Cairo University
Faculty of Computers and Artificial intelligence

Leukemia Detection and Classification Using Machine Learning and Deep Learning

Graduation Project

Implemented by

Name	ID
Esraa Ramadan Saber	20188003
Toka Mohamed ElDesoky	20188013
Samar SalahElDin Ghoneim	20188029
Nada Samir Mohamed	20188051
Somaya Yasser Galal	20188065

Supervised by

Dr. Ahmed Farouk

Dr. Hanaa Bayoumi

TA. Sarah El-Nady

2021-2022

Cairo University
Faculty of Computers and Artificial intelligence

Leukemia Detection and Classification Using Machine Learning and Deep Learning

Supervised by

Dr. Ahmed Farouk

Dr. Hanaa Bayoumi

TA. Sarah El-Nady

Implemented by

ID	Name
20188003	Esraa Ramadan Saber
20188013	Toka Mohamed ElDesoky
20188029	Samar SalahElDin AbdAlAziz Ghoneim
20188051	Nada Samir Mohamed ElBendary
20188065	Somaya Yasser Galal

Graduation Project

Academic Year 2021-2022

Acknowledgement

We would like to express our sincere gratitude to Faculty of Computers and Artificial Intelligence, Cairo University for letting us fulfill our dream of being a student there.

To our supervisors, Dr. Ahmed Farouk, Dr. Hanaa Bayoumi and TA. Sarah El-Nady. We are extremely grateful for your assistance and suggestions throughout our project. Thanks for your efforts and knowledge sharing through the past years and during working on this project with understanding, wisdom, patience, enthusiasm, encouragement, and constant guidance towards this project.

To all my friends and family for helping me survive all the stress from this year and not letting me give up.

Abstract

Leukemia is a blood cancer that can be fatal in most cases. Detection and classification of Leukemia into which type the patient has can take a lot of time and resources for such process to be done. There are multiple stages for the process to know whether the patient is affected or not and which type do they have to determine after the prognosis and which treatment is needed. Our study proposes a new approach for diagnosis of all types and subtypes of leukemia from microscopic blood images using convolutional neural networks (CNN), which requires a large training data set as it is a type of deep learning algorithm. Also, using Support Vector Machine (SVM) and Random Forest, which is a machine learning algorithm. We used two datasets one that had 5 publicly available data sources for each Leukemia type: Acute lymphocytic Leukemia, Chronic lymphocytic Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia, and normal blood cells. The second had the 4 leukemia types: Acute lymphocytic Leukemia, Chronic lymphocytic Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia. Data Augmentation was then applied when needed. Moreover, we also explored other well-known machine learning algorithms to evaluate our approach. Finally, we want to show the variety of performance between CNN architecture models and other well-known machine learning algorithms. To end with, we build a website so specialists can easily use to upload the samples to be predicted.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

Abbreviation	Word
ADAM	Adaptive Moment Estimation
ALL	Acute Lymphocytic Leukemia
AML	Acute Myelogenous Leukemia
CLL	Chronic Lymphocytic Leukemia
CML	Chronic Myelogenous Leukemia
CNN	Convolution Neural Network
CSS	Cascading Style Sheet
DL	Deep Learning
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
JPG	Joint Photographic Expert
JS	Javascript
KNN	K Nearest Neighbor
ML	Machine Learning
PNG	Portable Network Graphics
RGB	Red-Green-Blue

RGBA	Red-Green-Blue-Alpha
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
SW	Software
VGG	Visual Geometry Group from Oxford
VS Code	Visual Studio Code
WBCs	White Blood Cells

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Background

1.1.1 Overview

Leukemia is an aggressive disease related to the white blood cells (WBCs) and affects the bone marrow and blood of the human body. This disease can lead to destroying the immune system of the human body. This disease can also make the bone marrow produce excess number of abnormal WBCs, which do not function properly. Furthermore, these malignant WBCs can flow into the blood and cause damage of other parts of human body like Liver, kidney, Spleen, Brain and so on, which lead to other fatal forms of cancer. Leukemia is a blood cancer which is classified as either myelogenous (myeloid) or lymphocytic (lymphoid) depending on which types of white blood cells are affected.

When the abnormal/affected cells are mainly lymphocytes, the leukemia is classified as lymphocytic. When the abnormal/affected cells are primarily granulocytes or monocytes the leukemia is classified as myelogenous. The two main types are divided into two subtypes; acute and chronic. Acute Leukemia usually affects younger people, the cells are immature and it can be fatal if left untreated. Chronic Leukemia usually affects older people, the cells are mature and is usually less aggressive.

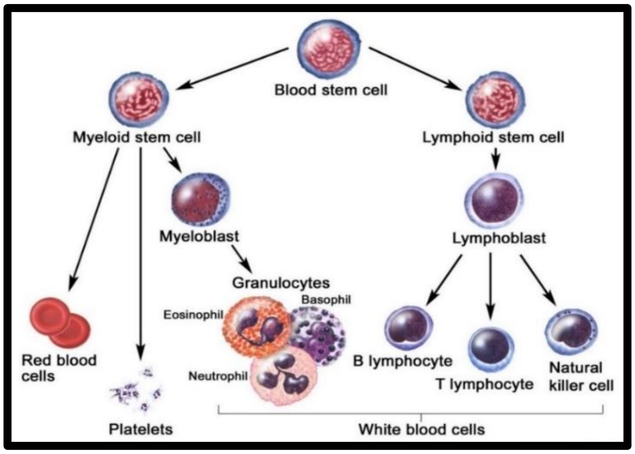


Figure 1: Types of Blood Stem Cell

1.1.2 Diagnosis Steps

In medical field, Leukemia is diagnosed through applying multiple steps; Complete blood count; measures the amounts of different cells in the blood, such as the red blood cells, white blood cells, and platelets. Routine microscopic exams; samples of blood or bone marrow are looked at under a microscope by a pathologist. Taking into consideration the shape and size of the cells. Cytochemistry; cells are exposed to chemical stains (dyes) that react with only some types of leukemia cells. Immunohistochemistry; used to determine which type of leukemia the patient has. Cytogenetics; compare the karyotype of the patient with a normal person. Molecular Tests; helps in finding genes changes. Prognosis; determining the aggressiveness of the leukemia the patient has and based on that, determining which treatment protocol to follow.

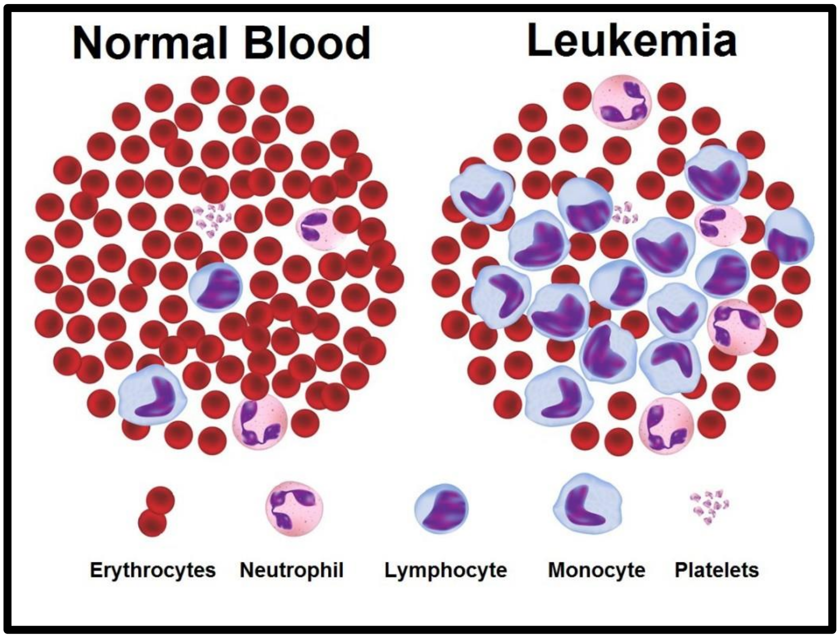


Figure 2. Difference Between Normal and Affected WBCS With Leukemia

1.1.3 Intersection with Machine

New approaches were created which are machine learning and deep learning; basically, they focus on the use of data and algorithms to imitate the way that humans learn, and gradually improving its efficiency. There are multiple algorithms for both machine learning and deep learning. Machine learning can be categorized into supervised and unsupervised algorithm/problem and is applied over small datasets. Supervised can be done using linear regression, support vector machine, logistic regression, naïve bayes, decision tree, K-nearest neighbour (KNN) and more. Unsupervised can include Kmeans, agglomerative and dimensionality reduction. Deep Learning is applied over large number datasets. It includes convolutional neural network, recurrent neural network and artificial neural network. Usually what is used in detecting images and classifying them is Classification algorithms of machine learning and CNN of deep learning.

1.2 Motivation

As shown, it requires a lot of steps to diagnose Leukemia through a molecular biologist view. While machine learning or deep learning takes less effort and less time to predict and classify the input images. This motivated us to build a model for Leukemia diagnosis using deep learning which enables us to save a lot of time and effort.

1.3 Problem definition:

As obvious, the traditional way to diagnose leukemia requires numerous amounts of steps which require a lot of time and resources. Therefore, we are taking a machine learning/deep learning approach to do these steps in less time and less energy taken. Given a dataset of microscopic blood images, we want to train and build several models than can detect Leukemia in these images and classify it as well. In addition, these models need to be compared and evaluated in order to find the most accurate model that can diagnose Leukemia.

1.4 Project Objective:

Through our study, we have worked on Detection of Leukemia (whether the patient is affected or not) and Classification into its types if affected (ALL, AML, CML and CLL). We approached the problem by applying machine learning and deep learning on our data. We used multiple machine learning algorithms as: SVM, Logistic Regression, Naïve Bayes, and Random Forest algorithms. Also, Multiple CNN Architecture as: MobileNet, VGG16, ResNet50, AlexNet and our own model. Some of the Leukemia types do not have large datasets therefore, we applied data augmentation to increase the number of images.

By using the model these models that have been created, specialists will save both time and effort for a primary detection and classification of the patient status regarding Leukemia. We took both an image processing approach and a bioinformatics one then compared the results of both of them.

1.5 Project Development Methodology

We used waterfall development methodology, where we performed requirement analysis and designed the system functionality. That was followed by data collection and the implementation and testing phase. We have first performed Machine Learning and Deep Learning on both of our datasets. For machine learning we used SVM, Logistic Regression and Random Forest. For deep learning we used CNN and known architectures like VGG16, AlexNet, MobileNet, and ResNet50. Finally, we built the website application and deployed the best obtained model in it.

1.6 Gantt Chart

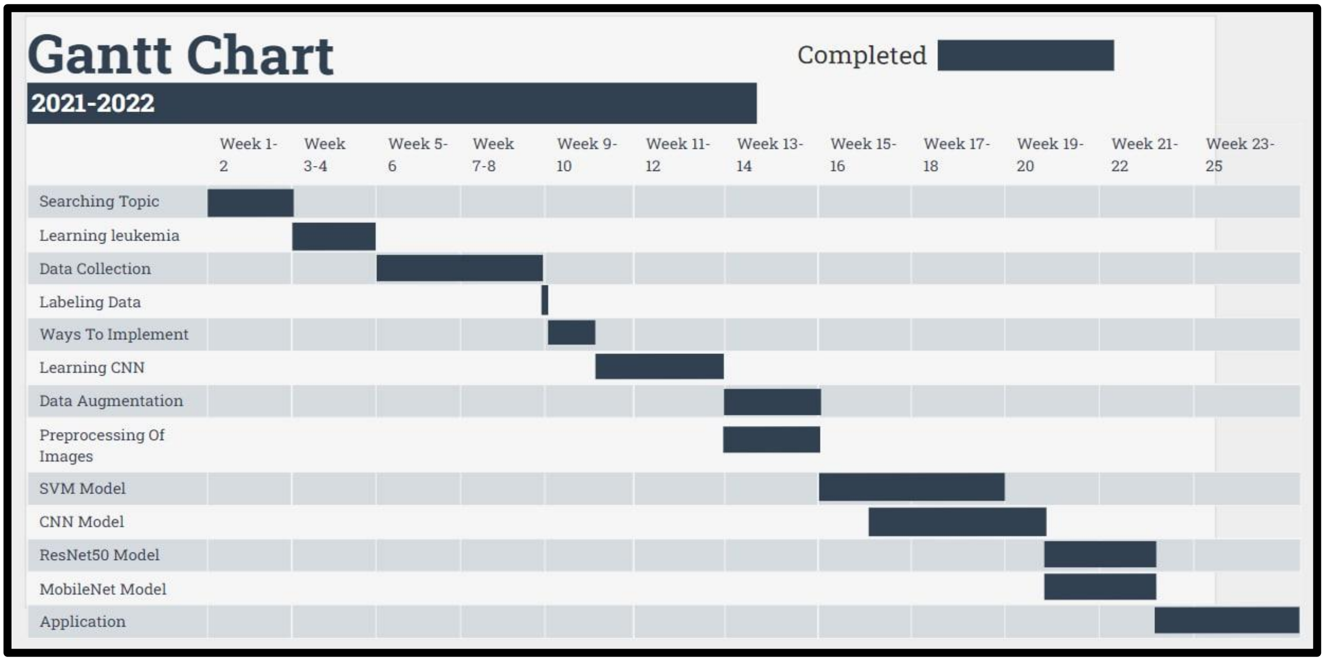


Figure 3. Gantt Chart

1.7 Used Tools

SW Tools:

1.7.1 PyCharm

PyCharm is an integrated development environment used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains.

We used PyCharm for python coding to build our models, read images from files, apply preprocessing on the read images, test different model combinations for both machine learning and deep learning, and different preprocessing for our data. We also used it to open a server using flask and build our database using SQLite.

1.7.2 VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. We used VSCode to build our website, we used different languages in it; Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (JS).

1.7.3 Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected.

We used python to code both our machine learning and deep learning models, and the different preprocessing for the images. We also used it to display graphs, read images, and displaying some of the images. We also used it to open a server using flask and build our database using SQLite. To build our Deep Learning models we used packages like keras and tensorflow. We also used other helpful packages like matplotlib, cv2, sklearn and numpy.

1.7.4 HTML

HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. We used it to build the structure of our pages in the website.

1.7.5 CSS

CSS allows you to create great-looking web pages, but how does it work under the hood? This article explains what CSS is with a simple syntax example and also covers some key terms about the language. We used it to make the style for our pages in the website.

1.7.6 JavaScript

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. We used it to make the pages interactive with the user clicks in the website.

1.7.7 SQLite

SQLite is a database engine written in the C language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. We used it to build our database in python environment.

1.7.8 Visual Paradigm

Visual Paradigm is a software application designed for software development teams to model business information systems and manage development processes. In addition to modeling support, this technology provides report generation and code engineering capabilities including code. We used it to make our diagrams; ERD, Sequence Diagram, Use case Diagram, and Component Diagram.

1.7.9 JQuery

JQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. We used it to add effect to the elements in the website and icons.

1.8 Report Organization

The project documentation is divided into 7 Chapters, describing all aspects of our project and an elaboration on each step and how we achieved it and the final results.

In Chapter 2 (Related Work) we will talk about the related papers to the field of interest and the topic of our paper, their accuracies and the difference between our project and their paper.

In Chapter 3 (System Analysis) we will mention our functional requirements, non-functional requirements, and the use case diagram for the website.

In Chapter 4 (System Design) we will show the diagrams made for our system, the ERD, the sequence diagram, the component diagram and the system class. We will also show the GUI for our website.

In Chapter 5 (Methodology) we will explain the steps of obtaining the best model, starting from collecting the datasets going through the architectures and ending with their accuracies.

In Chapter 6 (Implementation and Testing) we will discuss our experiments and will display how the system is running, the different test cases.

In Chapter 7 (Results) we will compare the results of each model, architecture and algorithm in both machine learning and deep learning.

Chapter 2

Related Work

Chapter 2

Related Work

2.1 Related Work:

2.1.1 AML, ALL, and CML classification and diagnosis based on bone marrow cell morphology combined with convolutional neural network Nov 2020

In this paper, they have worked on bone marrow smears microscopic images to classify three types of leukemia CML, AML and ALL. CNN model combined with transfer learning has been used to increase efficiency and effectiveness. After preprocessing of the images, they have used three frameworks (Resnt50, DenseNet121, Inception-V3). They have also trained models on Raw data and on the preprocessed images data. The Dense Net121 model that was used on the raw data produced prediction accuracy 74.8%, while the DenseNet121 model that was obtained by transfer learning optimization was 95.3%. The results showed that the leukemic cell morphology classification and diagnosis based on CNN combined with transfer learning is feasible. Compared with conventional manual microscopy, this method is more rapid, accurate, and objective.

2.1.2 Acute Myeloid Leukemia (AML) Detection Using AlexNet Model May 2021

In this paper, blood smear microscopic images were used to classify and detect AML using AlexNet and LeNet-5 architectures. The results showed that AlexNet was able to identify 88.9% of images correctly with 87.4% precision and 98.58% accuracy, whereas LeNet-5 correctly identified 85.3% of images with 83.6% precision and 96.25% accuracy.

2.1.3 Identification of Leukemia Subtypes from Microscopic Images Using Convolutional Neural Network Aug 2019

In this paper, they have used publicly available datasets for leukemia. They have applied seven different image transformation techniques as data augmentation. They have designed a CNN architecture capable of recognizing all subtypes of leukemia. Besides, they have also explored other well-known machine learning algorithms such as naive Bayes, support vector machine, k-nearest neighbor, and decision tree. To evaluate their approach, they have set up a set of experiments and used 5-fold cross-validation. The results they have obtained from experiments showed that their CNN model performance has 88.25% and 81.74% accuracy, in leukemia versus healthy and multi-class classification of all subtypes, respectively.

The **difference** between those papers and our project is that blood samples images will be used to classify the four types of leukemia, and detect normal cells from Leukemia ones. Machine Learning and Deep Learning will be used in this project unlike most papers which used Deep Learning only. For the Machine Learning approach Logistic Regression and SVM will be used, while in the Deep Learning approach CNN Models using ResNet-50, VGG-16, MobileNet and ImageNet architectures will be in use.

Chapter 3

System Analysis

Chapter 3

System Analysis

3.1 Project Specification:

3.1.1 Functional requirements

We will build a user-friendly application that enables doctors and specialists to use our trained Leukemia detection and classification model directly on their image dataset. This app will allow the following features:

- Registering in the application.
- Upload the image needed for prediction.
- Result of the model prediction.
- View information about leukemia and the website.

3.1.2 Non-functional requirements

- Reliability

As the system reliable enough to sustain in any condition. Should give consistently correct results.

- Maintainability:

It’s easy to add code to the existing system, and upgrade for new features and new technologies from time to time. It’s cost-effective and easy.

- Safety Requirements:

Information transmission should be securely transmitted to server without any changes.

- Security Requirements:

The system shall provide a certain type of security to make sure of the validity of data of the user. Information transmission should be securely transmitted to server without any changes.

- Usability:

As the system is friendly easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

- Performance:

The system must be interactive, and the delays involved must be less. In every action-response of the system, there are no immediate delays.

- Response Time:

The system response time is quick, it takes less than a second to process the information and output the results.

3.2 Use-case Diagram

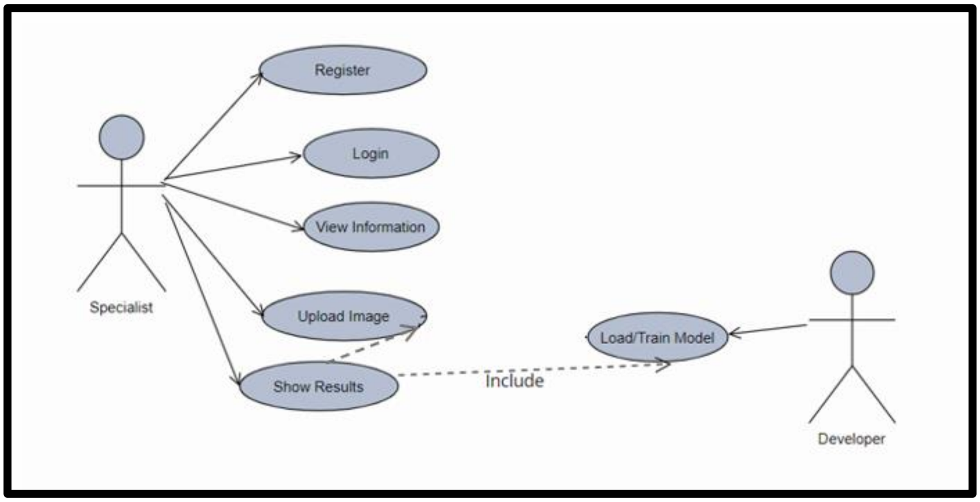


Figure 4 Use Case Diagram

Chapter 4

System Design

Chapter 4

System Design

4.1 System Component Diagram

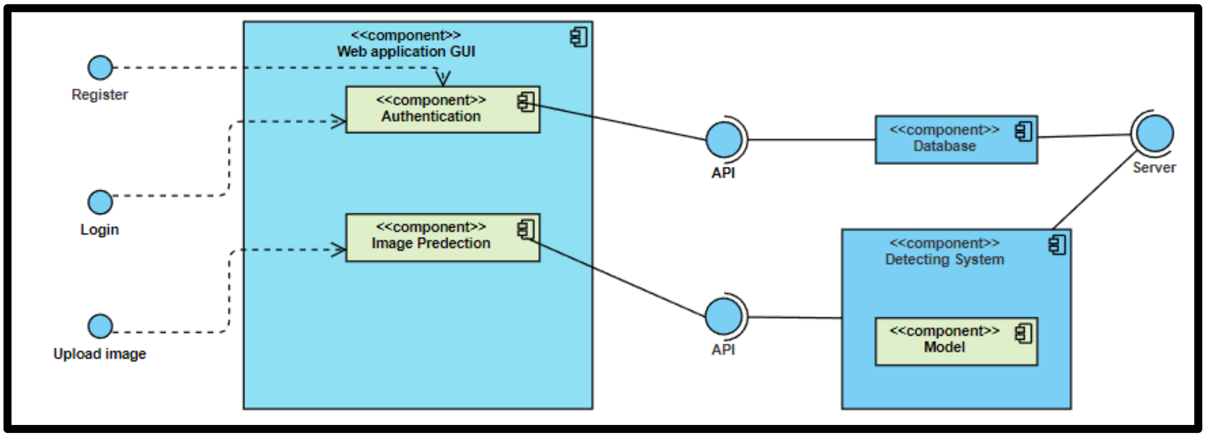


Figure 5. System Component Diagram

4.2 System Class Diagrams

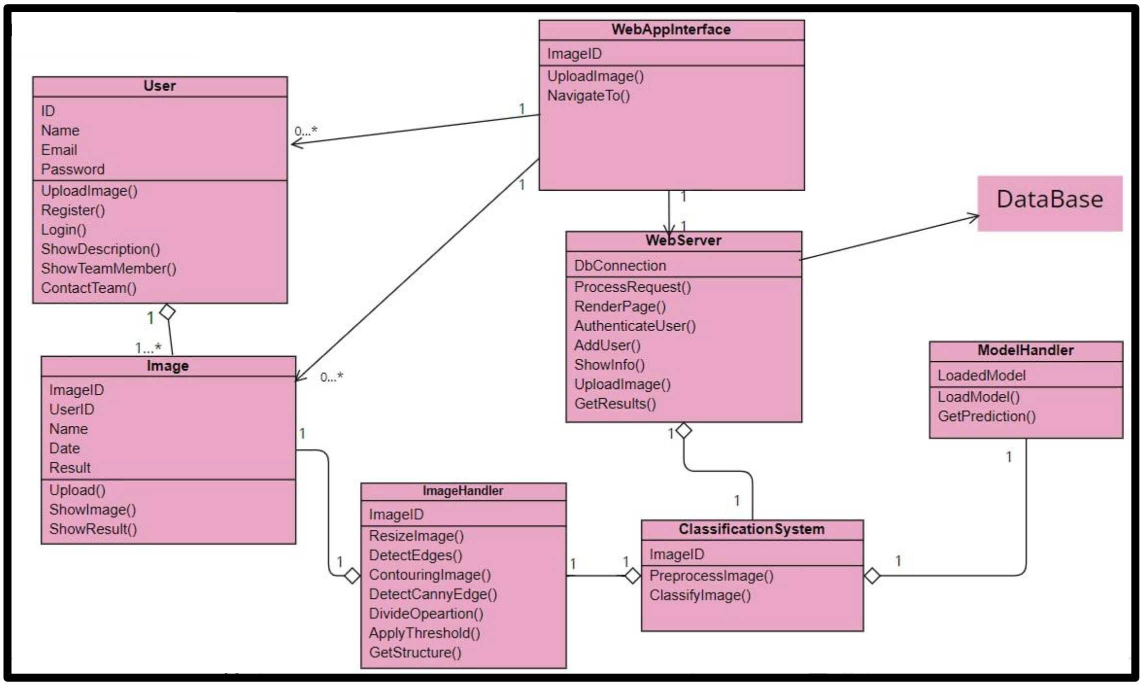


Figure 6. Class Diagram

4.3 Sequence Diagrams

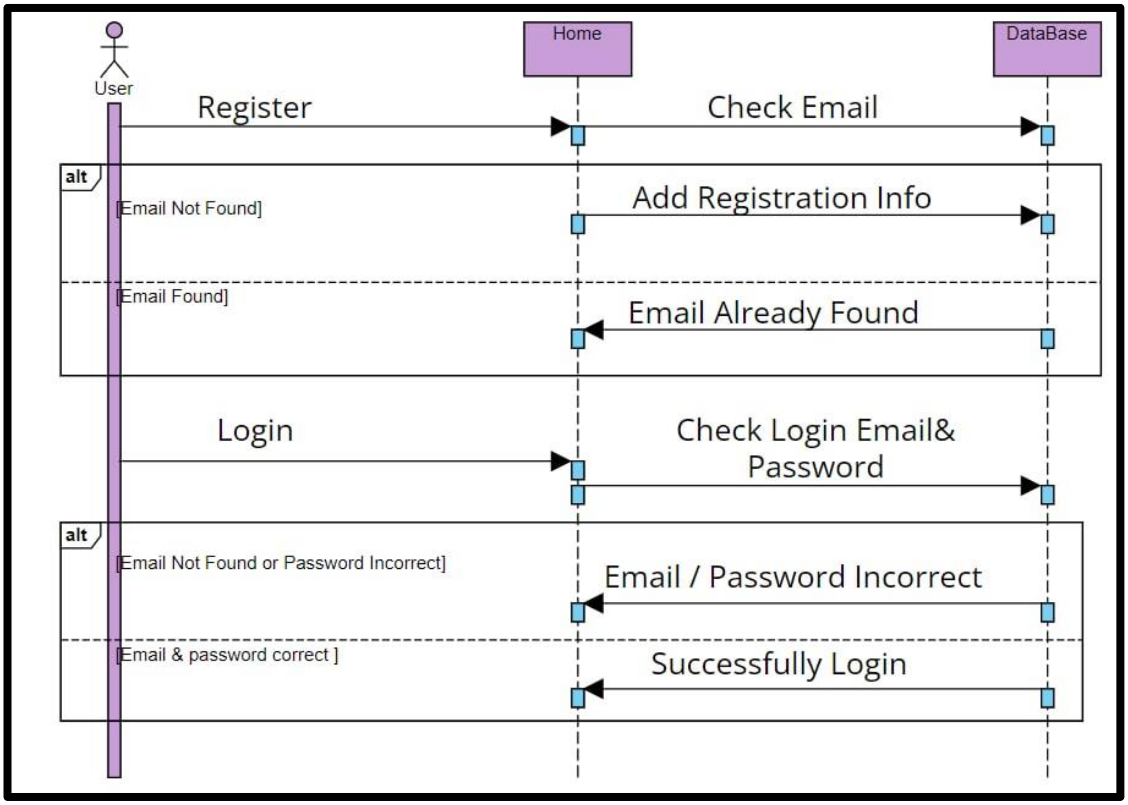


Figure 7. Sequence Diagram of Registration

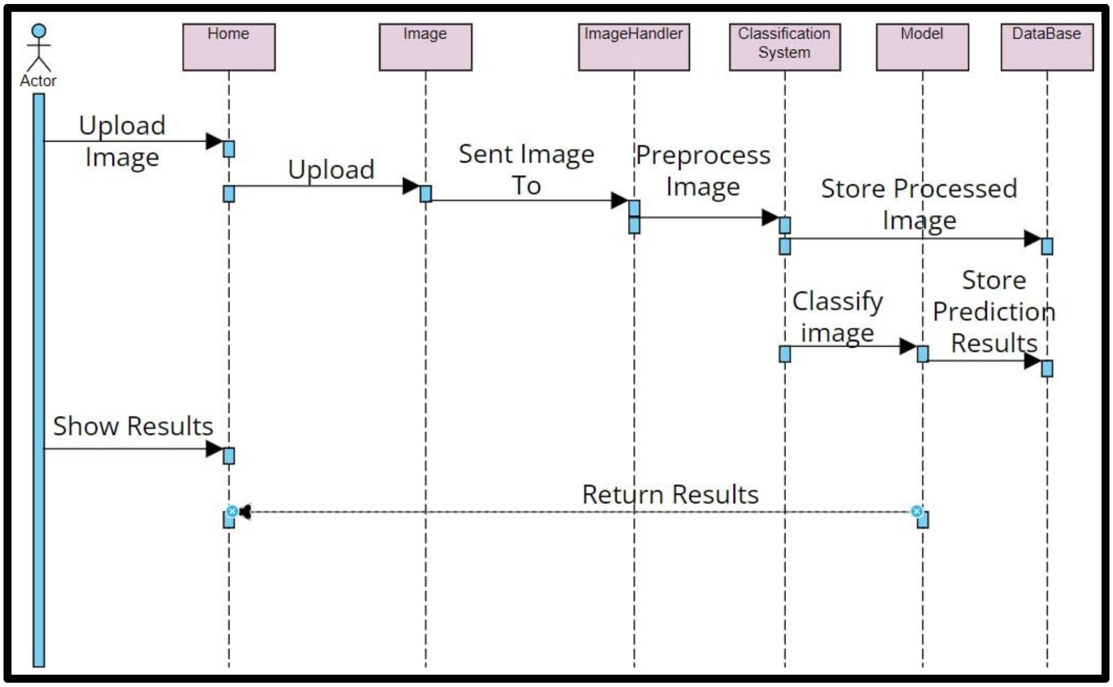


Figure 8. Sequence Diagram for Uploading Image Process

4.4 Project ERD

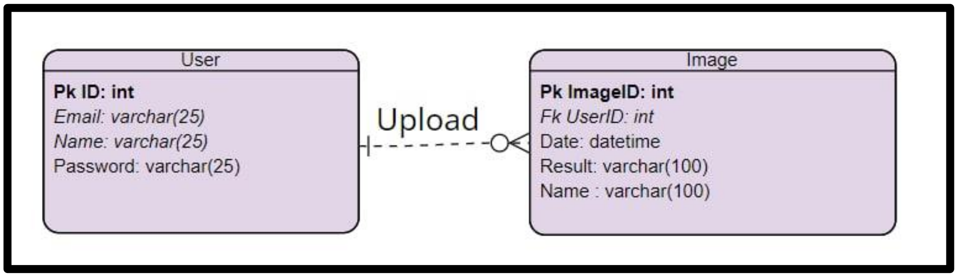
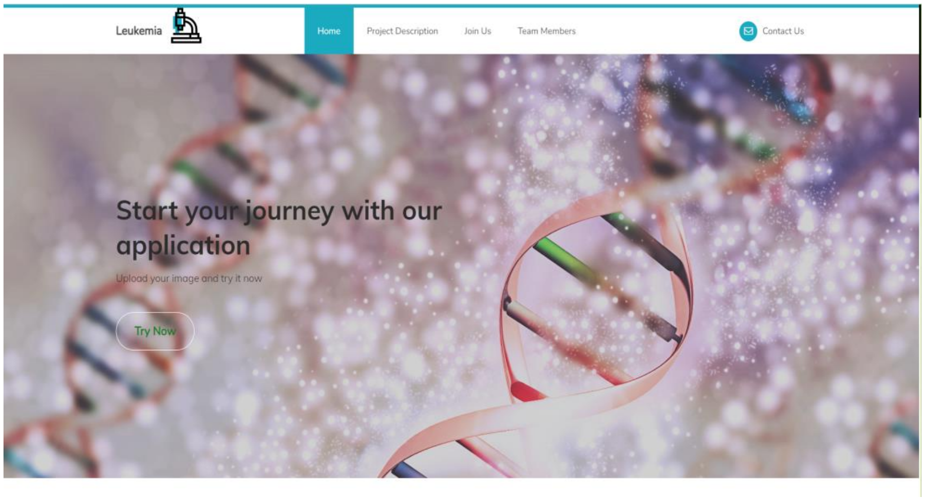


Figure 9 ER Diagram

4.5 System GUI Design

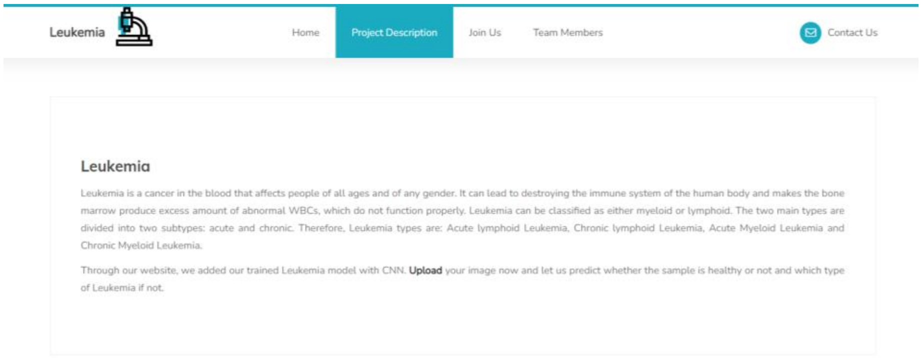
4.5.1 Home Page



Start your journey with our application

Upload your image and try it now

Figure 10. Home Page



Leukemia

Leukemia is a cancer in the blood that affects people of all ages and of any gender. It can lead to destroying the immune system of the human body and makes the bone marrow produce excess amount of abnormal WBCs, which do not function properly. Leukemia can be classified as either myeloid or lymphoid. The two main types are divided into two subtypes: acute and chronic. Therefore, Leukemia types are: Acute lymphoid Leukemia, Chronic lymphoid Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia.

Through our website, we added our trained Leukemia model with CNN. **Upload** your image now and let us predict whether the sample is healthy or not and which type of Leukemia if not.

Figure 11 About Page

4.5.2 Registration

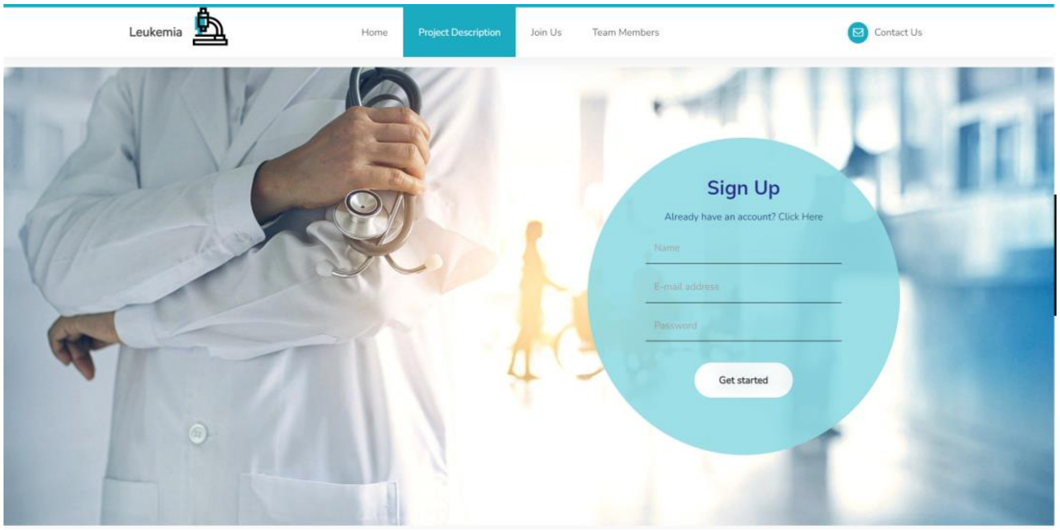
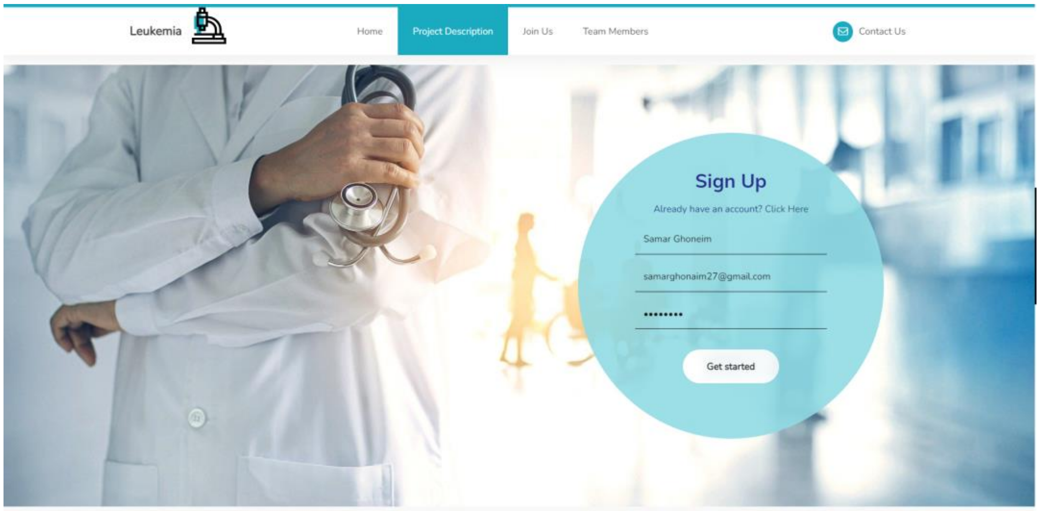


Figure 12. Registration Page



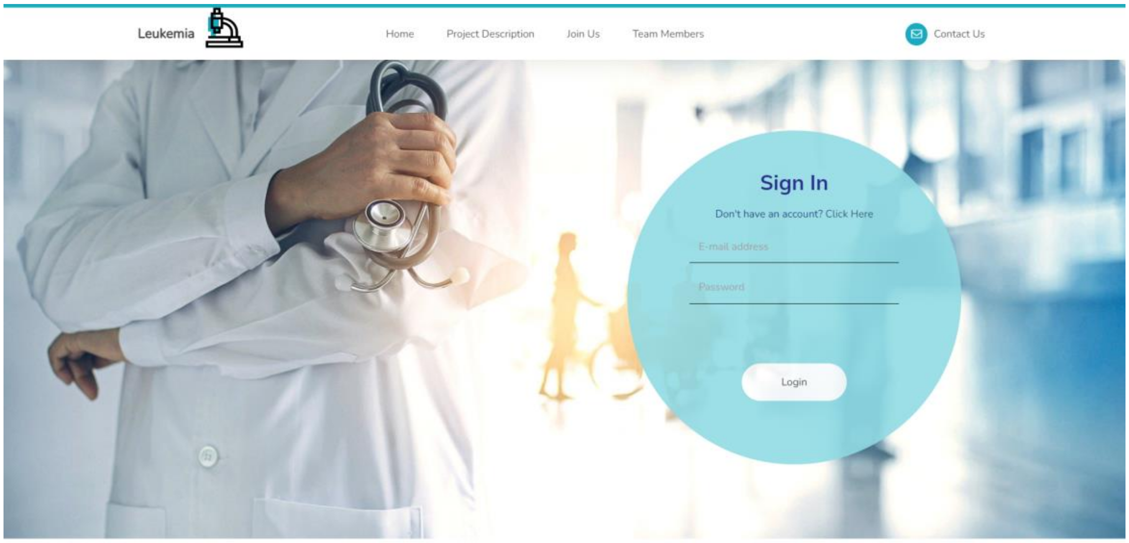
Samar Ghoneim

samarghonaim27@gmail.com

••••••••

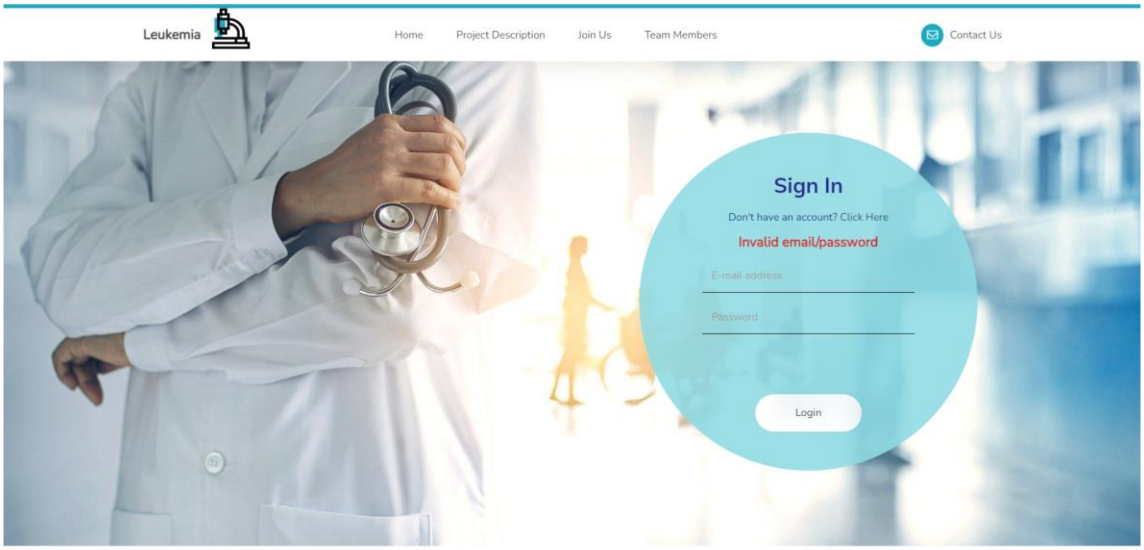
Figure 13 Registration Info

4.5.3 Login



Don't have an account? Click Here

Figure 14. Login Page

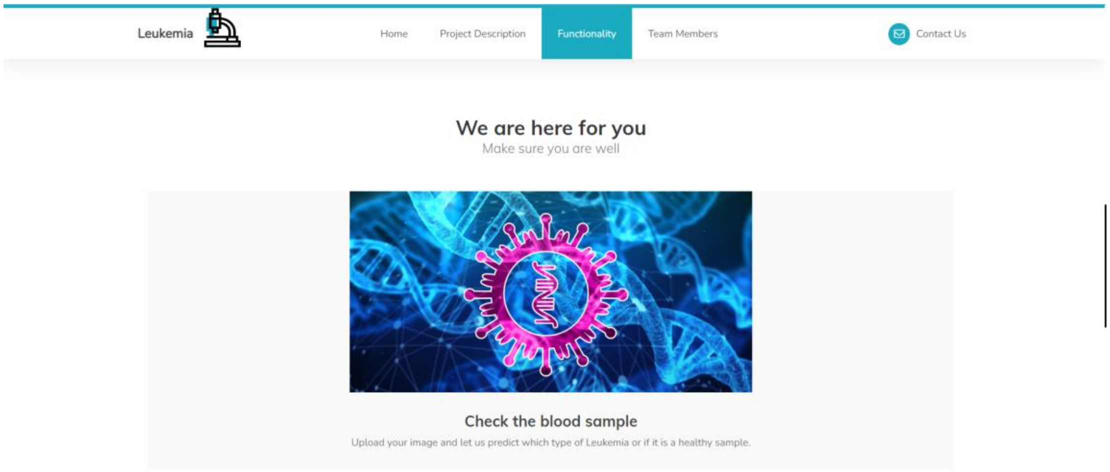


Don't have an account? Click Here

Invalid email/password

Figure 15. Error Handling

4.5.4 Functionality



We are here for you

Make sure you are well

Check the blood sample

Upload your image and let us predict which type of Leukemia or if it is a healthy sample.

Figure 16. Functionality Page

4.5.5 Upload

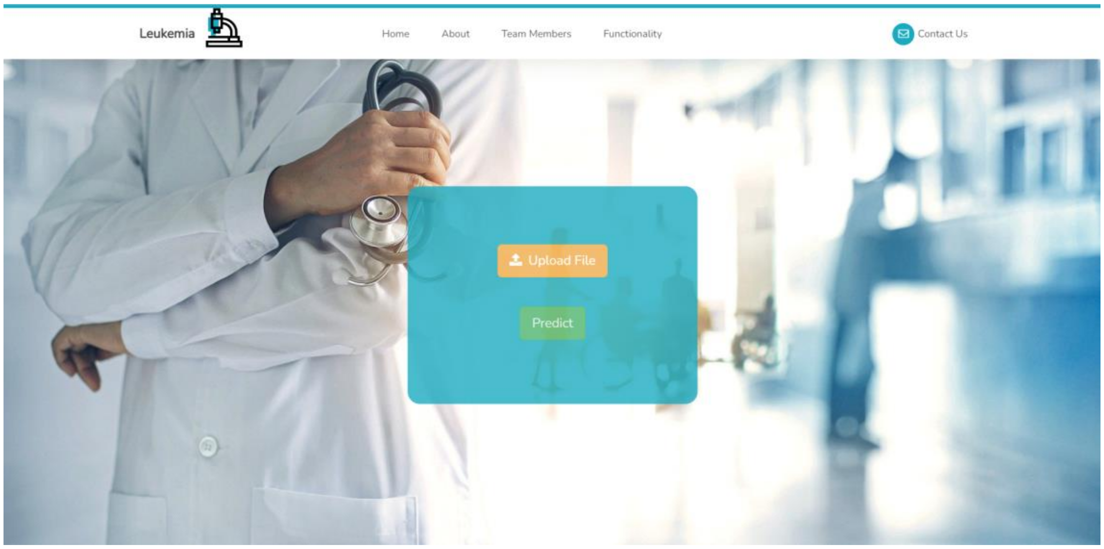
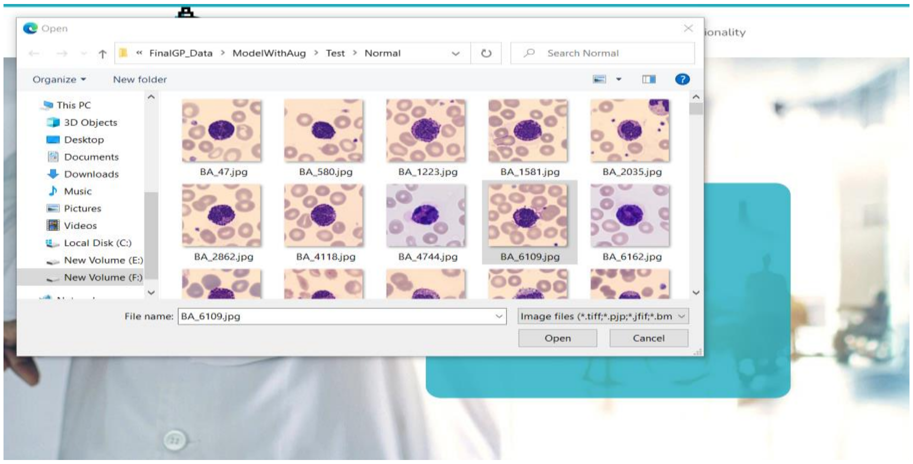


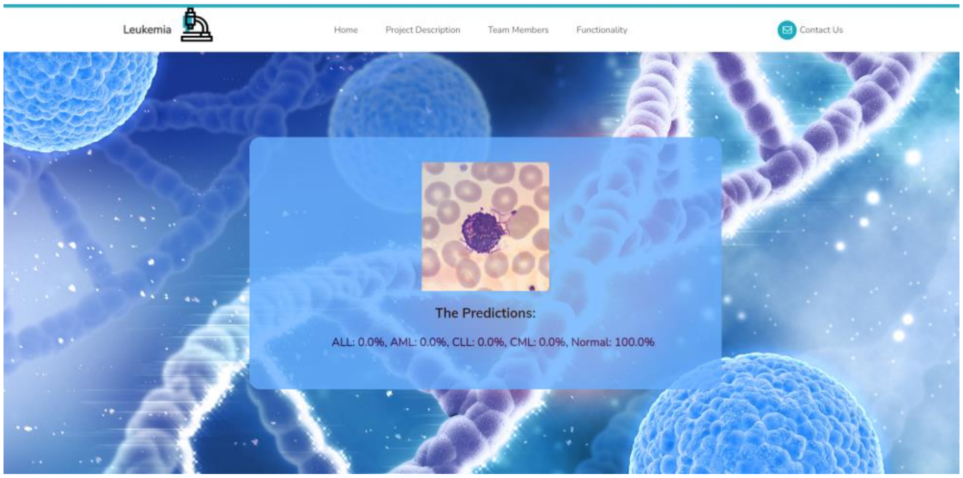
Figure 17. Upload Image Page



File name: BA_6109.jpg

Figure 18. Choose Image from Device

4.5.6 Result



The Predictions:

ALL: 0.0%, AML: 0.0%, CLL: 0.0%, CML: 0.0%, Normal: 100.0%

Figure 19. Result Page

Chapter 5

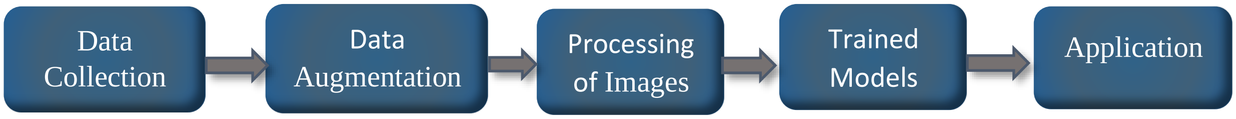
Methodology

Chapter 5

Methodology

5.1 Pipeline

The flow of implementing our models and architectures started with collecting our datasets then applying some steps to preprocess these datasets to be able to train our models in. After finding the best model we then used it to implement our website application. The steps are shown in the following figure:



5.2 Dataset Description

We have collected two datasets, the first had a total of over 30 thousand images, and was collected from different source. The second had a total of over 22 thousand images and was collected from one source, a medical laboratory.

5.2.1 First Dataset

We have collected multiple microscopic blood images for each type of Leukemia and for normal cells. ALL has about 10 thousand images, AML has about 10 thousand images, CLL has 10 images, CML has 20 images, and normal has about 10 thousand images.

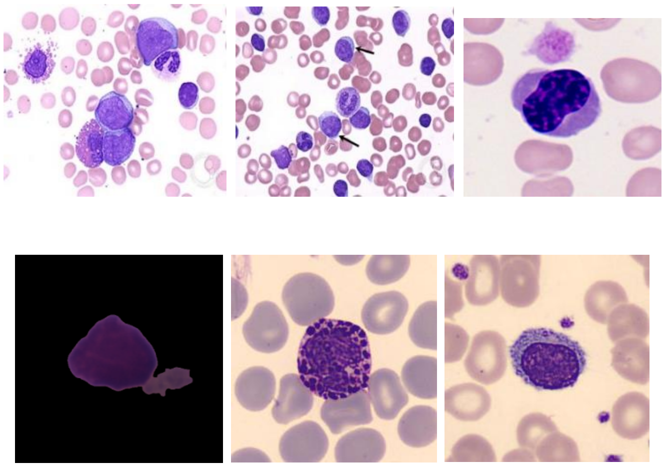


Figure 20. Samples Of First Data

5.2.2 Second Dataset

We have collected multiple microscopic blood images for each type of Leukemia. ALL has about 4 thousand images, AML has about 4 thousand images, CLL has about 3 thousand images, and CML has about 2 thousand images.

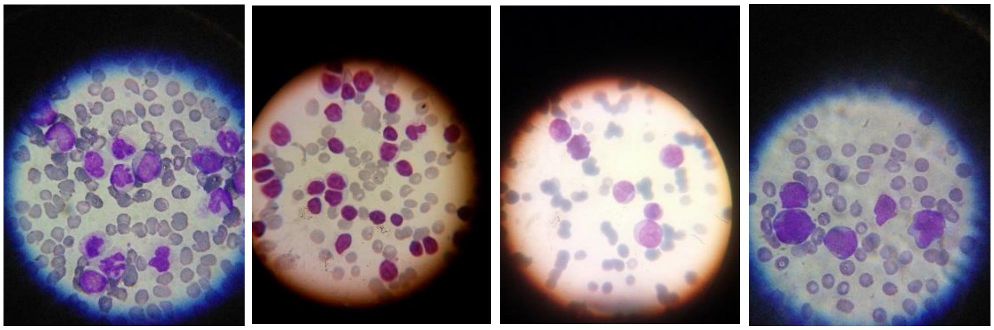


Figure 21. Samples Of Second Data

5.3 Data Augmentation

It is a technique of altering the existing data to create some more data for model training and testing processes. In other words, it is the process of artificially expanding the available dataset for training and testing a machine learning or a deep learning model.

We have only applied data augmentation on the first dataset due to the huge different in the number of images between CLL and CML types and the rest of the types (AML, ALL, and Normal). Unlike the second dataset where the number of images were closer to each other.

There are several ways to augment data, some of which we have applied; Flipping, Rotation, Translation, Scaling, Brightness, Saturation, Changing Color, and Cropping.

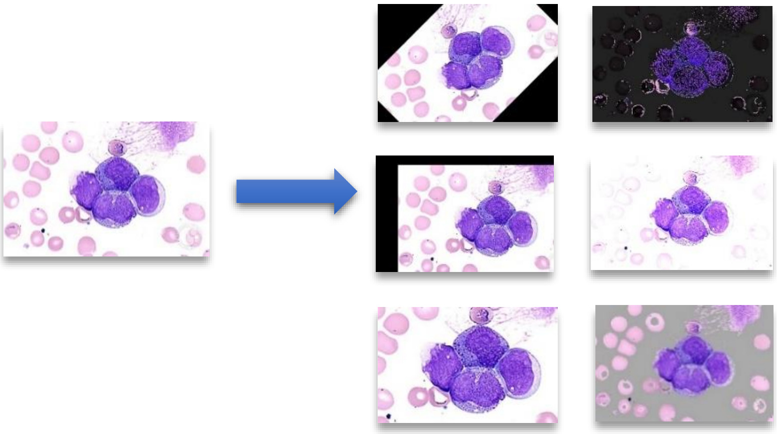


Figure 22 Data Augmentation of CML & CLL in first Dataset

5.4 Preprocessing

To be able to train the model, we first need to preprocess the images. First, we resized all images to be of the same dimensions, they ended up being 124x124 pixels. Then, converted them to the same file extension, jpg for the first datasets. Whereas for the second dataset, it already had the jpg extension. For the new dataset there was a lot of noise around the samples, so we removed it and converted them to png extension. For the deep learning models these steps were followed by the change of the images from RGBA to RGB for both datasets. However, for the machine learning models the preprocessing was as follows: Converting the images into grey scale, Feature Extraction using multiple methods (Edge detection using Gaussian blur and Canny Detection, Morphological transformation (divide, threshold, getStructure, etc.) and Image Contouring.) then images were flattened to enter the machine learning to be trained.

After converting, we applied data Augmentation to increase number of images. Subsequently, we flattened the array/matrix of the images and then sent it to the classifier in case of applying machine learning, the classifier will then output the result of the input images. However, in deep learning, we will go directly to the model layers, convolutional layer, max pooling, fully connected layer then to the output layer.

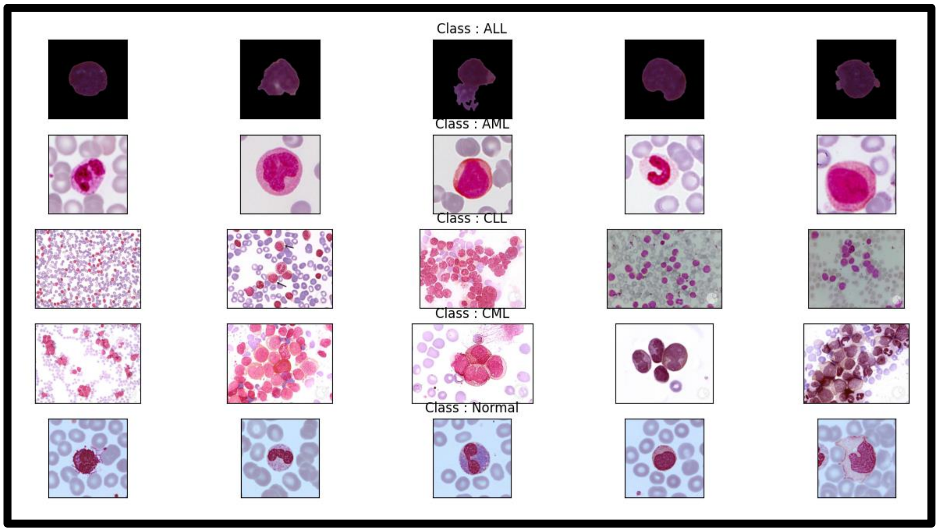


Figure 23. Preprocess Of First Data

5.5 CNN Architectures

5.5.1 Proposed CNN Architecture

The CNN architecture mainly consisted of convolution layers, pooling layers, flattening, and multilayer perceptron. CNNs performed automatic feature extraction from the input images and then classified them with fully connected neural networks. Feature extraction was performed by convolution and maxpooling layers. After applying filters on the image in these layers, the features were obtained and the classification stage was started. Figure NUM shows the details of the architecture we designed. The details of each phase are described below:

- *Convolutional Layer:* It was responsible for applying several feature detectors to explore many filters on the input image. The CNN we used had a 32 feature map with the size of 3×3. Convolution filters were applied to the image by sliding. The filter values were determined randomly. We used 2 convolution layers to avoid overfitting.

- *Max-Pooling Layer:* This layer was responsible for decreasing the dimension of the filtered image thus that it focused on the important feature/area or object in the image. In our network, we used a max-pooling layer with the size of 2×2. We also used two max pooling in this layer as well.

- *Flatten Layer:* This layer transformed a 2-dimensional max-pooled matrix into one dimensional array thus that each cell of this array could be used as an input node for the fully connected network.

- *Fully Connected Network:* This part was a naive connected full forward network that consisted of one input layer (the flattened layer in our case), a hidden layer, and an output layer. In our model, the hidden layer consisted of 128 nodes with 10% Dropout. Due to a simple calculation, the ReLU activation function had been implemented. In the output layer, we setup two types of optimizers SGD (stochastic gradient descent) and ADAM optimizers one type at a time. We added 5 output nodes (each node represents each leukemia type and Normal samples) and all of them were controlled by a SoftMax activation function.

Our CNN model was trained with 5 epochs since this setup was more suitable with the sample amount of dataset we used. Various numbers of epochs were experimented to obtain the best performance results. We tried to increase the number of epochs to 20, however, it took more running time without significant progress in accuracy.

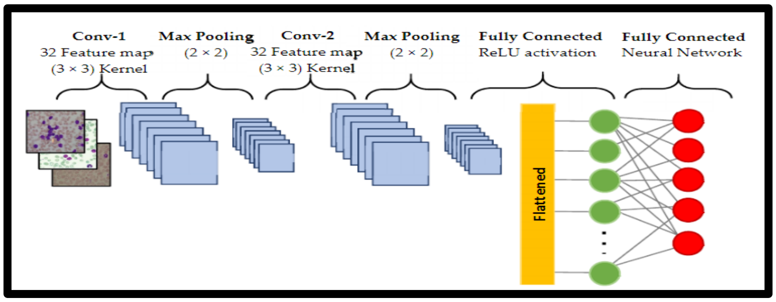


Figure 24. CNN Model Architecture

5.5.2 Alexnet

The Alexnet has eight layers with learnable parameters. The model consists of five layers with a combination of max pooling followed by 3 fully connected layers and they use Relu activation in each of these layers except the output layer.

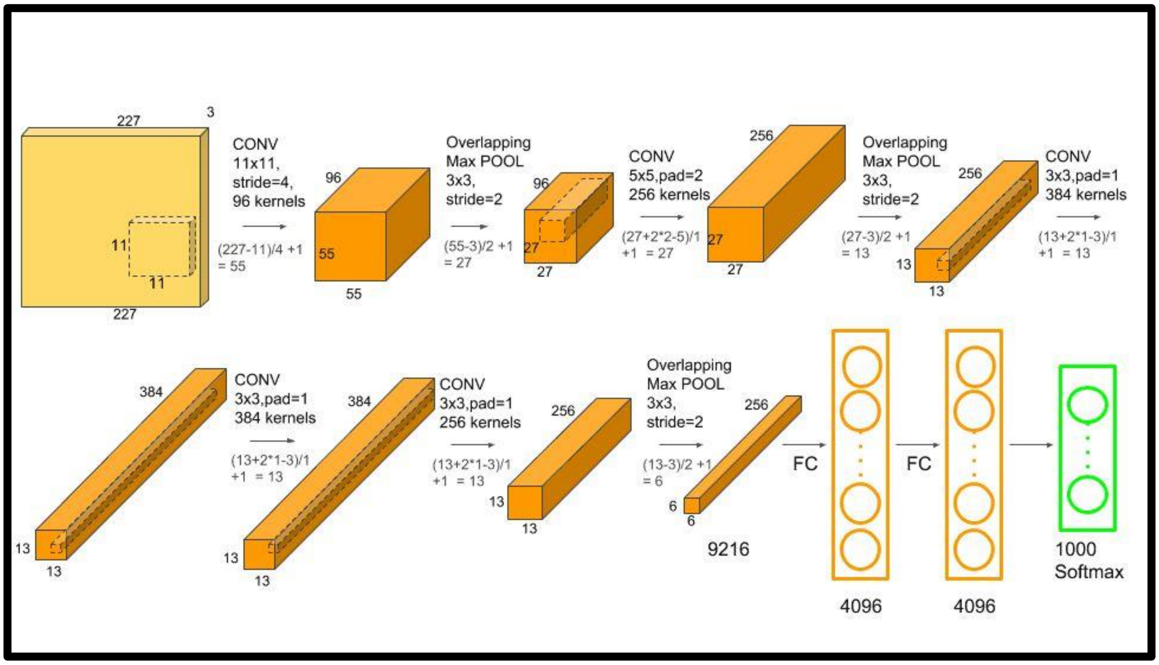


Figure 25. Alexnet Architecture

5.4.3 Resnet 50

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

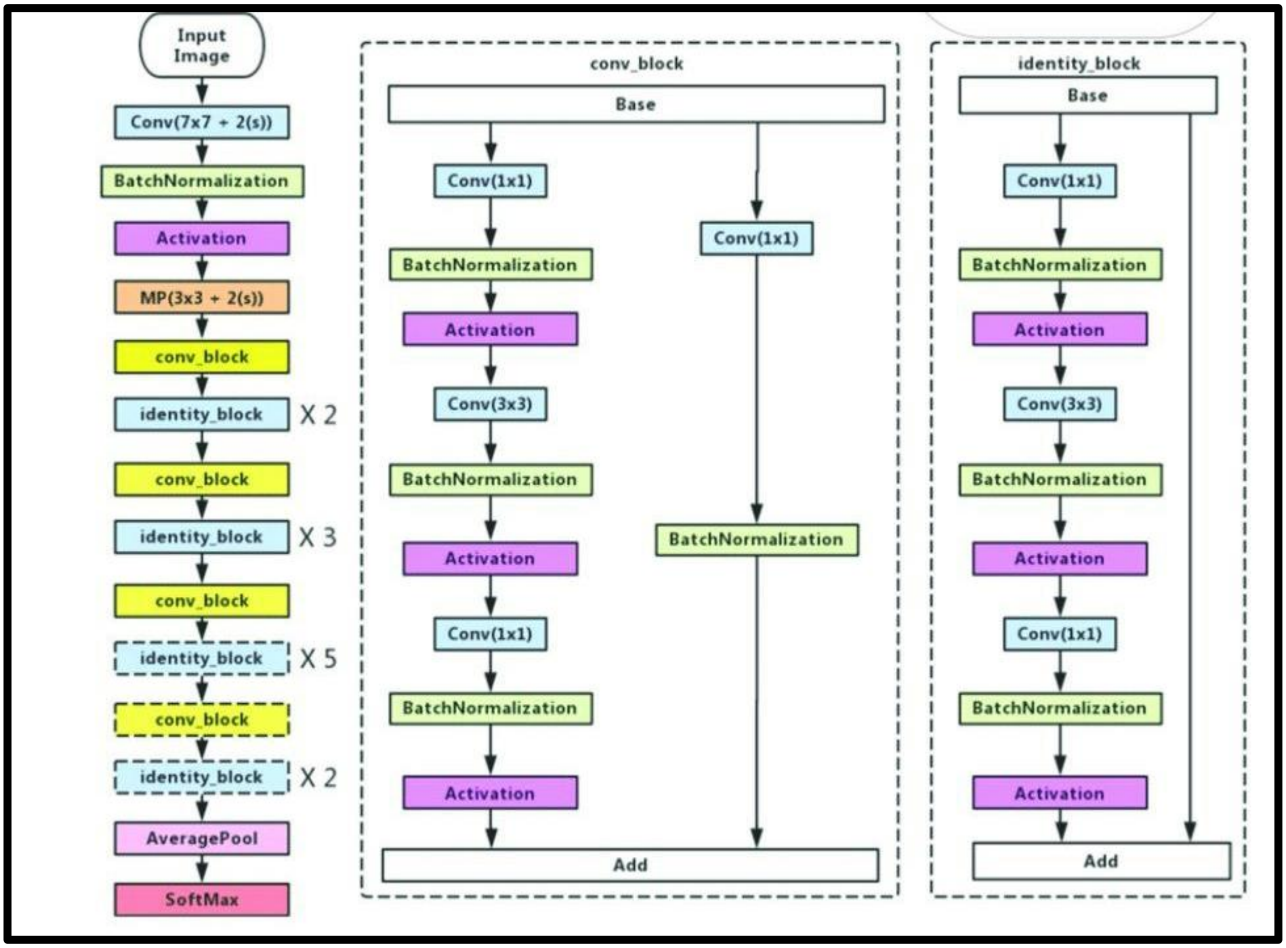


Figure 26. Resnet 50 Architecture

5.4.4 VGG-16

VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

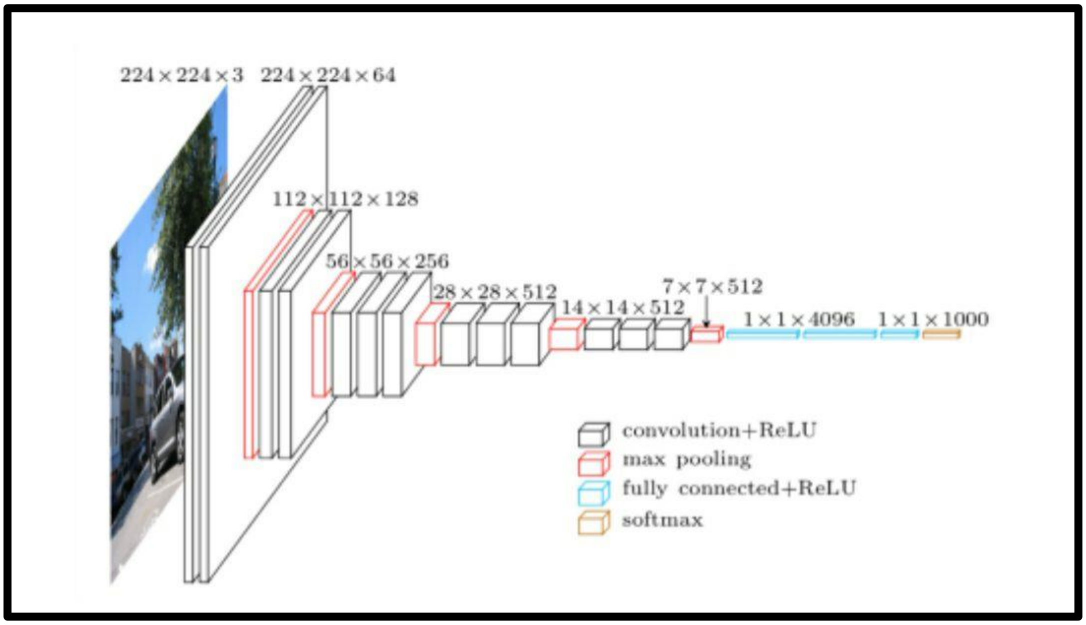


Figure 27. VGG-16 Architecture 1

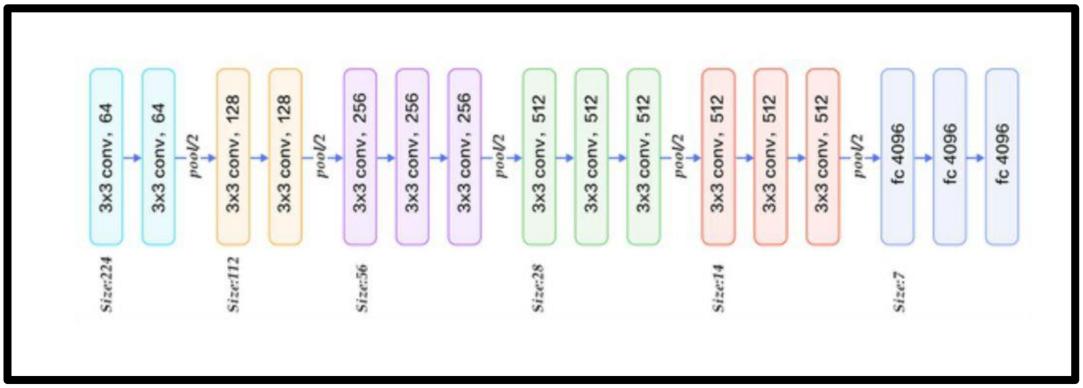


Figure 28. VGG-16 Architecture 2

5.4.5 Mobilenet

MobileNet model has 27 Convolutions layers which includes 13 depthwise Convolution, 1 Average Pool layer, 1 Fully Connected layer and 1 Softmax Layer.

In terms of Convolution layers, there are:

- 13 3x3 Depthwise Convolution
- 1 3x3 Convolution
- 13 1x1 Convolution

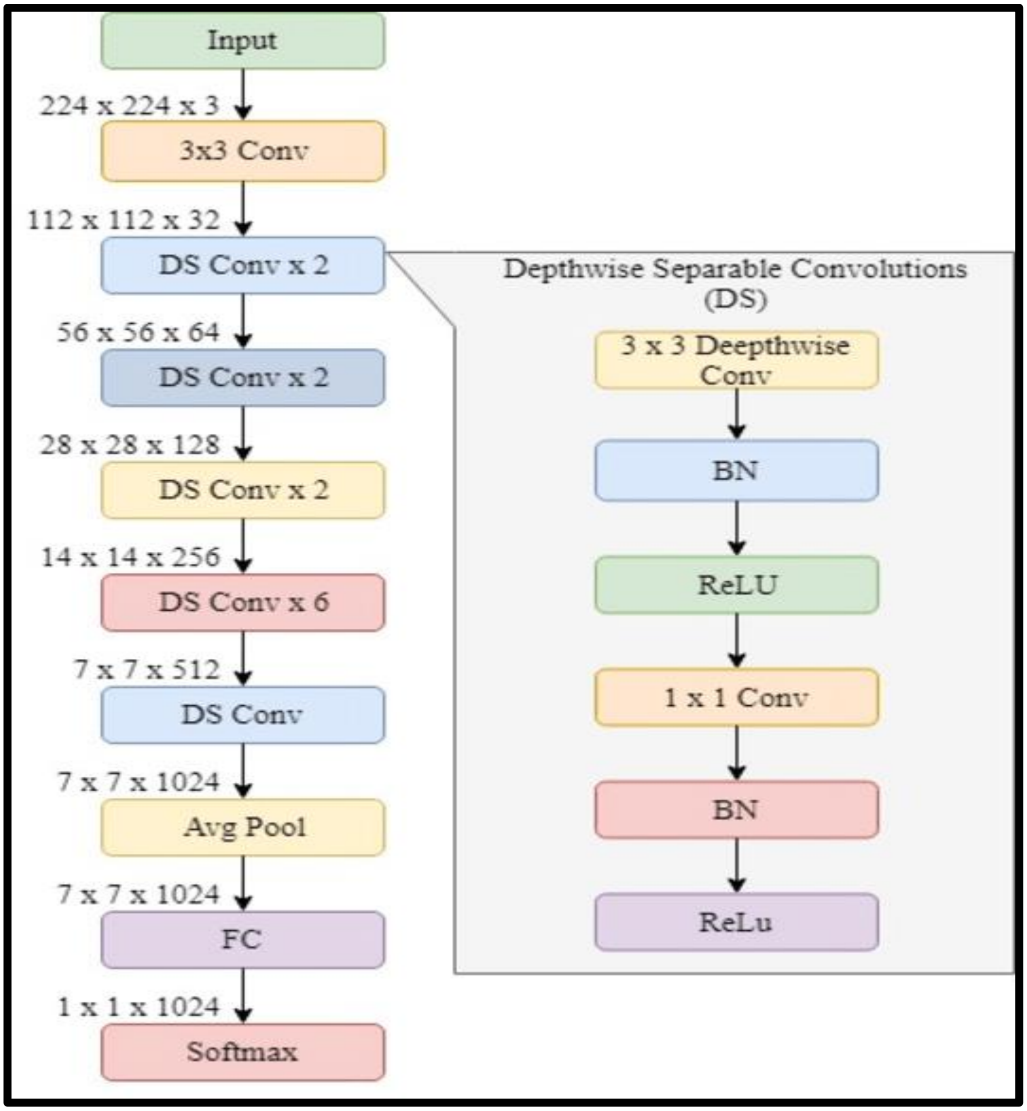


Figure 29. Mobilenet Architecture

Chapter 6

Implementation and Testing

Chapter 6

Implementation and Testing

6.1 Models Training and Testing

We have tried different datasets using different hyper parameters in both our machine learning models and out deep learning models. We tried different optimizers like SGD and ADAM. We changed the sizes of epoch from 2 to 25 and observe the difference in the learning process of the models. We have also tried different sizes for the feature map and for the filter size. We also tried binary classification on our datasets between normal and the leukemia types and observed the behaviour. Another experiment was mixing our datasets together and see if the model will be able to learn better and predict later on correctly. We also did a combination experiment between both our datasets and it gave us a high accuracy. Moreover, that combination helps us in being able to classify and detect any image that might be entered whether it is a direct image from the microscope or some preprocessing has been done or segmentation have been applied or even extracted single cells and testing it out. We also tried different preprocessing for the datasets. We have used different edge detection approaches like canny detection, gaussian blur and image contouring, also we applied some morphological transformation like getStructure, threshold, and divide to be used for features extraction.

We tested different machine learning algorithms like SVM, Random Forest, Logistic Regression, Decision Tree, Naïve Bayes, and KNN. For SVM, we kept changing the kernel type, the alpha, and the optimizer. For Random Forest, we tried different number for n_estimator and the depth of it and noticed the accuracies differences. For Logistic Regression, we changed the number of iterations and observed whether the model will be able to train better or not. For NB, we tried different types of it like Gaussian and Categorical and choose the best after that which was the Categorical one. For KNN, we altered the K size and chose the most suitable size based on accuracy, which was 3. We also tried cross validation and changed the split of the train and test.

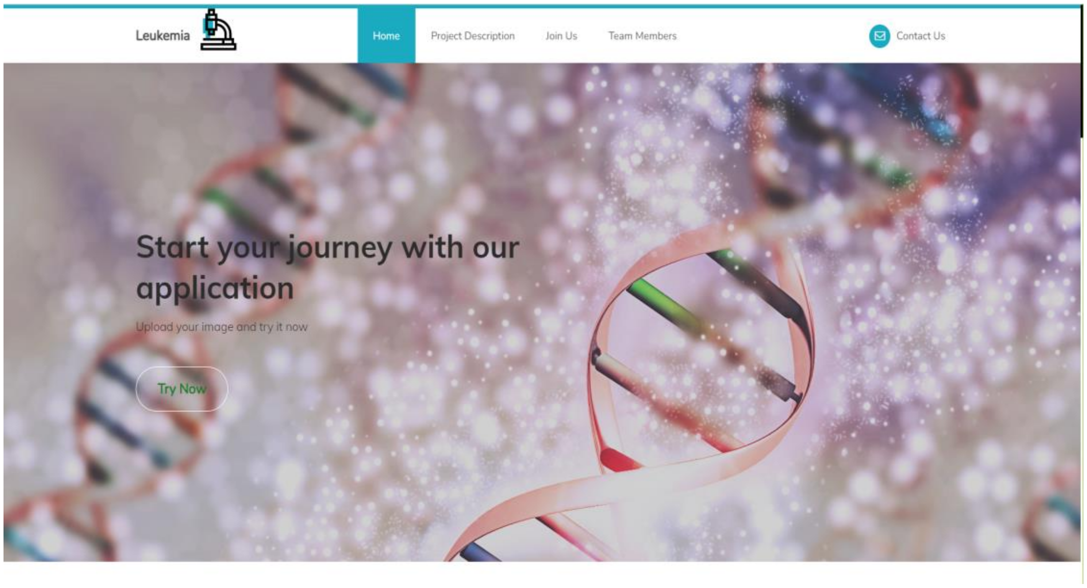
We tested different architecture with both built in architectures and a combination of layers for our CNN model. We tried AlexNet, ResNet50, VGG16, MobileNet, and normal CNN architecture. We tested different number of epochs, filter sizes, and filter number. We also tested different input sizes, 64x64, 124x124, and 224x224. The best was found to be 124x124 for input size, 2x2 for filter size, 64 for the number of filters, and number of epochs is 5.

For our CNN model, beside what was mentioned before, we also added different combinations of convolution layers, pooling layers, hidden layer, and dropout number. The best was 2 conv-layers, 2-max-pooling, 1 hidden layer with 128 nodes, dropout of 0.1, activation function for hidden layer is relu, and activation function for output is softmax.

6.2 Website Application

6.2.1 Main Page

The main page will have multiple tabs accessible by the user. Through it, the member can scroll through the application features, view the about of the website and the info about leukemia. They will also be able to view the team members and register.



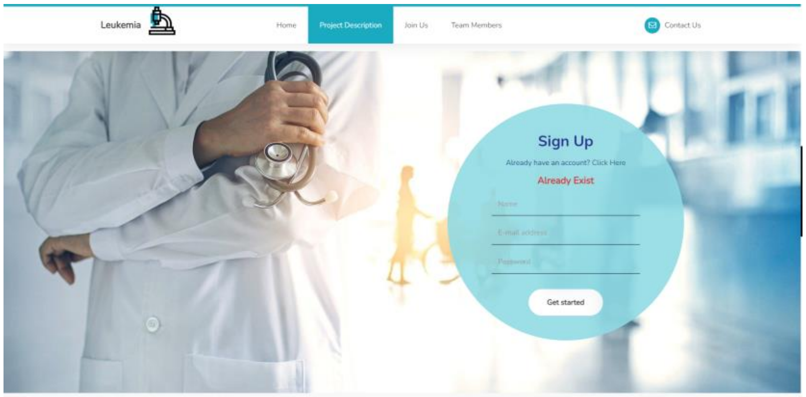
Start your journey with our application

Upload your image and try it now

Figure 30 Hope Page Try

6.2.2 Sign Up Page

The registration page requires Username, Email, and password of the user. Then we check that the email entered is not already saved in the database. If the email exists, an error message will be printed and showed to the user and will need to use another email or log in if it is their email. If the email does not exist, the user will be redirected to a page to be able to use the application.



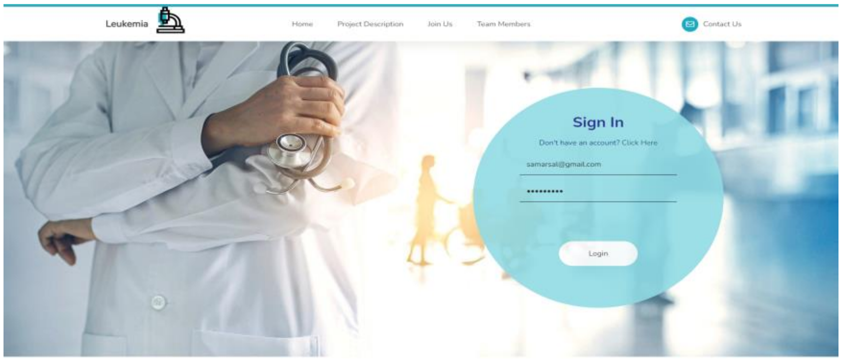
Already have an account? Click Here

Already Exist

Figure31 Error handling register

6.2.3 Login Page

The login page requires the email and password of the user. Then we take them and check if a user with that email and that password exist in the database. If it does, then the user will be redirected to our page for the use of our application. If it does not, then an error message will be printed that the email or password are incorrect. The user then will have the option to either register if they don't have an account or retry with the password and email if they already have an account.



Don't have an account? Click Here

samarsal@gmail.com

••••••••••

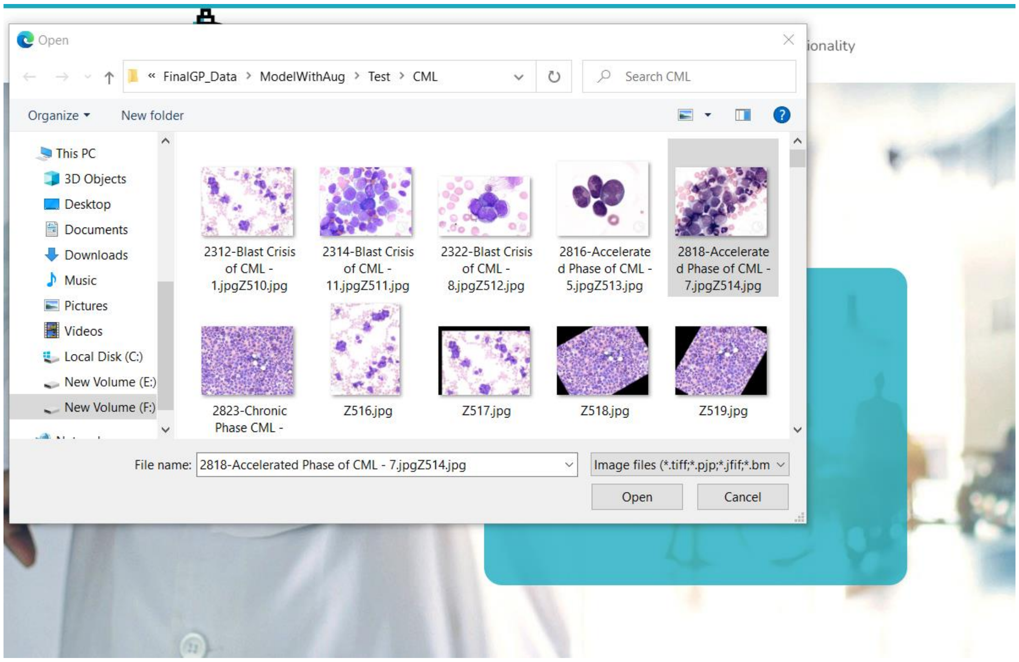
Figure32 Sign in page

6.2.4 Home Page

The main page will have multiple tabs accessible by the user. Through it, the member can scroll through the application features, view the about of the website and the info about leukemia. They will also be able to view the team members and access the website functionality.

6.2.5 Upload Page

The upload page will allow the user to upload an image and after pressing the predict button we will save some info about it. We will save the image, the user email, the result, and the date of uploading. The result will be retrieved after the image has gone through the needed preprocessing and been predicted by our used model.


Open
« FinalGP_Data > ModelWithAug > Test > CML
Search CML
Organize ▾
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
New Volume (E:)
New Volume (F:)
2312-Blast Crisis of CML - 1.jpgZ510.jpg
2314-Blast Crisis of CML - 11.jpgZ511.jpg
2322-Blast Crisis of CML - 8.jpgZ512.jpg
2816-Accelerated Phase of CML - 5.jpgZ513.jpg
2818-Accelerated Phase of CML - 7.jpgZ514.jpg
2823-Chronic Phase CML -
Z516.jpg
Z517.jpg
Z518.jpg
Z519.jpg
ionality
Open
Cancel

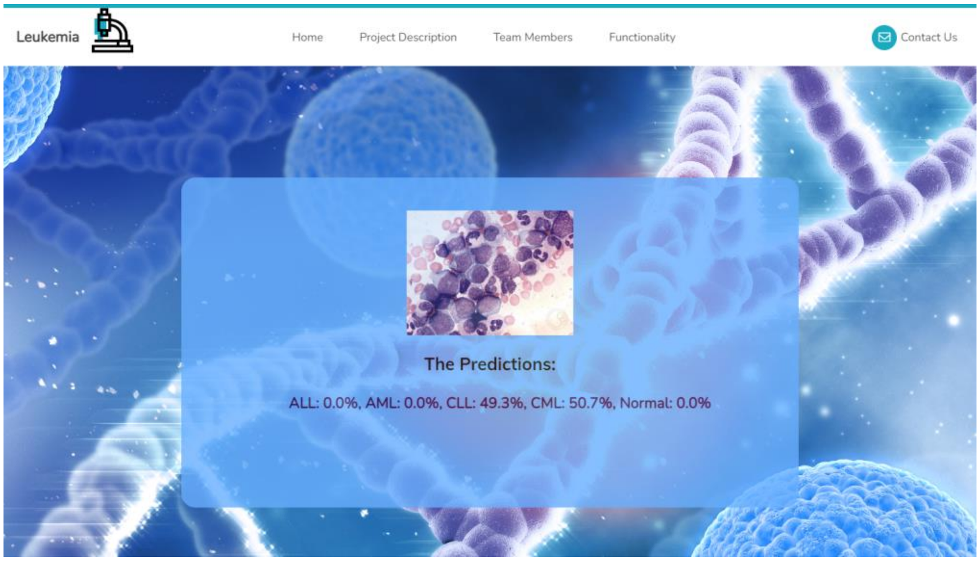
File name: 2818-Accelerated Phase of CML - 7.jpgZ514.jpg

Image files (*.tiff;*.pjp;*.jfif;*.bm

Figure 33. uploading an image

6.2.6 Result Page

In the result page, the image the user uploaded plus our model's accuracy will be printed back to the user.



The Predictions:

ALL: 0.0%, AML: 0.0%, CLL: 49.3%, CML: 50.7%, Normal: 0.0%

Figure 34. Prediction Result

Chapter 7

Results and Conclusion

Chapter 7

Results and Conclusion

7.1 Machine Learning Results

We noticed that the old dataset using machine learning produced higher accuracies, due to the finding the preprocessing that helped achieving that.

Table 1. Train Accuracies of ML

	Old Data	New Data
SVM	97.3%	97.5%
Logistic Regression	99%	96.5%
Random Forest	99.9%	98.8%
Naïve bayes	85.4%	73%

Table 2. Test Accuracies of ML

	Old Data	New Data
SVM	92%	71.6%
Logistic Regression	86%	69.5%
Random Forest	92.8%	73.3%
Naïve bayes	83.6%	61%

Table 3 Types Accuracies Old Data Machine Learning

	ALL	AML	CLL	CML	Normal
SVM	100%	90.1%	15%	67.2%	92.6%
Logistic Regression	99.6%	80.1%	22.5%	55.19%	73.5%
Random Forest	99.9%	91.8%	4%	60.9%	93.9%
Naïve bayes	100%	65.4%	2%	60.5%	93.2%

Table 4 Types Accuracies New Data Machine Learning

	ALL	AML	CLL	CML	Normal
SVM	91.7%	8.8%	7.3%	59.3%	99.7%
Logistic Regression	82.2%	11.5%	7.85%	60.6%	99.7%
Random Forest	98.4%	7.9%	8.1%	68.8%	99.7%
Naïve bayes	13%	54%	8.1%	2%	99.7%

7.2 Deep Learning Results

We noticed that the old dataset using deep learning produced higher accuracies.

Table 5. Train Accuracies of DL

	Old Data	New Data
CNN (32, 3x3)	99%	96%
CNN (64, 2x2)	99.8%	94.5%
AlexNet	99.8%	99.7%
MobileNet	99.9%	96.8%
ResNet50	91%	99%
VGG16	91%	88%

Table 6. Test Accuracies of DL

	Old Data	New Data
CNN (32, 3x3)	96%	82%
CNN (64, 2x2)	98.5%	90%
AlexNet	97%	83%
MobileNet	72.5%	70%
ResNet50	79%	89%
VGG16	80%	71%

Table 7 Types Accuracies Old Data Deep Learning

	ALL	AML	CLL	CML	Normal
CNN (64, 2x2)	99.8%	100%	6.8%	84.2%	98.5%
CNN (32, 3x3)	99.9%	99.7%	68.4%	21.8%	98.5%
AlexNet	100%	96%	9.4%	98.3%	99.6%

Table 8 Types Accuracies New Data Deep Learning

	ALL	AML	CLL	CML	Normal
CNN (64, 2x2)	48.2%	57.5%	0%	49.2%	99.7%
CNN (32, 3x3)	100%	12.3%	0%	58.4%	99.7%
AlexNet	100%	11.2%	7.1%	29.7%	99.7%

7.3 Best Architectures

The following graphs show the Bar Charts of the 3 best accuracies for train and test of each of the architectures and algorithms for Machine Learning (SVM, RFC, LR) and Deep Learning (CNN, AlexNet, ResNet50).

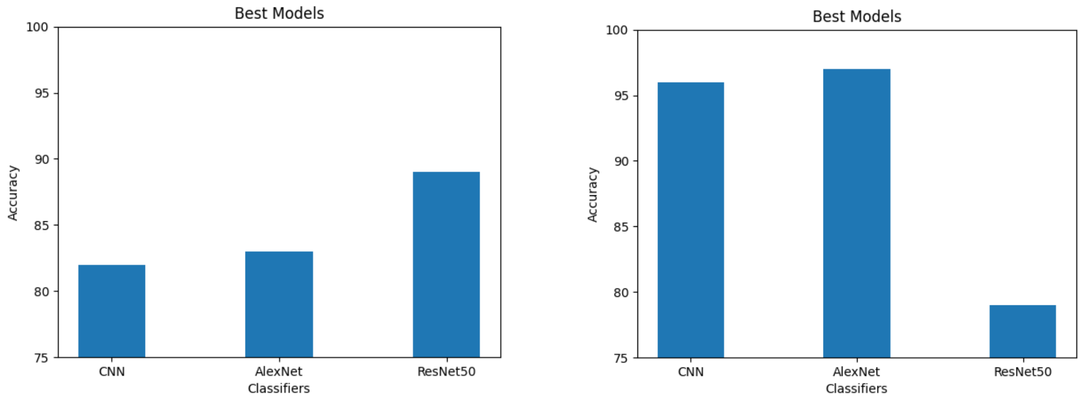


Figure 35 new and old data for deep learning

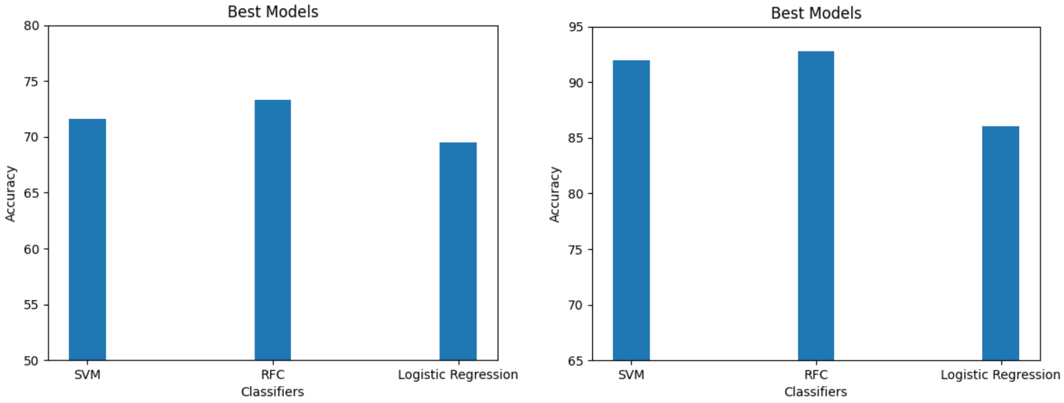


Figure 36 New and old data for Machine Learning

7.4 Conclusion

Leukemia is cancer of the body's blood-forming tissues. It can be fatal; thus, its early detection is important. Traditional Leukemia diagnosis and classifying takes more time, during which the patient’s situation could get worse. Machine learning and deep learning algorithms can be applied to microscopic blood images to detect leukemia, which will save time. In this project, we have trained, tested and evaluated several SVM and CNN models that aim at detecting and classifying leukemia.

After training our old and new data on different machine learning algorithms we found that Random Forest achieved the best accuracy. After training our old and new data on different deep learning models we found that on the old data Alexnet achieved the best accuracy, while the new data Resnet50 achieved the best accuracy.

For future work, we will keep up with any new available dataset to be able to easily detect any different type of datasets. Build mobile application to be easier for the specialist to upload the images, and be more usable.

References

- Huang, Furong PhDa; Guang, Peiwen MDa; Li, Fucui MDa; Liu, Xuewen MDa; Zhang, Weimin PhDb; Huang, Wendong PhDc, AML, ALL, and CML classification and diagnosis based on bone marrow cell morphology combined with convolutional neural network.

- Maneela Shaheen, Rafiullah Khan, R. R. Biswal, Mohib Ullah, Atif Khan, M. Irfan Uddin, Mahdi Zareei, and Abdul Waheed, Acute Myeloid Leukemia (AML) Detection Using AlexNet Model.

- C.E., IBM (2020, July 15). What is machine learning? IBM. Retrieved February 16, 2022, from https://www.ibm.com/eg-en/cloud/learn/machine-learning

- Gupta, A., & Gupta, R. (2019). ALL Challenge dataset of ISBI 2019 [Data set]. The Cancer Imaging Archive.

- Acevedo, Andrea; Merino, Anna; Alferez, Santiago; Molina, Ángel; Boldú, Laura; Rodellar, José (2020), "A dataset for microscopic peripheral blood cell images for development of automatic recognition systems".

- The American Society of Hematology. Available online: imagebank.hematology.org

- Matek, C., Schwarz, S., Marr, C., & Spiekermann, K. (2019). A Single-cell Morphological Dataset of Leukocytes from AML Patients and Non-malignant Controls [Data set]. The Cancer Imaging Archive.

- Ahmed, N., Yigit, A., Işik, Z., &amp; Alpkocak, A. (2019). Identification of leukemia subtypes from microscopic images using convolutional neural network: Semantic scholar.

- R. Free Data. Raabin Health Database. https://raabindata.com/free-data/#chronic-myelogenous-leukemia

Appendix A

Leukemia Detection and Classification

Esraa Ramdan, Nada Samir, Samar SalahElDin Ghoneim, Toka Mohamed, Somaya Yasser
Dr. Ahmed Fraouk, Dr. Hanaa Mobarz, TA. Sara ElNady

Abstract

Leukemia is a blood cancer that can be fatal in most cases. Detection and classification of Leukemia into which type the patient has can take a lot of time and resources for such process to be done. There are multiple stages for the process to know whether the patient is affected or not and which type do they have to determine after the prognosis and which treatment is needed.

We used two datasets one that had 5 publicly available data sources for each Leukemia type; Acute lymphocytic Leukemia, Chronic lymphocytic Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia, and normal blood cells. The second had 4 leukemia types; Acute lymphocytic Leukemia, Chronic lymphocytic Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia. Data Augmentation was then applied when needed. Finally, we want to show the variety of performance between CNN architecture models and other well-known machine learning algorithms.

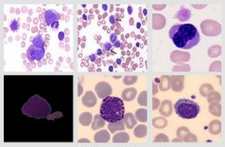
Introduction

Leukemia is a cancer in the blood that affects people of all ages and of any gender. It can lead to destroying the immune system of the human body and makes the bone marrow produce excess amount of abnormal WBCs, which do not function properly. Leukemia can be classified as either myeloid or lymphoid. The two main types are divided into two subtypes: acute and chronic. Therefore, Leukemia types are: Acute lymphoid Leukemia, Chronic lymphoid Leukemia, Acute Myeloid Leukemia and Chronic Myeloid Leukemia.

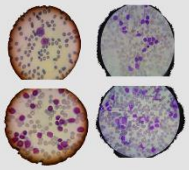
The traditional way to diagnose leukemia requires numerous amounts of steps which require a lot of time and resources. Therefore, we are taking a machine learning/deep learning approach to do these steps in less time and less energy taken. Given a dataset of microscopic blood images, we want to train and build several models than can detect Leukemia in these images and classify it as well.

Through our study, we are working on Detection of Leukemia (whether the patient is affected or not) and Classification into its types if affected (ALL, AML, CML and CLL). We are approaching the problem by applying machine learning and deep learning on our data. We are going to use SVM algorithm and CNN algorithm.

By using the model that is going to be created, specialists will save both time and effort for a primary detection and classification of the patient status regarding Leukemia. We are taking both an image processing approach and a bioinformatics one then comparing the results of both of them.



First dataset



Second dataset

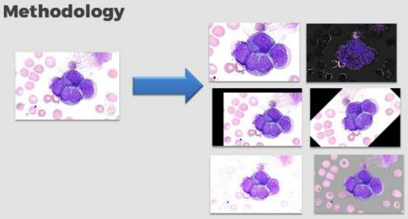
Methodology

Firstly, We have applied data augmentation on the first dataset only due to the huge different in the number of images between CLL and CML types and the rest of the types (AML, ALL, and Normal). Unlike the second dataset where the number of images were closer to each other. We used several ways to augment the data, such as; Flipping, Rotation, Translation, Scaling, Brightness, Saturation, Changing Color, and Cropping.

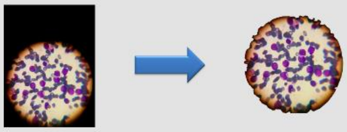
Secondly, We have applied some preprocessing on the image before it being trained on the models we used. We resized all the images to be of the same dimensions, they ended up to be 124x124 pixels. Then, converting them to the same file extension, jpg for the first datasets. Whereas for the second dataset, it already had the jpg extension. For the new dataset there was a lot of noise around the samples, so we removed it. For the deep learning models these steps were followed by the change of the images from RGBA to RGB for both datasets. However, for the machine learning models the preprocessing was as follows: Converting the images into grey scale, Feature Extraction using multiple methods (Edge detection using Gaussian blur and Canny Detection, Morphological transformation (divide, threshold, getStructure, etc.) and Image Contouring.)

We have performed Machine Learning and Deep Learning on both of our datasets. For machine learning we used SVM, Logistic Regression and Random Forest. For deep learning we used CNN and known architectures like VGG16, AlexNet, MobileNet, and ResNet50.

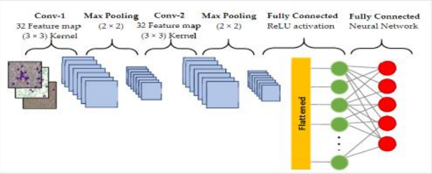
Methodology



Augmentation



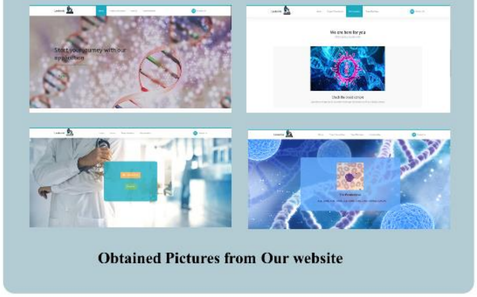
Preprocessing



CNN model architecture

Primary Design

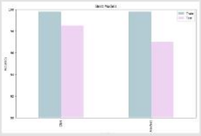
We built a user friendly website where the user can upload images of the samples needed for classification using our model.



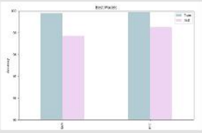
Obtained Pictures from Our website

Conclusion

After using multiple models and architectures, the best architecture for first dataset was using our **CNN model** with accuracy of **98.5%** followed by **AlexNet** with accuracy of **97%**. However, machine learning produced better results on the second dataset with **Random forest** yielding accuracy of **98.5%** followed by **SVM** with accuracy of **97.7%.**



Train/Test Accuracies
on First dataset



Train/Test Accuracies
on Second dataset