A Project Report

on

BONE MARROW CANCER DETECTION FROM LEUKOCYTES USINGNEURAL NETWORKS

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

19WH1A0578	**R. SHREYA**
19WH1A0582	**M. LEKHYA SRI**
19WH1A0583	**T. MILI PREETHIKA**

Under the esteemed guidance of

Ms. A. KRANTHI

Assistant Professor

Department of Computer Science Engineering

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Bachupally, Hyderabad – 500090

June, 2023

DECLARATION

We hereby declare that the work presented in this project entitled "**BONE MARROW CANCER DETECTION FROM LEUKOCYTES USING NEURAL NETWORKS**" submitted towards completion of Project Work in IV year of B.Tech., CSE at "BVRIT HYDERABAD College of Engineering For Women", Hyderabad is an authentic record of our original work carried out under the guidance of Ms. A. Kranthi, Assistant Professor, Department of CSE.

R. Shreya
(19WH1A0578)

M. Lekhya Sri
(19WH1A0582)

T. Mili Preethika
(19WH1A0583)

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Bachupally, Hyderabad - 500090

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the project work report on **"BONE MARROW CANCER DETECTION FROM LEUKOCYTES USING NEURAL NETWORKS"** is a bonafide work carried out by **R. Shreya (19WH1A0578), M. Lekhya Sri (19WH1A0582), T. Mili Preethika (19WH1A0583)** in partial fulfillment for the award of B.Tech degree in **Computer Science Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** under my guidance and supervision.

The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

Head of the Department
Dr. E. Venkateswara Reddy
Professor and HoD,
Department of CSE

Guide
Ms. A. Kranthi
Assistant Professor

External Examiner

ACKNOWLEDGEMENTS

We would like to express our sincere thanks to **Dr. K V N Sunitha**, **Principal, BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. E. Venkateswara Reddy, Professor & HOD, Department of CSE, BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Ms. A. Kranthi, Assistant Professor, Department of CSE, BVRIT HYDERABAD College of Engineering for Women** for her constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

R. Shreya
(19WH1A0578)

M. Lekhya Sri
(19WH1A0582)

T. Mili Preethika
(19WH1A0583)

CONTENTS

ABSTRACT

Leukocytes, produced in the bone marrow, make up around one percent of all blood cells. Uncontrolled growth of these white blood cells leads to the birth of blood cancer Acute lymphoblastic leukemia (ALL) is a type of cancer where the bone marrow forms too many lymphocytes. On the other hand, Multiple myeloma (MM), a different kind of cancer, causes cancer cells to accumulate in the bone marrow rather than releasing them into the bloodstream. The task of distinguishing lymphoblasts from normal white blood cells under the microscopeis however generally challenging, as morphologically the images of the two cell types appear similar. Our project is to predict where the cells are cancerous or normal using the neural networks. The Networks that would be used are ResNet50 and MobileNetV2.

LIST OF FIGURES

1. INTRODUCTION

Marrow is the sponge-like material inside your bones. Located deep within the marrow are stem cells, which can develop into red blood cells (RBCs), white blood cells (WBCs), or platelets. Each of them is made continuously in the bone marrow and released timely in the bloodstream. Bone marrow cancer happens when cells in the marrow begin to grow abnormally or at an accelerated rate. Cancer that starts in the bone marrow is called bone marrow cancer or blood cancer. Bone marrow cancer is distinct from bone cancer.

Bone marrow is an essential component of the body's hematopoietic system and is responsible for maintaining a healthy blood cell population. It plays a critical role in the immune system, oxygen transport, and blood clotting. Disorders or diseases affecting the bone marrow, such as leukemia, lymphoma, or myeloma, can disrupt the normal production of blood cells and have significant health implications.

Bone marrow is the infected area for a white blood cancer type cancer called Acute Lymphocytic Leukemia (ALL). Acute" indicates the fast progress of the disease, and if it does not get treated in the early stages, it might prove to be fatal within a short span. Acute Lymphoblastic Leukemia (ALL) is a fast-growing cancer of the blood and bone marrow, characterized by the development of many immature lymphocytes called lymphoblasts. It is the most common type of leukemia in children, particularly those between the ages of two and five, and the most common cause of death from pediatric cancers. Treatment depends on the type of ALL, age at diagnosis, and other factors.

Leukocytes, also known as white blood cells, are a crucial component of the immune system responsible for defending the body against infections, foreign substances, and abnormal cells, including cancer cells. They are produced in the bone marrow and circulate in the bloodstream, as well as other tissues and organs, playing a vital role in immune responses.
There are several types of leukocytes, each with distinct functions and characteristics:

Neutrophils: Neutrophils are the most abundant type of white blood cells and are essential for combating bacterial infections. They are highly mobile and can quickly migrate to sites of infection or inflammation. Neutrophils engulf and destroy bacteria and other foreign

substances through a process called phagocytosis.

Lymphocytes: Lymphocytes are responsible for adaptive immune responses and can be further classified into three main types: T cells, B cells, and natural killer (NK) cells. T cells play a role in cell-mediated immunity, coordinating immune responses and killing infected or cancerous cells. B cells produce antibodies that recognize and neutralize specific pathogens. NK cells are involved in the destruction of infected or abnormal cells without prior exposure.

Monocytes: Monocytes are large cells that can differentiate into macrophages or dendritic cells. Macrophages play a vital role in engulfing and digesting cellular debris, pathogens, and cancer cells. Dendritic cells present antigens to T cells, initiating immune responses.

Eosinophils: Eosinophils are involved in the immune response against parasites and play a role in allergic reactions. They release substances that can kill parasites and modulate inflammatory responses.

Basophils: Basophils are involved in allergic reactions and release substances such as histamine, which promotes inflammation and allergic responses.

Leukocytes can provide important diagnostic information in various medical conditions, including infections, autoimmune disorders, and cancers. Abnormalities in leukocyte count, distribution, or function can indicate underlying health issues. For example, leukocytosis, an elevated white blood cell count, may be a sign of infection or inflammation, while leukopenia, a decreased white blood cell count, can indicate immunodeficiency or bone marrow disorders.

In the context of bone marrow cancer prediction, analyzing the characteristics and behavior of leukocytes, such as their morphology, count, genetic alterations, and expression of specific markers, can help identify abnormal cell populations associated with cancer. Various techniques, including microscopic examination, flow cytometry, genetic analysis, and biomarker detection, can aid in the assessment and prediction of bone marrow cancers.

Understanding the functions and behaviors of leukocytes is essential for diagnosing and monitoring various medical conditions and plays a significant role in predicting and evaluating the progression of bone marrow cancer.

Deep Learning is known to demonstrate better functioning than accustomed Machine Learning for processing a large number of images. Convolution Neural Networks (CNNs) combine various multi-layer perceptron and display efficient results with a little pre-processing. CNN's themselves act as a feature extractor as each convolution layer of the network learns a new feature that is present in the images and hence produces a high activation.

This project aims to develop an automated classifier using neural networks to determine whether an individual is diagnosed with cancer or not.

1.1 Objective:

The objective of this project is to accurately identify the Cancer from White Blood Cells. Early detection and diagnosis are key to a good change for a cure. In order to reach a decisive conclusion for the diagnosis of the disease and the degree of progression, it is of paramount importance to identify malignant cells with high accuracy. Our motive is to proposea classifier that has best computational power.

1.2 Methodology:

1.2.1 Dataset

The dataset that was used is C-NMC 2019 Dataset [1] , was prepared at Laboratory Oncology, AIIMS, New Delhi, India.It consists of 10662 images of white blood cells with labels (normal versus cancer) that were segmented from microscopic images.

Images were collected from the data of 69 cancer subjects and 49 healthy subjects and were already split into training set, validation set, and test set.The percentages chosen were 60% for the training set, 20% for the validation set, and 20% for the test set. The images to be included in each of the sets were chosen randomly trying to maintain the data distribution of the original dataset. The “all” refers to the cancer images and “hem” refers to the normal images.

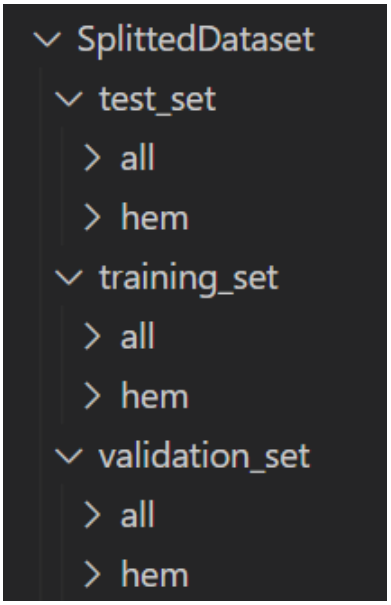
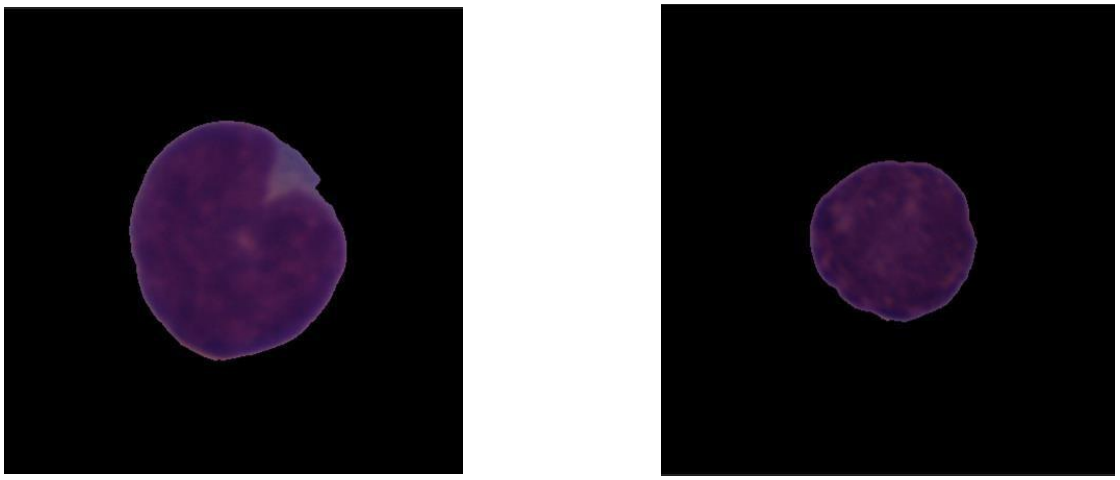


Fig 1 : **Dataset File Structure**

The sample structure of the directory is shown in figure 1.

Sample Data – The sample images of the training dataset are shown in figure 2.



Cancer cell **Normal cell**

Fig 2: Sample Data

1.2.2 The Proposed CNN Model

Bone marrow cancer, also known as hematological malignancy, refers to a group of cancers that affect the production and function of blood cells within the bone marrow. Early detection and accurate diagnosis of bone marrow cancer are crucial for effective treatment and improved patient outcomes. In recent years, neural networks have emerged as powerful tools for medical image analysis, including cancer detection. Two popular neural network architectures, ResNet50 and MobileNetV2, have shown promising results in various computer vision tasks and can be leveraged for bone marrow cancer prediction.

ResNet50:

ResNet50 is a deep convolutional neural network architecture that consists of 50 layers. It introduced the concept of residual connections, which enable the network to learn more efficiently and effectively. When applied to bone marrow cancer prediction, ResNet50 offers several benefits:

Deep architecture: The depth of ResNet50 allows it to capture intricate patterns and representations from bone marrow images, which can be crucial for accurate cancer detection. The network can learn hierarchical features that may correspond to subtle abnormalities associated with cancerous cells.

High accuracy: ResNet50 has achieved remarkable performance on large-scale image

classification tasks, including medical image analysis. Its depth and residual connections help mitigate the vanishing gradient problem, enabling the model to learn more complex representations and improve prediction accuracy.

Pre-trained weights: ResNet50 can leverage pre-trained weights from training on large-scale image datasets, such as ImageNet. This pre-training helps the model initialize with learned feature representations, making it easier to adapt and generalize to the bone marrow cancer detection task, even with limited training data.

MobileNetV2:

MobileNetV2 is a lightweight neural network architecture specifically designed for mobile and embedded devices. Despite its reduced complexity, MobileNetV2 offers several advantages for

bone marrow cancer prediction:

Efficiency: MobileNetV2 has a smaller model size and lower computational requirements compared to deeper architectures like ResNet50. This makes it suitable for resource-constrained environments or applications where real-time performance is crucial, such as mobile devices or point-of-care medical systems.

Speed: Due to its lightweight design, MobileNetV2 can perform inference faster than larger networks. This can be beneficial for processing large volumes of bone marrow images efficiently, enabling quicker diagnosis and reducing the turnaround time for patients.

Transferability: MobileNetV2 can also leverage pre-trained weights from larger datasets, facilitating transfer learning. This is particularly advantageous when training data for bone marrow cancer prediction is limited, as it allows the model to leverage general knowledge learned from other image datasets and improve performance.

Overall, both ResNet50 and MobileNetV2 offer unique advantages for bone marrow cancer prediction. ResNet50 excels in accuracy and capturing intricate patterns, while MobileNetV2 shines in efficiency and speed. The choice between the two models depends on the specific requirements of the application, available computational resources, and the trade-off between accuracy and efficiency.

1.3 Organization of Project

The technique which is developed is taking input as a image and extracts the features and filters from the bone marrow images, these features extracted are classified based on models trainings and experiences. The process followed is

Data Collection and Preprocessing:

- Gather a dataset of bone marrow images, including both cancerous and non-cancerous samples.
- Preprocess the images by resizing them to a consistent size, normalizing pixel values, and applying data augmentation techniques for increased diversity.

Model Selection and Setup:

- Choose between ResNet50 and MobileNetV2 based on requirements (accuracy vs. efficiency).
- Load the pre-trained weights of the selected model, if available, to leverage transfer

learning.

Model Architecture and Training:

- Design the architecture of the bone marrow cancer prediction model, incorporating the selected neural network (ResNet50 or MobileNetV2).
- Add any additional layers on top of the base model, such as fully connected layers or global pooling, to adapt it to your specific task.
- Split the dataset into training and validation sets.
- Train the model on the training set using appropriate loss functions and optimization algorithms.
- Monitor the training process and adjust hyperparameters as needed to optimize the model's performance.

Model Evaluation:

- Evaluate the trained model on the validation set to assess its performance in predicting bone marrow cancer.
- Compute evaluation metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC).
- Visualize and analyze the model's predictions and performance using confusion matrices or ROC curves.

And the image is finally classified as cancerous or not.

2. LITERATURE REVIEW

Title: Automated Diagnosis of Leukemia: A Comprehensive Review

Authors: Afshan shah , Syed saud naqvi , Khuram naveed , Nemasalem, Mohammad A. U. Khan and Khurram S. Alimgeer

Summary: The rapid production of abnormal white blood cells is called as leukemia. It will effect the blood and also damages the bone marrow in our body. We can even detect those by manual methods as well. As the manual methods are time-consuming the problems have been overcome by developing automated methods using CAD. Deep learning was used .by researches for automated diagnosis of leukemia.

Future Work: More efficient pre-processing and segmentation algorithms should be proposed that will work in a better way for high density noisy images which helps in achieving higher accuracy.

Title: Automatic Detection of white blood cancer from bone marrow microscopic images using convolutional neural networks

Authors: Deepika kumar, Nikita Jain, Aayush Khurana, Sweta Mittal, Suresh Chandra Satapathy, Roman Senkerik and Jude D. Hemanth

Summary: It is of two types - ALL and MM. In leukemia (ALL), the bone marrow produces an excessive number of lymphocytes. Multiple Myeloma (MM), a separate type of cancer, causes them to build up in the bone marrow. The model was trained with the optimized Dense Convolutional neural network framework (DCNN) and finally predicting the type of cancer present in the cells.

Future Work: A broader experimental study considering the dependence on the size of the databases should be performed.

Title: Automatic Diagnostic Tool for Predicting Cancer Grade in Bladder Cancer Patients Using Deep Learning

Authors: Rune Wetteland, Vebjørn Kvikstad, Trygve Eftestøl Erlend Tøssebro, Melinda lillesand , Emiel A. M. Janssen and Kjersti engan

Summary: Most common type of bladder cancer is Urothelial carcinoma, which is among the cancer types with the highest recurrence rate and lifetime treatment. The paper proposed a pipeline, called TRIgrade, that will identify diagnostic relevant regions in the whole-slide image (WSI) and collectively predict the grade of the current WSI

Future Work: The preprocessing steps will be added to deal with color variations, blur, and the tissue segmentation model will be updated with a new class for blur, providing a more generalized system.

Title: Feature Extraction of White Blood Cells Using CMYKMoment Localization and Deep Learning in Acute Myeloid Leukemia Blood Smear Microscopic Images

Authors: Tusneem Ahmed M. Elhassan, Mohd Shafry Mohd Rahim, Tan Tian Swee, Siti Zaiton Mohd Hashim and Mahmoud Aljurf

Summary: Acute myeloid leukemia (AML) diagnosis is a tedious protocol that is prone to human. In this study, a new hybrid feature extraction method was developed using image processing and deep learning methods. consists of two steps: 1) a region of interest (ROI) is extracted using the CMYK-moment localization method and 2) deep learning-basedfeatures are extracted using a CNN-based feature fusion method.

Future Work: The proposed model could be further improved by hybridization and parallelization methods to reduce the computation time and increase model accuracy.

Title: Exploiting the Multiscale Information Fusion Capabilities for Aiding the Leukemia Diagnosis Through White Blood Cells Segmentation

Authors: Nadeem Akram, Sharjeel Adnan, Muhammad Asif, Syed Muhammad Ali Imran, Muhammad Naveed Yasir, Rizwan Ali Naqvi and Dildar Hussain

Summary: Leukemia is one of the most terminal types of blood cancer.White blood cells (WBCs) have a significant association with leukemia diagnosis. WBC accurate segmentation enables to detect morphology and WBC count.Manual WBC assessment methods are tedious, subjective, and less accurate. To overcome these problems, they proposed a multi-scale

information fusion network (MIF-Net) for WBC segmentation. MIF-Netis a shallow architecture with internal and external spatial information fusion mechanisms.

Future Work: This paper further work on segmentaion of adjacent cells. In addition, we will also consider other types of cancers for computerassisted diagnosis.

Title: Short-Term Heart Rate Variability and Blood Biomarkers of Gastric Cancer Prognosis

Authors: Lili Wang, Bo Shi, Peng Li , Genxuan Zhang, Mulin Liu and Deli Chen

Summary: Inflammation, nutrition, and coagulation play significant roles in cancer prognosis**.** In this study, 61 patients who were first diagnosed with Gastric cancer were enrolled . Fasting blood samples were collected in the morning seven days before surgery. Blood CRP, prealbumin (PA), and fibrinogen (FIB) were used to assess the inflammation level, nutritional status, and coagulation function respectively. Five-minute resting electrocardiogram (ECG) signals were collected one day before surgical treatment. Short-term HRV time-series were extracted. The results of this study suggest that short-term HRV can potentially be used as a noninvasive biomarker for the comprehensive assessment of inflammation, nutrition, and coagulation in patients with GC.

Future Work: This study is of of great significance for further improving the effects of therapy for Gastric Cancer.

Title: Identification Tool for Gastric Cancer Based on Integration of 33 Clinical Available Blood Indices Through Deep Learning

Authors: Bowen Zhang, Long Cheng, Yuzhen Niu, Aming Wang, Pengyi Zhang, Tiantian Shen, Dekui Zhangand Shuyan Li

Summary: In cancer detection, liquid biopsy, as a noninvasive and rapid method, is growingin importance. This study integrated a variety of blood biochemical indices and established an identification system by means of deep learning under the H2O framework method. Compared with conventional diagnostic methods, the diagnostic model established in this study has the advantages of higher accuracy, noninvasiveness, and inexpensive detection. It has the potentialto screen a wide range of people and effectively reduces the pressure on the medical system.

Future Work: With the further collection and learning of blood test data, the performance of this diagnostic model can be further improved**.**

Title: Multiple Omics Analysis of the Rac3 Roles in Different Types of Human Cancer

Authors: Yipeng Song and Rongna Ma

Summary: Rac Family Small GTPase 3 (Rac3) is a member of the Rho family of small GTP-binding proteins which play critical roles in the occurrence, progression, and metastasis of various tumors. This study performed multiple omics analysis to investigate the roles of Rac3in differential expression, survival prognosis, mutative status, DNA methylation, functions, immune infiltration, and immunotherapy across 33 human cancer types. Widely expressed in various normal human tissues and cancer tissues, and Rac3 was remarkably overexpressed in different types of cancers. Rac3 expression was statistically correlate with immune and molecular subtypes, survival prognosis, mutative status, DNA methylation, immune cell infiltration, and immunotherapy in various cancers, which could contribute to understanding the Rac3 roles in the occurrence and development of human cancers, and explore new cancer treatment strategies.

Future Work: further studies are required to validate the paper findings before clinical application in various cancers.

Title: Dual-Illumination Ultrasound/ Photoacoustic System for Cervical Cancer Imaging[9]
Authors: Maryam Basji, Andrei Karpiouk , IraWiner, Stanislav Emelianov, and Mohammad Mehrmohammadi

Summary: Technologies for early cancer diagnosis can offer useful data and possibly lower the death rate brought on by cervical cancer. A small ultrasound and photoacoustic endoscopic device has been created in the earlier work to image the cervical tissue through the cervical canal. The dual-mode illumination system can provide more realistic information.

Future Work: Further research is required to evaluate the system performance on human cervical tissue.

Title: Non-Invasive Assessment of the Spatial and Temporal Distributions of Interstitial Fluid Pressure, Fluid Velocity and Fluid Flow in Cancers In Vivo

Authors: MD Tauhidul Islam, Songyuan Tang, Ennio Tasciotti, and Raffaella Righetti

Summary: For the diagnosis, prognosis, and treatment of cancer, interstitial fluid pressure, velocity, and associated measures have significant clinical value. We demonstrated the connection between these calculated values and the underlying IFP, IFV, fluid flow, and other factors within the tumors. Using finite element and ultrasound simulations for a variety of simulated phantoms the techniques were validated.

Future Work: The use of non-invasive poro elastography methods will be of great clinical significance for diagnosis, prognosis and treatment of cancers.

Title: Research on Recognition of Medical Image Detection Based on Neural Network
Authors: Shaoqiang Wang, Gewen he, Shudong Wang, Song Zhang and Fangfang fan

Summary: The fecal occult blood test (FOBT), used for regular detection of bowel cancer, as well as the high cost and discomfort of microscopy, are some of the limiting factors for this disease, which is easily influenced by food and medication. In this study, the expression spectrum is employed as an additional approach to detect medical images, the error back propagation neural network (BPNN) algorithm is applied, and a colorectal cancer (CRC) detection model based on neural network is created.

Title: Short-Term Heart Rate Variability and Blood Biomarkers of Gastric Cancer Prognosis
Authors: Lili Wang, Bo Shi, Peng Li, Genxuan Zhang , Mulin Liu and Deli Chen

Summary: In tumour development, autonomic function plays a significant role. An higher C-reactive protein (CRP) level, a serum marker for inflammation, has been linked to lower heart rate variability (HRV), a standard clinical test for evaluating autonomic function, according to earlier research. Given that lower HRV in GC patients is associated with increased inflammatory and coagulation loads as well as disturbed nutrition, it is possible that short-term HRV assessment could serve as a non invasive diagnostic for GC prognosis.

Future Work: Further improving the effects of therapy for GC.

Title: A Review on Traditional Machine Learning and Deep Learning Models for WBCs Classification in Blood Smear Images

Authors: Siraj Khan, Muhammad Sajjad, Tanveer Hussain, Amin Ullah and Ali Shariq Imran

Summary: Traditional machine learning (TML) and deep learning (DL) algorithms have considerably advanced medical image analysis (MIA). These techniques significantly enhanced the automatic diagnosis of brain tumours and leukemia/blood cancers and can the help haematologists.With a strong emphasis on leukocytes categorization in blood smear pictures and other medical imaging domains, such as magnetic resonance imaging (MRI), CTimages, X-rays, and ultrasounds, this paper provides an in-depth overview of the available TML and DL algorithms for MIA.

Future Work: In future these techniques will have tremendous contributions in the development of medical imaging, natural language processing and speech analysis.

Title: A Systematic Review on Recent Advancements in Deep and Machine Learning Based Detection and Classification of Acute Lymphoblastic Leukemia

Authors: Pradeep Kumar Das, Diya V A, Sukadev Meher, Rutuparna Panda and Ajith Abraham

Summary: Leukemia is a haematological condition that affects white blood cells and originates in the bone marrow (WBCs). The favoured method for an early identification of leukaemia is microscopic analysis of WBCs because it is less invasive and expensive. Here, multiple AI-based ALL detection methods are systematically evaluated with respect to their benefits and drawbacks.

Future Work: Further research can be carried out to efficiently classify benign and malignant (ALL) as well as classify ALL into its subtypes: L1, L2, and L3, which results in more accurate disease diagnosis.

3. THEORITICAL ANALYSIS OF PROPOSED PROJECT

3.1 REQUIREMENTS GATHERING

3.1.1 Software Requirements

- **Language Used:** Python 3
- **Operating System:** Windows
- **Software:** JupyterLab
- **Libraries Used**: cv2, Numpy, Shutil, Random, Tensorflow, Keras, Matplotlib

3.1.2 Hardware Requirements

- **Processor:** Intel Core i5
- **RAM:** 8GB

3.2 TECHNOLOGIES DESCRIPTION

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code part of this, and so is access to powerful constructs that avoid tedious repetition of code.

Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Shutil

It is a Python module that provides high-level file operations. It offers a convenient way to perform various file and directory-related tasks, such as copying files, moving files, removing files and directories, and archiving files.

Numpy

Numpy is a versatile package for array processing that offers a range of functionalities. It revolves around a high-performance multidimensional array object, providing efficient storage and manipulation of data. Numpy is widely used in scientific computing with Python, serving as a fundamental tool in this domain. Its key features include a powerful N- dimensional array object, advanced functions for broadcasting, support for integrating C/C++ and Fortran code, and capabilities for linear algebra, Fourier transforms, and random number generation. While Numpy is primarily known for its scientific applications, it can also serve as an effective container for generic data. This flexibility allows Numpy to seamlessly and rapidly integrate with various databases, enabling efficient data handling and processing.

TensorFlow

TensorFlow is an extensively used open-source software library that offers robust dataflow and differentiable programming capabilities. It is widely adopted for diverse technological applications, particularly in the field of machine learning, specifically neural networks. With its flexible design, TensorFlow caters to the needs of both research and production environments, making it a reliable choice for implementing cutting-edge technologies. Originally developed by the Google Brain team for internal purposes, TensorFlow was later released under the Apache 2.0 open-source license. This licensing choice expanded its accessibility, allowing a wider community to leverage its functionalities and contribute to its development. For deep learning tasks, TensorFlow provides a comprehensive API called

tensorflow.keras, which simplifies the process of building and training neural networks. This API encompasses a range of modules and functions tailored to meet the demands of constructing efficient and scalable deep learning models. The layers module offers a diverse set of layer types, including convolutional, dense, and pooling layers, enabling users to design complex network architectures. The Model module within tensorflow.keras empowers developers to create customized models by specifying the input and output layers, facilitating the construction of neural network topologies tailored to specific technological requirements. This flexibility allows for the incorporation of various architectures and configurations to achieve optimal performance.TensorFlow also provides specialized modules for data preprocessing and augmentation, streamlining the preparation of input data for training. The ImageDataGenerator module efficiently loads and preprocesses image data, while the preprocess_input function ensures compatibility of input images with specific neural network architectures, optimizing the training process. To enhance training efficiency and effectiveness, TensorFlow offers a range of callback functions. These functions, such as ModelCheckpoint and EarlyStopping, enable automatic model saving during training and early termination based on specified conditions, respectively. This proactive approach ensures model integrity and enhances training workflow.

Furthermore, TensorFlow facilitates the utilization of pre-trained models, enabling transfer learning for improved efficiency and performance. The load_model function allows for the integration of pre-trained weights and architectures, serving as a solid foundation for training on new datasets or solving related tasks. This transfer learning capability saves valuable time and computational resources. Overall, TensorFlow, with its versatile technological capabilities and the streamlined workflow provided by the tensorflow.keras API, empowers developers and researchers to implement state-of-the-art machine learning technologies with ease. Its robust features make it a preferred choice for various technological applications, driving advancements in the field of deep learning.

Keras

Keras is a high-level neural network API developed by google that provides a user-friendly and intuitive interface for building and training deep learning models. It is written in python and is used to make the implementation of neural networks easy and it also supports multiple backend neural network computation. It simplifies the process of developing deep learning models by offering a concise and easy-to-understand syntax. There are five frameworks supported by keras but tensor flow has adopted keras as its official high-level API. Keras is embedded in

TensorFlow and can be used to perform deep learning fast as it provides inbuilt modules for all

neural network computations. With Keras, users can quickly prototype and experiment with different network architectures, training configurations, and optimization algorithms.

CV2

The cv2 (OpenCV) library is a widely used computer vision library in machine learning projects. It provides a comprehensive set of functions and tools for image and video processing, analysis, and computer vision tasks. By importing the cv2 module, developers can leverage its capabilities to perform a variety of operations on visual data. One of the primary uses of cv2 in machine learning projects is image and video input/output. It allows loading and saving images and videos in different formats, providing flexibility in handling diverse datasets. The library supports various image file formats, such as JPEG, PNG, and BMP, as well as video formats like AVI and MP4.

Additionally, cv2 offers a wide range of image processing functions, including filtering, transformations, geometric operations, and color manipulation. These operations are instrumental in preprocessing visual data before feeding it into machine learning models. Examples of such operations include resizing images, applying filters, adjusting brightness and contrast, and extracting image features. Moreover, cv2 provides tools for object detection, image segmentation, and feature extraction, enabling advanced computer vision tasks. It includes pre-trained models and algorithms that can be utilized for tasks like face detection, object recognition, and motion analysis. These capabilities are particularly valuable in building machine learning applications that require visual understanding and interpretation.

Matplotlib

Matplotlib is a powerful 2D plotting library for Python that offers high-quality figures suitable for publication purposes. It supports a wide range of output formats and can be utilized in various environments, including Python scripts, IPython shells, Jupyter Notebooks, web application servers, and graphical user interface toolkits. Matplotlib aims to provide an intuitive and straightforward experience for simple plotting tasks while also offering advanced capabilities for complex visualizations.

With just a few lines of code, you can generate a diverse range of plots, histograms, power spectra, bar charts, error charts, scatter plots, and more. The pyplot module, especially when

combined with IPython, provides a user-friendly interface similar to MATLAB. For advanced users, Matplotlib offers extensive customization options, allowing precise control over line styles, font properties, axes properties, and more through an object-oriented interface or MATLAB-like functions. To explore the capabilities of Matplotlib, you can refer to the sample plots and thumbnail gallery provided by the library.

Sklearn

scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4. DESIGN

4.1 Introduction

Bone marrow cancer, also known as hematological malignancy, is a serious medical condition that requires early detection and accurate diagnosis for effective treatment. In recent years, advancements in deep learning techniques have paved the way for the development of powerful models capable of analyzing medical images and aiding in cancer detection. In this study, we propose a novel approach for bone marrow cancer prediction using ResNet50 and MobileNetV2, two state-of-the-art convolutional neural network (CNN) architectures widely recognized for their success in computer vision tasks.

The primary objective of this research is to leverage the capability of ResNet50 and MobileNetV2 to extract informative features from bone marrow images and classify them as cancerous or non-cancerous. By harnessing the deep convolutional layers of these pre-trained models, we aim to capture subtle patterns and characteristics within the images that may serve as indicative markers for bone marrow cancer. The models ability to learn from large-scale image datasets, such as ImageNet, through transfer learning further enhances their potential to recognize relevant features specific to bone marrow abnormalities.

Our methodology involves a systematic process, starting with the collection of a well-curated dataset comprising bone marrow images with corresponding labels indicating the presence or absence of cancer. The dataset is then preprocessed to ensure uniformity and improve the models' ability to generalize. Subsequently, we employ ResNet50 and MobileNetV2 as feature extractors, utilizing their deep convolutional layers to capture intricate details and spatial dependencies within the bone marrow images.

To facilitate the prediction of bone marrow cancer, additional layers are added on top of the extracted features to form a classification head. Through an iterative training process, the models are fine-tuned on the labeled dataset, optimizing their ability to distinguish between cancerous and non-cancerous samples. Extensive evaluation is conducted using various performance metrics to assess the models' accuracy, precision, recall, and overall efficacy in predicting bone marrow cancer.

By effectively leveraging the power of deep learning and transfer learning techniques, we anticipate that our proposed approach will contribute to the early detection and diagnosis of bone marrow cancer, assisting healthcare professionals in making informed decisions and providing patients with timely and appropriate treatment options. The results obtained from this study have the potential to significantly impact the field of oncology by augmenting the existing methods of cancer detection, ultimately leading to improved patient outcomes and a higher quality of life.

4.2 Architecture

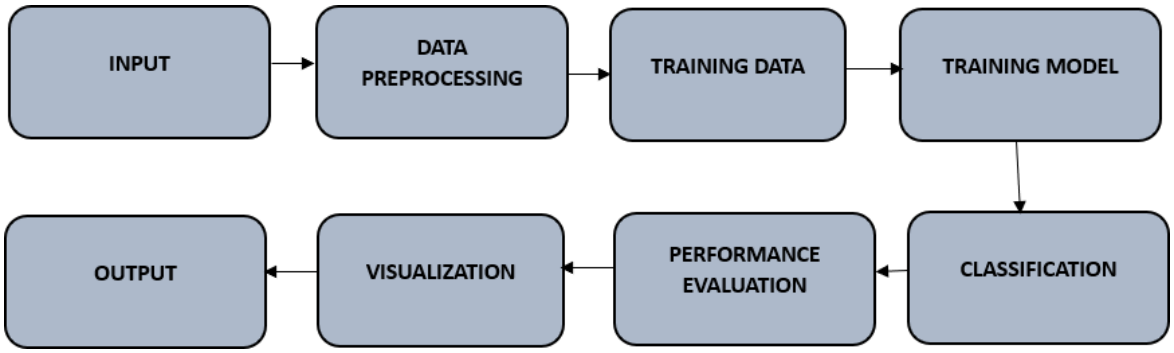


Fig 3 : **Architecture Design**

The architecture followed is shown in the figure 3. The (Microscopic) images are taken from the dataset as input. We are preprocessing the images the by Augmentation in order to equal the number of cancerous and non-cancerous images. We increased the image count by flipping and rotating the image. After augmentation we train the dataset by center cropping the image. We then training the dataset with two model one is Resnet50 and the other is Mobilenet v2. After training the models we do classification whether the model is predicting the image correctly or not. In Performance Evaluation we check the performance of the model by using evaluation matrix. Here we calculate the accuracy ,precision and recall. Finally visualizing the two models with the help of the graph.

Data Preprocessing

The training set, as has already been shown, is unbalanced since we have that about 69% of the images belong to the cancer class, while the remaining 31% belong to the normal class.

In order to solve this issue, a combination of subsampling and data augmentation techniques were applied so that the number of images of normal cells and of the ones of cancer cells were equalized to 3198.

In particular, from the 4432 cancer cell images, 3198 images were randomly selected. Furthermore, 1234 normal cell images were randomly selected in order to perform data augmentation on them. Noting that microscopic images are invariant to rotations and flips, new images were obtained from the originals by randomly performing a vertical or horizontal flip, or by applying a rotation of a random angle between 90°, 180°, and 270°.

Another thing that may be noticed when analyzing the dataset is that a large part of each image is consists of background of black color. Therefore, the size of the images can be greatly reduced without loss of information by performing center cropping. In order to do this, first the image is converted from RGB to grayscale and then a thresholding method is applied to locate the area of the image that is completely black. Finally, after removing as much of background as possible, black padding is added to make the image square again. In this way, any resizing will not affect the aspect ratio of the cell. As a result of this process, more than half of the images have been reduced to a size smaller than 224×224 and the vast majority are at most 300×300 in size. The sample output is shown in fig 4, 5.

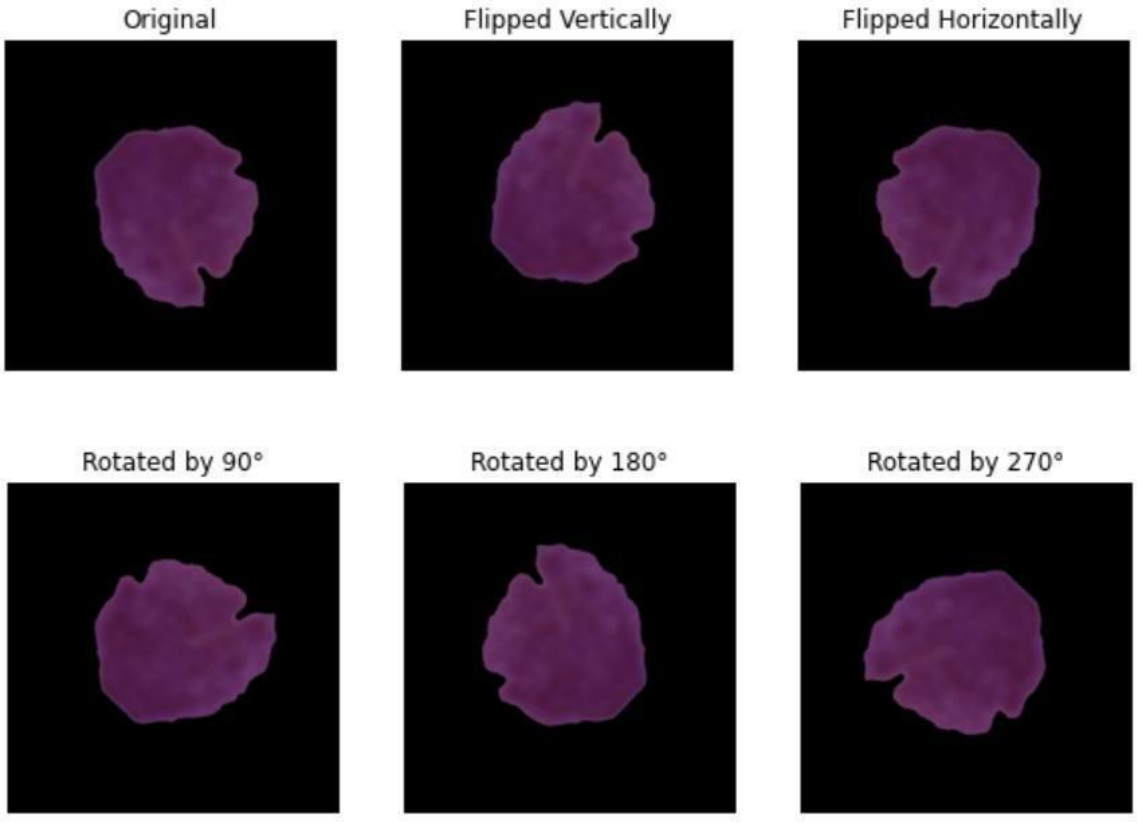


Fig 4: Data Augmentation of original Image

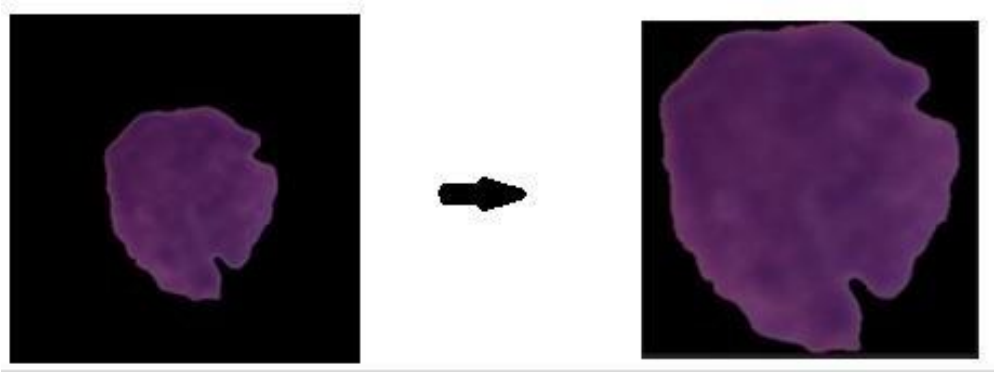


Fig 5: Center Cropping

4.3 Proposed Models

4.3.1 ResNet50

ResNet-50 is a convolutional neural network (CNN) architecture that was introduced in 2015 as part of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It is one of the variants of the ResNet (Residual Network) architecture, which was designed to overcome the problem of vanishing gradients in deep neural networks.

ResNet-50 consists of 50 layers, including a convolutional layer, followed by several residual blocks. Each residual block contains multiple convolutional layers, batch normalization layers, and rectified linear unit (ReLU) activation functions. The residual block also includes a shortcut connection that skips one or more layers to allow the network to learn the identity function, which helps to prevent the vanishing gradients problem.

The basic idea behind the ResNet-50 architecture is to use the residual blocks to learn residual functions instead of mapping the input directly to the output. The residual functions capture the difference between the input and output, making it easier for the network to learn the underlying patterns in the data.

ResNet-50 is a widely used architecture for image classification tasks, particularly in computer vision applications such as object detection and recognition. It has achieved state-of-the-art performance on several benchmark datasets and is a popular choice for transfer learning, where the pre-trained model is fine-tuned on a specific task. The architecture is shown in figure 6.
A detailed explanation of the ResNet-50 architecture:

The ResNet-50 architecture consists of 50 layers, including a convolutional layer, several residual blocks, and a fully connected layer at the end. Here's a breakdown of the architecture:

1. Input layer: The input to the network is an RGB image of size 224 x 224 pixels.

2. Convolutional layer: The first layer is a 7x7 convolutional layer with a stride of 2, which reduces the size of the image by a factor of 2. This layer is followed by a batch normalization layer and a ReLU activation function.

3. Max pooling layer: A max pooling layer with a 3x3 kernel and a stride of 2 is used to further reduce the size of the image by a factor of 2.

4. Residual blocks: The core of the ResNet-50 architecture consists of several residual blocks, each of which contains multiple convolutional layers, batch normalization layers, and ReLU activation functions. The residual block also includes a shortcut connection that skips one or more layers to allow the network to learn the identity function. The shortcut connection merges the output of the convolutional layers with the original input, which is then passed through a ReLU activation function. This helps to prevent the vanishing gradients problem and enables the network to learn deeper representations of the input.

5. Global average pooling layer: After the residual blocks, a global average pooling layer is used to generate a single feature map from the output of the last residual block. This layer averages the spatial dimensions of the feature maps to generate a single feature map with a depth equal to the number of filters in the previous layer.

6. Fully connected layer: Finally, a fully connected layer is used to generate the output of the network. This layer takes the feature map from the global average pooling layer and maps it to the number of classes in the dataset using a softmax activation function.

The ResNet architecture follows two basic design rules. First, the number of filters in each layer is the same depending on the size of the output feature map. Second, if the feature map's size is halved, it has double the number of filters to maintain the time complexity of each layer.

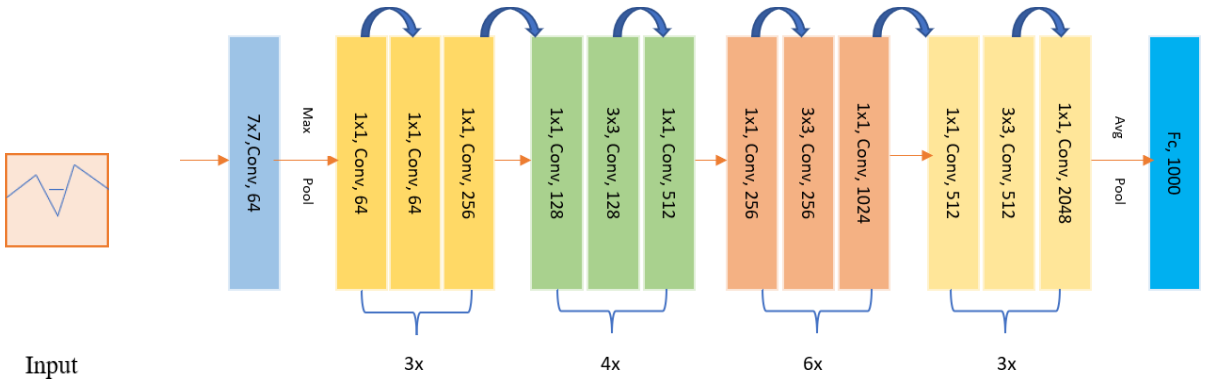


Fig 6: ResNet50 Architecture

The resnet 50 architecture contains the following element:

- A convolution with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2 giving us 1 layer.
- Next we see max pooling with also a stride size of 2.
- In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel andat last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us 9 layers in this step.
- Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of1 * 1,512 this step was repeated 4 time so giving us 12 layers in this step.
- After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is repeated 6 time giving us a total of 18 layers.
- And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.

After that we do a average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us 1 layer.

We don't actually count the activation functions and the max/ average pooling layers.

so totaling this it gives us a 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network.

Residual Network:

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections.The skip connection connects activations of a layer to further layers by skipping

some layers in between as shown in figure 7. This forms a residual block. Resnets are made by stacking these residual blocks together.

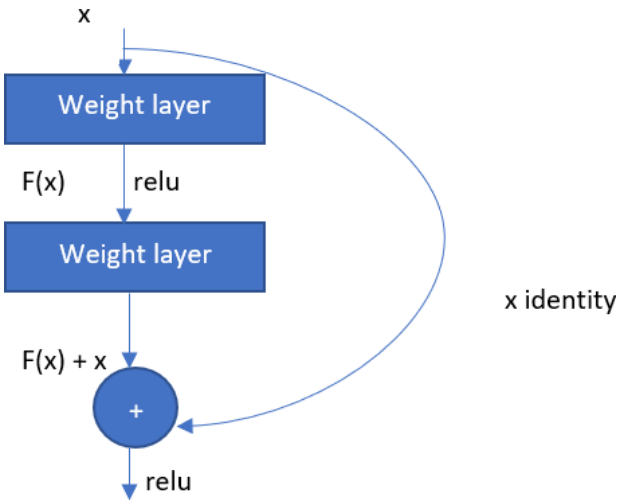


Fig 7 : Skip Connection

4.3.2 MobileNetV2

MobileNetv2 is a convolutional neural network architecture designed for mobile devices and other resource-constrained environments. It was introduced in 2018 by researchers at Google in a paper titled "MobileNetV2: Inverted Residuals and Linear Bottlenecks".

The key idea behind MobileNetv2 is to use a series of lightweight building blocks to reduce the number of parameters and computation required for inference while maintaining high accuracy. These building blocks include depthwise separable convolutions, linear bottlenecks, and inverted residuals.

Depthwise separable convolutions consist of a depthwise convolution, which applies a single filter to each input channel separately, followed by a pointwise convolution, which applies a 1x1 convolution to combine the outputs of the depthwise convolution. This reduces the number of parameters and computation required for each convolution.

Linear bottlenecks are used to reduce the dimensionality of the input features by projecting them into a lower-dimensional space and then expanding them back to their original dimensionality. This reduces the computational cost of subsequent convolutions.

Inverted residuals are used to improve the accuracy of the model by using skip connections and residual connections within each building block. This allows the model to learn

more complex representations and increases the accuracy of the model.

MobileNetv2 has been shown to achieve state-of-the-art performance on a range of computer vision tasks while requiring fewer parameters and less computation than other deep learning models.

MobileNet V2 model has 53 convolution layers and 1 AvgPool. It has two main components:

- Inverted Residual Block
- Bottleneck Residual Block

There are two types of Convolution layers in MobileNet V2 architecture:

- 1x1 Convolution
- 3x3 Depthwise Convolution

In MobileNetV2, there are two types of blocks. One is residual block with stride of 1. Another one is block with stride of 2 for downsizing.There are 3 layers for both types of blocks.The first layer is 1×1 convolution with ReLU6. The second layer is the depthwise convolution. The third layer is another 1×1 convolution but without any non-linearity. It is claimed that if ReLUis used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain. The tabular description is shown in figure 8.

And there is an expansion factor t. And t=6 for all main experiments.

If the input got 64 channels, the internal output would get 64×t=64×6=384 channels.

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

Fig 8 : Tablular Description of MobileNetV2

where t: expansion factor, c: number of output channels, n: repeating number, s: stride. 3×3 kernels are used for spatial convolution.

Each block has 3 different layers:

- 1x1 Convolution with Relu6
- Depthwise Convolution
- 1x1 Convolution without any linearity

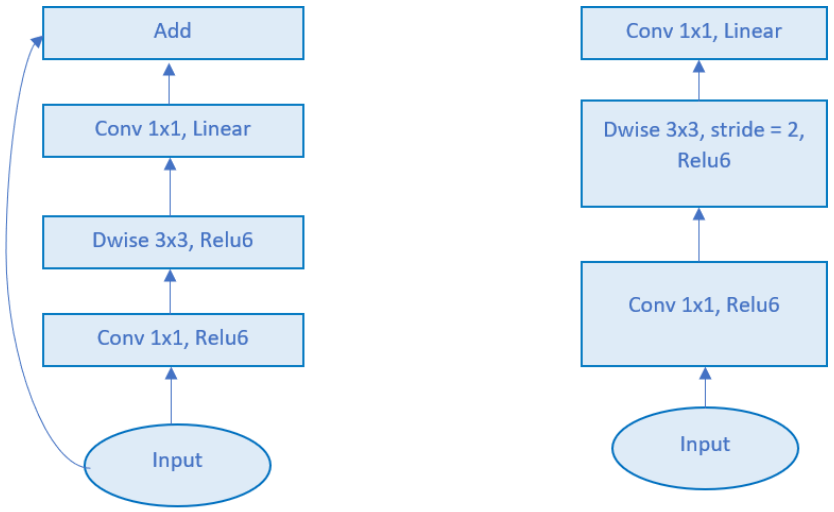
There are Stride 1 Blocks and Stride 2 Blocks. The internal components of the two blocks areas follows:

Stride 1 Block:

- Input
- 1x1 Convolution with Relu6
- Depthwise Convolution with Relu6
- 1x1 Convolution without any linearity
- Add

Stride 2 Block:

- Input
- 1x1 Convolution with Relu6
- Depthwise Convolution with stride=2 and Relu6
- 1x1 Convolution without any linearity



Stride = 1 block **Stride = 2 block**

Fig 9 : MobileNetV2 Architecture

Key Components

1x1 Convolution Layers: A 1 x 1 Convolution is a convolution with some special properties in that it can be used for dimensionality reduction, efficient low dimensional embeddings, and applying non-linearity after convolutions. It maps an input pixel with all its channels to an output pixel which can be squeezed to a desired output depth.

Depthwise convolution: Depthwise Convolution is a type of convolution where we apply a single convolutional filter for each input channel. In the regular 2D convolution performed over multiple input channels, the filter is as deep as the input and lets us freely mix channels to generate each element in the output.

Inverted Residual Block: An Inverted Residual Block, sometimes called an MBConv Block, is a type of residual block used for image models that uses an inverted structure for efficiency reasons. It was originally proposed for the MobileNetV2 CNN architecture. It has since been reused for several mobile-optimized CNNs.

Bottleneck Residual Block: A Bottleneck Residual Block is a variant of the residual block that utilises 1x1 convolutions to create a bottleneck. The use of a bottleneck reduces the number of parameters and matrix multiplications. The idea is to make residual blocks as thin as possible to increase depth and have less parameters.

5. IMPLEMENTATION

5.1 Coding

Git Link: https://github.com/Lekhya123/Bone-Marrow-Cancer-Detection-from-Leukocytes-using-Neural-Networks

5.2 Input

The training dataset is obtained from Kaggle. A total of 10662 images are divided as training, testing and validating datasets. Each having sub folders of cancerous and normal cells of microscopic cancer images. The percentages chosen were 60% for the training set, 20% for the validation set, and 20% for the test set. The division of images and the count of it is explained in the below figure 10. The bmp extension images are multiplied using data augmentation steps and are given to the model for training. A sample of the images used for training are shown in figure 2.

	Cancer Cells	**Normal Cells**	**Total**
Training	4432	1964	6396
Testing	1437	696	2133
Validation	1404	729	2133
Total	7273	3389	10662

Fig 10: Dataset Division

5.3 Evaluation Metrics

Checking the model efficiency before releasing it to the outside world is important. We can calculate the confusion matrix and from that we can get the accuracy and precision of the model we have designed. We calculate the this by calculating the ratio of correct prediction to the total number of input Samples. The formulas used from the figure 11 are shown below.

Accuracy - (TP + TN) / (TP + TN + FP + FN)

Precision - TP / (TP + FP)

Recall – TP / (TP + FN)

The final output is calculated based on the accuracy.

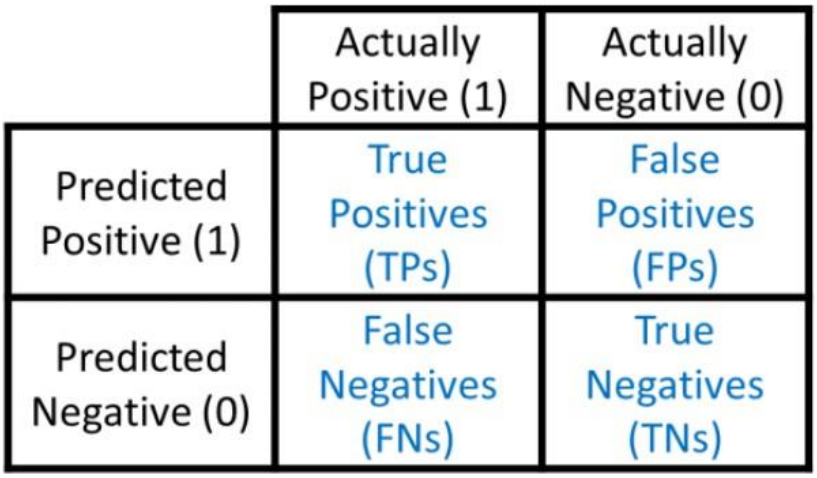


Fig 11: Confusion Matrix

5.4 Results

Given the model as shown in the figure 12 it classifies whether the cell is cancerous or not which can be used for the further diagnosis of the disease**.** The model_path here refers to the path of the pre-trained model (ResNet-50 or MobileNetV2). The accuracy we got for ResNet-50 is 93.48% and MobileNetV2 is 87.34%

```
model = tf.keras.models.load_model(model_path)

# Function to preprocess the image
def preprocess_image(image_path, target_size):
    img = load_img(image_path, target_size=target_size)
    img_array = img_to_array(img)
    img_array = img_array / 255.0  # Normalize the image
    img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension
    return img_array

# Provide the path of the image you want to predict
image_path = "Test2.bmp"

# Preprocess the image
target_size = (224,224)  # Use the same target size used during training
image_array = preprocess_image(image_path, target_size)

# Make the prediction
prediction = model.predict(image_array)
class_labels = ['hem','all']  # The class labels in the order used during training
predicted_class_index = np.argmax(prediction)
predicted_class = class_labels[predicted_class_index]

print("Predicted class:", predicted_class)
```

Fig 12: Model Predicting the Image Class

ResNet – 50 model's performance is shown in the figure 13

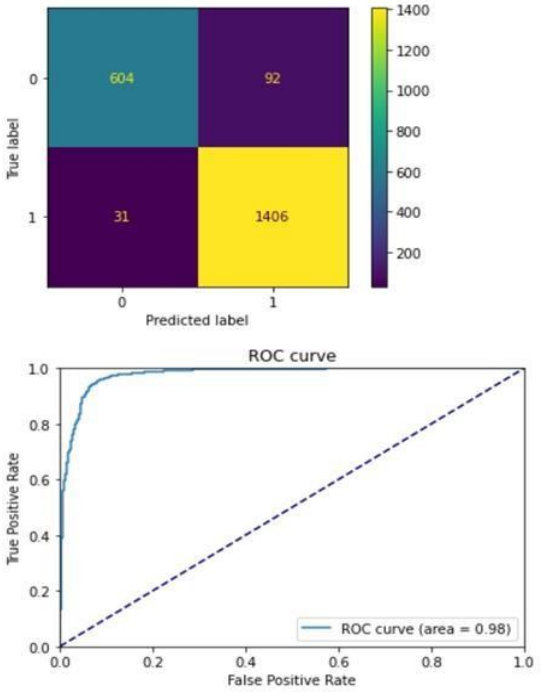


Fig 13: Confusion Matrix and ROC Curve for ResNet-50 Model

MobileNetV2 model's performance is shown in the figure 14

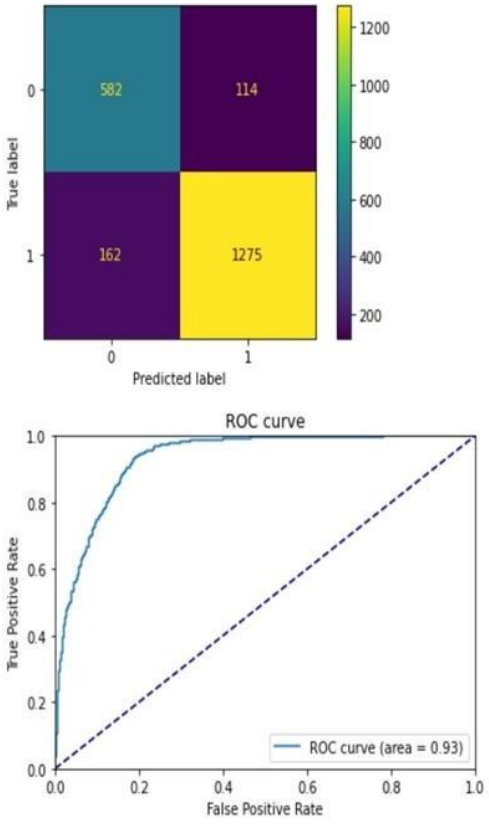


Fig 14: Confusion Matrix and ROC Curve for MobileNetV2 Model

The models ResNet-50 and MobileNetV2 are trained on the microscopic images of the cancerous cells. The expected cells are classified into cancerous or not. ResNet-50 performs better than the MobileNetV2 as per the comparision analysis done between them as shown in figure 15.

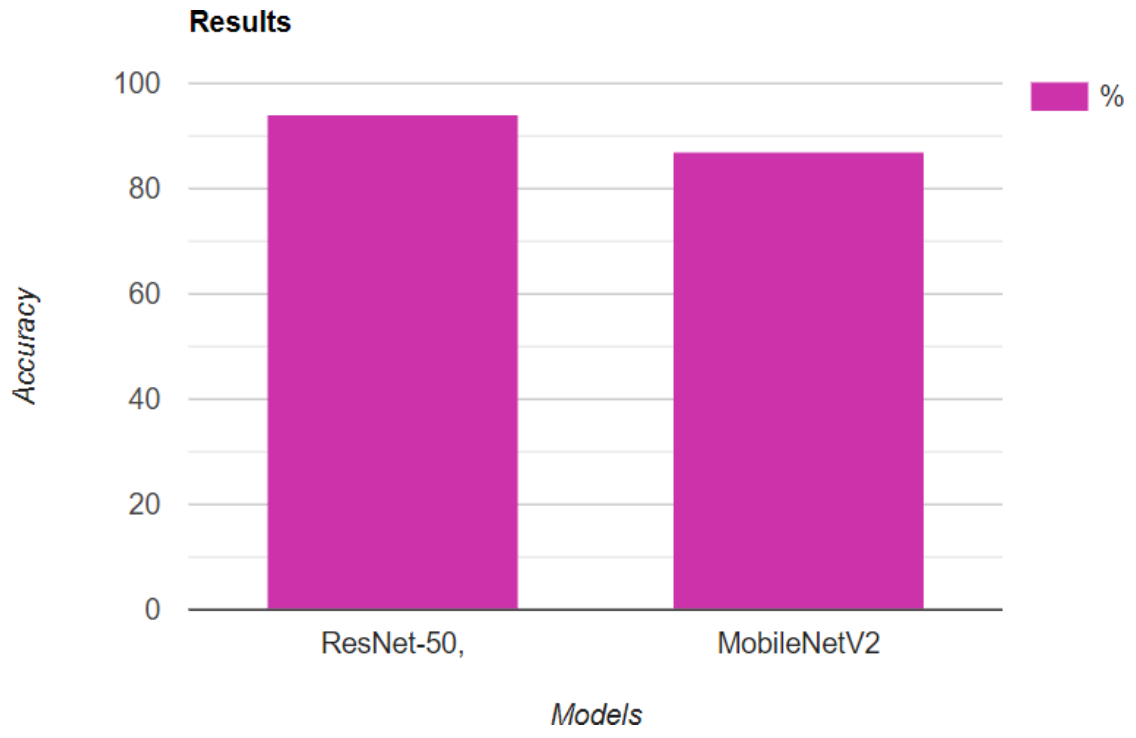


Fig 15: Comparative Analysis of ResNet-50 and MobileNetV2

The developed technique offers promising potential for early detection and diagnosis of bone marrow cancer. By utilizing ResNet50 and MobileNetV2, healthcare professionals can leverage these models to assist in making informed decisions and providing timely treatment options.

Overall, the proposed approach contributes to the advancement of cancer detection in the field of oncology, potentially leading to improved patient outcomes and a higher quality of life. The successful application of deep learning and CNNs in bone marrow cancer prediction highlights the potential for further advancements in medical image analysis and computer-aided diagnosis.

6. CONCLUSION AND FUTURE WORK

Predicting the existence of cells makes it easy to identify bone marrow cancer. The models are tested, validated, and trained using photos from datasets. The cells are predicted with the aid of the models ResNet50 and MobileNetV2. These computers aided systems help doctors in early diagnosis of bone marrow cancer leading to early-stage detection and faster recovery. ResNet50 has performed better compared to MobileNetV2. The shortcomings of the conventional prediction and categorization of images are overcome by these models, which improve the accuracy of properly predicting the cancer cells. Future work can be done on huge datasets with various other available deep learning techniques and further classification of the type of cancer can be done.

REFERENCES

[1] Dataset : https://www.kaggle.com/datasets/avk256/cnmc-leukemia

Afshan Shah , Syed Saud Naqvi, Khuram Naveed , Nema Salem ,Mohammad A. U. Khan , and Khurram S. Alimgeer, "Automated Diagnosis of Leukemia: A Comprehensive Review", IEEE, October 1, 2021.

Deepika Kumar, Nikita Jain, Aayush Khurana, Sweta Mittal, Suresh Chandra Satapathy, Roman Senkerik and Jude D. Hemanth, "Automatic Detection of White Blood Cancer From Bone Marrow Microscopic Images Using Convolutional Neural Networks", IEEE, August 14, 2020.

Rune Wetteland, Vebjørn Kvikstad, Trygve Eftestøl ,Erlend Tøssebro, Melinda Lillesand, Emiel A. M. Janssen, and Kjersti Engan, "Automatic Diagnostic Tool for Predicting Cancer Grade in Bladder Cancer Patients Using Deep Learning", IEEE, August 26, 2021.

Tusneem Ahmed M. Elhassan, Mohd Shafry Mohd Rahim, Tan Tian Swee, Siti Zaiton Mohd Hashim, and Mahmoud Aljurf, "Feature Extraction of White Blood Cells Using CMYK-Moment Localization and Deep Learning in Acute Myeloid Leukemia Blood Smear Microscopic Images", IEEE, Feb 15, 2022.

Nadeem Akram, Sharjeel Adnan, Muhammad Asif, Syed Muhammad Ali Imran, Muhammad Naveed Yasir, Rizwan Ali Naqvi, and Dildar Hussain, "Exploiting the Multiscale Information Fusion Capabilities for Aiding the Leukemia Diagnosis Through White Blood Cells Segmentation", IEEE, May 11, 2022.

Lili Wang, Bo Shi, Peng Li, Genxuan Zhang, Mulin Liu, and Deli Chen, "Short-Term Heart Rate Variability and Blood Biomarkers of Gastric Cancer Prognosis", IEEE, Janueary 27, 2020.

Bowen Zhang , Long Cheng, Yuzhen Niu, Aming Wang, Pengyi Zhang, Tiantian Shen, Lili XI6, Dekui Zhang, and Shuyan Li, "Identification Tool for Gastric Cancer Based on Integration of 33 Clinical Available Blood Indices Through Deep Learning", IEEE, October 2022.

Yipeng Song and Rongna Ma, "Multiple Omics Analysis of the Rac3 Roles in Different Types of Human Cancer", IEEE, September, 2022.

Maryam Basij, Andrei Karpiouk, Ira Winer, Stanislav Emelianov, Mohammad Mehrmohammadi, "Dual-Illumination Ultrasound/ Photoacoustic System for Cervical Cancer Imaging", IEEE Photonics Journal, Volume 13, Number 1, February 2021.

MD Tauhidul Islam, Songyuan Tang, Ennio Tasciotti, and Raffaella Righetti, "Non-Invasive Assessment of the Spatial and Temporal Distributions of Interstitial Fluid Pressure, Fluid Velocity and Fluid Flow in Cancers In Vivo", IEEE, June 29, 2021.

Shaoqiang Wang, Gewen He, Shudong Wang, Song Zhang, and Fangfang Fan, "Research on Recognition of Medical Image Detection Based on Neural Network", IEEE, June 2, 2020.

Siraj Khan, Muhammad Sajjad, Tanveer Hussain, Amin Ullah, and Ali Shariq Imran, "A Review on Traditional Machine Learning and Deep Learning Models for WBCs Classification in Blood Smear Images", IEEE, Dec 30, 2020.

Pradeep Kumar Das, Diya V A, Sukadev Meher , Rutuparna Panda, and Ajith Abraham, "A Systematic Review on Recent Advancements in Deep and Machine Learning Based Detection and Classification of Acute Lymphoblastic Leukemia", IEEE, August, 2022.

APPENDIX

Pseudo code for data augmentation

```
function random_flip_or_rotation(original_image):
    // Generate a random number to decide whether to flip or rotate
    flip_or_rotate = random_integer(0, 1)
    if flip_or_rotate == 1:
        // Flip the image
        horizontal = random_integer(0, 1)
        if horizontal == 1:
            new_image = flip_left_right(original_image)
        else:
            new_image = flip_up_down(original_image)
    else:
        // Rotate the image
        k = random_integer(1, 3)
        new_image = rotate_90(original_image, k)
    return new_image


function crop_and_resize_image(image):
    // Convert image to grayscale
    gray = convert_to_grayscale(image)
    // Apply thresholding to create a binary image
    thresholded_image = apply_thresholding(gray)
    // Find contours in the binary image
    contours, hierarchy = find_contours(thresholded_image)
    // Get the bounding rectangle of the first contour
    x, y, w, h = get_bounding_rectangle(contours[0])
    // Crop the image using the bounding rectangle
    cropped_image = crop_image(image, x, y, w, h)
   // Resize the cropped image to a square shape
    square_image = resize_image(cropped_image)
```

```
    return square_image
```

Pseudo code for loading dataset

```
# Load the training set
def load_training_set(image_height, image_width, batch_size):
    return image_dataset_from_directory(
        TRAINING_PATH,
        image_size=(image_height, image_width),
        batch_size=batch_size,
        class_names=['hem', 'all']
    )

# Load the validation set
def load_validation_set(image_height, image_width, batch_size):
    return image_dataset_from_directory(
        VALIDATION_PATH,
        image_size=(image_height, image_width),
        batch_size=batch_size,
        class_names=['hem', 'all']
    )

# Load the test set
def load_test_set(image_height, image_width, batch_size):
    return image_dataset_from_directory(
        TEST_PATH,
        image_size=(image_height, image_width),
        batch_size=batch_size,
        shuffle=False,
        class_names=['hem', 'all']
    )
```

Pseudo code for model training

```
# Compile the model
```

```
def compile_model(model, optimizer='adam', learning_rate=0.001):
    if optimizer == 'adam':
        model.compile(
            loss="binary_crossentropy",
            optimizer=optimizers.Adam(learning_rate=learning_rate),
            metrics=["accuracy"]
        )
    elif optimizer == 'rmsprop':
        model.compile(
            loss="binary_crossentropy",
            optimizer=optimizers.RMSprop(learning_rate=learning_rate),
            metrics=['accuracy']
        )
    model.summary()

# Run the model
def run_model(model, model_name, epochs=20, patience=5, monitor='val_loss'):
    save_path = LOCAL_MODELS_FOLDER + '/' + model_name + '.h5'
    callbacks_list = [
        keras.callbacks.EarlyStopping(monitor=monitor, patience=patience),
        keras.callbacks.ModelCheckpoint(
            filepath=save_path,
            monitor=monitor,
            verbose=1,
            save_best_only=True
        )
    ]
    history = model.fit(
        train_dataset,
        epochs=epochs,
        validation_data=validation_dataset,
        callbacks=callbacks_list
    )
```

```
    shutil.copy(save_path, GLOBAL_MODELS_FOLDER + '/' + model_name + '.h5')
    show_training_and_validation_performance(history)

# Evaluate the model
def evaluate_model(model):
    y_score = model.predict(test_dataset)
    y_pred = np.rint(y_score)
    y_true = tf.concat([labels_batch for data_batch, labels_batch in test_dataset], axis=0)

# Make the prediction
prediction = model.predict(image_array)
class_labels = ['hem', 'all'] # The class labels in the order used during training
predicted_class_index = np.argmax(prediction)
predicted_class = class_labels[predicted_class_index]
print("Predicted class:", predicted_class)
```