep convolutional networks for large-scale image recognition," in *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–10, 2017.

52. K. Simonyan, & A. Zisserma, “Deep Residual Learning for Image Recognition,"*3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc*, pp. 1–14, 2015.

53. E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, “Malicious Software Classification using Transfer Learning of ResNet-50 Deep Neural Network,”, pp. 6–12, 2017. doi: 10.1109/ICMLA.2017.00-19.

54. K. He and J. Sun, “Rethinking the InceptionV3 Architecture,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778. doi: 10.1109/CVPR.2016.90.

55. C. Szegedy, V. Vanhoucke, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

56. E. Chebet, L. Yujian, S. Njuki, and L. Yingchun, “Convolutional Neural Network deep learning models,”*Comput. Electron. Agric*, no. March, pp. 213-221, 2018. doi: 10.1016/j.compag.2018.03.032.

57. J. M. Celaya-padilla, J. I. Galván-tejada, H. Gamboa-rosales, and C. A. Olvera-olvera, “applied sciences Comparison of Convolutional Neural Network Inception Architectures for Classification of Tomato Plant Diseases”,2019.doi: 10.3390/app10041245.

58. C. Szegedy, S. Reed, P. Sermanet, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Comput. Electron. Agric*, March 2018, pp. 312-318. doi: 10.1016/j.compag.2018.03.032.

59. C. Szegedy, V. Vanhoucke, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp.2818–2826. doi: 10.1109/CVPR.2016.308.

60. J. Wang, G. R. M. Reddy, V. K. Prasad, and V. S. Reddy, “Rethinking the Inception Architecture for Computer Vision,” Singapore: Springer, 2019, pp.135–143. doi: 10.1007/978-981-13-3600-313.

61. M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, “MobileNetV2 : Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474 .

62. A. I. Journal, X. Yao, K. Sun, X. Bu, C. Zhao, and Y. Jin, “Classification of white blood cells using weighted optimized deformable convolutional neural networks convolutional neural networks,” *Artif. Cells, Nanomedicine, Biotechnol.*, vol. 49, no. 1, pp. 147–155, 2021. doi: 10.1080/21691401.2021.1879823.

63. M.Kayed, A.Anter and H.Mohamed , “Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5,” *ITCE,* 2020, vol. 11, pp. 353-359. doi: 1409.1556.2020.

64. https:// www. kaggle.com/ paultimothymooney/ blood- cells, Accessed 08.09.2020.

65. http://users.cecs.anu.edu.au/~hrezatofighi/Data/Leukocyte%20Data.htm, Accessed 08.09.2020.