Leukemia detection based on microscopic blood smear images using deep learning

Graduation Project Report

Team work:

Abd El_Mageed Ahmed Abd El_Mageed.

Ahmed Hussien Kamel.

Dalia Mostafa Farghal.

Supervisor: Dr.Eng. Mohamed Nagy Saad Elziftawy.

Acknowledgments

We have taken efforts in this graduation project. However, it would not have been possible without the kind support and help of many individuals and our faculty. We would like to extend our sincere thanks to all of them.

Firstly, we thank to Allah for helping us achieve this project successfully.

Secondly, we are highly indebted to Dr.Eng.: Mohamed Nagy Saad for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

Thirdly, we would like to express our gratitude towards all staff of Biomedical Engineering department for their kind co-operation and encouragement which help us in completion of this project.

Finally, our deep and sincere gratitude to our families for their continuous and unparalleled love, help and support. We forever indebted to our parents for giving us the opportunities and experiences that have made us who we are. This graduation project would not have been possible if not for them, and we dedicate this milestone to them.

Abstract

Leukemia induced death has been listed in the top ten most dangerous mortality cause for human being. Among many reasons, one of them is the slow decision-making process which make suitable medical treatment cannot be applied on time.

The hematologist faces difficulty in classifying white blood cells included cancerous leukemia cells. Currently, the diagnosis of hematological disorder was made by microscopic analysis of peripheral blood smear by hematological experts.

However, this kind of diagnosis is complex, costly process, time-consuming and inherently subjective, hence to avoid such problems a classification technique based on computer vision is required. In order to get the most effective treatment, the patient needs early diagnosis. Therefore, we need to have a system that supports early diagnosis to guide treatment of patients as soon as possible.

This project provides leukemia detection system using deep neural network in order to early diagnosis of leukemia in blood sample based on microscopic blood smear images. The system provides 97.3 % accuracy in classification of samples as cancerous or normal samples

List of Content

List of figures

List of tables

Abbreviation

NK	Natural Killer
RBC	Red Blood Cells
WBC	White Blood Cells
ALL	Acute Lymphoblastic (Lymphocytic) Leukemia
AML	Acute Myeloid (Myelogenous) Leukemia
CLL	Chronic Lymphocytic Leukemia
CML	Chronic Myeloid (Myelogenous) Leukemia
CBC	Complete Blood Count
DL	Deep Learning
AI	Artificial Intelligent
ANNS	Artificial Neural Networks
SGD	Stochastic Gradient Descent
CNNS	Convolutional Neural Networks
SVM	Support Vector Machine
HOG	Histogram of Oriented Gradient
LSTM	Long Short-Term Memory
RELU	Rectifier Linear Unit
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ILSVRC	ImageNet Large Scale Visual Recognition Competition
CT	Computed Tomography
AUC	Area Under Curve

ROC	Receiver Operating Characteristic
MTL	Multi Task Learning
DR	Diabetic Retinopathy
BOVW	Bag of Visual Word
LBP	Local Binary Pattern
CAD	Computed Aided Detection
SSAE	Stacked Sparse Autoencoder
LDDMM	Large Deformation Diffeomoptic Metric Mapping
GUI	Graphical User Interface
TK	Tkinter
USB	Universal Serial Bus

Chapter 1

Introduction

1.1 Objectives of project:

The main objective is to build a neural network that can classify human blood smear image and identify the status (diagnosis) of cell where it is normal or it is a leukemia cell, where human blood is the main source to detect diseases at earlier stage and can prevent it quickly, the system can be automated in order to produce lab results quickly, easily and efficiently**.**

1.2 Introduction

Medical imaging has become one of the most important visualization and interpretation methods in biology and medicine over the past decade. This time has witnessed a tremendous development of new, powerful instruments for detecting, storing, transmitting, analyzing, and displaying medical images. This has led to a huge growth in the application of digital image processing techniques for solving medical problems [1].

The most challenging aspect of medical imaging lies in the development of integrated systems for the use of the clinical sector. Design, implementation, and validation of complex medical systems require a tight interdisciplinary collaboration between physicians and engineers. Main objective of analyzing through images is to gather information, detection of diseases, diagnosis diseases, control and therapy, monitoring and evaluation [2]. At the moment, identification of blood disorders is done through visual inspection of microscopic images of blood cells. From the identification of blood disorders, classification of certain diseases could be reached. Leukemia is a type of cancer affecting blood, and if it is detected late, it will result in death. Leukemia occurs when a lot of abnormal white blood cells are produced by bone marrow. When abnormal white blood cells are a lot, the balance of the blood system will be disrupted. The existence of abnormal blood can be detected when the blood sample is taken and examined by hematologists.

1.3 Problem definition

Microscopic images will be inspected visually by hematologists and the process is time consuming and tiring [3 - 5]. The process require human expert and prone to errors due to emotion disturbance and human physical capability that is of course have its own limit. Moreover, it is difficult to get consistent results from visual inspection [3].

Visual inspection also can only give qualitative results for further research [3]. Studies show that most of the recent techniques use all information about blood for e.g. number of red blood cells, hemoglobin level, hematocrit level, mean volume corpuscle and many more as the parameter for classifying diseases such as thalassaemia, cancer and etc.

In order to know all information about blood, expensive testing and equipment of labs are required. Automatic image processing system is urgently needed and can overcome related constraints in visual inspection. The system to be developed will be based on microscopic

images to recognize leukemia cell in blood smear. The early and fast identification of the leukemia type greatly aids in providing the appropriate treatment for particular type of leukemia [6]. The currently used diagnostic methods rely on analyzing immunophenotyping, fluorescence in situ hybridization (FISH), cytogenetic analysis and cytochemistry. Sophisticated and expensive laboratories are required in order to run the diagnostic methods and it have been reported to provide a high ratio of misidentification, with this system, more images can be processed, reduce analyzing time, exclude the influence of subjective factors and increase the accuracy of identification process at the same time.

In machine learning the inspection and classification of leukemia will be based on texture, shape, size, color, and statistical analysis of white blood cells while deep learning makes it much deeper and get the whole features in the image as we will discuss this in latter chapters.

This project is applied to increase efficiency globally and at the same time can benefit and be a huge contribution in medical and pattern recognition field. The main objective is to enhance algorithms that can extract data from human blood where human blood is the main source to detect diseases at earlier stage and can prevent it quickly [8]. This system should be robust towards diversity that exists among individual, sample collection protocols, time and etc. this is an automated system that can produce lab results quickly, easily and efficiently.

Chapter2

What is leukemia?

2.1 About Marrow, Blood and Blood Cells

Marrow is the spongy center inside of bones where blood cells are produced. **Blood cells** are produced in the marrow. They begin as stem cells. Stem cells become red cells, white cells and platelets in the marrow. Then the red cells, white cells and platelets from the blood.

Platelets form plugs that help stop bleeding at the site of an injury.

Red cells carry oxygen around the body. When the number of red cells is below normal, the condition is called anemia. Anemia may make you feel tired or short of breath. It may make the skin look pale.

White cells fight infection in the body. There are two major types of white cells: germ-eating cells (neutrophils and monocytes) and infection-fighting lymphocytes (B cells, T cells and natural killer [NK] cells).

Plasma is the liquid part of the blood. It is mostly water. It also has some vitamins, minerals, proteins, hormones and other natural chemicals in it.

2.2 Normal Blood Cell Count Facts

The ranges of blood cell count below are for normal adults. They may be a little different from lab to lab and for children and teens [6].

2.2.1 Red blood cell (RBC) count

Men: 4.5 to 6 million red cells per microliter of blood.

Women: 4 to 5 million red cells per microliter of blood.

2.2.2 Hematocrit (the part of the blood made up of red cells)

Men: 42% to 50% and Women: 36% to 45%

2.2.3 Hemoglobin (amount of the red cell pigment that carries oxygen)

Men: 14 to 17 grams per 100 milliliters of blood.
Women: 12 to 15 grams per 100 milliliters of blood.

2.2.4 Platelet count

150,000 to 450,000 platelets per microliter of blood

2.2.5 White blood cell (WBC) count

4,500 to 11,000 white cells per microliter of blood

2.2.6 Differential (also called diff)

Shows the part of the blood made up of different types of white cells, The types of white cells counted are neutrophils, lymphocytes, monocytes, eosinophils and basophils. Adults usually have about 60% neutrophils, 30% lymphocytes, 5% monocytes, 4% eosinophils and less than 1% basophils in the blood.

2.3 What is Leukemia?

Leukemia is the general term for some different types of blood cancer. There are **four main types** of leukemia called:

1) **Acute lymphoblastic (lymphocytic) leukemia (ALL)**
2) **Acute myeloid (myelogenous) leukemia (AML)**
3) **Chronic lymphocytic leukemia (CLL)**
4) **Chronic myeloid (myelogenous) leukemia (CML).**

It is important to know that patients are affected and treated differently for each type of leukemia. These four types of leukemia do have one thing in common they begin in a cell in the bone marrow. The cell undergoes a change and becomes a type of leukemia cell. The marrow has two main jobs. The first job is to form myeloid cells.

Myeloid leukemia can begin in these cells. The second job is to form lymphocytes, which are a part of the immune system. Lymphocytic leukemia can arise in these cells. The leukemia is called **lymphocytic** or **lymphoblastic** if the cancerous change takes place in a type of marrow cell that forms **lymphocytes**.

The leukemia is called **myelogenous** or **myeloid** if the cell change takes place in a type of marrow cell that normally goes on to form red cells, some kinds of white cells and platelets [6]. For each type of leukemia, patients are affected and treated differently. ALL and AML (acute leukemias) are each composed of young cells, known as **lymphoblasts** or **myeloblasts** , These cells are sometimes called **blasts** , Acute leukemias progress rapidly without treatment. CLL and CML have few or no blast cells, CLL and CML often progress slowly compared to acute leukemias, even without immediate treatment

2.4 How Does Leukemia Develop?

Doctors do not know the causes of most cases of leukemia. Thy do know that once the marrow cell undergoes a leukemic change, the leukemia cells may grow and survive better than normal cells. Over time, the leukemia cells crowd out or suppress the development of normal cells, The rate at which leukemia progresses and how the cells replace the normal blood and marrow cells are different with each type of leukemia [6].

2.4.1 - AML and ALL (as shown in figures 2.1 and 2.2)

In these diseases, the original acute leukemia cell goes on to form about a trillion more leukemia cells. These cells are described as nonfunctional because they do not work like normal cells. They also crowd out the normal cells in the marrow. This causes a decrease in the number of new normal cells made in the marrow. This further results in low red cell counts (anemia), low platelet counts (bleeding risk) and low neutrophil counts (infection risk).

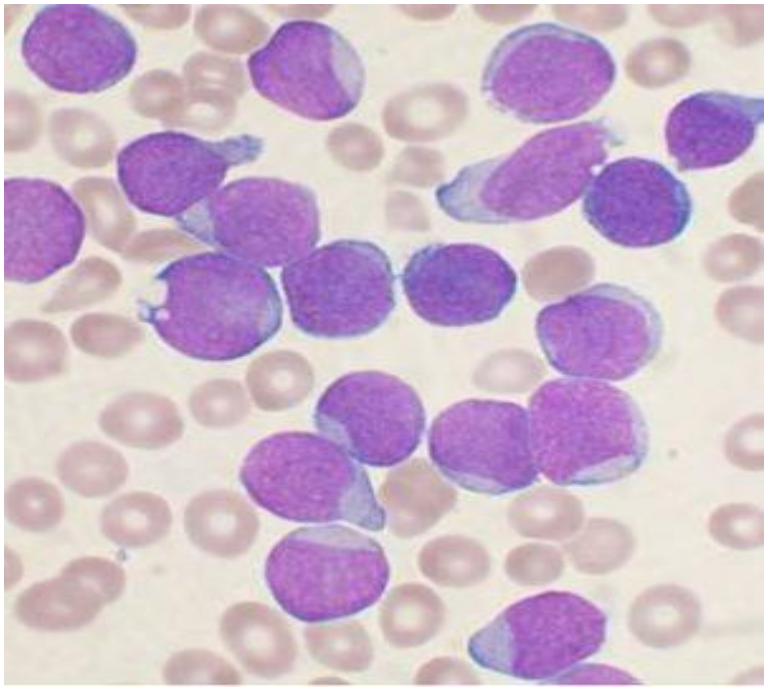


Figure 2.1: acute lymphoblastic leukemia (ALL)

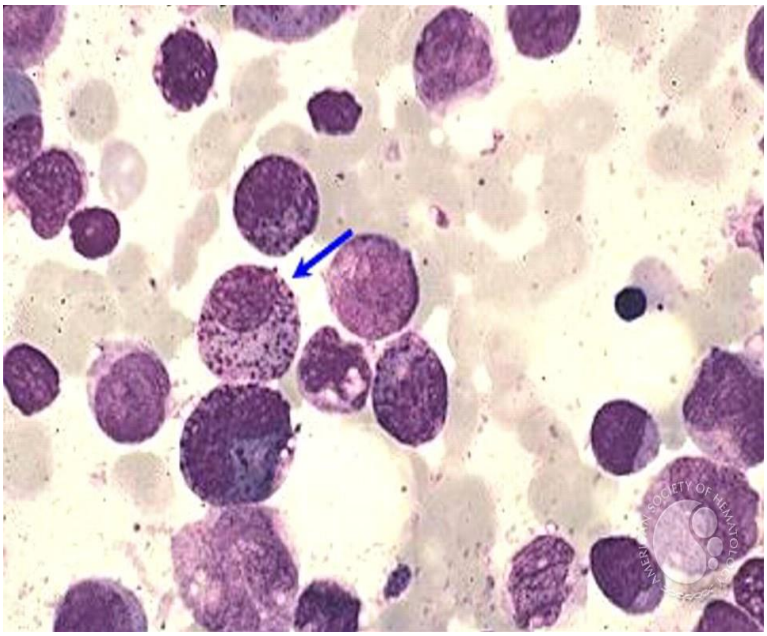


Figure 2.2: Acute myeloid leukemia (AML)

2.4.2 Chronic myeloid leukemia (CML)(figure2.3)

The leukemia cell that starts this disease makes blood cells (red cells, white cells and platelets) that function almost like normal cells. The number of red cells is usually less than normal, resulting in anemia. But many white cells and sometimes many platelets are still made. Even though the white cells are nearly normal in how they work, their counts are high and continue to rise. This can cause serious problems if the patient does not get treatment. If untreated, the white cell count can rise so high that blood flow slows down and anemia becomes severe.

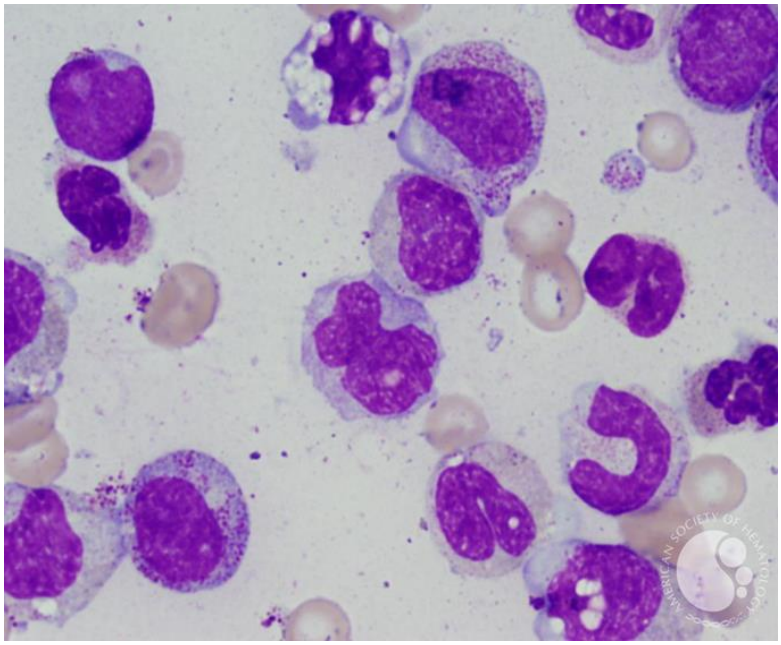


Figure2.3: chronic myeloid leukemia (CML)

2.4.3 Chronic lymphocytic leukemia (CLL) (as shown in figure 2.4)

The leukemia cell that starts this disease makes too many lymphocytes that do not function. These cells replace normal cells in the marrow and lymph nodes. They interfere with the work of normal lymphocytes, which weakens the patient's immune response. The high number of leukemia cells in the marrow may crowd out normal blood-forming cells and lead to a low red cell count (anemia). A very high number of leukemia cells building up in the marrow also can lead to low white cell (neutrophil) and platelet counts.

Unlike the other three types of leukemia, some patients with CLL may have disease that does not progress for a long time. Some people with CLL have such slight changes that they remain in good health and do not need treatment for long periods of time. Other patients require treatment at the time of diagnosis or soon after.

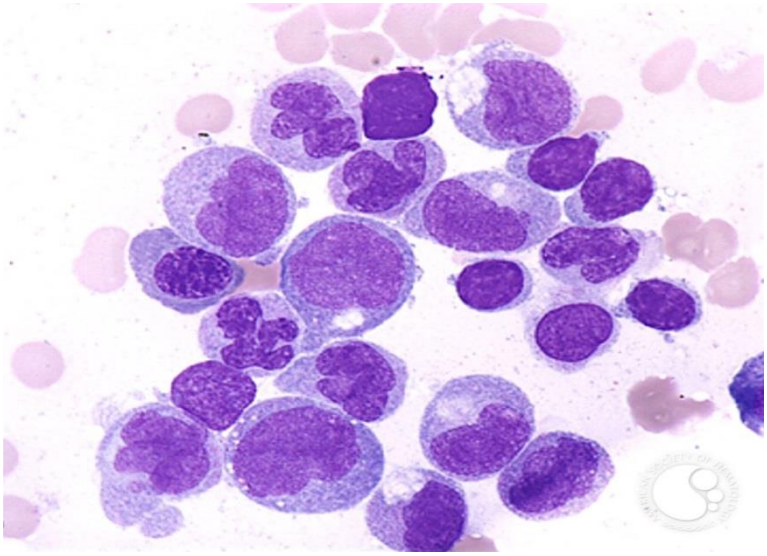


Figure 2.4: chronic lymphocytic leukemia (CLL)

2.5 Who's at Risk?

People can get leukemia at any age. It is most common in people over age 60. The most common types in adults are AML and CLL. Each year, about 3,811 children develop leukemia. ALL is the most common form of leukemia in children [6].

The term risk factor is used to describe something that may increase the chance that a person will develop leukemia, for most types of leukemia, the risk factors and possible causes are not known, For AML specific risk factors have been found, but most people with AML do not have these risk factors.

Some of those factors are:

1) Some types of chemotherapies
2) Down syndrome and some other genetic diseases
3) Chronic exposure to benzene. The majority of benzene in the environment comes from petroleum products, however, half of the personal exposure is from cigarette smoke
4) Radiation therapy used to treat cancer

Exposure to high doses of radiation therapy is also a risk factor for ALL and CML. In CLL, while it is not common, some families have more than one blood relative with the disease. Doctors are studying why some families have a higher rate of CLL. Other possible risk factors

for the four types of leukemia are continually under study. Leukemia is not contagious (catching).

2.6 Signs and Symptoms

Some signs or symptoms of leukemia are similar to other more common and less severe illnesses. Specific blood tests and bone marrow tests are needed to make a diagnosis.
Signs and symptoms vary based on the type of leukemia.

For acute leukemia, they include:

1) Tiredness or no energy
2) Shortness of breath during physical activity
3) Pale skin
4) Mild fever or night sweats
5) Slow healing of cuts and excess bleeding
6) Black-and-blue marks (bruises) for no clear reason
7) Pinhead-size red spots under the skin
8) Aches in bones or joints (for example, knees, hips or shoulders)
9) Low white cell counts, especially monocytes or neutrophils.

People with CLL or CML may not have any symptoms. Some patients learn they have CLL or CML after a blood test as part of a regular checkup. Sometimes, a person with CLL may notice enlarged lymph nodes in the neck, armpit or groin. The person may feel tired or short of breath (from anemia) or have frequent infections if the CLL is more severe. In these cases, a blood test may show an increase in the lymphocyte count. CML signs and symptoms tend to develop slowly. People with CML may feel tired and short of breath while doing everyday activities. Thy may also have an enlarged spleen (leading to a “dragging” feeling on the upper left side of the belly), night sweats and weight loss [6].

2.7 Diagnosis

A **CBC (complete blood count)** is used to diagnose leukemia. A CBC is a test that is also used to diagnose and manage many other diseases. This blood test may show high or low levels of white cells and show leukemia cells in the blood. Sometimes, platelet counts and red cell counts are low normal range (4.5: 6 million per microliter of blood for men and 4:5 million for women).

Bone marrow tests (aspiration and biopsy) are often done to confirm the diagnosis and to look for chromosome abnormalities. These tests identify the leukemia cell type. A complete blood exam and a number of other tests are used to diagnose the type of leukemia. These tests can be repeated after treatment begins to measure how well the treatment is working [7].

2.7.1 How Are Blood and Bone Marrow Tests Done?

2.7.1.1 - Blood tests

Usually a small amount of blood is taken from the person's arm with a needle. The blood is collected in tubes and sent to a lab.

2.7.1.2 Bonemarrow

A liquid sample of cells is taken from **aspiration** the marrow through needle. The cells are then looked at under a microscope.

2.7.1.3 Bonemarrow biopsy

A very small amount of bone filed with marrow cells is removed through a needle. The cells are then looked at under a microscope; each main type of leukemia also has different **subtypes**. In other words, patients with the same main type of leukemia may have different forms of the disease. A patient's age, general health and subtype may play a role in determining the best treatment plan. Blood tests and bone marrow tests are used to identify AML, ALL, CML or CLL subtypes.

2.8 Treatment

It is important to get medical care at a center where doctors are experienced in treating patients with leukemia. The aim of leukemia treatment is to bring about a complete remission. This means that after treatment, there is no sign of the disease and the patient returns to good health. Today, more and more leukemia patients are in complete remission at least five years after treatment [6].

2.8.1 Acute Leukemia.

Treatment for patients with acute leukemia may include chemotherapy, stem cell transplantation or new approaches under study (clinical trials). Patients with an acute leukemia ALL and AML need to start treatment soon after diagnosis. Usually, they begin treatment with chemotherapy, which is often given in the hospital. The first part of treatment is called induction therapy. More inpatient treatment is usually needed even after a patient is in remission. This is called post remission therapy and consists of consolidation (intensification) therapy and in some cases maintenance therapy. This part of treatment may include chemotherapy with or without stem cell transplantation (sometimes called bone marrow transplantation).

2.8.2 Chronic Myeloid Leukemia (CML).

Patients with CML need treatment soon after diagnosis. There are three drugs approved for newly diagnosed patients. These drugs are imatinib mesylate (Gleevec®), dasatinib (Sprycel®) or nilotinib (Tasigna®). If one of these drugs is not effctive, one of the other drugs can be tried. All three of these drugs are taken by mouth. Gleevec, Sprycel and Tasigna do not cure CML. But they keep CML under control for many patients for as long as they take it. Allogeneic stem cell

transplantation is another treatment option that is only done if CML is not responding as expected to drug therapy.

2.8.3 Chronic Lymphocytic Leukemia (CLL)

Some CLL patients do not need treatment for long periods of time after diagnosis, sometimes called watch and wait. Patients who need treatment may receive chemotherapy or monoclonal antibody therapy alone or in combination. Allogeneic stem cell transplantation is a treatment option for certain patients, but usually not as the first choice of therapy.

2.9 Side Effects of Leukemia Treatment [6]

The term side effect is used to describe the way that treatment affects healthy cells. **People react to treatments in different ways**. Sometimes they have mild side effects. Many treatment side effects go away when treatment ends or become less noticeable over time. Most can be handled without the need to stop the drug. Other side effects may be serious and lasting. For example, people who have side effects from monoclonal antibody therapy may experience the side effects while getting the IV treatment.

Side effects from chemotherapy, such as nausea or changes to normal blood cells, may occur for a period of time after the treatment.

2.9.1 Acute leukemias ALL& AML

Common side effects may include:

1) Changes to blood counts
2) Mouth sores
3) Nausea
4) Vomiting
5) Diarrhea
6) Hair loss
7) Rash
8) Fever.

2.9.2 Chronic lymphocytic leukemia (CLL).

Common side effects may include:

1) Extreme tiredness
2) Hair loss

3) Changes to blood counts
4) Upset stomach
5) Mouth sores
6) Diarrhea.

2.9.3 Chronic myeloid leukemia (CML).

Common side effects from tyrosine kinase inhibitor therapy such as Gleevec may include:

1) Changes to blood counts
2) Diarrhea
3) Muscle cramps and joint aches
4) Nausea
5) Swelling or fluid retention.

.

Chapter3

Deep learning

and

Convolutional neural networks

3.1 What is deep learning?

“Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but nonlinear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. , The key aspect of deep learning is that these layers are not designed by human engineers: they are learned from data using a general-purpose learning procedure” – Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, Nature 2015 [7].

Deep learning is a subfield of machine learning, which is, in turn, a subfield of artificial intelligence (AI). For a graphical depiction of this relationship, please refer to Figure 2.1. The central goal of AI is to provide a set of algorithms and techniques that can be used to solve problems that humans perform intuitively and near automatically, but are otherwise very challenging for computers, A great example of such a class of AI problems is interpreting and understanding the contents of an image – this task is something that a human can do with little-to-no effort, but it has proven to be extremely difficult for machines to accomplish. While AI embodies a large, diverse set of work related to automatic machine reasoning (inference, planning, heuristics, etc.), the machine learning subfield tends to be specifically interested in pattern recognition and learning from data.

Artificial Neural Networks (ANNs) are a class of machine learning algorithms that learn from data and specialize in pattern recognition, inspired by the structure and function of the brain. As we’ll find out, deep learning belongs to the family of ANN algorithms, and in most cases, the two terms can be used interchangeably. In fact, you may be surprised to learn that the deep learning field has been around for over 60 years, going by different names and incarnations based on research trends, available hardware and datasets, and popular options of prominent researchers at the time. In the remainder of this chapter, we’ll review a brief history of deep learning, discuss what makes a neural network “deep”, and discover the concept of “hierarchical learning” and how it has made deep learning one of the major success stories in modern day machine learning and computer vision.

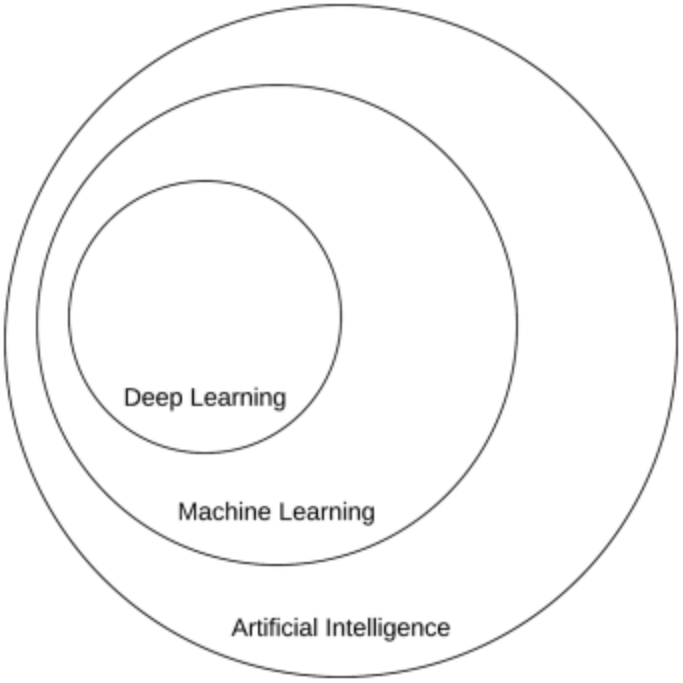


Figure 3.1: A Venn diagram describing deep learning as a subfield of machine learning which is in turn a subfield of artificial intelligence

3.2 A Concise History of Neural Networks and Deep Learning

The history of neural networks and deep learning is a long, somewhat confusing one. It may surprise you to know that "deep learning" has existed since the 1940s undergoing various name changes, including cybernetics, connectionism, and the most familiar, Artificial Neural Networks (ANNs). While inspired by the human brain and how its neurons interact with each other, ANNs are not meant to be realistic models of the brain. Instead, they are an inspiration, allowing us to draw parallels between a very basic model of the brain and how we can mimic some of this behavior through ANNs.

The first neural network model came from McCulloch and Pitts in 1943 [8]. This network was a binary classifier, capable of recognizing two different categories based on some input. The problem was that the weights used to determine the class label for a given input needed to be manually tuned by a human – this type of model clearly does not scale well if a human operator is required to intervene. Then, in the 1950s the seminal Perceptron algorithm was published by Rosenblatt [9][10] – this model could automatically learn the weights required to classify an input (no human intervention required). An example of the Perceptron architecture can be seen in Figure 3.2.

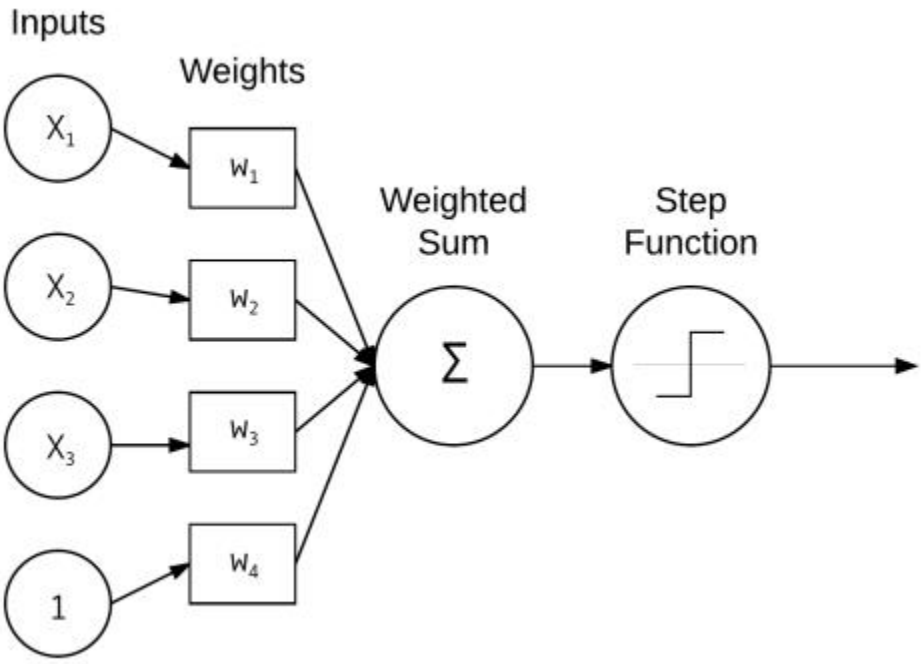


Figure 3.2: An example of the simple Perceptron network architecture that accepts a number of inputs, computes a weighted sum, and applies a step function to obtain the final prediction.

In fact, this automatic training procedure formed the basis of Stochastic Gradient Descent (SGD) which is still used to train very deep neural networks today. During this time period, Perceptron-based techniques were all the rage in the neural network community. However, a 1969 publication by Minsky and Paper [11] effectively stagnated neural network research for nearly a decade. Their work demonstrated that a Perceptron with a linear activation function (regardless of depth) was merely a linear classifier, unable to solve nonlinear problems. The canonical example of a nonlinear problem is the XOR dataset in Figure 3.3.

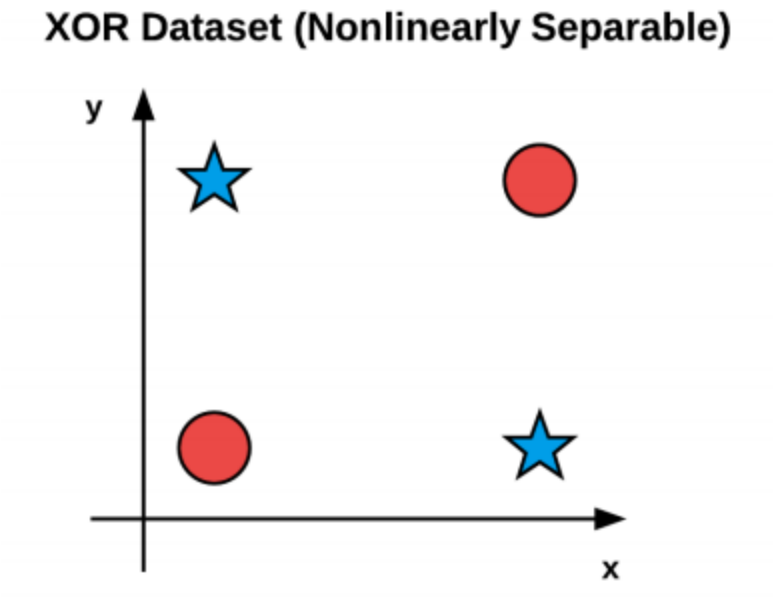


Figure3.3: The XOR dataset is an example of a nonlinear separable problem that the Perceptron cannot solve. Take a second to convince yourself that it is impossible to draw a single line that separates the blue stars from the red circles.

Furthermore, the authors argued that (at the time) we did not have the computational resources required to construct large, deep neural networks (in hindsight, they were absolutely correct). This single paper alone almost killed neural network research. Luckily, the backpropagation algorithm and the research by Warbots (1974) [12], Rinehart (1986) [13], and LeCun (1998) [14] were able to resuscitate neural networks from what could have been an early demise. Their research in the backpropagation algorithm enabled multi-layer feedforward neural networks to be trained (Figure 3.4).

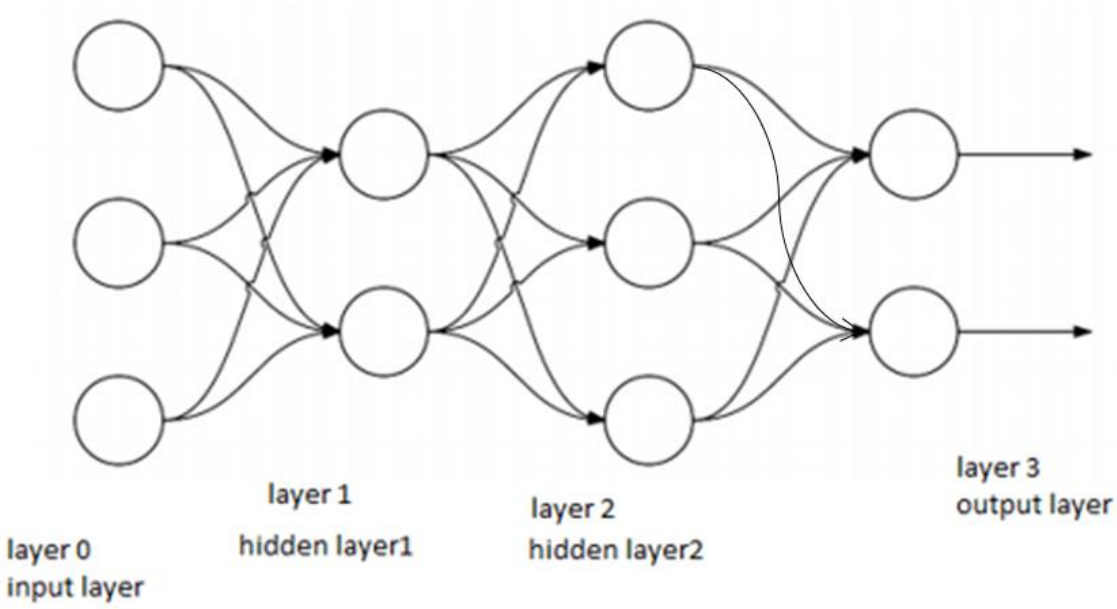


Figure 3.4: A multi-layer, feedforward network architecture with an input layer (3 nodes), two hidden layers (2 nodes in the first layer and 3 nodes in the second layer), and an output layer (2 nodes).

Combined with nonlinear activation functions, researchers could now learn nonlinear functions and solve the XOR problem, opening the gates to an entirely new area of research in neural networks. Further research demonstrated that neural networks are universal approximators [15], capable of approximating any continuous function (but placing no guarantee on whether or not the network can actually learn the parameters required to represent a function).The backpropagation algorithm is the cornerstone of modern day neural networks allowing us to efficiently train neural networks and "teach" them to learn from their mistakes. But even so, at this time, due to

(1) Slow computers (compared to modern day machines)

(2) Lack of large, labeled training sets, researchers were unable to (reliably) train neural networks that had more than two hidden layers – it was simply computationally infeasible. Today, the latest incarnation of neural networks as we know it is called **deep learning**. What sets deep learning apart from its previous incarnations is that we have faster, specialized hardware with more available training data. We can now train networks with many more hidden layers that are capable of hierarchical learning where simple concepts are learned in the lower layers and more abstract patterns in the higher layers of the network.

Perhaps the quintessential example of applied deep learning to feature learning is the Convolutional Neural Network (LeCun 1988) [15] applied to handwritten character recognition

which automatically learns discriminating patterns (called “filters”) from images by sequentially stacking layers on top of each other.

Filters in lower levels of the network represent edges and corners, while higher level layers use the edges and corners to learn more abstract concepts useful for discriminating between image classes. In many applications, CNNs are now considered the most powerful image classifier and are currently responsible for pushing the state-of-the-art forward in computer vision subfields that leverage machine learning.

3.3 Hierarchical Feature Learning

Machine learning algorithms (generally) fall into three camps – supervised, unsupervised, and semi-supervised learning. We’ll discuss supervised and unsupervised learning in this chapter while saving semi-supervised learning for a future discussion.
In the supervised case, a machine learning algorithm is given both a set of inputs and target outputs. The algorithm then tries to learn patterns that can be used to automatically map input data points to their correct target output. Supervised learning is similar to having a teacher watching you take a test. Given your previous knowledge, you do your best to mark the correct answer on your exam; however, if you are incorrect, your teacher guides you toward a better, more educated guesses the next time.In an unsupervised case, machine learning algorithms try to automatically discover discriminating features without any hints as to what the inputs are. In this scenario, our student tries to group similar questions and answers together, even though the student does not know what the correct answer is and the teacher is not there to provide them with the true answer.

Unsupervised learning is clearly a more challenging problem than supervised learning – by knowing the answers (i.e., target outputs), we can more easily define discriminate patterns that can map input data to the correct target classification. In the context of machine learning applied to image classification, the goal of a machine learning algorithm is to take these sets of images and identify patterns that can be used to discriminate various image classes/objects from one another. In the past, we used hand-engineered features to quantify the contents of an image – we rarely used raw pixel intensities as inputs to our machine learning models, as is now common with deep learning. For each image in our dataset, we performed feature extraction, or the process of taking an input image, quantifying it according to some algorithm (called a feature extractor or image descriptor), and returning a vector (i.e., a list of numbers) that aimed to quantify the contents of an image. Figure 3.5 below depicts the process of quantifying an image containing prescription pill medication via a series of Blackbox color, texture, and shape image descriptors.

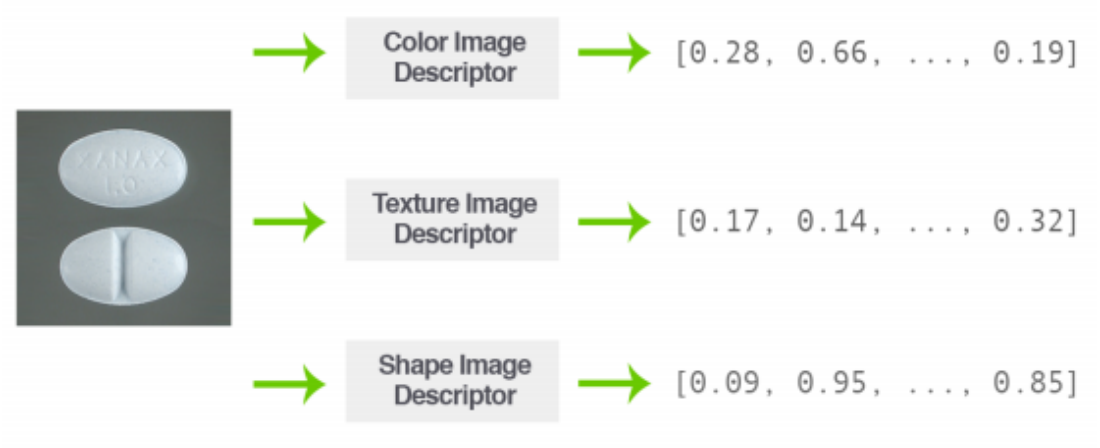


Figure 3.5: Quantifying the contents of an image containing a prescription pill medication via a series of Blackbox color, texture, and shape image descriptors.

Our hand-engineered features attempted to encode texture (Local Binary Patterns [17], Horlick texture [18]), shape (Hu Moments [19], Zernike Moments [20]), and color (color moments, color histograms, color correlograms [21]). Other methods such as key point detectors [22], Harris [23], Dog [28], to name a few) and local invariant descriptors (SIFT [24], SURF [25], BRIEF [26], ORB [27], etc.) describe salient (i.e., the most "interesting") regions of an image.

Other methods such as Histogram of Oriented Gradients (HOG) [28] proved to be very good at detecting objects in images when the viewpoint angle of our image did not vary dramatically from what our classifier was trained on. An example of using the HOG + Linear SVM detector method can be seen in Figure 3.6 where we detect the presence of stop signs in images.

For a while, research in object detection in images was guided by HOG and its variants, including computationally expensive methods such as the Deformable Parts Model [29] and Exemplar SVMs [30].In each of these situations, an algorithm was hand-defined to quantify and encode a particular aspect of an image (i.e., shape, texture, color, etc.). Given an input image of pixels, we would apply our hand-defined algorithm to the pixels, and in return receive a feature vector quantifying the image contents – the image pixels themselves did not serve a purpose other than being inputs to our feature extraction process. The feature vectors that resulted from feature extraction were what we were truly interested in as they served as inputs to our machine learning models. Deep learning, and specifically CNNs, take a different approach. Instead of hand-defining a set of rules and algorithms to extract features from an image, **these features are instead automatically learned from the training process**.

Again, let's return to the goal of machine learning: computers should be able to learn from experience (i.e., examples) of the problem they are trying to solve. Using deep learning, we try to understand the problem in terms of a hierarchy of concepts. Each concept builds on top of the

others. Concepts in the lower level layers of the network encode some basic representation of the problem, whereas higher level layers use these basic layers to form more abstract concepts. This hierarchical learning allows us to completely remove the hand-designed feature extraction process and treat CNNs as end-to-end learners. Given an image, we supply the pixel intensity values as **inputs** to the CNN. A series of **hidden layers** are used to extract features from our input image. These hidden layers build upon each other in a hierarchal fashion. At first, only edge-like regions are detected in the lower level layers of the network. These edge regions are used to define corners (where edges intersect) and contours (outlines of objects).

Combining corners and contours can lead to abstract "object parts" in the next layer. Again, keep in mind that the types of concepts these filters are learning to detect are automatically learned – there is no intervention by us in the learning process. Finally, **output** layer is used to classify the image and obtain the output class label – the output layer is either directly or indirectly influenced by every other node in the network. We can view this process as hierarchical learning: each layer in the network uses the output of previous layers as "building blocks" to construct increasingly more abstract concepts.

These layers are learned automatically – there is no hand-crafted feature engineering taking place in our network. Figure 3.6 compares classic image classification algorithms using hand-crafted features to representation learning via deep learning and CNNs.

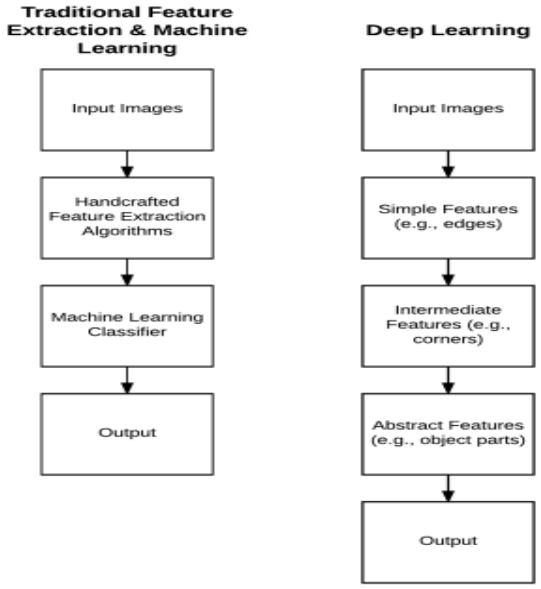


Figure 3.6: **Left:** Traditional process of taking an input set of images, applying hand-designed feature extraction algorithms, followed by training a machine learning classifier on the features. **Right:** Deep learning approach of stacking layers on top of each other that automatically learn re complex, abstract, and discriminating features.

One of the primary benefits of deep learning CNNs is that it allows us to skip the feature extraction step and instead focus on process of training our network to learn these filters. However, training a network to obtain reasonable accuracy on a given image dataset isn't always an easy task.

3.4 How "Deep" Is Deep?

To quote Jeff Dean from his 2016 talk, Deep Learning for Building Intelligent Computer Systems [31]: "When you hear the term deep learning, just think of a large, deep neural net. Deep refers to the number of layers typically and so this kind of the popular term that's been adopted in the press."

This is an excellent quote as it allows us to conceptualize deep learning as large neural networks where layers build on top of each other, gradually increasing in depth. **The problem is we still don't have a concrete answer to the question, "How many layers does a neural network need to be considered deep?"** The short answer is there is **no consensus** amongst experts on the depth of a network to be considered deep [10].

And now we need to look at the question of network type. By definition, a CNN is a type of deep learning algorithm. But suppose we had a CNN with only one convolutional layer – is a network that is shallow, but yet still belongs to a family of algorithms inside the deep learning camp considered being "deep"? We could not easily train networks with more than two hidden layers during the 1980s and 1990s.

In fact, Geoff Hinton supports this sentiment in his 2016 talk, Deep Learning [32], where he discussed why the previous incarnations of deep learning (ANNs) did not take off during the 1990s phase:

1. Our labeled datasets were thousands of times too small.
2. Our computers were millions of times too slow.
3. We initialized the network weights in a stupid way.
4. We used the wrong type of nonlinearity activation function.

All of these reasons point to the fact that training networks with a depth larger than two hidden layers were a futile, if not a computational, impossibility.

In the current incarnation we can see that the tides have changed. We now have:

1) Faster computers
2) Highly optimized hardware (i.e., GPUs
3) Large, labeled datasets in the order of millions of images
4) A better understanding of weight initialization functions and what does/does not work
5) Superior activation functions and an understanding regarding why previous nonlinearity functions stagnated research

Paraphrasing Andrew Ng from his 2013 talk, Deep Learning, Self-Taught Learning and Unsupervised Feature Learning [33], we are now able to construct deeper neural networks and train them with more data. As the depth of the network increases, so does the classification accuracy. This behavior is different from traditional machine learning algorithms (i.e., logistic regression, SVMs, decision trees, etc.) where we reach a plateau in performance even as available training data increases. A plot inspired by Andrew Ng's 2015 talk, what data scientists should know about deep learning, [34] can be seen in Figure 3.7, providing an example of this behavior.

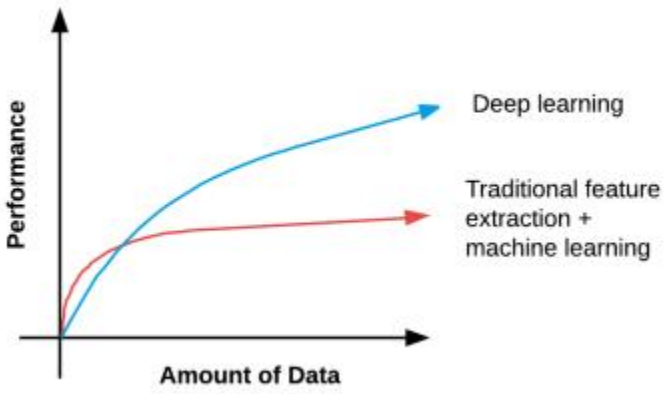


Figure 3.7: As the amount of data available to deep learning algorithms increases, accuracy does as well, substantially outperforming traditional feature extraction + machine learning approaches.

As the amount of training data increases, our neural network algorithms obtain higher classification accuracy, whereas previous methods plateau at a certain point. Because of the relationship between higher accuracy and more data, we tend to associate deep learning with large datasets as well.
When working on your own deep learning applications, its preferred to use the following **rule of thumb** to determine if your given neural

Network is deep:

1) Are you using specialized network architecture such as CNNs, RNNs, or Long Short-Term Memory (LSTM) networks? If so, yes, you are performing deep learning.
2) Does your network have a depth > 2? If yes, you are doing deep learning.
3) Does your network have a depth > 10? If so, you are performing very deep learning [35].

All that said; try not to get caught up in the buzzwords surrounding deep learning and what is/is not deep learning.

At the very core, deep learning has gone through a number of different incarnations over the past 60 years based on various schools of thought – but each of these schools of thought centralize around artificial neural networks inspired by the structure and function of the brain. Regardless of network depth, width, or specialized network architecture, you're still performing machine learning using artificial neural networks

3.5 What is Convolution Neural Network "CNN"?

A convolution is defined as an operation on two functions. In image analysis, one function consists of input values (e.g. pixel values) at a position in the image, and the second function is a filter (or kernel) as shown in fig 3.8; each can be represented as array of numbers. Computing the dot product between the two functions gives an output [36].

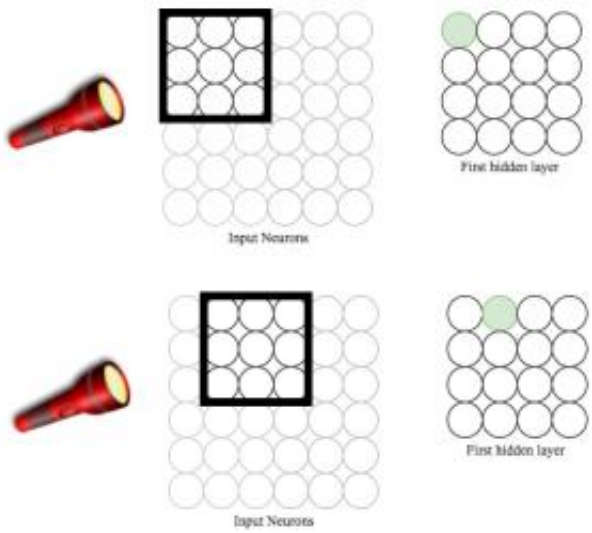


Fig3.8 Process of convolution

3.5.1 History of CNN:

Work on CNNs has been done since the late seventies (Fukushima, 1980) and they were already applied to medical image analysis in 1995 by Lo et al. (1995). They saw their first successful real-world application in Lent (LeCun et al., 1998) for hand-written digit recognition. Despite these initial successes, the use of CNNs did not gather momentum until various new techniques were developed for efficiently training deep networks, and advances were made in core computing systems. The watershed was the contribution of Izhevsk et al. (2012) to the ImageNet challenge in December 2012. The proposed CNN, called Alex Net, won that competition by a large margin. In subsequent years, further progress has been made using related but deeper

architectures (Breshkovsky et al., 2014). In computer vision, deep convolutional networks have now become the technique of choice [37].

The convolutional neural network (CNN) has been widely used in image classification, video recognition, and object detection, and it has achieved the excellent performance in these areas. The CNN usually contains convolutional layers, pooling layers, one or more fully connected layers, and a SoftMax layer. The convolutional layers combined with the pooling layers are used for extracting features. The SoftMax layer is regarded as the classifier [38].

3.6 CNN Layers:

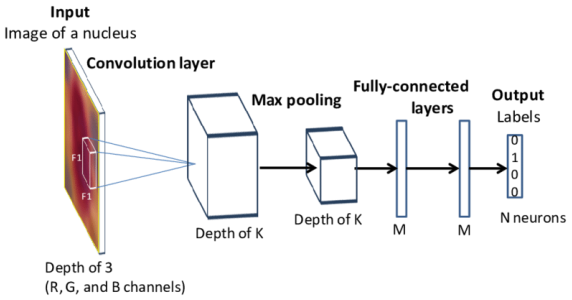


Fig3.9 Schematic representation of the architecture of convolutional neural network

3.6.1 Convolutional layer

The most important part or layer of a convolutional neural network is a convolutional layer. There may be one or more convolutional layer. Generally, the first layer in CNN model is a convolutional layer. Deep convolutional neural networks involve a high number of convolutional layers. These layers are responsible for identifying low level as well as high-level complex features in a given input the process to compute a single output matrix is defined as follows:

$$A_j = f\left(\sum_{i=1}^{N} I_i * K_{i,j} + B_j\right) \qquad (1)$$

Each input matrix I-I is convolved with a corresponding kernel matrix Kim, and summed with a bias value Bj at each element. Finally, a non-linear activation function is applied to each element [40].

3.6.2 Neuron Activation

The implementation of CNN is a discriminative trained model that uses standard back-propagation algorithm using a sigmoid (Equation 2), (Rectified Linear Units (ReLU) (Equation 3) as activation function [39], and tanh (Equation 4) [38]

$$F(x) = 1/(1 + e^{-x}) \qquad (2)$$

$$F(x) = \max(0, x) \qquad (3)$$

$$F(x) = \tanh(x) \quad (4)$$

3.6.3 Pooling layer

Pooling layers are generally used just after the convolutional layers. Their purpose is to simplify information outputted by the convolutional layer [4]. In sum, it prepares a condensed feature map from the feature map generated in the previous step. One of the most common procedures for pooling is max-pooling. Let us consider the following example.[41]

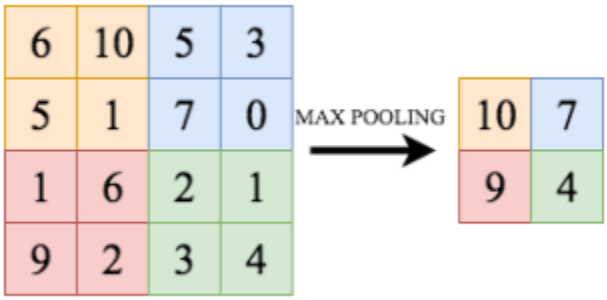


Fig 3.10: Max pooling layer

3.6.4 Fully-connected layer

This layer is the last layer of the CNN model. Its job is to input 3-dimensional volume and output an N dimensional vector where N is the number of classes [40].

3.6.5 SoftMax.

A SoftMax layer can predict n different classes through computing the probability of belonging to each category. In the last layer within network, the feature will rasterize into x, this is a column feature vector [38]. :

$$p(y = j|x, \theta) = \frac{e^{\theta_j^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} \quad (5)$$

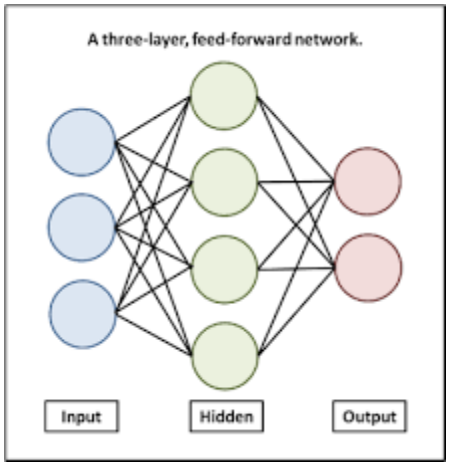


Fig 3.11: fully connected layer and SoftMax

3.7 What is Transfer Learning?

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task [41].

3.8 General CNN Architectures

Architectural Innovations in CNN Different improvements in CNN architecture have been made from 1989 to date. These improvements can be categorized as parameters optimizations, regularization, structural reformulation, etc. However, it is observed that the main upgradation in CNN performance was mainly due to the restructuring of processing units and designing of new blocks. All of the innovations in CNN have been made in relation with depth and spatial exploitation. Depending upon the type of architectural modification, CNN can be broadly categorized into seven different classes' namely spatial exploitation, depth, multi-path, width, channel boosting, feature map exploitation, and attention based CNNs. Taxonomy of deep CNN architectures is shown in Figure 3 [42].

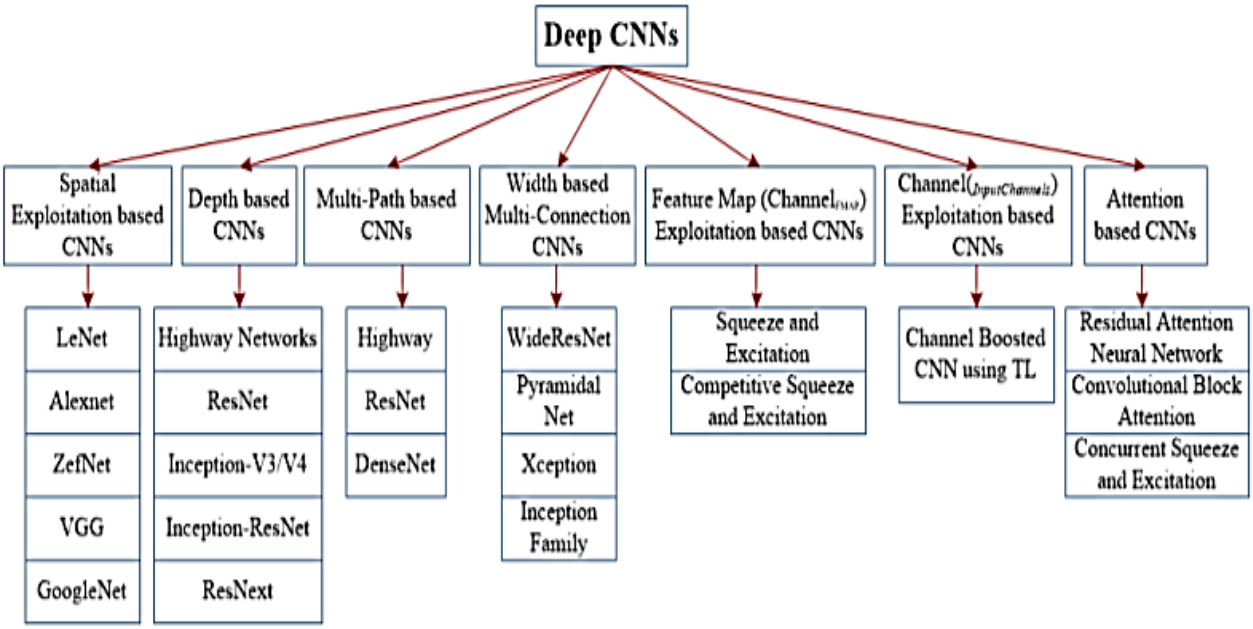


Fig 3.12: Taxonomy of deep CNN architectures

LeNet (LeCun et al., 1998) as shown in fig.3.13 and **AlexNet** (Krizhevsky et al., 2012) as shown in fig.3.14, introduced over a decade later, were in essence very similar models. Both networks were relatively shallow, consisting of two and five convolutional layers, respectively, and employed kernels with large receptive fields in layers close to the input and smaller kernels closer to the output.

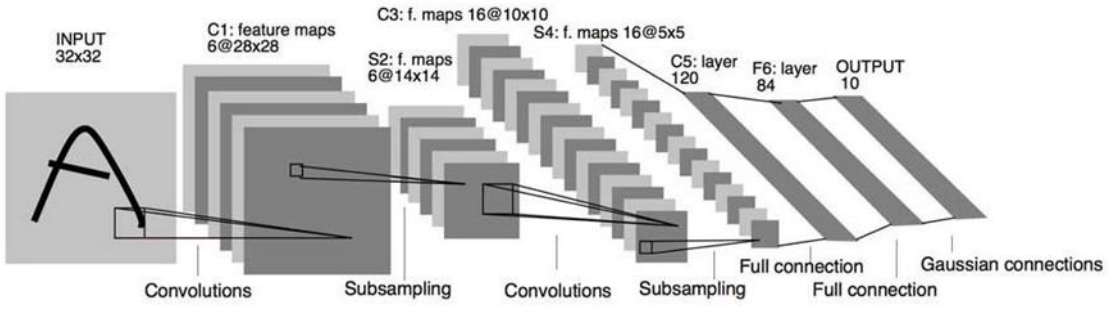


Fig 3.13: basic layout of LENET architecture.

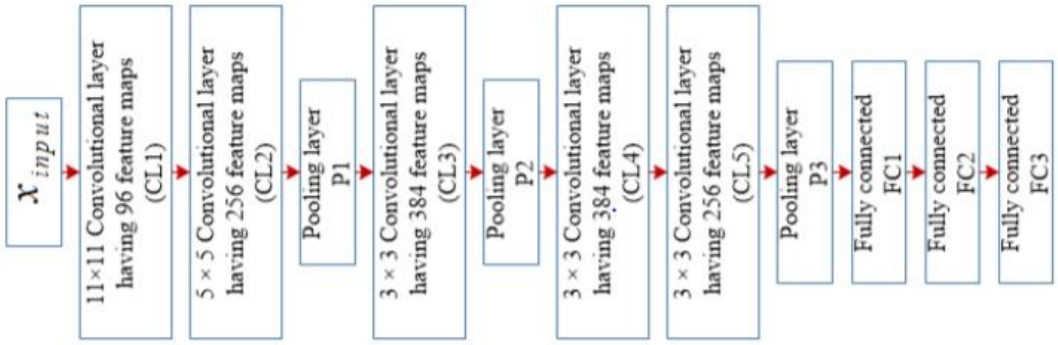


Fig3.14 basic layout of ALEXNET architecture .

These deeper architectures generally have a lower memory footprint during inference, which enable their deployment on mobile computing devices such as smartphones. Simonyan and Zisserman (2014) were the first to explore much deeper networks, and employed small, fixed size kernels in each layer. A 19-layer model often referred to as **VGG19 or OxfordNet** as shown in fig 3.15**,** won the ImageNet challenge of 2014.

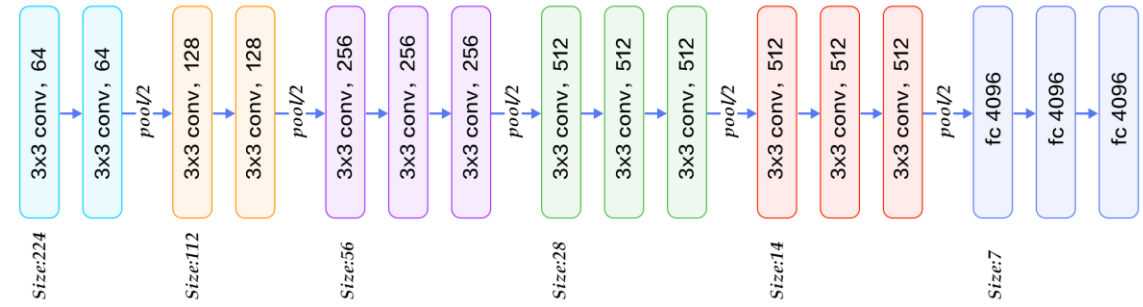


Fig3.15: basic layout of VGG architecture .

On top of the deeper networks, more complex building blocks have been introduced that improve the efficiency of the training procedure and again reduce the amount of parameters. Szegedy et al. (2014) introduced a 22-layer network named **GoogLeNet** as shown in fig.3.16.

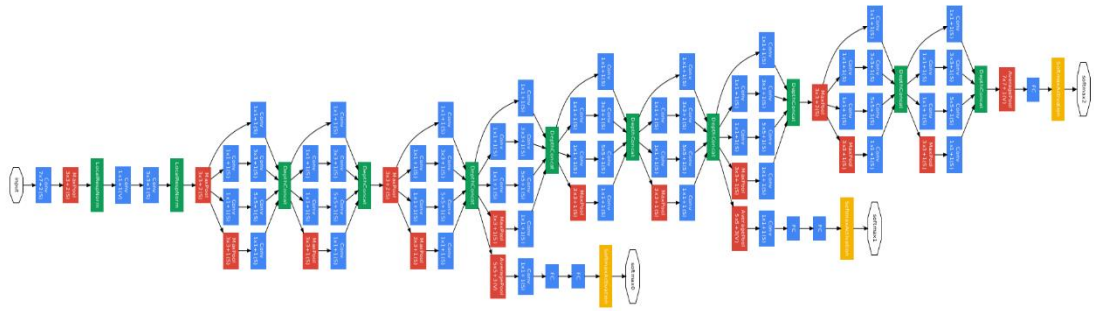


Fig 3.16: basic layout of GoogLeNet architecture .

The **ResNet** architecture (He et al., 2015) as shown in fig 3.17 won the ImageNet challenge in 2015 and consisted of so-called ResNet-blocks. Rather than learning a function, the residual block only learns the residual and is thereby pre-conditioned towards learning mappings in each layer that are close to the identity function. This way, even deeper models can be trained effectively [37].

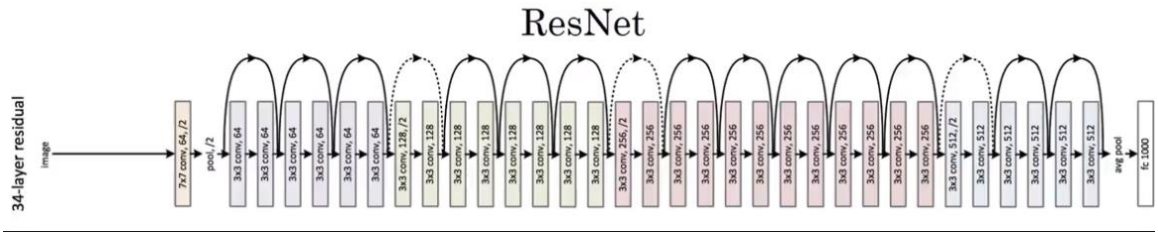


Fig3.17 basic layout of ResNet architecture .

3.9 Performance Metrics for Deep Learning Models:

After doing the usual Feature Engineering, Selection, and of course, implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective is the model based on some metric using test datasets. Different performance metrics are used to evaluate different Machine Learning Algorithms [42].

Definitions:

- True Positives (TP): The total number of accurate predictions that were “positive.”
- False Positives (FP): The total number of inaccurate predictions that were “positive.”
- True Negative (TN): The total number of accurate predictions that were “negative
- False Negative (FN): The total number of inaccurate predictions that were “negative [43].

3.9.1 Training and validation accuracy:

It is the number of correct predictions made by the model over all kind's predictions made?

Accuracy is a good measure when the target variable classes in the data are nearly balanced as shown in fig 3.18 & equation (6) [42].

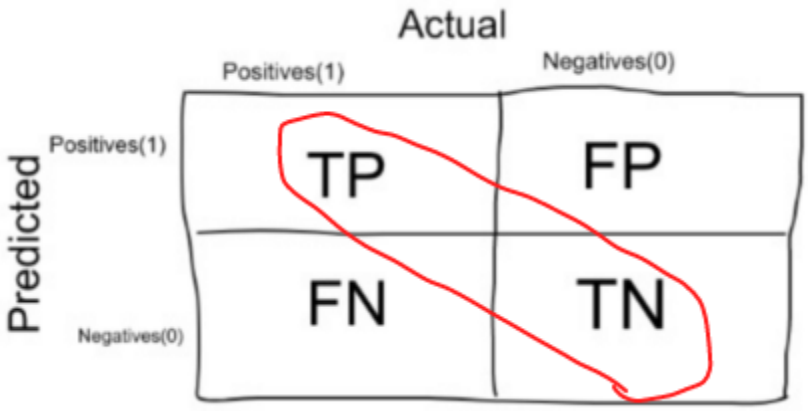


Fig 3.18 accuracy

$$\text{Accuracy} = (TP + TN)/ (TP + FP + TN + FN) \quad (6)$$

3.9.2 Precision:

It is the number of correct positive results divided by the number of positive results predicted by the classifier as shown in fig 3.19& equation (7) [44].

$$\text{Precision} = TP / (TP + FP) \quad (7)$$

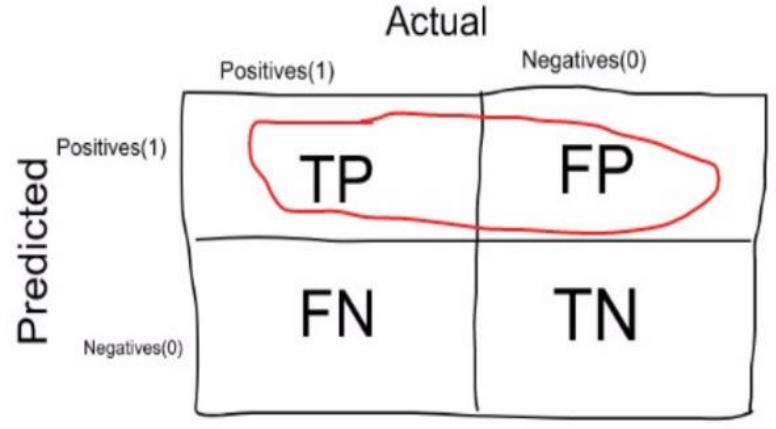


Fig 3.19 precision

3.9.3 Recall:

It is the number of correct positive results divided by the number of **all** relevant samples (all samples that should have been identified as positive) equation (8) [44].

$$\text{Recall} = TP / (TP + FN) \quad (8)$$

3.9.4 F1 Score:

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances) equation (9) [44].

$$(2 * (\text{Precision} * \text{Recall}))/ (\text{Precision} + \text{Recall}) \quad (9)$$

3.9.5 Confusion matrix:

The Confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes [42].

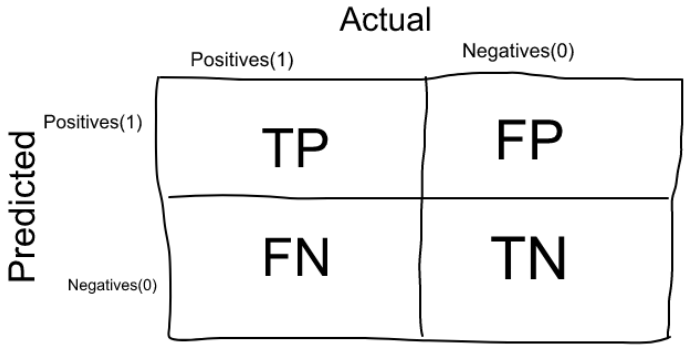


Fig 3.20 confusion matrix

3.9.6 Training and validation Loss:

The lower the **loss,** the better a model . The loss is calculated on **training** and **validation** and its interoperation is how well the model is doing for these two sets. Unlike accuracy, loss is not a percentage. It is a summation of the errors made for each example in training or validation sets.

In machine learning and deep learning there are basically three cases

1) Underfitting

This is the only case where **training loss > validation loss**, but only slightly.

2) Overfitting

Training loss << validation loss

This means that your model is fitting very nicely the training data but not at all the validation data, in other words it's not generalizing correctly to unseen data

3) Perfect fitting

Training loss == validation loss

If both values end up being roughly the same and also if the values are converging (plot the loss over time) then chances are very high that you are doing it right.

3.10 Data for CNN:

The data used to build the final model usually comes from multiple datasets. In particular, three data sets are commonly used in different stages of the creation of the model:

1. **Training dataset :**
   It is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model. The model is trained on the training dataset using a supervised learning method. In practice, the training dataset often consist of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), which is commonly denoted as the target (or label).
2. **Validation dataset :**
   The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters. Validation datasets can be used for regularization by early stopping: stop training when the error on the validation dataset increases, as this is a sign of overfitting to the training dataset .

3. **Test dataset:**
   It is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset. If the data in the test dataset has never been used in training the test dataset is also called a **holdout dataset** [45].

3.11 Hyperparameters optimization in Deep Learning:

Model optimization is one of the toughest challenges in the implementation of machine learning solutions. Hyperparameters are settings that can be tuned to control the behavior of a machine learning algorithm. The number and diversity of hyperparameters in machine learning algorithms is very specific to each model. However, there some classic hyperparameters that we should always keep our eyes on:

1. **Learning Rate:**
   The mother of all hyperparameters, the learning rate quantifies the learning progress of a model in a way that can be used to optimize its capacity.
2. **Number of Hidden Units:**
   A classic hyperparameter in deep learning algorithms, the number of hidden units is key to regulate the representational capacity of a model.
3. **Convolution Kernel Width:**
   In convolutional Neural Networks (CNNs), the Kernel Width influences the number of parameters in a model which, in turns, influences its capacity [46]
4. **Number of epochs**
   Number of epochs is the number of times the whole training data is shown to the network while training.
5. **Batch size**
   Mini batch size is the number of sub samples given to the network after which parameter update happens.
6. **Dropout for regularization**
   Dropout is a preferable regularization technique to avoid overfitting in deep neural networks. The method simply drops out units in neural network according to the desired probability. A default value of 0.5 is a good choice to test with.
7. **Activation functions:**
   Activation functions are used to **introduce nonlinearity** to models, which allows deep learning models to learn nonlinear prediction boundaries [47].
8. **Number of fitters:**
   Is the core hyperparameter for CNN, effects on training parameters and running time.

Chapter 4

Medical applications using CNN

4.1 Overview

Machine learning algorithms have the potential to be invested deeply in all fields of medicine, from drug discovery to clinical decision making, significantly altering the way medicine is practiced. The success of machine learning algorithms at computer vision tasks in recent years comes at an opportune time when medical records are increasingly digitalized [48].

Medical image analysis is an active field of research for machine learning, partly because the data is relatively structured and labeled, and it is likely that this will be the area where patients first interact with functioning, practical artificial intelligence systems. This is significant for two reasons. Firstly, in terms of actual patient metrics, medical image analysis is a litmus test as to whether artificial intelligence systems will actually improve patient outcomes and survival. Secondly, it provides a test bed for human-AI interaction, of how receptive patients will be towards health-altering choices being made, or assisted by a non-human actor [48].

A tremendous interest in deep learning has emerged in recent years. The most established algorithm among various deep learning models is CNN, a class of artificial neural networks that has been a dominant method in computer vision tasks since the astonishing results were shared on the object recognition competition known as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012.

Medical research is no exception, as CNN has achieved expert-level performances in various fields. Models demonstrated the potential of deep learning for diabetic retinopathy screening, skin lesion classification, and lymph node metastasis detection, respectively.

Needless to say, there has been a surge of interest in the potential of CNN among radiology researchers, and several studies have already been published in areas such as lesion detection [49], classification [50], segmentation [51], image reconstruction [52],[53], and natural language processing [54]. Familiarity with this state-of-the-art methodology would help not only researchers who apply CNN to their tasks in radiology and medical imaging, but also clinical radiologists, as deep learning may influence their practice in the near future. Here are some medial image analysis applications using CNN:

4.2 Classifications

In medical image analysis, classification with deep learning usually utilizes target lesions depicted in medical images, and these lesions are classified into two or more classes. For example

4.2.1 Classifications in CT images

Deep learning is frequently used for the classification of lung nodules on computed tomography (CT) images as benign or malignant (**Fig.4.1).** As shown, it is necessary to prepare a large number of training data with corresponding labels for efficient classification using CNN. For lung nodule classification, CT images of lung nodules and their labels (i.e., benign or cancerous) are used as training data. Fig.4.1b show two examples of training data of lung nodule classification between benign lung nodule and primary lung cancer. After training CNN,

the target lesions of medical images can be specified in the deployment phase by medical doctors or computer-aided detection (CADe) systems [55].

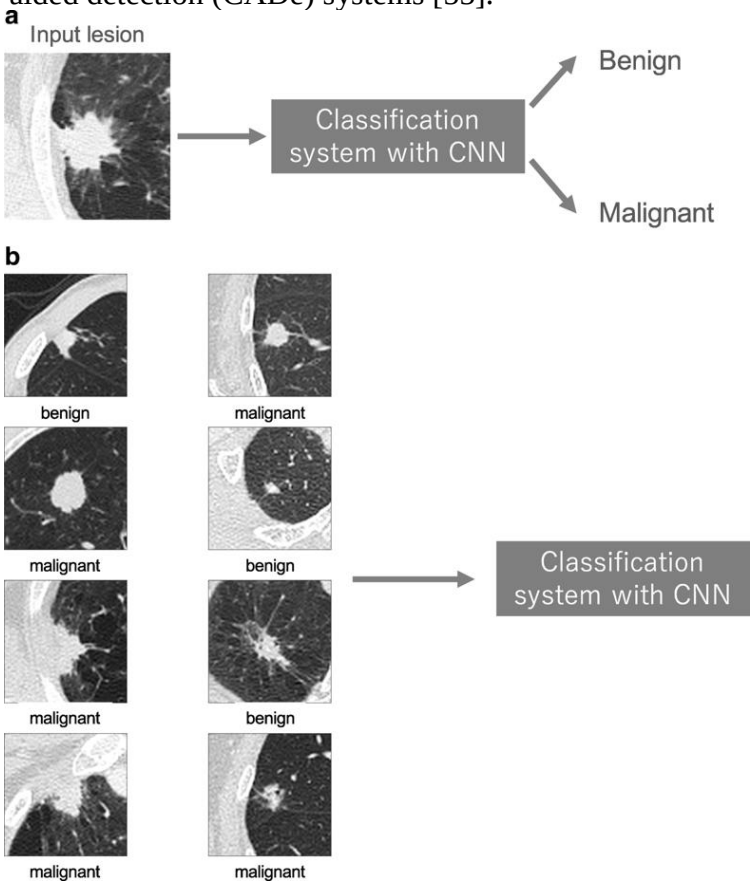


Fig.4.1

A schematic illustration of a classification system with CNN and representative examples of its training data. a) Classification system with CNN in the deployment phase. b) Training data used in training phase

4.2.2 Classification as Computer-Aided Diagnosis (CADx)

Lo et al. described a CNN to detect lung nodules on chest X-rays as far back as 1995 [51]. They used 55 chest x-rays and a CNN with 2 hidden layers to output whether or not a region had a lung nodule. The relative availability of chest x-ray images has likely accelerated deep learning progress in this modality.

Rajkomar et al. [56] augmented 1850 chest x-ray images into 150,000 training samples. Using a modified pre-trained GoogLeNet CNN, they classified the orientation of the images into frontal or lateral views with near 100% accuracy. Although this task of identifying the orientation of the chest x-ray is of limited clinical use, it does demonstrate the effectiveness of pre-training, and data augmentation in learning the relevant image metadata, as part of an eventually fully-automated diagnostic work-flow. Pneumonia or chest infection is a common health-problem world-wide that is eminently treatable.

Rajpurkar et al. [57] employed a modified DenseNet with 121 convolutional layers called CheXNet to classify 14 different diseases seen on the chest x-rays, using 112,000 images from the ChestXray14 dataset. CheXNet achieved state of the art performance in classifying the 14 diseases; pneumonia classification in particular achieved an Area Under Curve (AUC) score of 0.7632 with Receiver Operating Characteristics (ROC) analysis. Moreover, on a test set of 420 images, CheXNet matched or bettered the performance of 4 individual radiologists, and also the performance of a panel comprising of 3 radiologists.

4.2.3 Skin lesion identification

It is a key step toward dermatological diagnosis. When describing a skin lesion, it is very important to note its body site distribution as many skin diseases commonly affect particular parts of the body. A deep multi-task learning (MTL) framework is built to jointly optimize skin lesion classification and body location classification uses the state of the art ImageNet pre trained model with specialized loss functions for the two related tasks [58].

A trained contemporary CNN architecture using a large-scale dataset (23000 images) for diagnosis targeted skin disease classification [59]. A prior art CNN architecture with a relatively small dataset (1300 images) model can classify skin lesions for non dermoscopic images [60].

4.2.4 Diabetic retinopathy (DR)

It can also be diagnosed using CNNs. Using digital photographs of the fundus of the eye, a trained a CNN with 10 convolutional layers and 3 fully connected layers on approximately 90,000 fundus images. They classified DR into 5 clinically used classifications of DR severity, with 75% accuracy [50].

4.2.5 Unsupervised learning methods in MRI

They are also an active area of research using in MRI. Model used Deep Belief Networks to extract features from functional fMRI (fMRI) images, and MRI scans of patients with Huntington Disease and Schizophrenia [63]. Other model classified fMRI images into diagnoses of Healthy or Mild Cognitive Impairment, using a stacked architecture of RBMs to learn hierarchal functional relationships between different brain regions [61]. Looking outside the usual CNN models, a model compared the performance of the well-known CNNs Alexnet and VGGNet to other techniques, namely Bag of Visual Words (BOVW) and Local Binary Patterns (LBP) [62].

4.3 Detection

Detection, sometimes known as Computer-Aided Detection (CADe) is a keen area of study as missing a lesion on a scan can have drastic consequences for both patient and clinician. It is a

common task for radiologists to detect abnormalities within medical images. Abnormalities can be rare and they must be detected among many normal cases

4.3.1 The detection of CT scans

The detection of CT scans cancerous lung nodules on CT lung scans. Approximately 2000 CT scans were used and achieved a logarithmic loss score of 0.399, by using a 3-D CNN inspired by U-Net architecture to isolate local patches first for nodule detection. Then this output was fed into a second stage consisting of 2 fully connected layers for classification of cancer probability Fanzhou [63].

Another model evaluated five well-known CNN architectures in detecting thoraco-abdominal lymph nodes and interstitial lung disease on CT scans Shin et al. [64]. Detecting lymph nodes is important as they can be a marker of infection or cancer. They achieved a mediastina lymph node detection AUC score of 0.95 with a sensitivity of 85% using GoogLeNet, which was state of the art.

4.3.2 Detecting malignant skin cells.

Model used 130,000 dermatological photographs and dermoscopic images to train a GoogLeNet Inception V3 CNN, with no hand-crafting of features. The CNN outperformed human dermatologists in classifying the images as benign, malignant or non-neoplastic lesions, reaching an accuracy of 72% compared to the 65% and 66% accuracies obtained by 2 human dermatologists [65].

The CNN again bettered 21 human dermatologists at deciding treatment plans for two types of skins cancers: carcinoma and melanoma. This task involved 376 biopsy-proven images, and the CNN achieved AUC scores between 0.91 to 0.96.

4.3.3 Histopathological images

Histopathological images are increasingly digitized Currently these images are laboriously read by human pathologists who look for markers of malignancy such as: increased nucleus to cytoplasm ratios, increased numbers of mitotic figures indicating increased cell replication, atypical cellular architecture, signs of cellular necrosis, high cell proliferation index from molecular markers.

A histopathological slide can contain hundreds to thousands of cells, and wading through them at high magnification carries the risk of missing aberrant neoplastic areas. A model used 11–13 layer CNNs to identify mitotic figures in 50 breast histology images from the MITOS dataset. Their approach achieved precision and recall scores of 0.88 and 0.70 respectively [66].

Another model achieved accuracies of 97–98% in classifying kidney cancer histopathological images into tumor or non- tumor, using CNNs that were 5–7 layers deep [67].

Nuclei in breast cancer histological slides were also identified. A model used a Stacked Sparse Autoencoder (SSAE) instead. Their model obtained precision and recall scores of 0.89 and 0.83 respectively, showing that unsupervised learning methods can also be successfully employed in this field [68].

4.3.4 2D CNN on detecting chest radiographs

One previous study investigated the usefulness of 2D-CNN for detecting tuberculosis on chest radiographs [69]. The study utilized two different types of 2D-CNN, AlexNet and GoogLeNet , to detect pulmonary tuberculosis on chest radiographs.By using dataset of 1007 chest radiographs . According to the results, the best area under the curve of receiver operating characteristic curves for detecting pulmonary tuberculosis from healthy cases was 0.99, which was obtained by ensemble of the AlexNet and GoogLeNet 2D-CNNs.

4.4 Segmentation

Convolutional neural networks: an overview and application in radiology

Segmentation of organs or anatomical structures is a fundamental image processing technique for medical image analysis, such as quantitative evaluation of clinical parameters (organ volume and shape) and computer-aided diagnosis (CAD) system.

In the previous section, classification depends on the segmentation of lesions of interest. Segmentation can be performed manually by radiologists or dedicated personnel, a time-consuming process. However, one can also apply CNN to this task as well. **Fig.4.2a** shows a representative example of segmentation of the uterus with a malignant tumor on MRI [51]. In most cases, a segmentation system directly receives an entire image and outputs its segmentation result.

Training data for the segmentation system consist of the medical images containing the organ or structure of interest and the segmentation result; the latter is mainly obtained from previously performed manual segmentation. **Fig.4.2b** shows a representative example of training data for the segmentation system of a uterus with a malignant tumor. In contrast to classification, because an entire image is inputted to the segmentation system, it is necessary for the system to capture the global spatial context of the entire image for efficient segmentation.

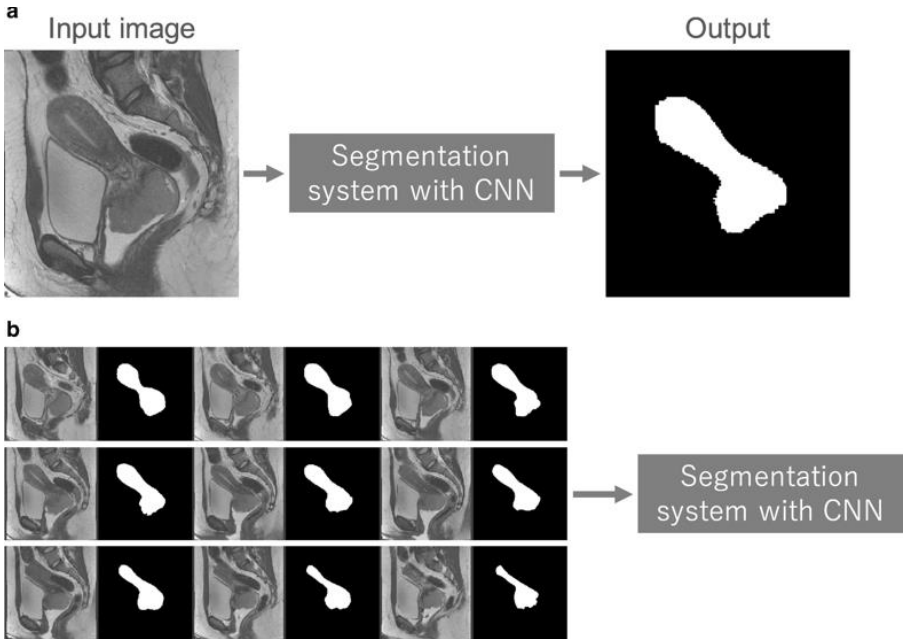


Fig.4. 2
A schematic illustration of the system for segmenting a uterus with a malignant tumor and representative examples of its training data.

a) Segmentation system with CNN in deployment phase. **b**)Training data used in the training phase. Note that original images and corresponding manual segmentations are arranged next to each other

One way to perform segmentation is to use a CNN classifier for calculating the probability of an organ or anatomical structure. In this approach, the segmentation process is divided into two steps; the first step is construction of the probability map of the organ or anatomical structure using CNN and image patches, and the second is a refinement step where the global context of images and the probability map are utilized.

One previous study used a 3D-CNN classifier for segmentation of the liver on 3D CT images [56]. The input of 3D-CNN were 3D image patches collected from entire 3D CT images, and the 3D-CNN calculated probabilities for the liver from the image patches. By calculating the probabilities of the liver being present for each image patch, a 3D probability map of the liver was obtained.

Then, an algorithm called graph cut was used for refinement of liver segmentation, based on the probability map of the liver. In this method, the local context of CT images was evaluated by 3D-CNN and the global context was evaluated by the graph cut algorithm.

One potential approach of using U-net in radiology is to extend U-net for 3D radiological images, as shown in classification. For example, V-net was suggested as an extension of U-net for segmentation of the prostate on volumetric MRI images [70]. In the study, V-net utilized a

loss function based on the Dice coefficient between segmentation results and ground truth, which directly reflected the quality of prostate segmentation[71].

Another study [72] utilized two types of 3D U-net for segmenting liver and liver mass on 3D CT images, which was named cascaded fully convolutional neural networks; one type of U-net was used for segmentation of the liver and the other type for the segmentation of liver mass using the liver segmentation results. Because the second type of 3D U-net focused on the segmentation of liver mass, the segmentation of liver mass was more efficiently performed than single 3D U-net.

CT and MRI image segmentation research covers a variety of organs such as liver, prostate and knee cartilage, but a large amount of work has focused on brain segmentation, including tumor segmentation. The latter is especially important in surgical planning to determine the exact boundaries of the tumor in order to direct surgical resection. Sacrificing too much of eloquent brain areas during surgery would cause neurological deficits such as limb weakness, numbness and cognitive impairment.

Traditionally, medical anatomical segmentation was done by hand, with a clinician drawing outlines slice by slice through an entire MRI or CT volume stack; therefore it is ideal to implement a solution that automates this laborious task. Pereira et al. [73] applied deliberately small filters of 3×3 size, to allow the design of a deeper 11 convolution layer CNN, and to reduce overfitting. Their CNN was trained on 274 MRI brain tumor scans of gliomas, a type of brain tumor with significant malignant potential, obtaining first place in the BRATS 2013 and second place in the BRATS 2015 challenge.

Havaei et al. [74] also looked at gliomas, and explored various 2-dimensional CNN architectures on the BRATS 2013 dataset. Their algorithm performed better than the BRATS 2013 winner, and took 3 minutes to run, compared to 100 minutes. Their Input Cascade CNN had a cascaded architecture, with the output of a first CNN being fed into a second CNN.

4.5 Localization

Localization of normal anatomy is less likely to interest the practicing clinician although applications may arise in anatomy education. Alternatively, localization may find use in fully automated end-to-end applications, whereby the radiological image is autonomously analyzed and reported without any human intervention.

Another looked at transverse CT image slices and constructed a two stage CNN where the first stage identified local patches, and the second stage discriminated the local patches by various body organs, achieving better results than a standard CNN [75].

4.5.1 Localization on CT images

A model trained a CNN with 5 convolution layers to discriminate approximately 4000 transverse axial CT images into one of 5 categories: neck, lung, liver, pelvis, legs. He was able to achieve a

5.9% classification error rate and an AUC score of 0.998, after data augmentation techniques Roth et al. [76].

4.5.2 Localization on MRI images

Other used stacked autoencoders on 78 contrast-enhanced MRI scans of the abdominal region containing liver or kidney metastatic tumors, to detect the locations of the liver, heart, kidney and spleen. Hierarchal features were learned over the spatial and temporal domains, giving detection accuracies of between 62% and 79%, depending on the organ Shin et al. [77].

4.6 Registration

Image registration is the process of aligning two or more images of the same scene. This process involves designating one image as the reference image, also called the fixed image, and applying geometric transformations or local displacements to the other images so that they align with the reference. Images can be misaligned for a variety of reasons. Commonly, images are captured under variable conditions that can change the camera perspective or the content of the scene. Misalignment can also result from lens and sensor distortions or differences between capture devices.

4.6.1 Image registration in is neurosurgery

Image registration is applied in neurosurgery or spinal surgery, to localize a tumor or spinal bony landmark, in order to facilitate surgical tumor removal or spinal screw implant placement. A reference image is aligned to a second image, called a sense image and various similarity measures and reference points are calculated to align the images, which can be 2 or 3-dimensional. The reference image may be a pre-operative MRI brain scan and the sense image may be an intraoperative MRI brain scan done after a first-pass resection, to determine if there is remnant tumor and if further resection is required.

4.6.2 Registration on MRI images

Using MRI brain scans from the OASIS dataset, a model stacked convolution layers in an encoder-decoder fashion, to predict how an input pixel would morph into its final configuration [78]. They invoked the use of a large deformation diffeomorphic metric mapping (LDDMM) registration model.

4.6.3 Registration on X rays images

Another trained a 5 layer CNN on synthetic X-ray images in order to register 3-dimensional models of a knee implant, a hand implant, and a trans-esophageal probe onto 2-dimensional X-ray images, in order to estimate their pose [79]. Their method obtained successful registrations 79–99% of the time, and took 0.1 seconds, a significant improvement over traditional intensity-based registration methods.

Chapter 5

System description

5-1 Data description

The main problem that bioinformatics researchers face when finding solutions for detecting and diagnosing Leukemia diseases is a lack of dataset because medical images are private. Besides, the more Leukemia images CNN can handle, the higher accuracy achieved. Therefore, the need for a large enough dataset to build an effective CNN architecture in the diagnosis of Leukemia is extremely urgent.

Images that had been used in this project were downloaded from internet and are available in ALL IDB[80], ASH Image Bank Hematology[81], Stock photo, vectors and Royalty free Images[82], Shutter stock[83], Atlas of Hematology [84], Atlas of blood smear analysis[85], Blue Histology and American society of Hematology [86], This dataset is composed of 630 image , contains 480 cancer images and 150 normal images .

5.2 Data preprocessing

5.2.1 Remove duplication

As dataset is collected from various resources, we found that there are some repetitions, some images contain a watermark and other contains websites 's logo totally about 43 images, so now data set is become 587 images.

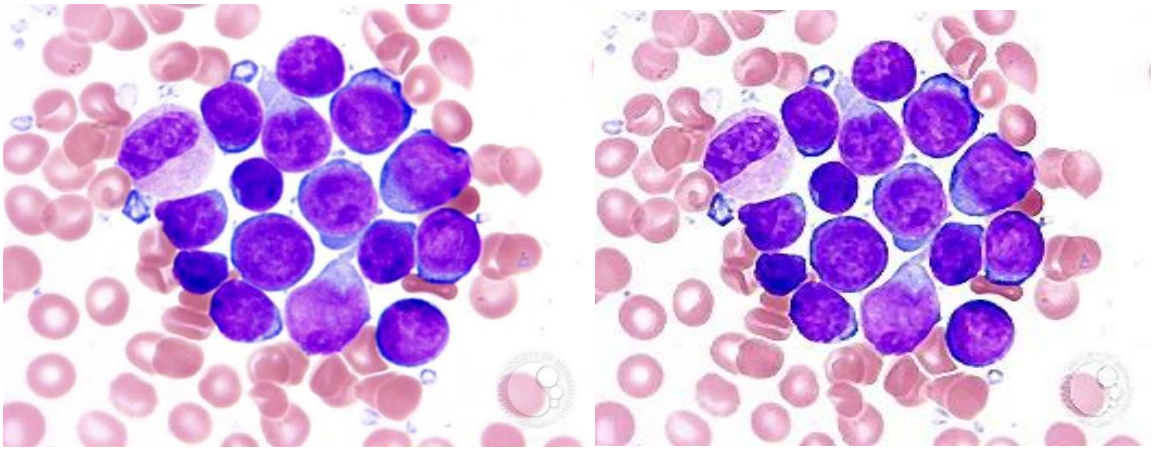
5.2.2 Resizing of images

As dataset has different distribution of size, and for the purpose of training CNN model it was needed to make all images in dataset has the same size, so we had applied a resizing technique and make all image 128 x 128 pixels to reduce the training time. as shown in figure (5.1)

5.2.3 Filtering images

Before processing stage, we need to remove noise and enhance line structures in images and this is available by applying median filter (3 x3) and sharpening image (3 x3) using equation:

Sharp filter = resized image - .9 * Laplacian image (1) [87] as shown in figure (5.1)



(a) (b)

Fig 5.1: Image enhancement process (a) original image (size 512*512), (b) image resized by 128*128 and filtered by median and sharpen filters.

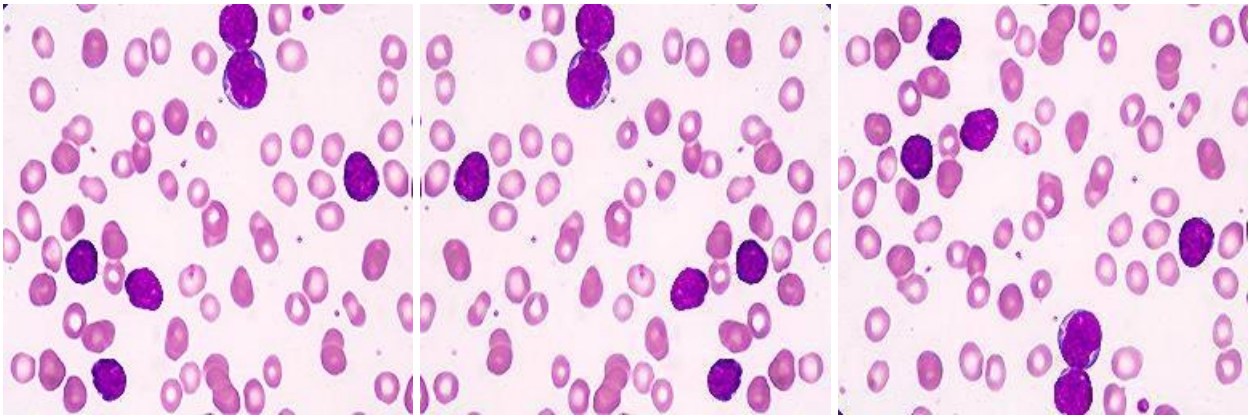
5.2.4 Data augmentation

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset [88]. Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class [89].

In this project we use different types of augmentation techniques:

5.2.4.1 Flipping

An image flip means reversing the rows or columns of pixels in the case of a vertical or horizontal flip respectively [83] . (As shown in figure5.2)

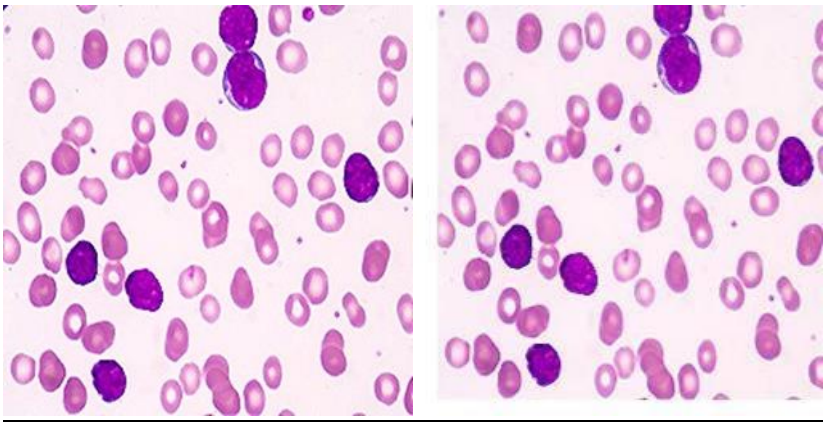


(a) (b) (c)

Fig.5.2: a) original image, b) horizontal flipping ,c) vertical flipping.

5.2.4.2. Horizontal and Vertical Shift Augmentation

A shift to an image means moving all pixels of the image in one direction, such as horizontally or vertically, while keeping the image dimensions the same. This means that some of the pixels will be clipped off the image and there will be a region of the image where new pixel values will have to be specified [90]. (as shown in figure 5.3)

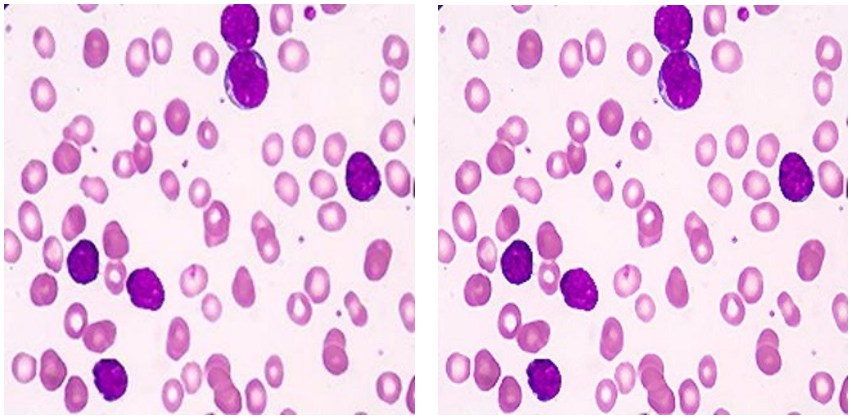


(a) (b)

Fig.5.3: a) original image, b) width shifted image

5.2.4.3 Random Zoom Augmentation

A zoom augmentation randomly zooms the image in and either adds new pixel values around the image or interpolates pixel values respectively [90]. (as shown in figure 5.4)

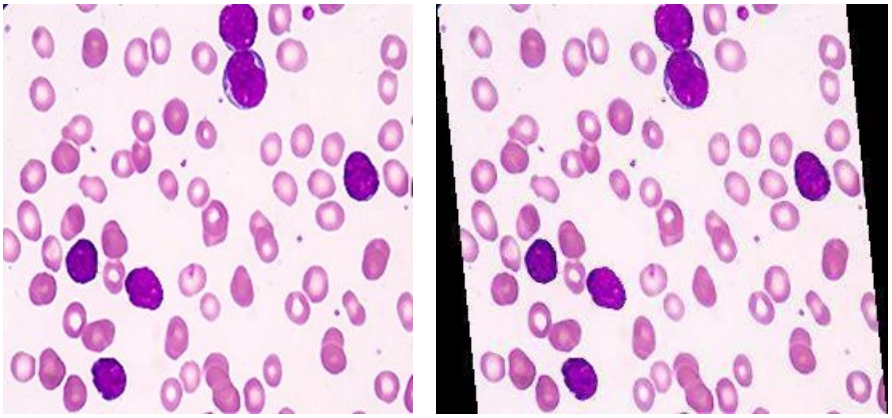


(a) (b)

Fig.5.4: a) original image, b) zoomed image

5.2.4.4 Shearing

Shearing will automatically crop the correct area from the sheared image, so that we have an image with no black space or padding [91]. (as shown in figure 5.5)



(a) (b)

Fig.5.5: a) original image, b) sheared image

5.2.4.5 Interpolation (nearest)

Interpolation is a method of constructing new data points within the range of a discrete set of known data points [92]

Nearest neighbor interpolation is the simplest approach to interpolation. Rather than calculate an average value by some weighting criteria or generate an intermediate value based on complicated rules, this method simply determines the "nearest" neighboring pixel, and assumes the intensity value of it [93]. (As shown in figures 5.2, 5.3, 5.4, and 5.5).

After augmentation processes our data become 1900 images for cancer and 1568 for normal. To fit data to models we divided it through coding into three data sets: training set, validation set and test set by ratios 60%, 20% and 20% respectively.

5.3 processing stage

In order to get high accuracy as possible we worked in different model depending on transfer learning principle. Transfer learning is a machine learning technique in which a model trained for some specific task can be used to learn new task by transfer of knowledge. This technique can be effective, fast, and convincing, when we don't have enough data to train the model from scratch.

5.3.1. Basic CNN model

We designed a model where the input images are (RGB) color images with resolution of 128x128 pixels. It consists of 3 convolutional layers with max polling layers. Each convolutional layer is followed by rectified linear unit (relu). We used a constant filter size(3x3), the number of filters (128) and stride of ones (equal 1) and fully connected layer trained for two categories classification using sigmoid activation function. Where we classified the data set into leukemia cell or normal cell. The figure 5.6 below indicates the block diagram of this model.

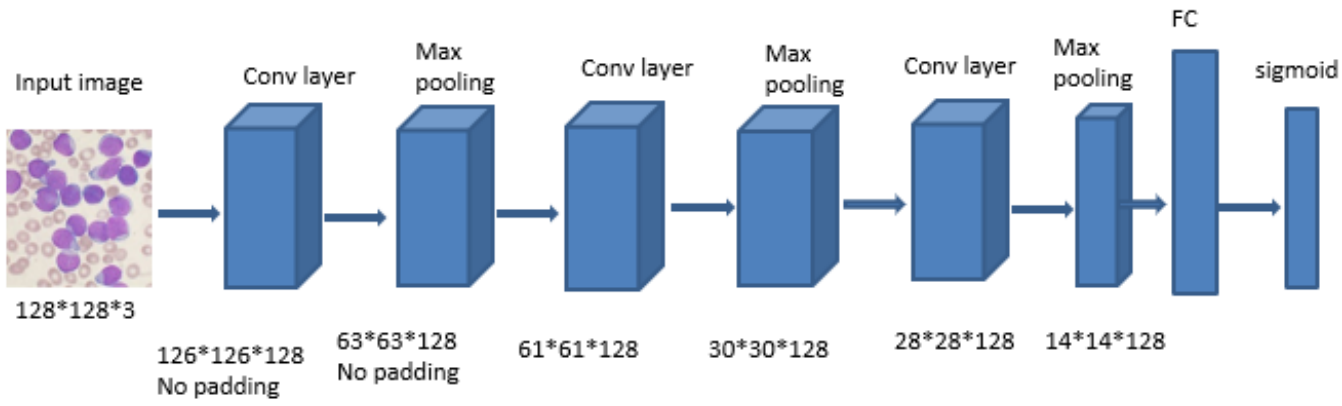


Figure 5.6: Model's block diagram

5.3.2. Alex net architecture

In this study, we deployed the pretrained AlexNet .This architecture was proposed by Krizhevsky et al,9 who deployed this architecture for ImageNet Large Scale Visual Recognition Challenge 2012,20 and won the challenge in first place. input images were Red Green Blue (RGB) color images with resolution of 227 x 227 pixels. It consists of 5 convolutional layers with 3 max polling layers. Each convolutional layer in AlexNet architecture is followed by rectified linear unit (ReLU). All the parameters including the filter size, the number of filters,

and stride for each layer are illustrated in Figure 5.7, we replace SoftMax layer with a sigmoid layer as we want to classify the input image into only two type.

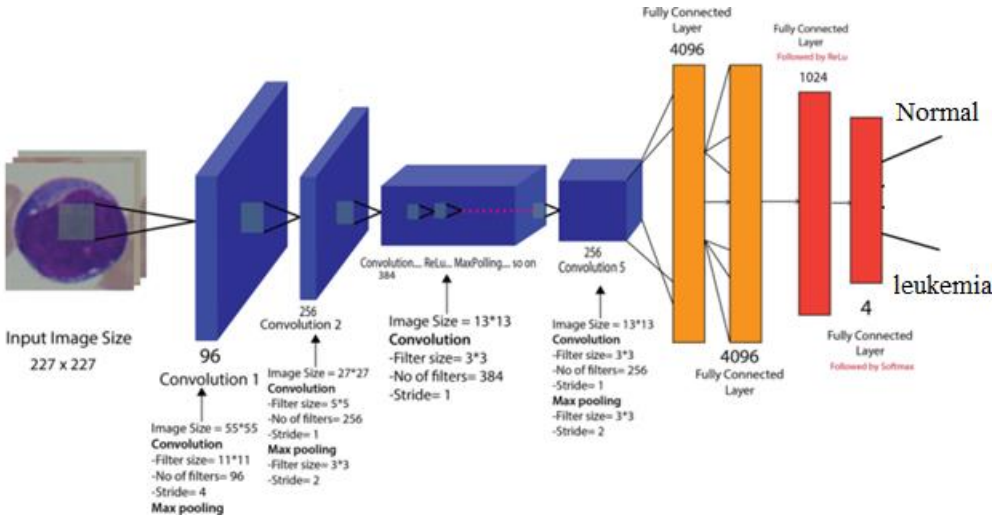


Figure 5.7: AlexNet architecture for leukemia detection.

5.3.3. Modification of model used in Thanh et al paper

In this model we used a retrained model that had been used in a published paper [94] shown in figure 5.8, and we changed values of hyperparameter to become as shown in figure 5.9, In this work we use a network containing 7 layers. The first 5 layers perform feature extraction and the other 2 layers (fully connected and SoftMax) classify the extracted features. The input image has the size of 128x128x3. In the convolution layer 1, we used a constant filter size of 5x5 and a total of 16 different filters. The stride is 1 and no zero-padding was applied. The second and third convolution layers have the same structure with the first one but different number of filters, 32 and 64, respectively. We used pooling layer with filter size 2, stride 2 to decrease the volume spatially. During the learning, the chosen size of the mini batch was 100. REL is used as the activation function.

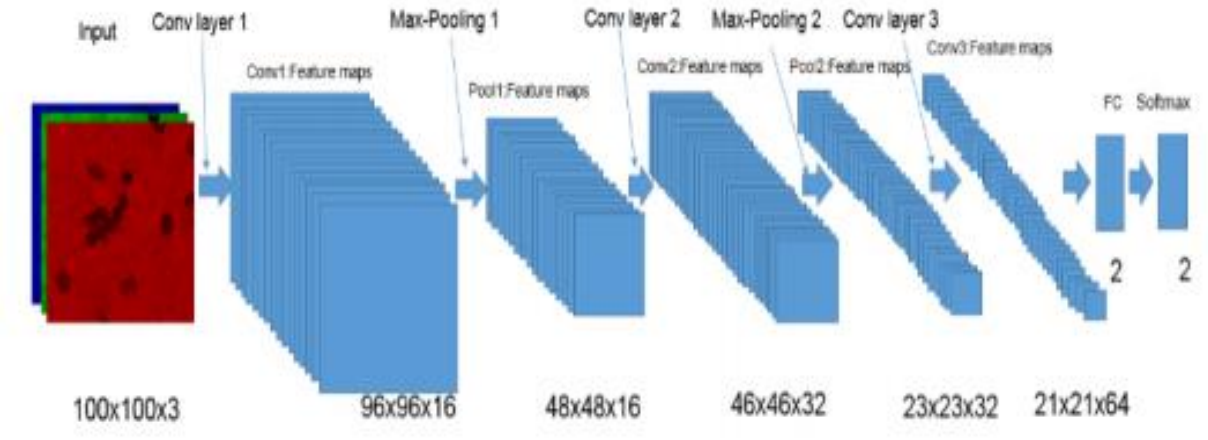


Figure 5.8: the original architecture of CNN in the mentioned paper.

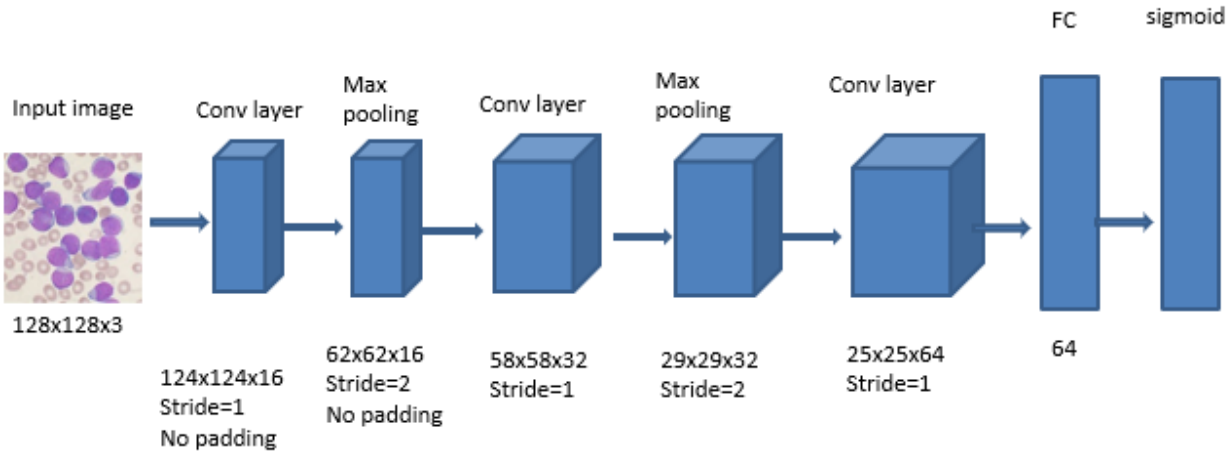


Figure 5.9: architecture of CNN after changes in hyperparameter

5.4 Graphical user interface (GUI):

We made a user-friendly GUI to assist physicists perform classification with few clicks in buttons; this GUI is performed by programming language python using Tkinter (TK) module.

Out of all the GUI methods, TK is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with TK outputs the fastest and easiest way to create the GUI applications [95].

5.4.1 Advantages of TK [96]:

a) It's simple to learn.

b) Bundled with Python.

c) Highly portable.
d) Can look [kind of] native.
e) It's fast enough.
f) Mature and stable
g) Free for commercial use

5.4.2 Our implementation in tkinter

We made a user-friendly GUI to assist physicists perform classification with few clicks in buttons. Our GUI consists of 3 windows which are connected with each other by buttons

5.4.2.1 Login window

Login window is the First window, which used for logging in our application, makes it secure and even difficult for anyone to access it except the user of this program. Log in window consists of 2 connected windows, if this is the first time to use the program, the user will select creating an account and input user name and password which will be saved in the sqlite3 program as database as shown in fig.5.10, If user has an account, he should enter name and password, and the program will compare these data with the stored data. If there is an error, a message error will appear, and if it is not, it will be allowed to access the program easily as shown in fig.5.11.

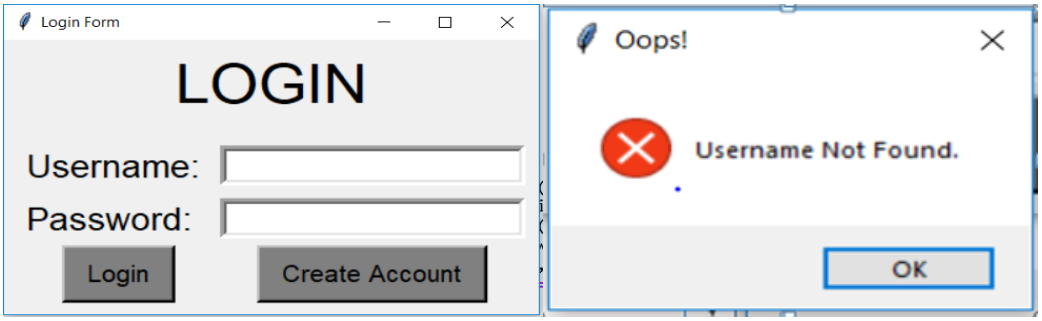


Fig.5.10: login window and a message error

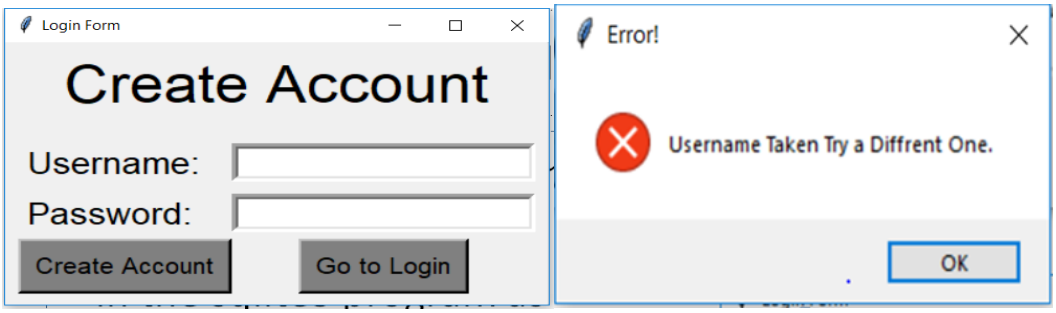


Fig.5.11: create account window and a massage error

5.4.2.2 Leukemia Detection and classification window:

Leukemia Detection and classification window is the second window. It consists of 6 parts to make software user friendly as shown in fig.5.12.

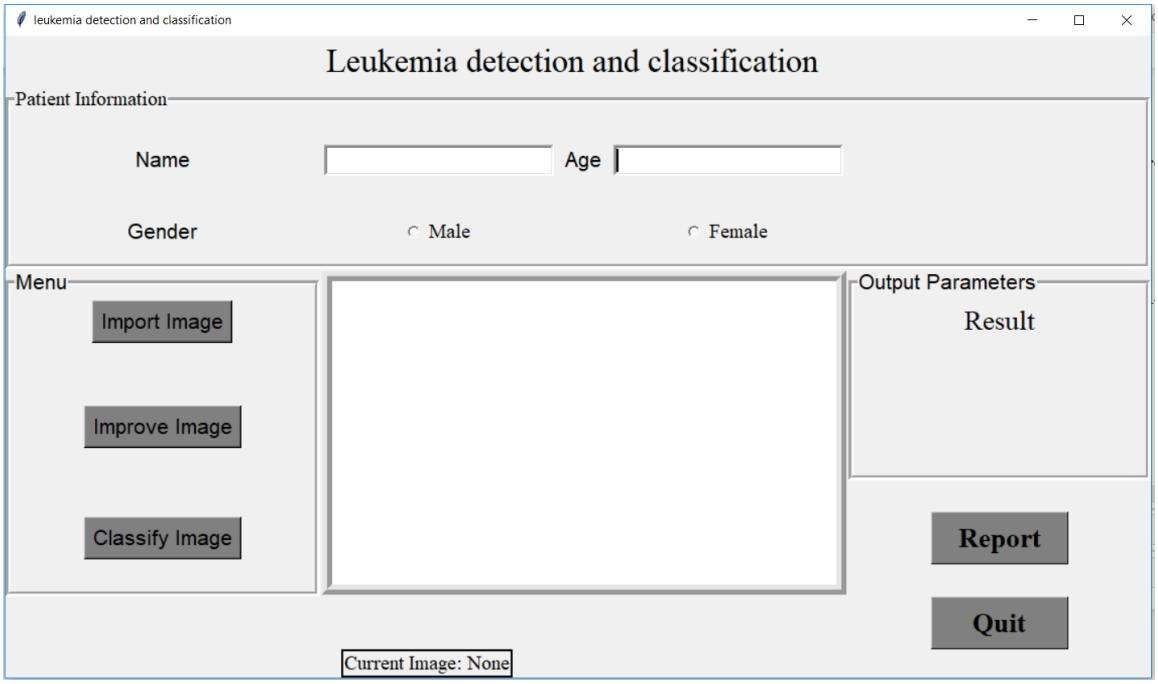


Fig 5.12: Leukemia Detection and classification window

Main window contains many parts:

1-Patient information part:

Which includes important data about patient as: name, age, gender. These data are stored in sqlite3 program as database, allow to be used again.

2-Menu part:

Which include some buttons:

a) Import image button opens a file dialog to allow user select the image he wants to classify, when clicking in import button, user dialog is opened as shown in figure 5.13.

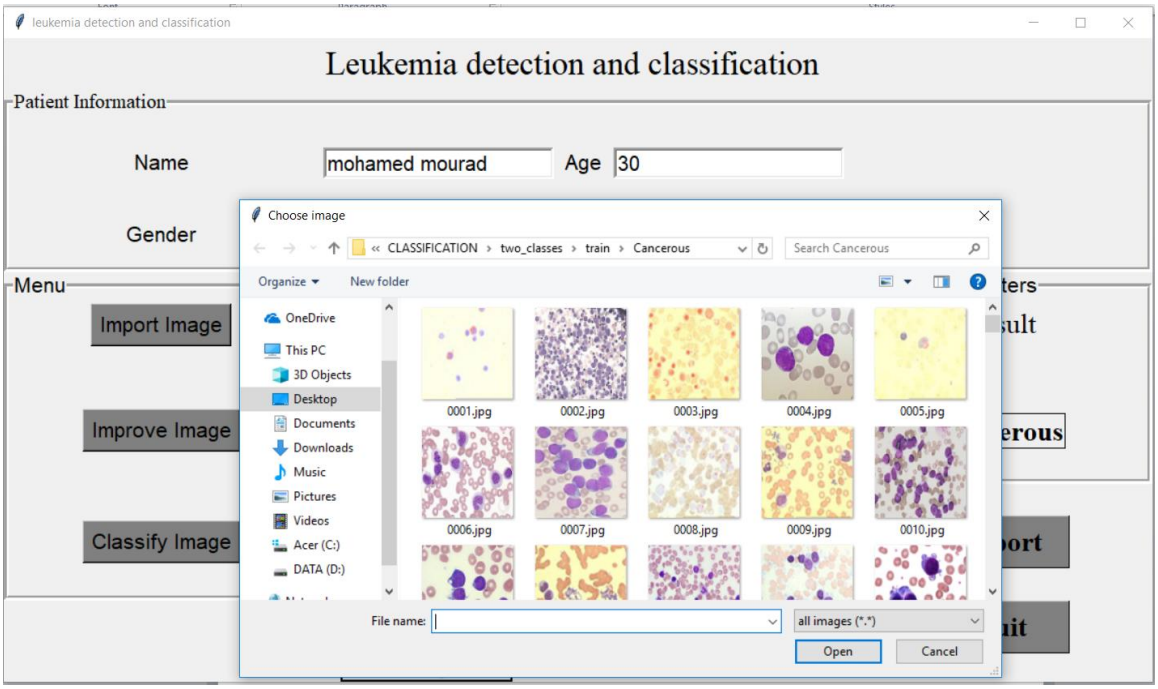


Fig.5.13: when clicking in import button, a user dialog is opened

b) Improve image button has a set of filters which is carefully chosen after many attempts to improve the quality of the image as shown in fig.5.14, if user click improve button without choosing an image, a massage error is appeared as shown in fig 5.15.

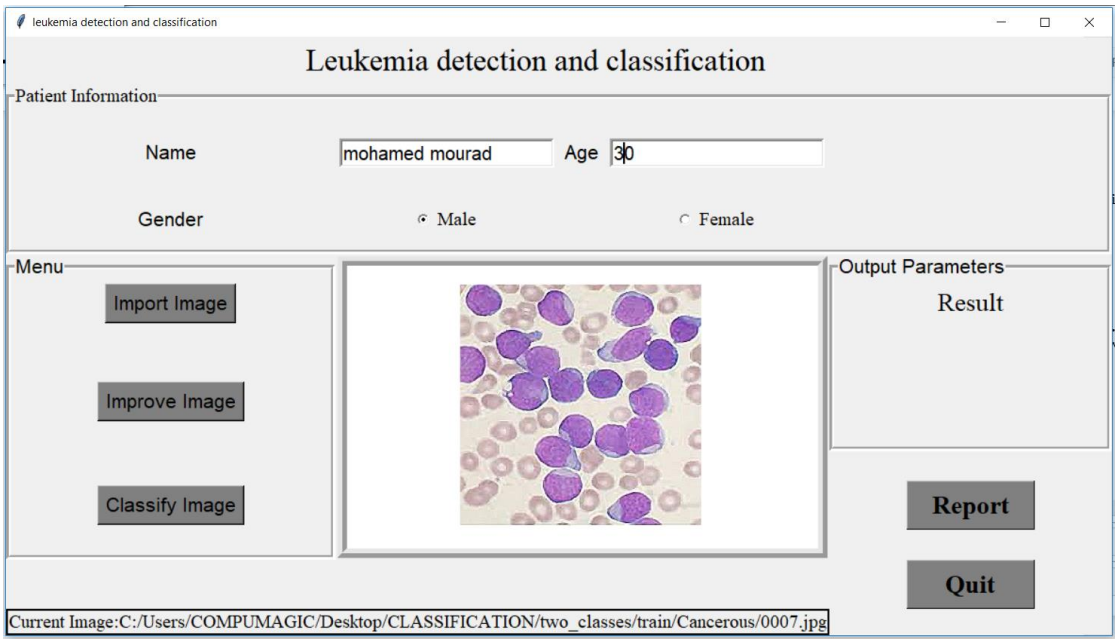


Fig.5.14 : when clicking on improve image

c) Classify image button, it has a convolution neural network architect model with accuracy reaches 98% to classify image as normal or abnormal according to exist of leukemia in it as shown in fig.5.15, if user click classify button without choosing an image, a massage error is appeared as shown in fig 5.16.

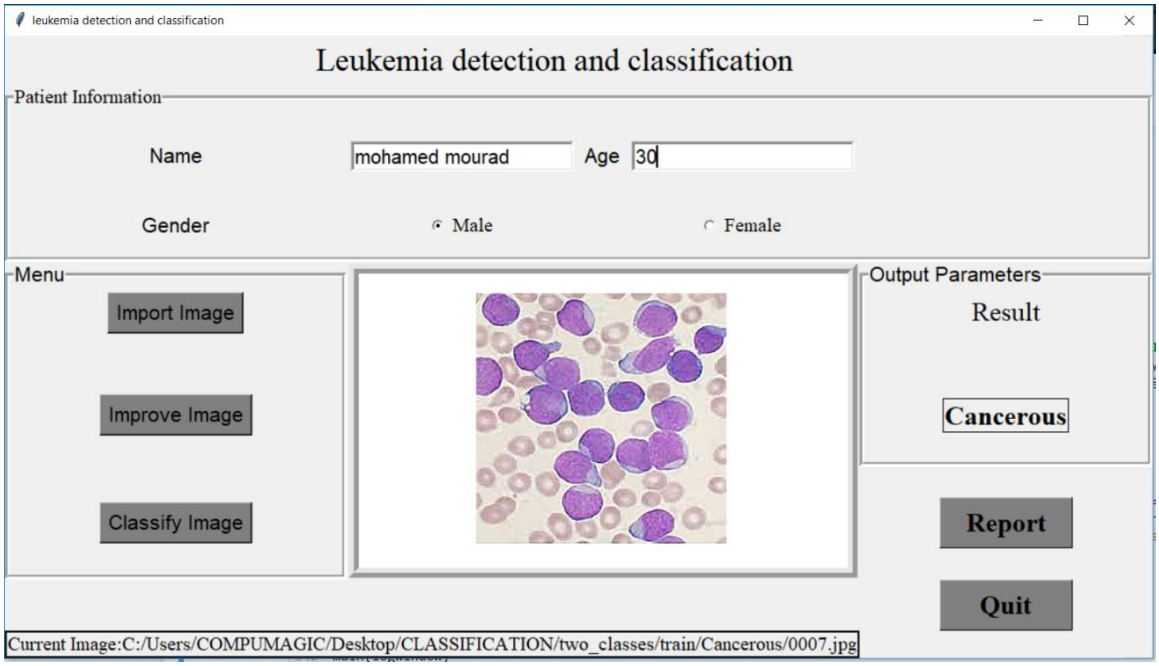


Fig.5.15, When clicking on classify button.

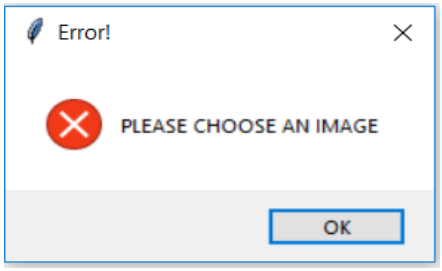


Fig.5.16 : Massage error when click improve or classify button without choose image

3- Image part, a place where the image is shown after importing or filtering it.

4- Output parameter; include the result of classifying the image as normal or cancerous.

5- Quit button, to close the program, a massage confirmation is appeared when clicking quit button as shown in fig.5.17.

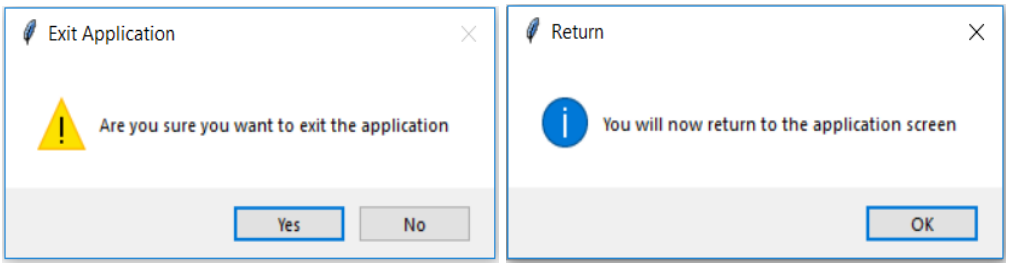


Fig.5.17, Massage confirmation when click quit button

6- Report button, which connect the second window with the third one

5.4.2.3 Report window

Report window is the third window, which presents summary of second window. It includes patient name, age and result. Make it easier for technical printing it so patient can present it to doctor who observe his condition as shown in fig.5.18.

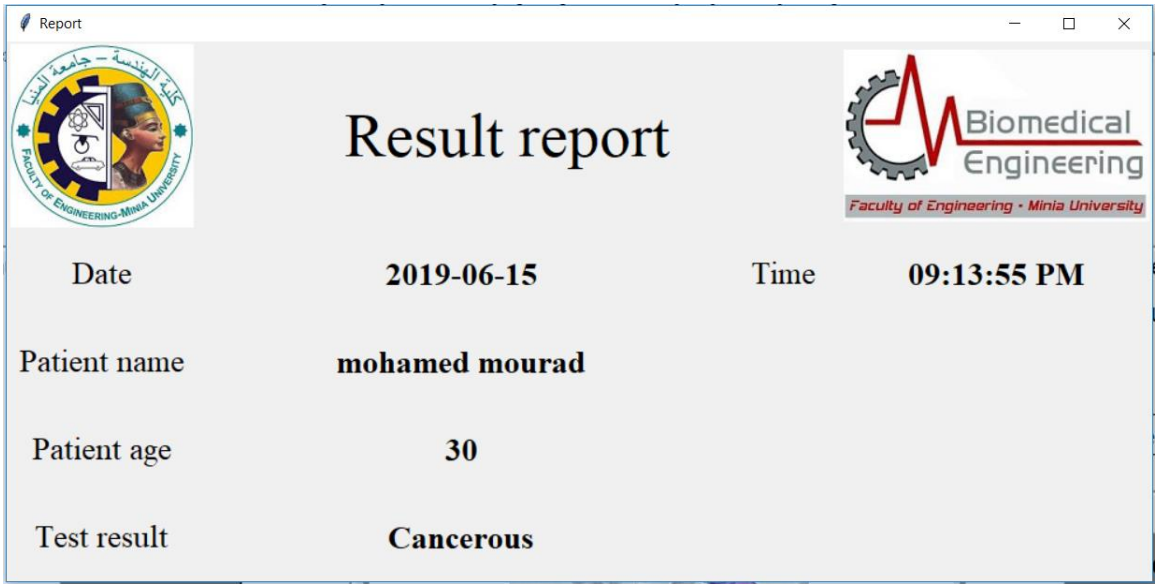


Fig.5.18 Report window

5.4.3 Widget used in Tkinter

There's widget we used and the usage of them:

1) The **Label Frame** widget is a variant of the Tk Frame widget. By default, it draws a border around its child widgets, and it can also display a title. We used it to group a number of related widgets, such as radio buttons, labels and buttons
2) The **Label** widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want [97]. We used it to display some static text as name, age, gender, result, title, username and password.
3) The **radio button** widget implements a multiple-choice button, which is a way to offer many possible selections to the user and lets user choose only one of them [98]. We used it in gender to select one on male or female.
4) The **Button** widget can contain text or images, and you can associate a Python function or method with each button. When the button is pressed, Tk automatically calls that function or method [99]. We used it in import image, improve image, classify image and quit buttons.
5) The **entry** widgets are the basic widgets of Tk used to get input, i.e. text strings, from the user of an application. This widget allows the user to enter a single line of text. If the user enters a string, which is longer than the available display space of the widget, the content will be scrolled. This means that the string cannot be seen in its entirety. The arrow keys can be used to move to the invisible parts of the string. If you want to enter multiple lines of text, you have to use the text widget. An entry widget is also limited to single font [100]. we used it in first and second window.

5.5 Hardware:

We employ a Ceti Vulcan LED Compound Microscope exist in our laboratory to be a digital one; using simpler component and lower costs. Microscope now can be used in different studies and researches related to histology images. Images taken from the microscope could be saved in micro card or could be sent into a computer by USB to be processed. Our hardware consists of a microscope and a magnification camera connected to computer by an USB. We utilize it as an

acquisition tool to capture images of leukemia and normal blood smear then send them to computer so image can be classified by our software program to be cancerous or normal.

5.5.1 Ceti Vulcan LED Compound Microscope

It is the microscope exists in our lab shown in fig. 5.19.

Fig.5.19: Ceti Vulcan LED Compound Microscope.

5.5.1.1 Microscope features [101]:

(a) Mechanical stage with low X/Y controls and Gorilla Glass insert for smooth operation.
(b) Super wide-field 10X, FN22 eyepieces for maximum field of view .
(c) Quintuple, reversed nosepiece with five Infinity Corrected N-Plan Achromatic objectives.
(d) Ergonomic, tilting trinocular eyepiece tube for comfortable viewing over long periods.
(e) Automatic Illumination System with auto intensity control on each objective.
(f) Full Köhler illumination and a wide range of accessories.

5.5.1.2 Microscope specifications [101]:

Ceti Vulcan LED Compound Microscope specifications are shown in table.5.1.

Table .5.1: Ceti Vulcan LED Compound Microscope specifications

Product Code	3200.9100LED	3100.9300LED
Optical Head	Ergonomic tilting trinocular with photo port, adjustable from 0° to 35°	
	3-position beam splitter; 0:100, 100:0, 80:20	
	Interpupillary Distance Adjustment: 47 to 78mm	
	Dioptric adjustment on both eyepieces	
Eyepieces	10x/22mm, super wide-field	
Nosepiece	Revolving sextuple nosepiece on ball-bearings with 25mm objectives mounting threads	Motorised with remote switching pad on base
		Nosepiece rotating buttons behind focus knobs
Objectives	4x, 10x, 20x, 40x, 100x oil Infinity Corrected N-Plan Achromatic	
Condenser	In height adjustable N.A. 0.90/1.25 Abbe swing-out condenser with numerical aperture identification marks (for brightfield)	Pre-condenser with field diaphragm
	In height adjustable N.A. 1.25 Zernike phase contrast condenser with phase annuli for 10/20/40x and S100x-oil objectives (for phase contrast)	Motorised Abbe - NA 1.25 with automatic swing-out when 4x objective is selected
	-	Focusable and centrable, iris diaphragm
Stage	Double layer mechanical stage, 190 x 152mm with Gorilla Glass Insert and slide holder	
	78 x 32mm movement, Vernier scale	
Focusing Knobs	Low position coaxial coarse and fine with stop	
	Fine focus knobs can be swapped	
	Tension control	
Mains Power	220 - 240V/50 - 60Hz	
Illumination	3W S-LED with Fly-lens, intensity control and Automatic Illumination System (AIS)	
	ECO function switches illumination off after 30 minutes non-use	
Supplied With	Power cord, microscope oil, and cover	

5.5.2 Koolertron 4.3" LCD Digital USB Microscope Magnifier

It is LCD Digital USB Microscope Magnifier with Adjustable Stand 1-600X Continuous Magnification Zoom,8 LED Adjustable Light Source, Rechargeable Lithium Battery, Camera and Video Recorder as shown in fig.5.20.

Fig.5.20: Koolertron 4.3" LCD Digital USB Microscope Magnifier

5.5.2.1 LCD Digital USB Microscope Magnifier Features [102]:

(a) Adjustable Stand.
(b) Pixel: HD 3 .6MP CCD.
(c) Display: 4.3 inches HD OLED display .
(d) Digital zoom: 1-600X continuous magnification system (HDdisplay 22 inches).
(e) Object distance: 2cm-10cm .
(f) Photo resolution：5M,2M,1.3M.
(g) Video resolution：640x480With built-in lithium battery, continuous working for 6 hours
(h) Languages: English, Spanish, Russian, Korean, Japanese, Thai, Hebrew, Portuguese, German, French, Italy, Turkey, Czech, Poland, Traditional Chinese, simplified Chinese
(i) Power DC interface (5P MINI).
(j) Memory card's socket (Micro SD), supporting 1-64GB (not include).
(k) Brightness Adjustment button.
(l) High brightness 8 LEDs, available for up to 100 thousand hours.

5.5.2.2 LCD Digital USB Microscope Magnifier Specifications [102]:

Koolertron 4.3" LCD Digital USB Microscope Magnifier specifications are shown in table5.2.

Table5.2: Specifications of Koolertron 4.3" LCD Digital USB Microscope Magnifier

Package Dimensions	8.4 x 6.8 x 3.4 inches
Item Weight	1.42 pounds
Shipping Weight	1.42 pounds
ASIN	B07FRRTD9L
Item model number	AS-SMXW31
Batteries	1 CR2 batteries required.

5.5.3 Project hardware

Our hard ware implementation as shown in fig.5.21 and 5.22.; consists of 2 microscopies which are connected to labtop with an USB, image taken from microscope sent to labtop to the GUI in order to be classified as normal or abnormal image.

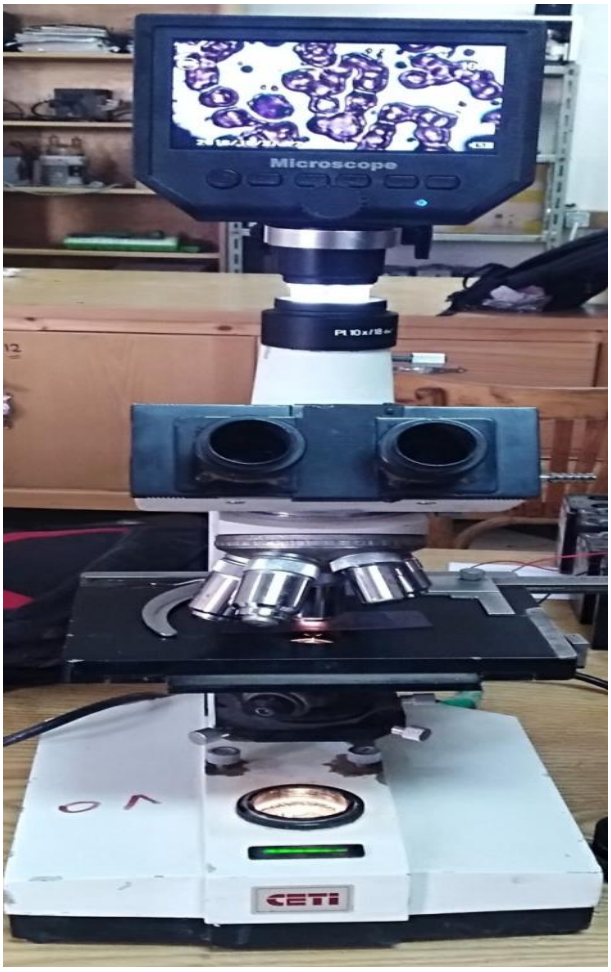


fig.5.21: Final project hardware body

Fig.5.22:Our hardware implementation

Chapter 6

Result

6.1 Measured parameters

In order to evaluate each model and clarify the best one; we compare between them by some measured parameters. Our measured parameters are:

6.1.1Accuracy

Accuracy is the ratio of number of correct predictions to the total number of input samples. In our models accuracy is the optimizing parameter.

6.1.2 Train accuracy

For basic CNN model train accuracy comes out to 90.99% after 17 epochs; that means our leukemia classifier is doing a good job of classification as shown in fig.6.1a. For AlexNet architecture, accuracy achieved its maximum accuracy of 56% after 11 epochs; that mean our model is terrible on leukemia classification as shown in fig.6.1 b but Modification of model used in Thanh et al paper [94] achieved the maximum accuracy over all models 97.73 % after 10 epochs as shown in fig.6.1c.

6.1.3 Validation accuracy

For basic CNN Model validation accuracy comes out to 85% after 17 epochs; as shown in fig.6.1a. This means that we expect our model to perform with ~85% accuracy on new data. For AlexNet architecture, accuracy achieved its maximum accuracy of 53.6% after 11 epochs; that mean our model is bad on leukemia classification as shown in fig.6.1 b; This means that we expect our model to perform with ~53.6% accuracy on new data. But in Modification of model used in Thanh et al paper [94] achieved the maximum validation accuracy over all models 94.3 % after 10 epochs as shown in fig.6.1c. This means that we expect our model to perform with ~94.3 % accuracy on new data.

We notice that as epochs increase, your train metric increases, while validation accuracy metric decreases. This means that our model is fitting the training set better, but slightly losing its ability to predict on new data, indicating that ours models are beginning to overfitting.

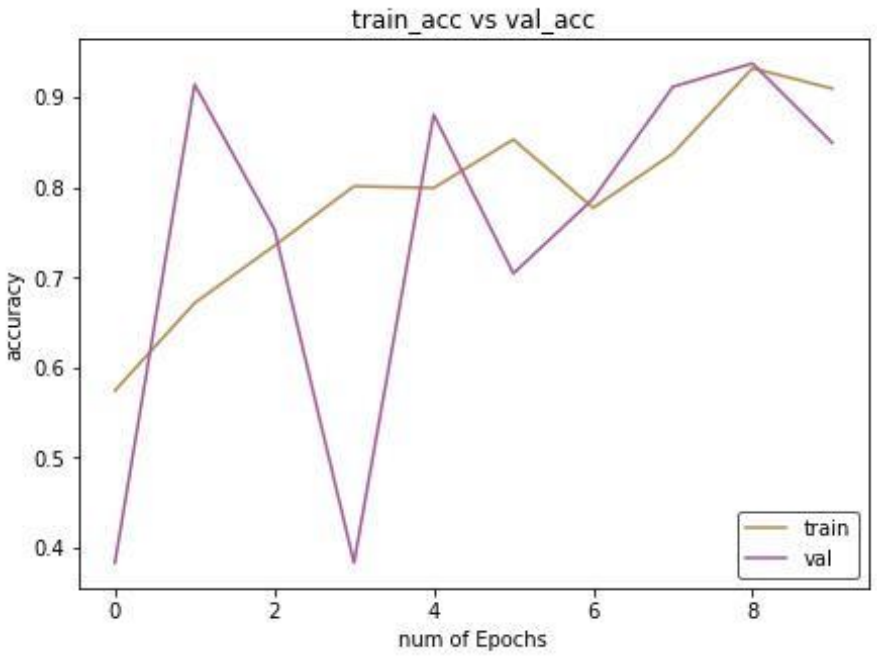


Fig.6.1a, curve of validation accuracy & train accuracy for basic CNN model

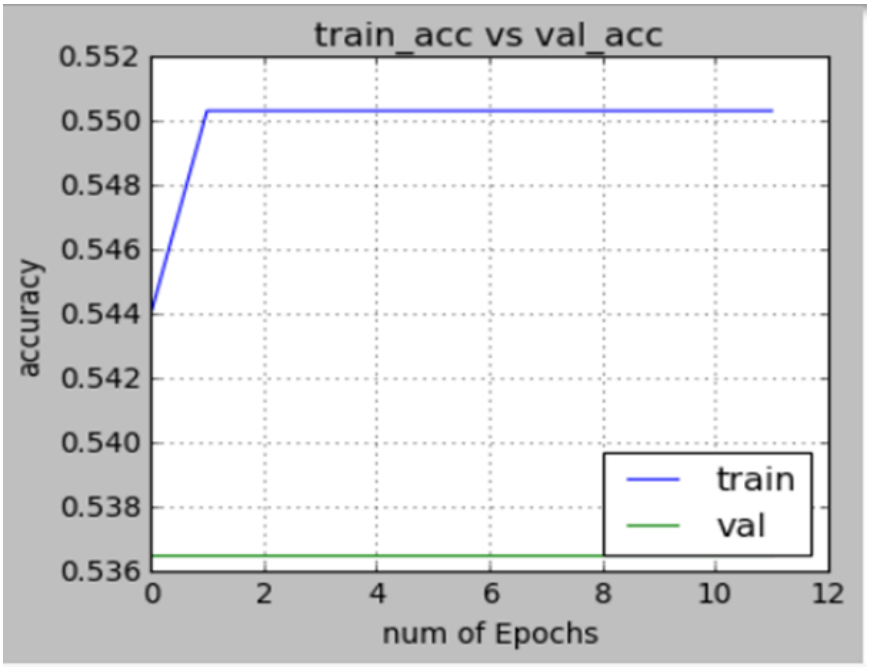


Fig.6.2b, curve of validation accuracy & train accuracy for AlexNet architecture

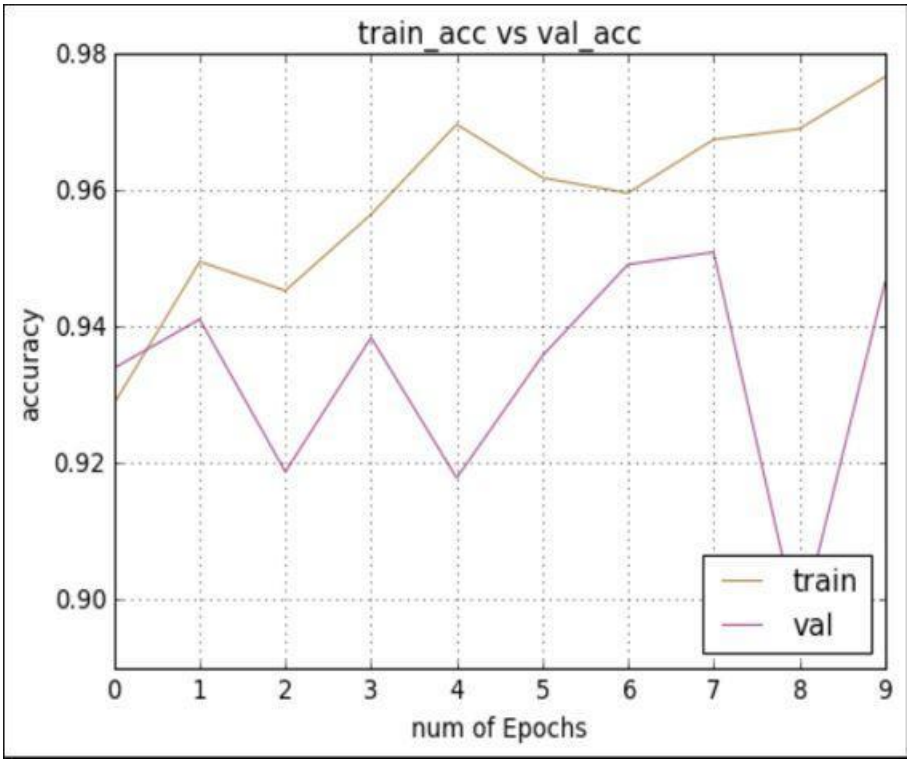


Fig.6.2c, curve of validation accuracy & train accuracy for Modification of model used in Thanh et al paper [94]

6.1.4 Validation loss & Train loss

The lower the **loss,** the better a model. Validation loss is the same metric as training loss, but it is not used to update the weights. It is calculated in the same way - by running the network forward over inputs X(i) and comparing the network outputs yi with the ground truth values y^i using a loss function e.g. $J = \frac{1}{n} L(y\hat{}_i, yi)$ where L is the individual loss function based somehow on the difference between predicted value and target.

As shown in fig.6.3a for basic CNN model max validation loss is ~ 37% & train loss ~31% , fig.6.3b for AlexNet architecture max validation loss is ~ 69.1% & train loss ~68.9 %and fig.6.3c for Modification of model used in Thanh et al paper [94] max validation loss is ~ 7% & train loss ~17%. That mean the Modification of model used in published paper [94] is the best model due to its minimum loss comparing to other models.

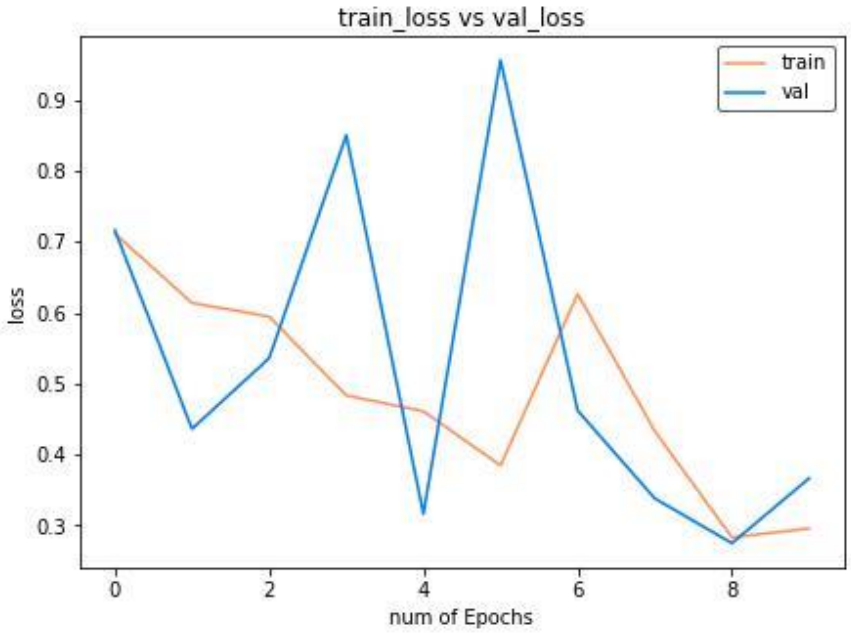


Fig.6.3a, curve of validation loss & train loss for basic CNN model

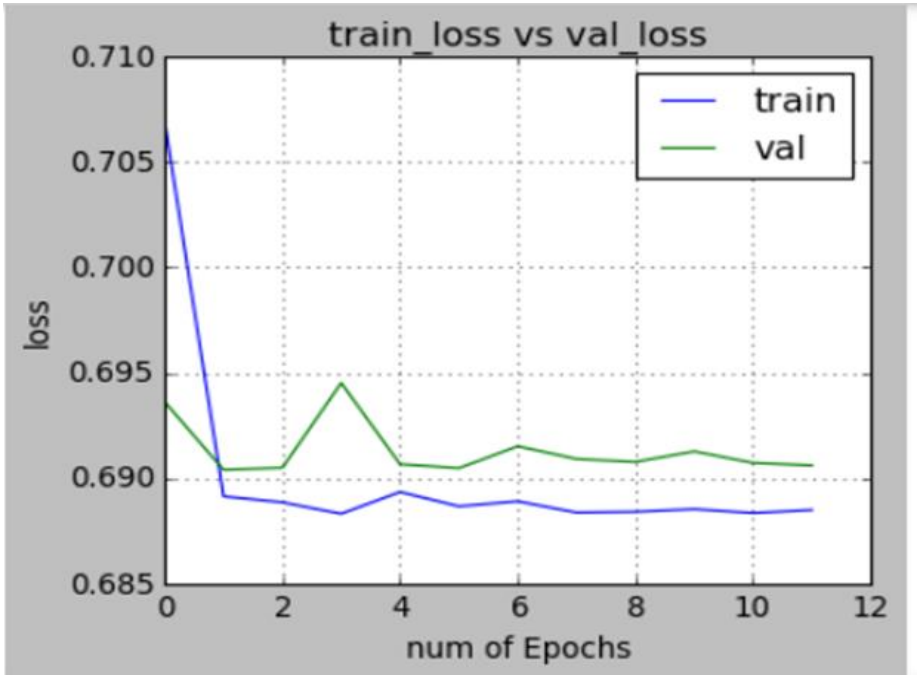


Fig.6.3b, curve of validation loss & train loss for AlexNet architecture

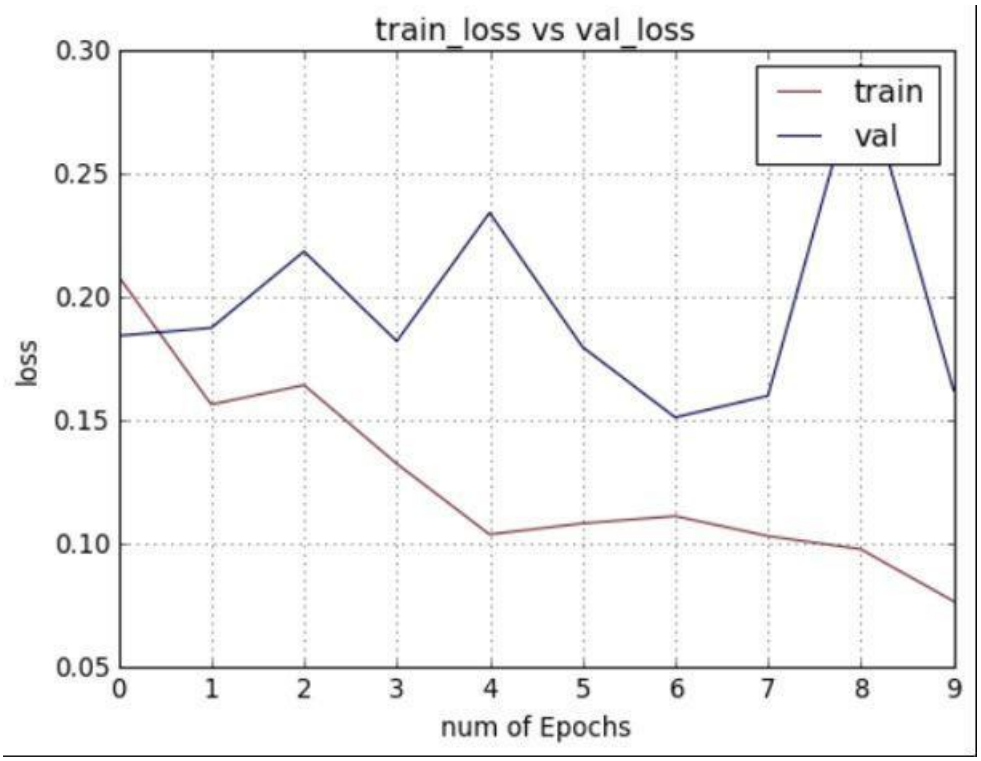


Fig.6.3c, curve of validation loss & train loss for Modification of model used in Thanh et al paper [94].

6.1.5 Mean absolute error

Mean Absolute Error is the average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output. However, they don't give us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data. Mathematically, it is represented as :

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

Mean absolute error for models as shown in fig.6.4a for Our CNN model is 12%, fig.6.4b for AlexNet architecture is 4.8% and fig.6.4c for Modification of model used in Thanh et al paper [94] is 0.549%.

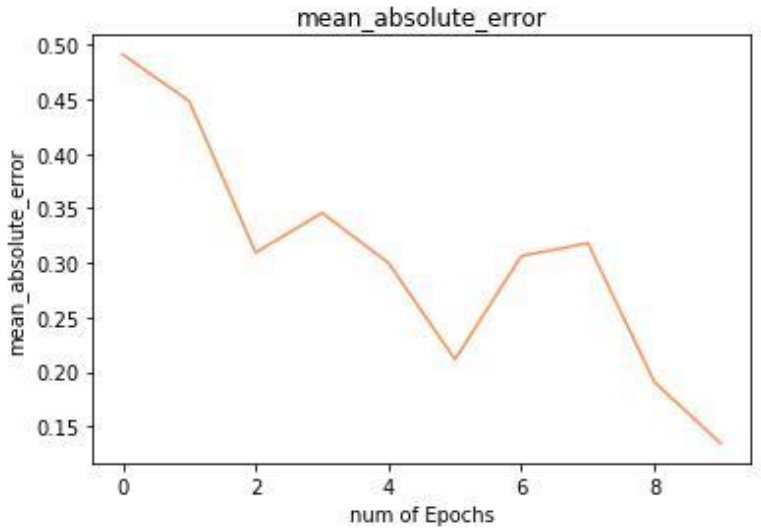


Fig.6.4a, curve of mean absolute error for basic CNN model

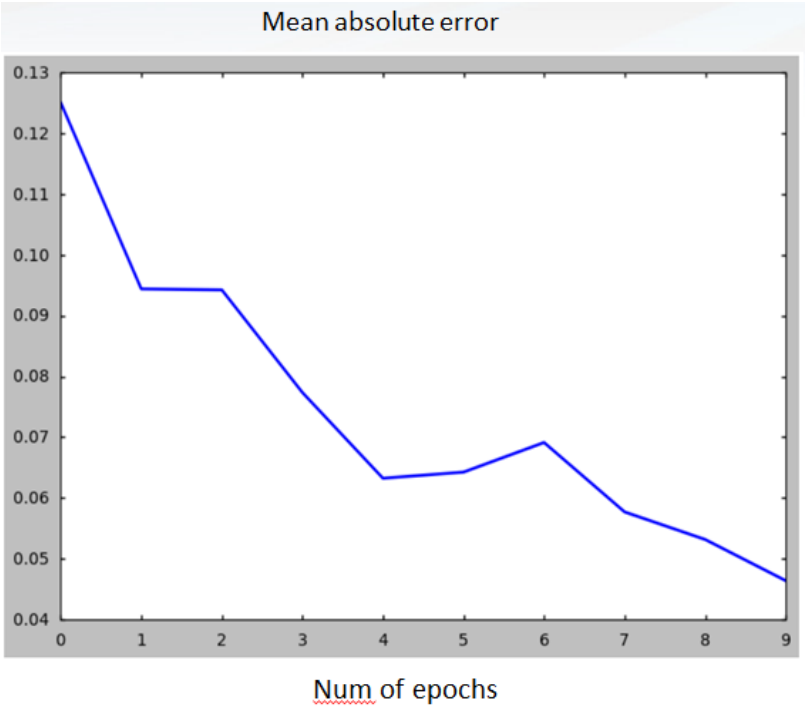


Fig.6.4b, curve of mean absolute error for AlexNet architecture

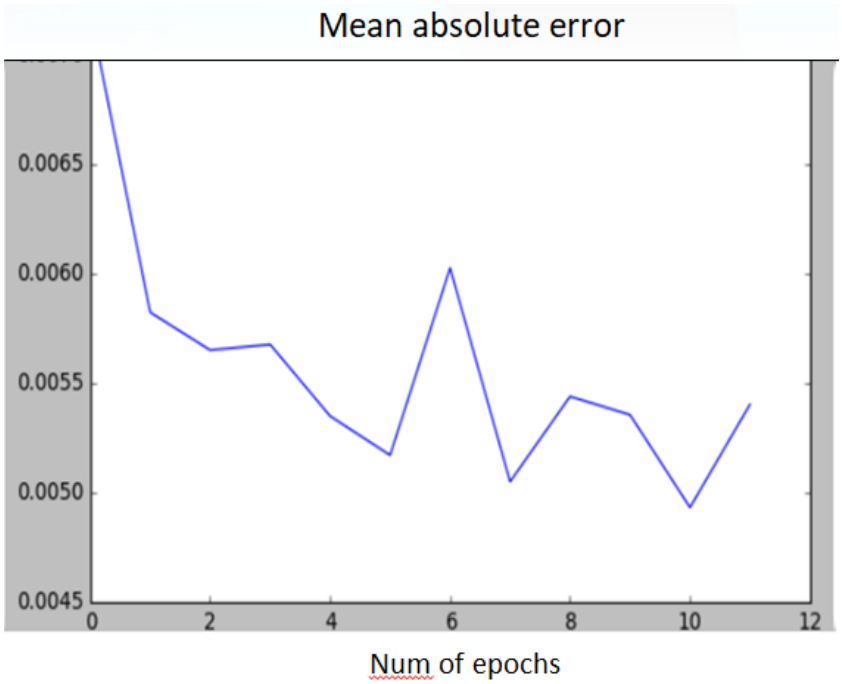


Fig.6.4c, Curve of mean absolute error for Modification of model used in Thanh et al paper [94].

6.1.6 Confusion matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. It is a good option to reporting results in M-class classification problems because it is possible to observe the relations between the classifier outputs and the true ones.

For basic CNN model; numbers of leukemia images which is predicted as leukemia is: 372, numbers of leukemia images which is predicted as normal: 8, number of normal images which is predicted as normal: 269 and number of normal images which is predicted as leukemia: 51; as shown in fig.6.5a. These accuracies show that this model good at prediction leukemia images but bad at prediction normal images.

For AlexNet architecture; numbers of leukemia images which is predicted as leukemia: 0, numbers of leukemia images which is predicted as normal: 380, number of normal images which is predicted as normal: 157 and number of normal images which is predicted as leukemia: 163; as shown in fig.6.5b. These accuracies show that this model is terrible at prediction normal images.

For Modification of model used in published paper [94] number of leukemia images which is predicted as leukemia is: 369, numbers of leukemia images which is predicted as normal: 11, number of normal images which predicted as normal: 301 and number of normal images which is predicted as leukemia: 19; as shown in fig.6.5c. These accuracies show that this model done a great job at prediction normal images.

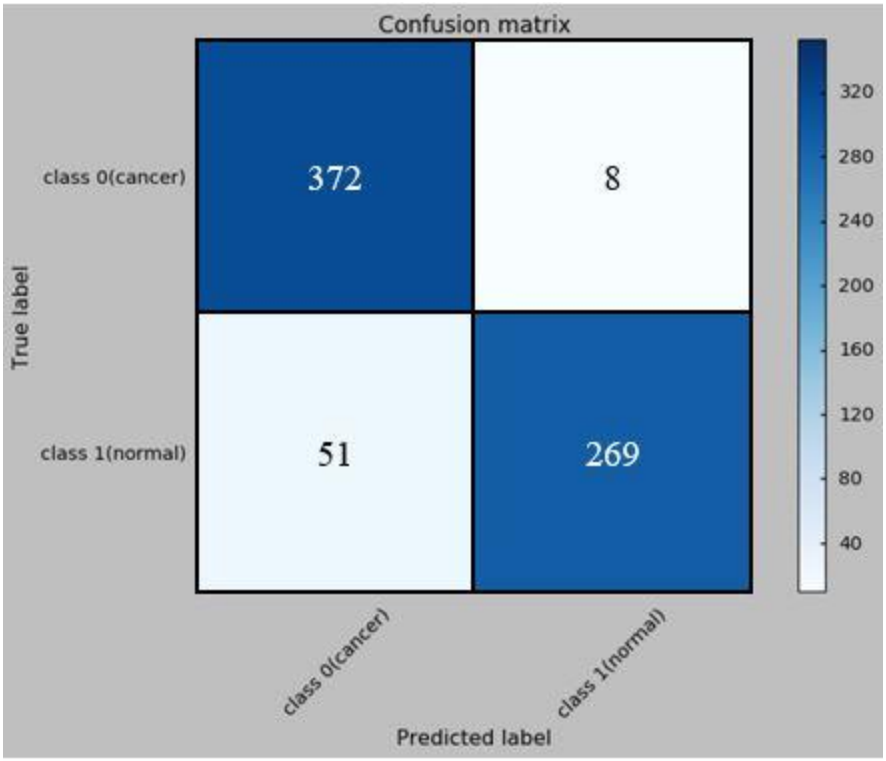


Fig.6.5a, curve of Confusion matrix for basic CNN model

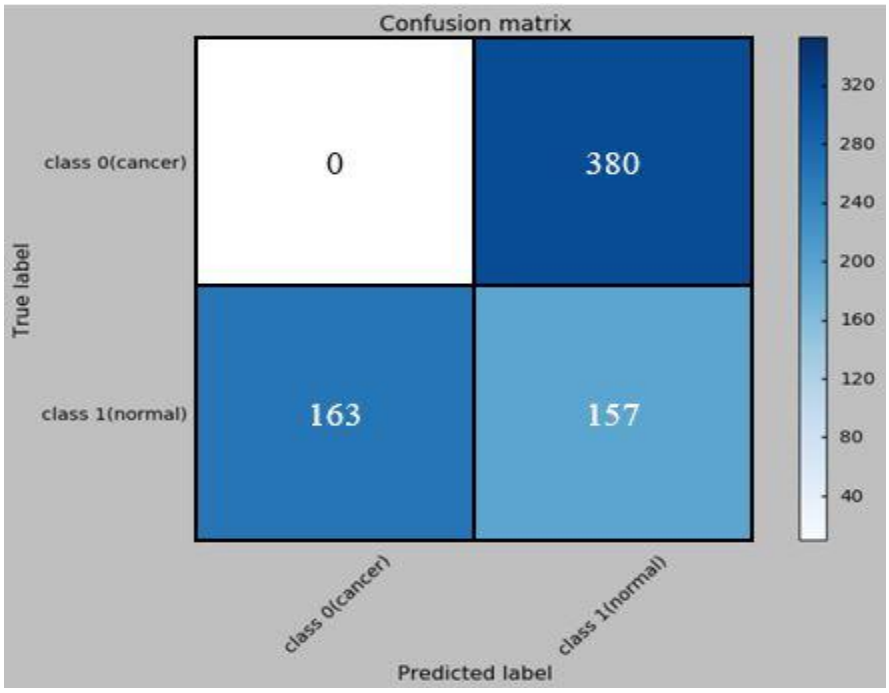


Fig.6.5b, curve of Confusion matrix for AlexNet architecture

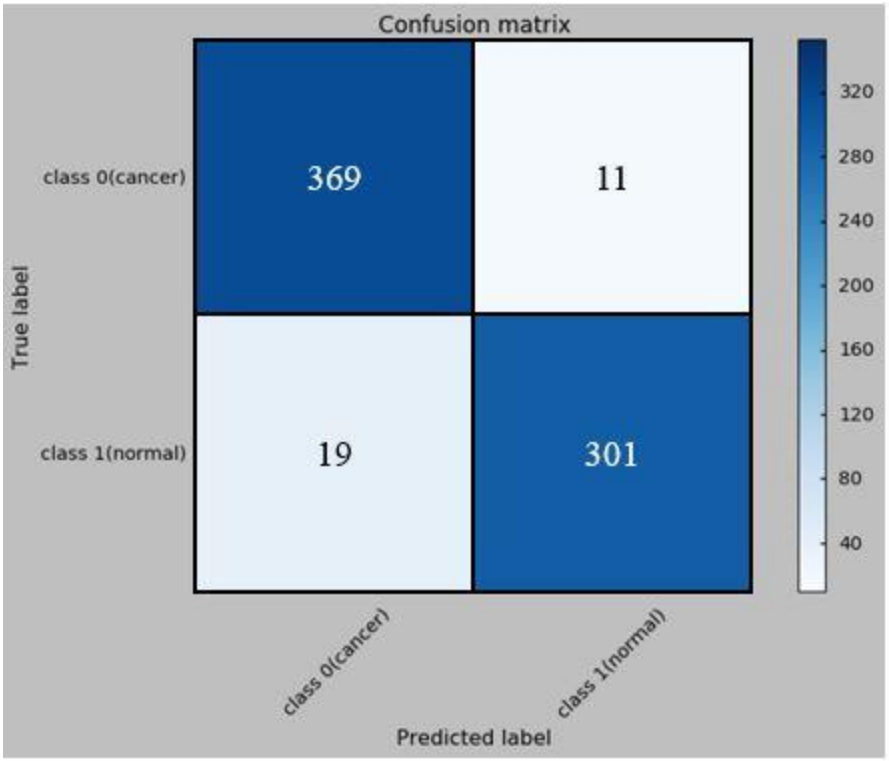


Fig.6.5c, curve of confusion matrix for Modification of model used in Thanh et al paper [94]

6.1.7 Precisions

It is the number of correct positive results divided by the number of positive results predicted by the classifier. As shown in fig.6.6a for Our CNN model has medium precision, fig.6.6b for AlexNet architecture has a very low precision and fig.6.6c for Modification of model used in Thanh et al paper [94] has a good precision due to its goodness method.

6.1.8 Recall

It is the number of correct positive results divided by the number of **all** relevant samples (all samples that should have been identified as positive) ,As shown in fig.6.6a for Our CNN model, fig.6.6b for AlexNet architecture and fig.6.6c for Modification of model used in Thanh et al paper [94]. Perfect model regarded to recall is the third model

In first CNN model in class 1 it is High recall, low precision; This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives. But in class 0 **Low recall, high precision ;**This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP).

6.1.9 F1 score

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances) As shown in fig.6.6a for Our CNN model, fig.6.6b for AlexNet architecture and fig.6.6c for Modification of model used in Thanh et al paper [94]. These Figures show that Modification of model used in Thanh et al paper [94] is a precise and robust.

6.1.10 Support

Support is the number of samples of the true response that lie in that class, it tells us about the actual counts of each class in test data. As shown in fig.6.6a for basic CNN model, fig.6.6b for AlexNet architecture and fig.6.6c for Modification of model used in Thanh et al paper [94].

	precision	recall	f1-score	support
class 0(Cancerous)	0.98	0.62	0.76	78
class 1(Normal)	0.84	0.99	0.91	163

Fig.6.6a, values of precision, recall, f1 score and support for our CNN model

	precision	recall	f1-score	support
class 1(cancer)	0.00	0.00	0.00	178
class 0(normal)	0.49	1.00	0.66	172

Fig.6.6b, values of precision, recall, f1 score and support for AlexNet architecture

	precision	recall	f1-score	support
class 0(cancer)	0.97	0.95	0.96	373
class 1(normal)	0.94	0.97	0.95	327

Fig.6.6c, values of precision, recall, f1 score and support for Modification of model used in Thanh et al paper [94]

Chapter 7

Discussion and conclusion

7.1- Discussion

Our optimizing parameters were accuracy and validation accuracy, by using CNN we trained 3 models: First model consists of 3 convolutional layers with max polling layers. Its accuracy was 90% and 84.97 % validation accuracy. It was bad with our dataset due to its little layers so we trained another model, second model was AlexNet, this architecture proves its efficiency in CNN models, so we trained it with our data, input is (RGB) color images with resolution of 227 x 227 pixels. It consists of 5 convolutional layers with 3 max polling layers. These models achieved 55.35% accuracy and 49.76 % validation accuracy. We found that it doesn't fit our dataset. So, we still have the same problem of low accuracy and keep looking for another model, in the last model we used a retrained model that had been used in a published paper [94], it contain7 layers. The first 5 layers perform feature extraction and the other 2 layers (fully connected and SoftMax) classify the extracted features. The input image has the size of 128x128x3. This architect has accuracy of 97.73 % validation accuracy is 94.64 %, finally we found that this model fit our data and we used in our GUI.

Many studies and publication were done in leukemia and its types; however, no study can achieve our accuracy. We will discuss some of these studies. A model proposed leukemia detection and classification method in which they used histogram equalization and linear contrast stretching for preprocessing. On their specific data set, they are able to achieve 97% accuracy for leukemia detection and 95.6% accuracy for ALL subtype classification [103].

Another leukemia detection method was proposed by **Putzu et al** in which they deployed zack algorithm for the segmentation of leukocytes. After that, SVM classifier was used to classify normal and cancerous cells on the given features. They are able to achieve 92% accuracy [104]. Also another model deployed global thresholding using Otsu threshold technique for segmentation of lymphocytes. After extracting shape-based features, k nearest neighbor classifier was trained to achieve an accuracy of 93% [105].

Other model investigated an automated method for WBC segmentation and leukemia classification. They had used contrast enhancement and histogram equalization for preprocessing of microscopic blood images, then Otsu threshold method was used for white cell segmentation. After extracting shapebased features, K nearest neighbor was trained over those features for classification of normal and blast cells. They were able to achieve an accuracy of 93% [106].

Another proposed a novel method for leukemia segmentation and classification where they used histogram equalization and median filter for preprocessing. For segmentation of lymphocytes, 2 methods including fuzzy c-mean and k-mean were compared. Fuzzy c-means performed better than k-mean clustering for segmentation of lymphocytes. Then support vector machine (SVM) was used for classification of normal and blast cells [108].

A dual threshold method for lymphocyte segmentations was proposed by **Li et al** where they had achieved significant accuracy for segmentation of lymphocytes. They improved traditional single-threshold method by using golden section search to discover optimal threshold value of lymphocytes. **Sarmad Shafique, and Samabia Tehsin:** they used retrained model from CNN archetictures trained on ALL-Image DataBase (IDB) data set Acute lymphoblastic leukemia-IDB1consistedof108images, Acute lymphoblastic leukemia-IDB 2 data set consisted of 260 images and 50 images from google then used augmentation technique to become 760 images. To reach suitable results they converted all datasets to different color formats (RGB,HSV, YCbC,HCbCr), they achieve average accuracy : 96.06% [109].

Thanh, Caleb Vununu, Sukhrob Atoev : they used DCNN consists of 7 layers ,trained on ALL-IDB1 image database which consists of 108 cell image which augmented to become 1188 pictures to achieve accuracy 96.6 % [110].

Another leukemia study authors propose a Convolutional Neural Network (CNN) based method to distinguish normal and abnormal blood cell images. The proposed method achieves accuracy up to 96.6% with the dataset including 1188 blood cell images.

7.2- Conclusion

In this project , we investigated an application of deep CNNs in which we deployed pretrained model for the detection and classification of blood sample to normal and abnormal samples using microscopic blood sample images and convolutional neural networks classification algorithms , the system was built by deep learning which using all features in microscopic images not only examining changes of specific features as a classifier input.

we have performed the pretrained model in a largely augmented dataset in order to confirm the accuracy and reliability of the system, by performing data augmentation we are able to achieved 97.3% accuracy , the system have high accuracy , less processing time (show result in less than 30 second) ,smaller error and early identification of leukemia yields in providing the appropriate treatment to the patient.

Detection system was built in three parts a) acquisition part which consist of a digital camera that have been installed at the top of eyepiece of microscope, b) pretrained CNN model the responsible for classification system and c) graphical user interface to display the image that is obtained from camera and show the classification.

7.3- Problem we faced

The dataset hadn’t the same conditions as it was collected from various resources and it was too small to use with DL as it deals with huge dataset up to millions of images. To overcome this

problem, we thought to collect real images from laboratories and hospitals, as The National Institute of Oncology and 75375's hospital but unfortunately they don't respond us. So we had to use the power of data augmentation, this solution was suitable to us, our data before augmentation was 260 images and after augmentation become 3030 images.

Images used in researches and studying should have high resolutions, so it should be taken with a high efficiency microscope. Because images of leukemia must be too precise to be used and we need lot of images, we attempt to access cells of leukemia from a laboratory of oncology and photograph them with a microscope in our lab attached with a magnifying camera. Unfortunately, images weren't accurate to be used, but we use this microscope as a prototype.

7.4- Future Work

For future work, researchers could expand their focus in a classifying the subtypes of leukemia cells such as Acute Myeloid Leukemia or AML, Chronic Myeloid Leukemia or CML, Acute Lymphoid Leukemia or ALL and Chronic Lymphoid Leukemai or CLL not only separating between cancerous and non-cancerous cell, and develop a convenient environment to construct a big leukemia dataset as this topic of research suffer from leaks in images

References:

1. C.R., Valencio, M.N., Tronco, A.C.B., Domingos, C.R.B., "Knowledge Extraction Using Visualization of Hemoglobin Parameters to Identify Thalassemia", Proceedings of the 17th IEE Symposium on Computer Based Medical Systems, 2004, pp. 1-6.

2. R., Adollah, M.Y., Mashor, N.F.M, Nasir, H., Rosline, H., Mahsin, H., Adilah, "Blood Cell Image Segmentation: A Review", Biomed 2008, Proceedings 21, 2008, pp. 141-144.

3. N., Ritter, J., Cooper, "Segmentation and Border Identification of Cells in Images of Peripheral Blood Smear Slides", 30th Australasian Computer Science Conference, Conference in Research and Practice in Information Technology, Vol. 62, 2007, pp. 161-169.

4. D.M.U., Sabino, L.D.F., Costa, L.D.F., E.G., Rizzatti, M.A., Zago, "A Texture Approach to Leukocyte Recognition", Real Time Imaging, Vol. 10, 2004, pp. 205-206.

5. M.C., Colunga, O.S., Siordia, S.J., Maybank, "Leukocyte Recognition Using EMAlgorithm", MICAI 2009, LNAI 5845, Springer Verlag Berlin Heidelberg, 2009, pp. 545-555.

6. J. Walter," understanding leukemia", 2009, pp- 8-18

7. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: Nature 521.7553 (2015), pages 436–444.

8. Warren S. McCulloch and Walter Pitts. "Neurocomputing: Foundations of Research". In: edited by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, pages 15–27. ISBN: 0-262-01097-6. URL: http://dl.acm.org/citation.cfm? id=65669.104377.

9. F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain". In: Psychological Review (1958), pages 65–386.

10. F. Rosenblatt. Principles of Neurodynamic: Perceptron's and the Theory of Brain Mechanisms. Spartan, 1962.
11. M. Minsky and S. Paper. Perceptron's. Cambridge, MA: MIT Press, 1969 (cited on pages 22,129).

12. P. J. Werbos. “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences”. PhD thesis. Harvard University, 1974 .

13. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Neurocomputing: Foundations of Research”. In: edited by James A. Anderson and Edward Rosenfeld. Cambridge,MA, USA: MIT Press, 1988. Chapter Learning Representations by Back-propagating Errors, pages 696–699. ISBN: 0-262-01097-6. URL: http://dl.acm.org/citation.cfm?id=65669.104451 .

14. Yann LeCun et al. “Efficient BackProp”. In: Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop. London, UK, UK: Springer-Verlag, 1998, pages 9–50. ISBN: 3-540-65311-2. URL: http://dl.acm.org/citation.cfm?id=645754.668382 .

15. Balázs Csanád Csáji. “Approximation with Artificial Neural Networks”. In: MSc Thesis, Eötvös Loránd University (ELTE), Budapest, Hungary (2001) .

16. Yann Lecun et al. “Gradient-based learning applied to document recognition”. In: Proceedings of the IEEE. 1998, pages 2278–2324 .

17. Jason Brownlee. What is Deep Learning? http://machinelearningmastery.com/what-is-deep-learning/. 2016 .

18. T. Ojala, M. Pietikainen, and T. Maenpaa. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7 (2002), pages 971–987 .

19. Robert M. Haralick, K. Shanmugam, and Its’Hak Dinstein. “Textural Features for Image Classification”. In: IEEE Transactions on Systems, Man, and Cybernetics SMC-3.6 (Nov. 1973), pages 610–621. ISSN: 0018-9472. DOI: 10.1109/tsmc.1973.4309314. URL: http://dx.doi.org/10.1109/tsmc.1973.4309314 .

20. Ming-Kuei Hu. “Visual pattern recognition by moment invariants”. In: Information Theory, IRE Transactions on 8.2 (Feb. 1962), pages 179–187. ISSN: 0096-1000.

21. A. Khotanzad and Y. H. Hong. “Invariant Image Recognition by Zernike Moments”. In: IEEE Trans. Pattern Anal. Mach. Intell. 12.5 (May 1990), pages 489–497. ISSN: 0162-8828. DOI: 10.1109/34.55109. URL: http://dx.doi.org/10.1109/34.55109 .

22. Jing Huang et al. “Image Indexing Using Color Correlograms”. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR ’97). CVPR ’97. Washington, DC, USA: IEEE Computer Society, 1997, pages 762–. ISBN: 0-8186-7822-4.URL: http://dl.acm.org/citation.cfm?id=794189.794514 .

23. Edward Rosten and Tom Drummond. “Fusing Points and Lines for High Performance Tracking”. In: Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2. ICCV ’05. Washington, DC, USA: IEEE Computer Society, 2005, pages 1508–1515. ISBN: 0-7695-2334-X-02. DOI: 10 . 1109 / ICCV . 2005 . 104. URL: http://dx.doi.org/10.1109/ICCV.2005.104 .

24. Chris Harris and Mike Stephens. “A combined corner and edge detector”. In: In Proc. of Fourth Alvey Vision Conference. 1988, pages 147–151 .

25. David G. Lowe. “Object Recognition from Local Scale-Invariant Features”. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. ICCV ’99. Washington, DC, USA: IEEE Computer Society, 1999, pages 1150–. ISBN: 0-7695-0164-8. URL: http://dl.acm.org/citation.cfm?id=850924.851523

26. Herbert Bay et al. “Speeded-Up Robust Features (SURF)”. In: Comput. Vis. Image Underst. 110.3 (June 2008), pages 346–359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09. 014. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014

27. Michael Calonder et al. “BRIEF: Binary Robust Independent Elementary Features”. In: Proceedings of the 11th European Conference on Computer Vision: Part IV. ECCV’10. Heraklion, Crete, Greece: Springer-Verlag, 2010, pages 778–792. ISBN: 3-642-15560-X, 978-3-642-15560-4. URL: http://dl.acm.org/citation.cfm?id=1888089.1888148

28. Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection”. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01. CVPR ’05. Washington, DC, USA: IEEE Computer Society, 2005, pages 886–893. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.177. URL: http://dx.doi.org/10.1109/CVPR.2005.177

29. Pedro F. Felzenszwalb et al. “Object Detection with Discriminatively Trained Part-Based Models”. In: IEEE Trans. Pattern Anal. Mach. Intell. 32.9 (Sept. 2010), pages 1627–1645. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.167. URL: http://dx.doi.org/10. 1109/TPAMI.2009.167

30. Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. “Ensemble of Exemplar-SVMs for Object Detection and Beyond”. In: ICCV. 2011

31. Jeff Dean. Results Get Better With More Data, Larger Models, More Compute. http: //static.googleusercontent.com/media/research.google.com/en//people/ jeff/BayLearn2015.pdf. 2016

32. Geoffrey Hinton. What Was Actually Wrong With Backpropagation in 1986? https : //www.youtube.com/watch? v=VhmE_UXDOGs.2016

33. Andrew Ng. Deep Learning, Self-Taught Learning and Unsupervised Feature Learning. https://www.youtube.com/watch?v=n1ViNeWhC24. 2013

34. Andrew Ng. What data scientists should know about deep learning. https : / / www . slideshare.net/ExtractConf. 2015

35. Jürgen Schmidhuber. "Deep Learning in Neural Networks: An Overview". In: CoRR abs/1404.7828 (2014). URL: http://arxiv.org/abs/1404.7828

36. JUSTIN KER, LIPO WANG, JAI RAO, AND TCHOYOSON LIM "Deep Learning Applications in Medical Image Analysis", Digital Object Identifier 10.1109/ACCESS.2017.2788044, published in IEEE at December 29, 2017.

37. Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, Clara I. Sanchez ´ ,Diagnostic Image Analysis Group ,Radboud University Medical Center ,Nijmegen, The Netherlands "A Survey on Deep Learning in Medical Image Analysis"

38. ZhiFei Lai and HuiFang Deng , "Medical Image Classification Based on Deep Features Extracted by Deep Model and Statistic Feature Fusion with Multilayer Perceptron" Hindawi Computational Intelligence and Neuroscience Volume 2018, Article ID 2061516, 13 pages ,https://doi.org/10.1155/2018/2061516 .

39. Kamran Kowsari1, Rasoul Sali1, Marium N. Khan3, William Adorno1, S. Asad Ali4, Sean R. Moore3, Beatrice C. Amadi5, Paul Kelly5;6, Sana Syed2;4;∗, and Donald E. Brown1;2;∗ "Diagnosis of Celiac Disease and Environmental Enteropathy on Biopsy Images Using Color Balancing on Convolutional Neural Networks"

40. Purvil Bambharolia , "OVERVIEW OF CONVOLUTIONALNEURAL NETWORKS" ,international conference on academic research in engineering and management ,30th april 2017 .

41. Jason Brownlee "A Gentle Introduction to Transfer Learning for Deep Learning"on December 20, 2017 in Deep Learning for Computer Vision, https://machinelearningmastery.com/transfer-learning-for-deep-learning/

42. https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

43. https://becominghuman.ai/understand-classification-performance-metrics-cad56f2da3aa

44. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

45. https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

46. https://towardsdatascience.com/understanding-hyperparameters-optimization-in-deep-learning-models-concepts-and-tools-357002a3338a

47. https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

48. Asifullah Khan1, 2*, Anabia Sohail1, Umme Zahoora1, and Aqsa Saeed Qureshi1 "A Survey of the Recent Architectures of Deep Convolutional Neural Networks"

49. Ker, J., Wang, L., Rao, J., & Lim, T. (2018). Deep Learning Applications in Medical Image Analysis. IEEE Access, 6, 9375–9389.

50. Kooi T, Litjens G, van Ginneken B et al (2017) Large scale deep learning for computer aided detection of mammographic lesions. Med Image Anal 35:303–312

51. H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, Y. Zheng, "Convolutional neural networks for diabetic retinopa-thy", Procedia Comput. Sci., vol. 90, pp. 200-205, Jul. 201

52. Kurata Y, Nishio M, Fujimoto K et al (2018) Automatic segmentation of uterus with malignant tumor on MRI using U-net. In: Proceedings of the Computer Assisted Radiology and Surgery (CARS) 2018 congress (accepted)

53. Kim KH, Choi SH, Park SH (2018) Improving arterial spin labeling by using deep learning. Radiology 287:658–666.

54. Liu F, Jang H, Kijowski R, Bradshaw T, McMillan AB (2018) Deep learning MR imaging-based attenuation correction for PET/MR imaging. Radiology 286:676–684

55. Chen MC, Ball RL, Yang L et al (2018) Deep learning to classify radiology free-text reports. Radiology 286:845–852

56. Liang M, Tang W, Xu DM et al (2016) Low-dose CT screen-ing for lung cancer: computer-aided detection of missed lung cancers. Radiology 281:279–288

57. Lu F, Wu F, Hu P, Peng Z, Kong D (2017) Automatic 3D liver location and segmentation via convolutional neural network and graph cut. Int J Comput Assist Radiol Surg 12:171–182

58. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A (eds) Proceedings of Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015

59. Haofu Liao, Jiebo Luo A Deep Multi-Task Learning Approach to Skin Lesion Classification

60. Kawahara, J.; BenTaieb, A.; and Hamarneh, G. 2016. Deep features to classify skin lesions. In 13th IEEE International Symposium on Biomedical Imaging, ISBI 2016, Prague, Czech Republic, April 13-16, 2016, 1397–1400

61. Esteva, A.; Kuprel, B.; and Thrun, S. 2015. Deep networks for early stage skin disease and skin cancer classification.

62. H. I. Suk, C. Y. Wee, S. W. Lee, D. Shen, "State-space model with deep learning for functional dynamics estimation in resting-state fMRI", Neuroimage, vol. 129, pp. 292-307, Apr. 2016

63. M. D. Kumar, M. Babaie, S. Zhu, S. Kalra, H. R. Tizhoosh, A comparative study of CNN BOVW and LBP for classification of histopathological images, Sep. 2017

64. F. Liao, M. Liang, Z. Li, X. Hu, S. Song, Evaluate the malignancy of pulmonary nodules using the 3D deep leaky noisy-or network, 2017

65. H.-C. Shin et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures dataset characteristics and transfer learning", IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1285-1298, May 2016.

66. A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks", Nature, vol. 542, no. 7639, pp. 115-118, 2017.

67. D. C. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks", Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent., pp. 411-418, 2013.

68. X. Yang et al., "A deep learning approach for tumor tissue image classification", Proc. Int. Conf. Biomed. Eng., 2016,

69. J. Xu et al., "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images", IEEE Trans. Med. Imag., vol. 35, no. 1, pp. 119-130, Jan. 2016.

70. Lakhani P, Sundaram B (2017) Deep learning at chest radi-ography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284:574–582

71. Milletari F, Navab N, Ahmadi S-A (2016) V-net: fully con-volutional neural networks for volumetric medical image seg-mentation. In: Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV).

72. Christ PF, Elshaer MEA, Ettlinger F et al (2016) Automat-ic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In: Ourselin S, Joskowicz L, Sabuncu M, Unal G, Wells W (eds) Proceedings of Medical image computing and computer-assisted intervention – MICCAI 2016.

73. S. Pereira, A. Pinto, V. Alves, C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images", IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1240-1251, May 2016.

74. M. Havaei et al., "Brain tumor segmentation with deep neural networks", Med. Image Anal., vol. 35, pp. 18-31, Jan. 2017.

75. Z. Yan et al., "Bodypart recognition using multi-stage deep learning" in Information Processing in Medical Imaging, Cham, Switzerland:Springer, vol. 24, pp. 449-461, Jun. 2015.

76. H. R. Roth et al., "Anatomy-specific classification of medical images using deep convolutional nets", Proc. IEEE 12th Int. Symp. Biomed. Imag. (ISBI), pp. 101-104, Apr. 2015.

77. H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1930-1943, Aug. 2013.

78. X. Yang, R. Kwitt, M. Styner, M. Niethammer, "Quicksilver: Fast predictive image registration—A deep learning ap-proach", Neuroimage, vol. 158, pp. 378-396, Jul. 2017.

79. S. Miao, Z. J. Wang, R. Liao, "A CNN regression approach for real-time 2D/3D registration", IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1352-1363, May 2016

80. https://homes.di.unimi.it/scotti/all/

81. https://imagebank.hematology.org

82. www.123rf.com

83. www.shutterstock.com

84. http//hemotologyatlas.co

85. http://www.chronolab.com/atlas/hema

86. https://www.hematology.org/

87. Digital Image ProcessingThird EditionRafael C. GonzalezUniversity of TennesseeRichard WoodsMedData Interactive

88. https://towardsdatascience.com/advanced-data-augmentation-strategies-383226cd11ba

89. https://blog.paperspace.com/data-augmentation-for-bounding-boxes

90. https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

91. https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

92. http://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid358544.pdf

93. https://github.com/mdbloice/Augmentor

94. T. T. P. Thanh, Caleb Vununu, Sukhrob Atoev, Suk-Hwan Lee, and Ki-Ryong Kwon "Leukemia Blood Cell Image Classification Using Convolutional Neural Network". In: International Journal of Computer Theory and Engineering, Vol. 10, No. 2, April 2018.

95. https://www.geeksforgeeks.org/python-gui-tkinter/

96. https://notitle.victordomingos.com/images/GUI_programming_python_tkinter.pdf

97. https://www.tutorialspoint.com/python/tk_label.htm

98. https://www.tutorialspoint.com/python/tk_radiobutton.htm

99. http://effbot.org/tkinterbook/button.htm

100. https://www.python-course.eu/tkinter_entry_widgets.php

101. ceti_vulcan_led_compound_microscopes.pdf

102. https://www.amazon.com/Koolertron-Microscope-Adjustable-Magnification Rechargeable/dp/B0752RGX6N/ref=pd_day0_hl_21_4/147-5828423-5543558?_encoding=UTF8&pd_rd_i=B0752RGX6N&pd_rd_r=12feb585-8dda-11e9-b9cc-f314601f96fe&pd_rd_w=BX6QS&pd_rd_wg=GF5Rk&pf_rd_p=ad07871c-e646-4161-82c7-5ed0d4c85b07&pf_rd_r=WDW9DR6WANCKK7JF4NCP&psc=1&refRID=WDW9DR6WANCKK7JF4NCP

103.

104. MoradiAmin, M, Samadzadehaghdam, N, Kermani, S, Talebi, A. Enhanced recognition of acute lymphoblastic leukemia cells in microscopic images based on feature reduction using principle component analysis. Front Biomed Technol. 2015;2(3):128–136.

105. Putzu, L, Caocci, G, Di Ruberto, C. Leucocyte classification for leukaemia detection using image processing techniques. Artif Intell Med. 2014;62(3):179–191.

106. Chatap, N, Shibu, S. Analysis of blood samples for counting leukemia cells using support vector machine and nearest neighbour. IOSR J Comput Eng. 2014;16(5):79–87.

107. Joshi, MD, Karode, AH, Suralkar, S. White blood cells segmentation and classification to detect acute leukemia. Int J Emerging Trends Technol Computer Sci (IJETICS). 2013;2(3):147–151.

108. Karthikeyan, T, Poornima, N. Microscopic image segmentation using fuzzy c means for leukemia diagnosis. Leukemia. 2017;4(1):3136–3142.

109. Li, Y, Zhu, R, Mi, L, Cao, Y, Yao, D. Segmentation of white blood cell from acute lymphoblastic leukemia images using dual threshold method. Comput Math Methods Med. 2016;2016:9514707.

110. Acute Lymphoblastic Leukemia Detection and Classification of Its Subtypes Using Pretrained Deep Convolutional Neural Networks Sarmad Shafique, MS1 and Samabia Tehsin, PhD1