A PROJECT ON

An Effective Multimodal Framework for Sickle Cell Anemia Detection using Transfer Learning and Explainable AI

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

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING
Submitted by:

Tanishka Katiyar	**2019177**
Samridhi Naithani	**2019067**
Lovely Upadhyay	**2018908**
Ishita Rawat	**2018844**

Under the Guidance of
Dr. Guru Prasad M S
Professor

Project Team ID: MP24CSE200

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
June-2025

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the Project Report entitled **"An Effective Multimodal Framework for Sickle Cell Anemia Detection using Transfer Learning and Explainable AI"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and submitted in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun is an authentic record of my own work carried out during a period from **August-2024 to May-2025** under the supervision of **Dr. Guru Prasad M S**, **Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University).

The matter presented in this dissertation has not been submitted by me/us for the award of any other degree of this or any other Institute/University.

Tanishka Katiyar	2019177
Samridhi Naithani	2019067
Lovely Upadhyay	2018908
Ishita Rawat	2018844

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Supervisor **Head of the Department**

External Viva

Name of the Examiners: **Signature with Date**

1.

2.

Acknowledgement

Any achievement, be in scholastic or otherwise does not depend solely on the individual effort but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities in their own capacity have helped me in carrying out this project work.

Our sincere thanks to project guide **Dr. Guru Prasad M S, Professor,** Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable guidance and support throughout the course of project work and for being a constant source of inspiration.

We extend our thanks to **Dr. Deepak Gaur**, Project coordinator, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable suggestions throughout all the phases of the Project Work.

We are extremely grateful to **Prof. (Dr.) D. P. Singh**, HOD of the Computer Science and Engineering Department, Graphic Era (Deemed to be University), for his moral support and encouragement.

We thank the **management of Graphic Era (Deemed to be University)** for the support throughout the course of our Bachelor's Degree and for all the facilities they have provided.

Last, but certainly not least we thank all teaching and non-teaching staff of Graphic Era (Deemed to be University) for guiding us in the right path. Most importantly we wish to thank our parents for their support and encouragement.

Tanishka Katiyar	2019177
Samridhi Naithani	2019067
Lovely Upadhyay	2018908
Ishita Rawat	2018844

Abstract

Sickle cell anemia is a genetic blood disorder where red blood cells become rigid and sickle-shaped, leading to blocked blood flow and severe pain. It is caused by a mutation in the HBB gene affecting hemoglobin production. However, AI research on sickle cell anemia is limited by scarce annotated datasets and lack of multimodal models combining image and clinical data. The Objectives of this research are to overcome the limitations of a Blackbox model and lack of wide variety of dataset. Numerical and image dataset has been used to create a combined model and LIME/Grad Cam has been used to increase explainability. Transfer learning was used for image training where the model was first trained on a general dataset for all types of anemia then after that using the trained model for specifically Sickle Cell Anemia and after trying four different models for both numerical and image dataset training-testing, SVM and EfficientNet-B0 showed best accuracy for Numerical and Image data respectively. Developing AI models for sickle cell anemia can enable faster, more accurate diagnosis and reduce the burden on medical professionals. It can also improve patient outcomes through early detection, personalized treatment, and wider accessibility in low-resource settings.

Keywords: Multimodal Model, LIME, SVM, Transfer Learning, Grad Cam

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the following sections, a brief introduction and the problem statement for the work has been included.

1.1 Project Introduction

Sickle cell anemia (SCA) is a genetic blood disorder caused by mutations in hemoglobin-producing genes, resulting in red blood cells (RBCs) adopting an abnormal sickle shape. These deformed cells obstruct blood vessels, impair oxygen delivery, and lead to severe complications, significantly affecting patient quality of life [2]. Early and precise diagnosis is crucial to managing the disease, yet traditional methods involving manual examination of blood smear images are time-intensive, require expertise, and are prone to error [5].

Advancements in artificial intelligence (AI) have opened new avenues for automating SCA detection, making diagnostic processes faster and more accurate. In this work, we utilize transfer learning with pre-trained models to identify and classify RBC abnormalities, offering improved efficiency and accuracy in distinguishing healthy cells from sickle-shaped ones [2]. These models benefit from robust preprocessing techniques, such as histogram equalization and thresholding, to enhance image quality and address challenges like cell overlap and variability in morphology [4].

Despite these advances, limitations in dataset size and model interpretability remain significant barriers. To address the data scarcity, we employ augmentation techniques—such as rotation, flipping, and brightness adjustment—to expand the dataset and improve model generalization. Additionally, explainable AI (XAI) methodologies, including Grad Cam, LIME, and attention maps, provide clinicians with insights into AI-driven predictions, fostering trust and transparency in diagnostic outcomes [3].

This project aims to develop a comprehensive system for SCA detection by integrating transfer learning for image analysis, SVM for processing numerical or text-based clinical data, and explainable AI techniques. The system will combine image-based and numerical data to enhance diagnostic precision while employing generative AI to augment datasets and XAI techniques to provide transparency in predictions. This approach seeks to bridge the gap between automated disease detection and clinical applicability, offering a scalable and reliable solution to improve diagnostic outcomes in diverse healthcare settings.

1.2 Problem Statement

The problem statement for the present work can be stated as follows:

The project aims to develop an AI-driven diagnostic system for detecting sickle cell anemia (SCA) by leveraging advanced transfer learning models enhanced with image augmentation for data enrichment and Explainable AI (XAI) techniques for interpretability. Sickle cell anemia is a genetic blood disorder characterized by abnormally shaped red blood cells that impede blood flow, leading to severe complications such as acute pain, organ damage, and stroke. Early and accurate diagnosis is critical to improving patient outcomes, but traditional diagnostic methods—such as blood smear microscopy, hemoglobin electrophoresis, and genetic testing—are often time-intensive, require expert interpretation, and are limited by the availability of balanced datasets. Recent advancements in artificial intelligence offer transformative potential for medical diagnostics. However, the adoption of AI-based systems for detecting SCA faces two key challenges:

1. **Limited Data Availability**: Medical datasets for rare conditions like SCA are scarce due to privacy constraints and the difficulty of obtaining labeled data.

2. **Lack of Transparency**: Traditional "black-box" AI models provide little insight into their decision-making processes, hindering clinical trust and acceptance.

To address these issues, the proposed system will:

1. **Build a Robust Diagnostic Model**: Utilize transfer learning with pre-trained models to detect sickle cells in blood smear images, integrating Support Vector Machine (SVM) for analyzing numerical or text-based data to improve overall diagnostic accuracy.

2. **Enhance Data Using Augmentation Techniques:** Apply image augmentation methods such as rotation, flipping, and brightness adjustments to increase dataset diversity and improve model robustness and generalization.

3. **Implement Explainability Through XAI**: Incorporate techniques like Grad-CAM, LIME, and attention maps to provide clinicians with interpretable insights into the model's predictions, building transparency and trust.

By combining transfer learning, data augmentation, Support Vector Machine (SVM) and explainable AI, this project aims to create an accessible, scalable, and interpretable system for sickle cell anemia detection, ultimately enhancing patient care and clinical decision-making.

1.3 Objectives

The objectives of the proposed work are as follows:

1. To apply image augmentation techniques (such as rotation, flipping, and brightness adjustment) to artificially expand the blood smear image dataset and improve the robustness and generalization of the model.

2. To develop an accurate detection system for sickle cell anemia using transfer learning with pre-trained models for blood smear image classification, and Support Vector Machine (SVM) for analyzing numerical or text-based clinical data.

3. To apply XAI methods such as Grad-CAM, LIME, and attention mechanisms.

4. To develop a user-friendly interface that allows healthcare professionals to input test samples and receive AI-driven predictions along with understandable explanations for informed decision-making.

Chapter 2

Literature Survey/ Background

In the present times, research work is going on in context of Sickle Cell Anemia Detection. In this chapter some of the major work existing in these areas has been reviewed.

Sickle Cell Anemia (SCA) is a hereditary blood disorder characterized by the presence of abnormally shaped red blood cells, which can impede blood flow and oxygen delivery throughout the body. Over the years, multiple machine learning and deep learning-based approaches have been proposed for the detection and classification of SCA. This section provides a comprehensive literature review of relevant studies focusing on automated detection techniques.

Ette et al. [1] proposed a geometric feature-based classification method for detecting anemia using Decision Tree classifiers. Their approach involved a detailed preprocessing pipeline (grayscale conversion, Gaussian blur, binary conversion, and noise removal) followed by feature extraction and classification of RBCs into different abnormal forms. This technique efficiently distinguished types of anemia based on the shapes of RBCs present in blood smear images.

Singh and Thakkar [2] explored the prediction of hydroxyurea (HU) dosage in SCA patients using LSTM and Extreme Learning Machines (ELM). Their study used 12 pathological features and concluded that LSTM outperformed ELM in dosage classification accuracy, highlighting the importance of temporal dependencies in medical data.

Das et al. [3] introduced two deep learning models—ACDSSNet-I and ACDSSNet-II—leveraging atrous convolution and DeepLabV3+ architectures. These models enhanced semantic segmentation accuracy for SCA by incorporating MobileNetV2 and ResNet50, demonstrating state-of-the-art performance with 98.21% accuracy and 99.00% specificity.

Mohamad et al. [4] implemented a CNN model in MATLAB for classifying red blood cells into normal and sickle-shaped types. A multi-layer SVM classifier achieved 95.83% accuracy in training. The study emphasized the advantages of CNN in handling overlapped and complex-shaped cells for SCD diagnosis.

Sen et al. [5] applied traditional machine learning techniques—Random Forest, Naïve Bayes, Logistic Regression, and SVM—for classifying RBCs based on geometrical, statistical, and textural features. Otsu thresholding and Watershed segmentation techniques were used for preprocessing and separating overlapped cells, respectively.

Tengshe et al. [6] proposed a lightweight CNN model using only five convolutional layers. Preprocessing involved histogram equalization and data augmentation. The model classified RBCs into three classes—normal, sickle, and others—with a testing accuracy of 94.57%. This work underscored the efficiency of deep learning in overcoming feature engineering limitations.

Goswami et al. [7] focused on using deep neural networks and explainable artificial intelligence for detecting sickle cell disease. They employed transfer learning techniques using three prominent models—GoogLeNet, ResNet18, and ResNet50—to classify red blood cells. Among these, ResNet50 delivered the highest accuracy of 94.90%. The study also integrated Grad-CAM as part of explainable AI to visualize the decision-making process, thereby improving trust and interpretability for pathologists. Their approach offers enhanced speed, precision, and reliability in diagnosing sickle cell disease.

Below table is a comparative summary of the works that have been reviewed outlining the techniques, preprocessing, models used, and accuracy achieved.

Table 1: Comparative summary of the reviewed works

Ref	**Year**	**Methodology**	**Model/Algorithm**	**Preprocessing Techniques**	**Accuracy**
[1]	2024	RBC shape classification	Decision Tree	Grayscale, Gaussian Blur, Binary Threshold, Noise Removal	Not Specified
[2]	2021	HU Dosage Prediction	LSTM, ELM	Pathological Data (12 features)	LSTM > ELM
[3]	2024	Semantic Segmentation	ACDSSNet-I, ACDSSNet-II	Atrous Conv, DeepLabV3+, MobileNetV2, ResNet50	98.21%
[4]	2023	RBC Classification	CNN + Multi-layer SVM	Morphometry, CNN Feature Extraction	95.83%
[5]	2021	RBC Shape Classification	RF, SVM, LR, NB	Otsu Thresholding, Watershed, Feature Extraction	Not Specified
[6]	2021	RBC Classification	CNN (5 conv layers)	Histogram Equalization, Data Augmentation	94.57%

Chapter 3

Software Design

This chapter describes the overall architecture, module breakdown, data flows, and technology stack for the Multi-Modal Sickle-Cell Detection System.

3.1 Architectural Overview

1. Presentation Layer
   - Streamlit App (app.py)

     i. Renders UI controls for text entry, image upload, and mode selection.

     ii. Displays model predictions and visual explanations.
2. Application Layer
   - Controller (predict.py)

     i. Orchestrates calls to text and image inference engines.

     ii. Applies ensemble logic when both inputs are present.
   - Explainability Engines

     i. LIME Explainer (utils/lime_explainer.py)

     ii. Grad-CAM Utils (utils/gradcam_utils.py)
3. Domain Layer
   - Text Model Service

     i. Loads svm_model.pkl and scaler.pkl

     ii. Performs feature scaling, SVM prediction, LIME explanation
   - Image Model Service

     i. Loads efficientnet_b0.pth

     ii. Performs image preprocessing, inference, Grad-CAM heatmap
   - Ensemble Service

     i. Combines probability vectors via soft-voting
4. Data Layer
   - Local Filesystem (models/ folder)

     i. Holds pre-downloaded model artifacts and scaler

     ii. Holds dataset for training and testing

3.2 Modules

The table shown below presents the modules used throughout the project, what was their responsibilities and their respective file names.

Table 2: Modules

Module	Responsibility	Key Files
UI / Presentation	Render inputs, outputs, and explainability visualizations	app.py
Prediction Controller	Route user inputs to correct predictor(s)	utils/predict.py
Text Predictor	Scale features, run SVM, return probabilities and label	utils/predict.py
Image Predictor	Preprocess image, run EfficientNet, return probabilities	utils/predict.py
LIME Explainer	Instantiate LimeTabularExplainer, generate local explanations	utils/lime_explainer.py
Grad-CAM Explainer	Hook model layers, compute and overlay activation heatmap	utils/gradcam_utils.py
Data Management	Load / cache models from models/ folder	utils/predict.py

3.3 Data Flow

1. User Input
   - Text → dictionary of nine CBC features
   - Image → blood smear JPEG/PNG
2. Routing (in app.py)
   - If text-only: call Text Predictor → then LIME Explainer
   - If image-only: call Image Predictor → then Grad-CAM Explainer
   - If both: call both predictors → Ensemble Service → display both explanations
3. Prediction & Explanation
   - Predictor returns (probabilities, label)
   - Explainer returns visualization object (matplotlib)
4. UI Rendering

- Streamlit displays label + numeric score + interactive LIME table or overlaid heatmap

3.4 Technology Stack

Multiple technologies have been used in the model and below table depicts all of them and how they have utilized.

Table 3: Technology Stack

Layer	Technology / Library
Presentation	Python 3.12, Streamlit
Machine Learning	scikit-learn (SVM), PyTorch (CNNs)
Explainability	LIME, pytorch-grad-cam
Data Handling	pandas, joblib
Image I/O	Pillow, OpenCV
Packaging & Env.	venv, pip, requirements.txt

3.5 Design Principles

1. **Separation of Concerns**: UI, prediction logic, and explainability code live in distinct modules.

2. **Modularity**: Each explainability technique encapsulated in its own utility file.

3. **Reproducibility**: All dependencies declared in requirements.txt; models versioned in models.

4. **Extensibility**: New models (e.g., XGBoost text, object-detection image) can be added by extending predict.py and UI controls.

Chapter 4

Requirements and Methodology

4.1 Requirements

4.1.1 Hardware Requirements

This table outlines the minimum and recommended hardware requirements for optimal performance in tasks like model training and fine-tuning. It emphasizes the need for a multi-core CPU, sufficient RAM (at least 8–16 GB), SSD storage for faster access, and a capable GPU like the NVIDIA RTX 2060 or Tesla T4 to accelerate deep learning processes.

Table 4: Hardware Requirements

Component	Minimum	Recommended
CPU	Quad-core 2.0 GHz	Quad-core 3.0 GHz or better
RAM	8 GB	16 GB or higher
Storage	10 GB free HDD	10 GB free SSD
GPU		NVIDIA RTX 2060 / Tesla T4 or better for faster model training/fine-tuning
Network	Broadband (~10 Mbps)	Broadband (>50 Mbps) for model downloads and remote datasets

4.1.2 Software Requirements

1. Operating System

- Windows 10/11 (64-bit)
- Ubuntu 20.04 LTS or higher
- macOS 12 Monterey or higher

2. Language and Runtime

- Python 3.8–3.12

3. Python Packages

All installed via pip install -r requirements.txt

Core libraries:

- numpy
- pandas
- scikit-learn
- matplotlib
- opencv-python
- torch
- torchvision
- joblib

XAI tools:

- lime
- grad-cam

Web framework:

- streamlit

4.2 Methodology

4.2.1 Image based model development

1. Data Collection

- AneRBC Dataset (Phase I)

  12,000 cropped RBC images (306 × 320 pixels) derived from 1,000 high-resolution slides (600 anemic, 600 healthy).
- Sickle-Specific Dataset (Phase II)

  422 positive (unlabelled) images containing sickle-cell morphology. 147 negative images (clear healthy RBCs).

2. Data Preprocessing and Splitting

- Image Transforms

Resize to model-specific input:

Pretrained backbones (ResNet, DenseNet, EfficientNet): 224 × 224

Custom CNN: 306 × 320

Normalization (ImageNet mean/std).

- Phase I Split (AneRBC)

  80% train / 20% validation split applied at tile level.

- Phase II Split (Sickle-Specific)

  70% train / 15% validation / 15% test split on full images.

3. Two-Phase Training Strategy

Phase I – Pretraining on AneRBC

- Purpose: Learn generic RBC morphology features (normocytes vs. anemic variations).
- Setup: Four CNNs initialized with ImageNet weights (ResNet-18, DenseNet-121, EfficientNet-B0) plus a custom 4-layer CNN.
- Training:
  - Batch size 16, Adam optimizer (lr 1e-4), Cross-Entropy loss.
  - Data augmentations: random horizontal/vertical flips, rotations ±10°, color jitter, Gaussian blur.
- Outcome: All models converge on AneRBC, achieving ≥ 88% val accuracy.

Phase II – Fine-Tuning on Sickle-Specific Data

- Purpose: Adapt pretrained feature extractors to sickle-cell detection.
- Dataset Augmentation (Negative Class only):
  - Each of 147 originals → 2 augmentations (rotations, flips, brightness/contrast jitter) → 441 total negatives.
  - Positives left at 422 to approximate class balance.
- Fine-Tuning:
  - Lower learning rate (1e-4), same optimizer and loss.
  - Early stopping on val accuracy; best checkpoints saved.

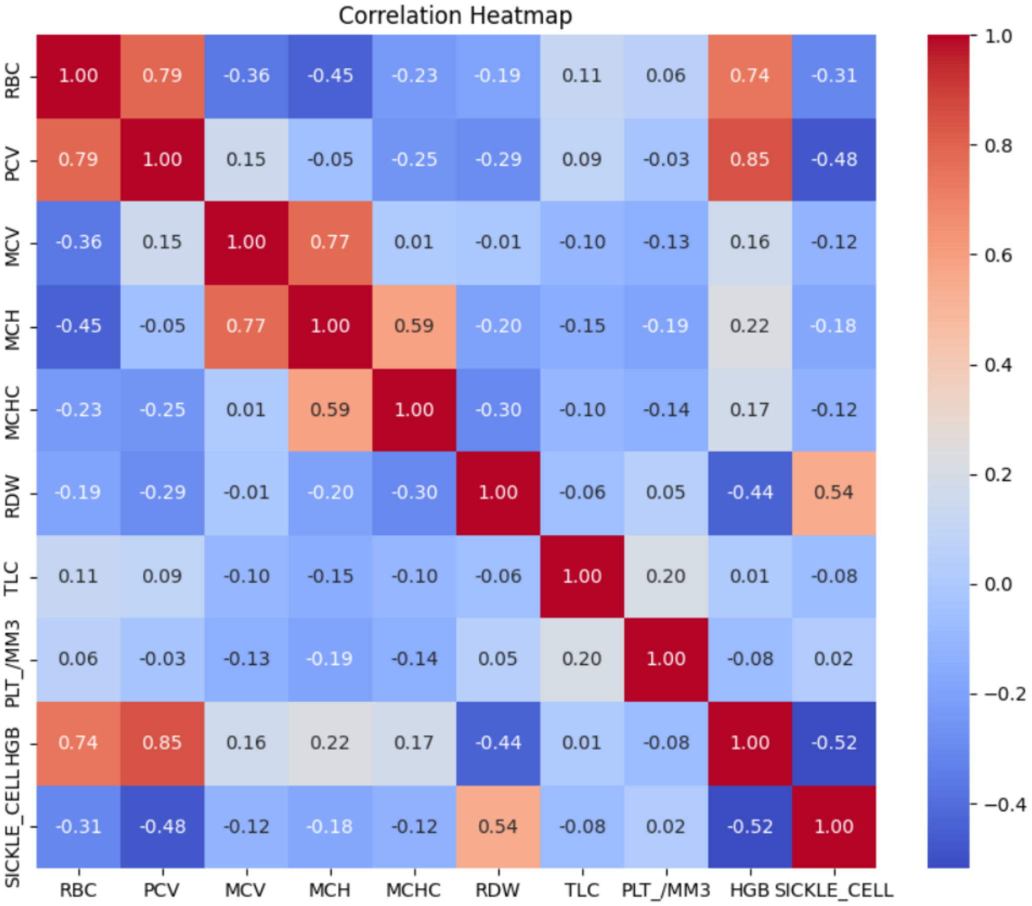
4.2.2 Text Based Model Development

For numerical data: The features selected for detecting sickle cell disease in this model are:

1. Red Blood Cell count (RBC) – Indicates the number of red blood cells in the blood.

2. Packed Cell Volume (PCV) – Represents the proportion of blood volume occupied by red blood cells.
3. Mean Corpuscular Volume (MCV) – Measures the average size of red blood cells.
4. Mean Corpuscular Hemoglobin (MCH) – Calculates the average hemoglobin content in a red blood cell.
5. Mean Corpuscular Hemoglobin Concentration (MCHC) – Determines the concentration of hemoglobin in red blood cells.
6. Red Cell Distribution Width (RDW) – Assesses the variation in red blood cell size.
7. Total Leukocyte Count (TLC) – Measures the total number of white blood cells in the blood.
8. Platelet Count (PLT/MM^3) – Evaluates the number of platelets responsible for blood clotting.
9. H emoglobin level (HGB) – Indicates the amount of hemoglobin present in the blood, essential for oxygen transport.

This correlation heatmap(Fig-1) visualizes the relationships between key blood parameters such as RBC, PCV, HGB, and SICKLE_CELL. Strong positive correlations (e.g., between PCV and HGB, 0.85) and negative correlations (e.g., SICKLE_CELL with HGB, -0.52) highlight how certain features vary together, which can help identify potential biomarkers or predictors in medical data analysis.



	RBC	PCV	MCV	MCH	MCHC	RDW	TLC	PLT_/MM3	HGB	SICKLE_CELL
RBC	1.00	0.79	-0.36	-0.45	-0.23	-0.19	0.11	0.06	0.74	-0.31
PCV	0.79	1.00	0.15	-0.05	-0.25	-0.29	0.09	-0.03	0.85	-0.48
MCV	-0.36	0.15	1.00	0.77	0.01	-0.01	-0.10	-0.13	0.16	-0.12
MCH	-0.45	-0.05	0.77	1.00	0.59	-0.20	-0.15	-0.19	0.22	-0.18
MCHC	-0.23	-0.25	0.01	0.59	1.00	-0.30	-0.10	-0.14	0.17	-0.12
RDW	-0.19	-0.29	-0.01	-0.20	-0.30	1.00	-0.06	0.05	-0.44	0.54
TLC	0.11	0.09	-0.10	-0.15	-0.10	-0.06	1.00	0.20	0.01	-0.08
PLT_/MM3	0.06	-0.03	-0.13	-0.19	-0.14	0.05	0.20	1.00	-0.08	0.02
HGB	0.74	0.85	0.16	0.22	0.17	-0.44	0.01	-0.08	1.00	-0.52
SICKLE_CELL	-0.31	-0.48	-0.12	-0.18	-0.12	0.54	-0.08	0.02	-0.52	1.00

Fig.1 Correlation Heatmap

These features are critical for diagnosing anemia, abnormal red blood cell morphology, and other hematological conditions associated with sickle cell disease.

SVM Model

1. Target Variable Creation:

   A custom rule-based function is_sickle_cell() is defined to generate the target variable SICKLE_CELL based on clinical indicators:

   - Low Haemoglobin (HGB < 11) : indicates Anemia
   - High Red Cell Distribution Width (RDW > 15): Indicates anisocytosis.
   - Low Mean Corpuscular Volume (MCV < 80): Implies microcytic anemia.
   - High Platelet Count (PLT_/MM3 > 450): May indicate reactive thrombocytosis.

The function returns 1 if any of the above conditions are met (indicating potential sickle cell disease), otherwise 0.

2. Splitting the Dataset:

- The dataset is cleaned and filtered to retain only relevant blood parameters (RBC, PCV, MCV, MCH, MCHC, RDW, TLC, PLT_/MM3, and HGB), which are stored as features (X), while the generated SICKLE_CELL column is the target (y).
- The data is split into training and testing sets using train_test_split() with 80% data for training and 20% for testing.
- Stratification is applied to maintain the same distribution of the target variable across both sets.

3. Feature Scaling:

- A StandardScaler is applied to the training data to standardize features by removing the mean and scaling to unit variance.
- The fitted scaler is also used to transform the test data, ensuring consistency.
- The trained scaler is saved using joblib.dump() for reuse during inference.

4. Handling Class Imbalance:

- The Synthetic Minority Over-Sampling Technique (SMOTE) is used on the training data to address class imbalance by generating synthetic examples of the minority class.
- This ensures the model sees an equal representation of both normal and sickle cell cases during training.

The textual dataset is pretty unevenly distributed as show in Fig-2 with majority of data belonging to Positive Sickle cell anemia class and the Negative class in minority.

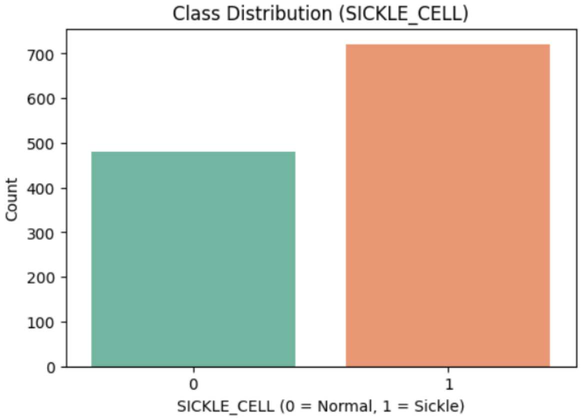


Fig.2 Class Distribution

After using SMOTE the class imbalanced has been addressed as show in Fig-3 this helped in increasing the accuracy of the SVM model.

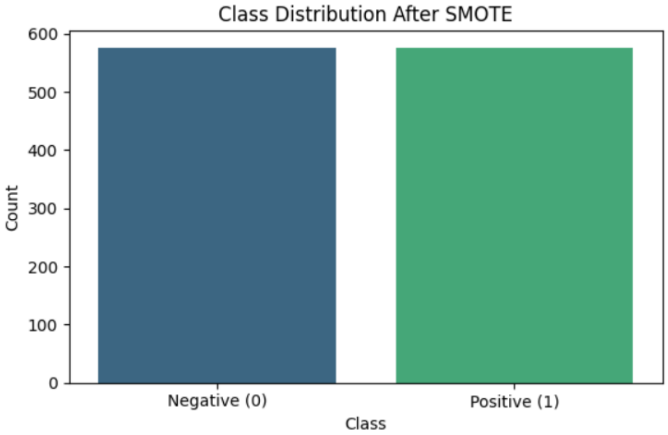


Fig. 3 Class Distribution After SMOTE

5. Model Training:

- A Support Vector Machine (SVM) classifier is trained on the balanced and scaled training data with the following configuration:
- Kernel: Default (rbf), allowing non-linear separation of classes.
- Probability Estimates: Enabled (probability=True) to obtain confidence scores for predictions.
- The model is fitted using the resampled and scaled training data.

6. Saving the Model and Preprocessing Artifacts:

- The trained SVM model is saved using joblib.dump() for later reuse.
- The same scaler is reused for prediction on unseen inputs, ensuring the same preprocessing pipeline is applied.

7. Model Evaluation:

- The model is evaluated on the test set using accuracy, classification report (precision, recall, F1-score), and confusion matrix to assess its performance.
- A class distribution plot is also generated to visualize the balance of the dataset after resampling.

8. Explainability (LIME):

- The LIME (Local Interpretable Model-Agnostic Explanations) framework is used to interpret model predictions.

- A LimeTabularExplainer is initialized with the training data and used to generate an explanation for the new input.
- This helps identify which features contributed most to the model's decision, improving transparency in medical decision support.

4.2.3 Integrate Explainable (XAI) Techniques

1. LIME Integration with SVM (Text-Based Model)

The Support Vector Machine (SVM) model, trained on numerical features such as RDW, MCV, MCH, HGB, and others, was integrated with LIME (Local Interpretable Model-agnostic Explanations) to explain its decisions. LIME works by creating perturbations around a specific input instance and then fitting an interpretable surrogate model (like a linear regression) locally to approximate the SVM's behavior.

- Process:
    - Input patient data is passed to the SVM model to obtain a prediction (Anemic/Healthy).
    - LIME perturbs the feature space and evaluates the impact of each feature on the prediction.
    - A bar plot is generated highlighting which features most influenced the decision.
- Outcome: This helps clinicians understand which clinical indicators (e.g., low haemoglobin or abnormal MCV) were most responsible for classifying a patient as sickle cell positive.

2. Grad-CAM Integration with EfficientNet-B0 (Image-Based Model)

The image-based EfficientNet-B0 model was integrated with Grad-CAM (Gradient-weighted Class Activation Mapping) to visualize which parts of a red blood cell image were most influential in the classification.

- Process:
    - When an input image is passed through EfficientNet-B0, Grad-CAM uses the gradients flowing into the last convolutional layer to produce a heatmap.
    - This heatmap is overlaid on the original image to highlight regions of interest—such as distorted or sickle-shaped areas of the red blood cells.
- Outcome: This allows users and healthcare professionals to visually inspect what the CNN model is focusing on while making predictions, increasing interpretability and model transparency.

4.2.4 UI Integration and Multimodal Prediction

The UI was integrated using a probabilistic logic and applied Ensemble Learning, and made into a Multimodal predictor, which accepts three types of input namely Text-only Input, Image-only input and a combined input of both kinds as shown through the flowchart in Fig-7.

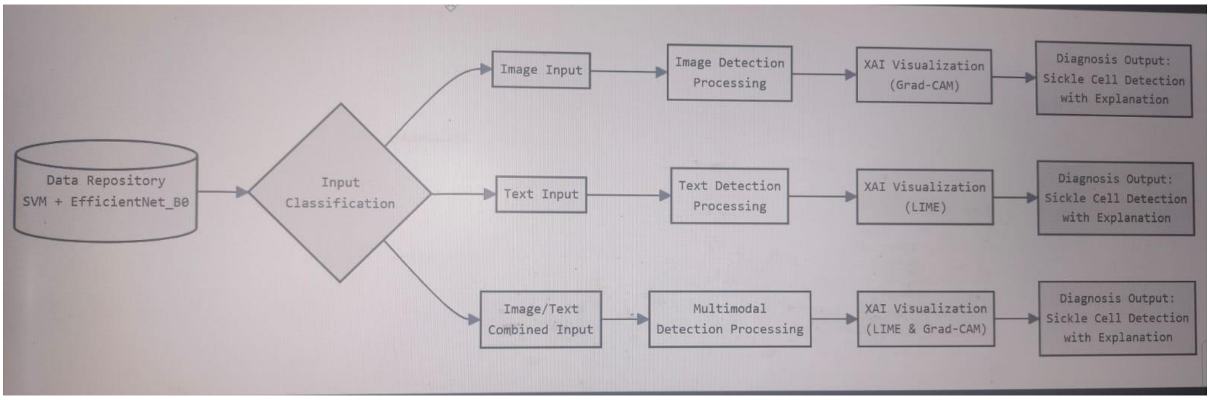

Data Repository
SVM + EfficientNet_B0
Input Classification
Image Input
Image Detection Processing
XAI Visualization (Grad-CAM)
Diagnosis Output: Sickle Cell Detection with Explanation
Text Input
Text Detection Processing
XAI Visualization (LIME)
Diagnosis Output: Sickle Cell Detection with Explanation
Image/Text Combined Input
Multimodal Detection Processing
XAI Visualization (LIME & Grad-CAM)
Diagnosis Output: Sickle Cell Detection with Explanation

Fig.4 Ensemble Learning Flowchart

Above image shows the perfect flow of our model and our methodology.

Chapter 5

Code Templates

5.1 EfficientNet-B0 Formula

Below I have attached the formulas used in the working of the EfficientNet-B0 model:

1) **Compound Scaling Formula**

   This method is used to uniformly scale the network:

   depth: $d=\alpha^{\Phi}$ width: $w=\beta^{\Phi}$, resolution: $r=\gamma^{\Phi}$

   Subject to the constraint:

   $\alpha\cdot\beta^{2}\cdot\gamma^{2}\approx 2$ (to roughly double the computation)

2) **Mobile Inverted Bottleneck Convolution (MBConv)**

   EfficientNet uses MBConv blocks, which are depth wise separable convolutions with skip connections and squeeze-and-excitation (SE) blocks. The block can be broken down into:

   MBConv Block Computation:

- Expand: F_{expand}=Conv1x1
- Depth wise Convolution: F_{dw}=DWConv(F_{expand})
- Squeeze-and-Excitation:
  - Squeeze: s=GlobalAvgPool(F_{dw})
  - Excite: $e = \sigma(W_2\cdot\delta(W_1\cdot s))$
  - Scale: $F_{se} = F_{dw}\cdot e$
- Project: $F_{project}$ = Conv1x1(F_{se})
- Residual connection: if input and output shapes match, output is:

$$Y = x + F_{project}$$

3) **Swish Activation Function**

   EfficientNet uses the Swish activation function:

$$\text{Swish}(x) = x\cdot\sigma(x)=\frac{x}{1+e^{-x}}$$

4) **Parameters and FLOPs**

   EfficientNet-B0 is designed to be optimal in terms of the number of parameters and FLOPs (floating-point operations):

- EfficientNet-B0:
  - ~5.3 million parameters
  - ~0.39 billion FLOPs

Table 5: Summary Table of Efficient-B0 Properties:

Scaling Component	**Value**
Depth coefficient (α)	1.2
Width coefficient (β)	1.1
Resolution coefficient (γ)	1.15
Baseline ϕ	0

5.2 SVM Formula

Below I have attached the formulas used in the working of the SVM model:

1) Decision Function

The SVM's decision function for prediction is:

$$f(x) = \sum_{i=1}^{\mathbb{N}} a_i y_i \, (x_i, x) + b$$

- α_i : Lagrange multipliers learned from training
- $y_i \in \{-1,1\}$: true labels of training samples
- x_i : support vectors
- x: input sample
- K(·,·): kernel function
- b: bias term

2) Kernel Function Used — RBF Kernel

Since svc() uses RBF (Gaussian) kernel by default:

$$K(x_i,x_j)=\exp(-\gamma \| x_i - x_j \|^2)$$

Where:

$$\gamma = \frac{1}{n_{features} \cdot Var(X)}$$

3) **Optimization Objective (Dual Form)**

The model solves the dual optimization problem:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

Subject to:

$0 \leq \alpha_i \leq C$, $\sum_{i=1}^{N} \alpha_i y_i = 0$

4) Prediction Rule

Once trained, the predicted class is:

$\hat{y}$ = sign(f(x))

In the code, probability=True has been used, so the probability output is computed using Platt scaling, which fits a sigmoid to the decision function output:

$$P(y = 1|x) = \frac{1}{1 + \exp\,(Af(x) + B}$$

Where A and B are learned from the training set during calibration.

5.3 User Interface code snippet

Below is the code used in the prediction logic of the Multimodal UI:

```
if predict_triggered:
        use_text = input_type in ["Text Only", "Text + Image"]
        use_image = input_type in ["Image Only", "Text + Image"]
        if use_text:
                probs_text, pred_text, input_scaled = predict_text(user_input, svm, scaler)
                st.subheader("Text Prediction")
                st.markdown(f"<div style='font-size:18px;'>Prediction: <b
                style='color:#FFA500'>{'Sickle Cell' if pred_text else 'Normal'}</b></div>",
                unsafe_allow_html=True)
                lime_exp, lime_fig = explain_with_lime(input_scaled, svm, explainer)
```

```
          st.markdown("*LIME Explanation:*")
          st.pyplot(lime_fig, clear_figure=True)
if use_image and image_file is not None:
          temp_path = "temp_uploaded.jpg"
          with open(temp_path, "wb") as f:
                    f.write(image_file.read())

          probs_img, pred_img, model, input_tensor, image_pil =
          predict_image(temp_path, image_model_path)
          st.subheader("Image Prediction")
          st.markdown(f"<div style='font-size:18px;'>Prediction: <b
          style='color:#FFA500'>{'Sickle Cell' if pred_img else 'Normal'}</b></div>",
          unsafe_allow_html=True)

          cam_img = generate_gradcam(model, input_tensor, image_pil, target_class=1)
          st.image(cam_img, caption="Grad-CAM Explanation", width=400)

          os.remove(temp_path)
if use_text and use_image:
          st.subheader("Combined Prediction")
          final_probs = (np.array(probs_text) + np.array(probs_img)) / 2
          final_pred = np.argmax(final_probs)
          st.markdown(f"<div style='font-size:18px;'>Combined Prediction: <b
          style='color:#FFA500'>{'Sickle Cell' if final_pred else 'Normal'}</b></div>",
          unsafe_allow_html=True)
```

Chapter 6

Testing

We have tested 4 models for image dataset and compared them based on their efficiency:

6.1 ResNet18

The ResNet-18 model shows strong performance in sickle cell anemia detection, achieving around 91.2% accuracy by correctly identifying 78 positive and 25 negative cases. However, it misclassified 6 sickle cell cases as negative, which could be critical in a medical context. While the model is effective overall, improving its sensitivity to reduce false negatives would enhance its reliability for clinical use.

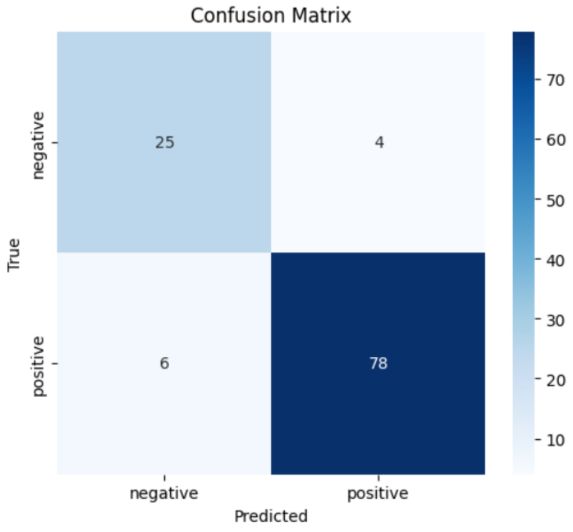
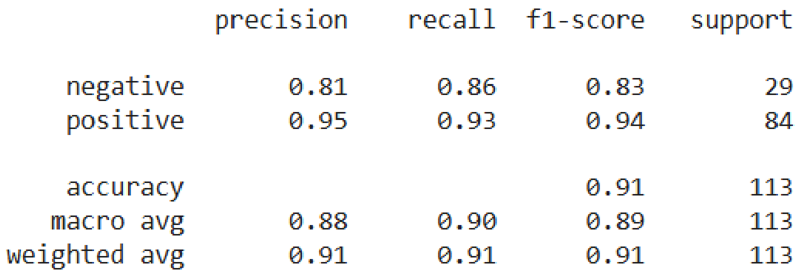


Fig.5 ResNet18 Confusion Matrix

Based on the classification report(Fig-9), the ResNet-18 model achieved a high **overall F1-score of 0.91**, with especially strong performance on positive (sickle cell) cases. While precision and recall for negative cases are slightly lower (0.81 and 0.86), the model maintains a good balance across both classes, making it effective for **sickle cell anemia detection**.



```
              precision    recall  f1-score   support

    negative       0.81      0.86      0.83        29
    positive       0.95      0.93      0.94        84

    accuracy                           0.91       113
   macro avg       0.88      0.90      0.89       113
weighted avg       0.91      0.91      0.91       113
```

Fig.6 ResNet18 performance evaluation

6.2DenseNet121

The DenseNet-121 model for sickle cell anemia detection achieved an accuracy of 88.5%, slightly lower than ResNet-18's 91%. In the confusion matrix, DenseNet-121 misclassified 6 samples from both classes, whereas ResNet-18 had fewer false negatives and false positives (4 and 6, respectively). This indicates that ResNet-18 had a marginally better ability to distinguish between positive and negative cases.(Refer Fig-10)

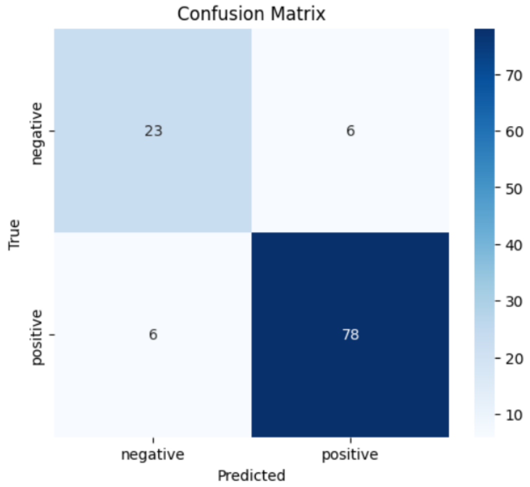
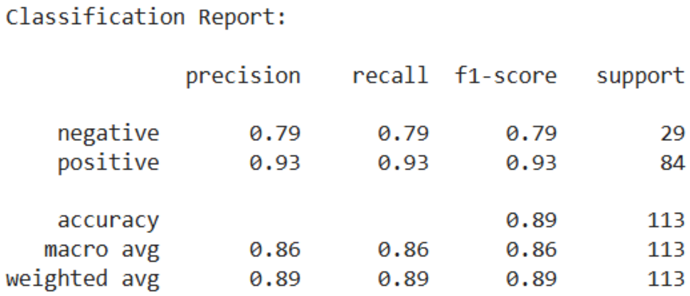


Fig.7 DenseNet121 Confusion Matrix

The classification report(Fig-11) shows DenseNet-121 has a slightly lower F1-score for the negative class (0.79) compared to ResNet-18 (0.83), while both perform similarly for the positive class. Overall, ResNet-18 offers more balanced performance across both classes.



```
Classification Report:

              precision    recall  f1-score   support

    negative       0.79      0.79      0.79        29
    positive       0.93      0.93      0.93        84

    accuracy                           0.89       113
   macro avg       0.86      0.86      0.86       113
weighted avg       0.89      0.89      0.89       113
```

Fig.8 DenseNet121 Performance Evaluation

Overall, while both ResNet-18 and DenseNet-121 perform well in detecting sickle cell anemia, ResNet-18 shows slightly better accuracy and balance between classes.

6.3EfficientNet_B0

The EfficientNet model demonstrates excellent performance in detecting sickle cell anemia, achieving a high accuracy of 99.1% as shown in the confusion matrix(Fig-12). It correctly identified 83 positive and 29 negative cases, with only one false negative and no false positives—indicating near-perfect sensitivity and specificity. Compared to ResNet-18 and DenseNet-121, EfficientNet shows significantly fewer misclassifications, making it highly suitable for clinical diagnostics.

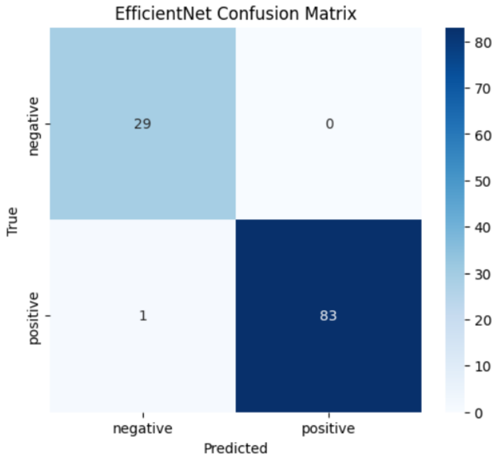
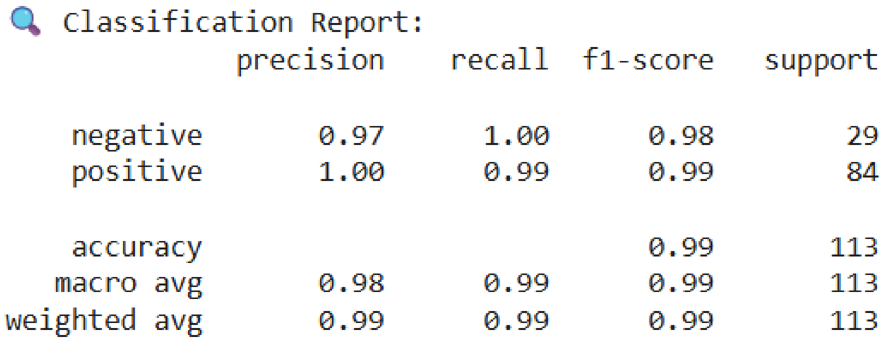


Fig.9 EfficientNet_B0 Confusion Matrix

From the classification report(Fig-13), EfficientNet achieved an impressive F1-score of 0.98 for the negative class and 0.99 for the positive class, outperforming both ResNet-18 and DenseNet-121. Its perfect recall for the negative class (1.00) and nearly perfect metrics overall indicate superior and more balanced classification capability.



```
🔍 Classification Report:
              precision    recall  f1-score   support

    negative       0.97      1.00      0.98        29
    positive       1.00      0.99      0.99        84

    accuracy                           0.99       113
   macro avg       0.98      0.99      0.99       113
weighted avg       0.99      0.99      0.99       113
```

Fig.10 EfficientNet_B0 Performance Evaluation

Overall the model seems very reliable and strong and balanced for Sickle Cell Anemia Detection.

6.4Custom CNN

The Custom CNN model achieved 81% accuracy in sickle cell anemia detection, correctly identifying 82 positive and 10 negative cases(Fig-14). However, it misclassified 19 negatives as positives and 2 positives as negatives, showing poor specificity. Compared to ResNet-18 and EfficientNet, its higher false positive rate limits clinical reliability.

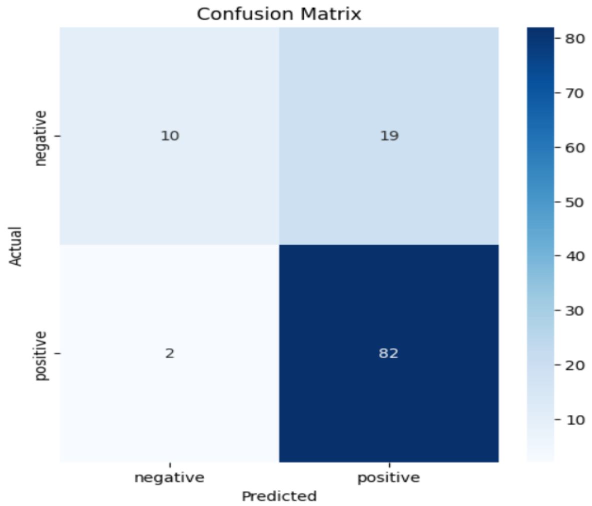
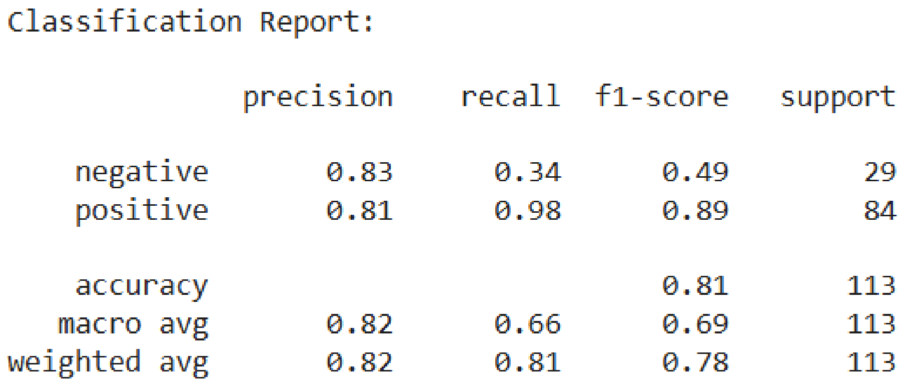


Fig.11 Custom CNN Confusion Matrix

As shown in the classification report(Fig-15), the model's recall for the negative class is just 0.34 with an F1-score of 0.49, far lower than the others. While it performs well on positive cases, the imbalance limits its clinical reliability.



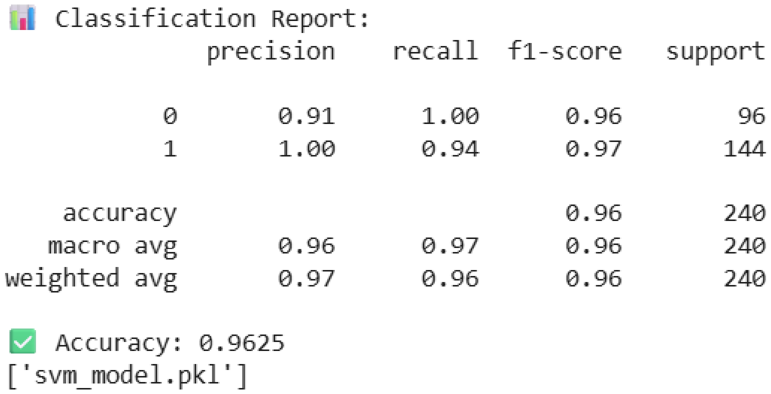
```
Classification Report:

              precision    recall  f1-score   support

    negative       0.83      0.34      0.49        29
    positive       0.81      0.98      0.89        84

    accuracy                           0.81       113
   macro avg       0.82      0.66      0.69       113
weighted avg       0.82      0.81      0.78       113
```

Fig.12 Custom CNN Performance Evaluation

EfficientNet_B0 achieves the highest accuracy and balanced performance, making it most suitable for clinical use. ResNet-18 and DenseNet-121 perform well but with slightly lower accuracy and balance. The Custom CNN shows weaker specificity, limiting its reliability.

6.5 SVM

The SVM model that is used for textual data shows high accuracy of 96% with high values in performance metrics.



```
📊 Classification Report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.96        96
           1       1.00      0.94      0.97       144

    accuracy                           0.96       240
   macro avg       0.96      0.97      0.96       240
weighted avg       0.97      0.96      0.96       240

✅ Accuracy: 0.9625
['svm_model.pkl']
```

Fig.13 SVM Performance Evaluation

This classification report indicates that the SVM model performs very well in detecting Sickle Cell Anemia from textual data. It has high precision, recall, and f1-scores for both the positive and negative classes, resulting in an overall high accuracy. The model is particularly strong at correctly identifying instances that have Sickle Cell Anemia (precision of 1.00). The slight difference between precision and recall for class 1 suggests a minor trade-off where the model might miss a few actual positive cases. However, overall, the performance appears robust.

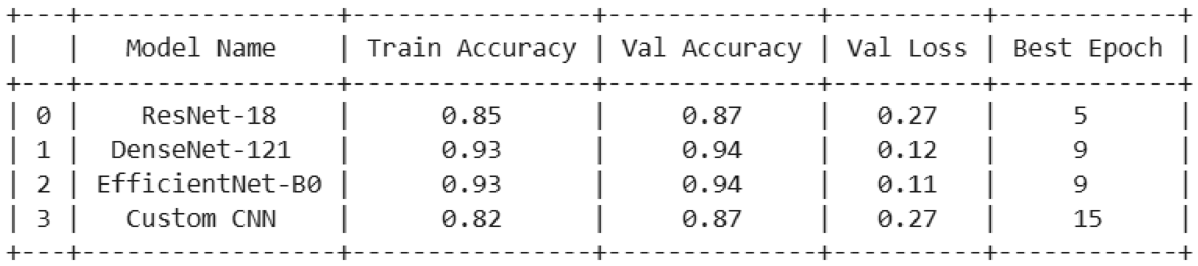
Chapter 7

Results and Discussion

7.1 Results

7.1.1 Model Evaluation & Comparison

Below are the results of the Four different models that were trained and tested and their Train Accuracy, Validation Accuracy, Validation Loss, Best epochs were checked as shown in Fig-14 to evaluate their performance and identifying potential issues like underfitting and overfitting.



```
+---+-----------------+----------------+--------------+----------+------------+
|   |   Model Name    | Train Accuracy | Val Accuracy | Val Loss | Best Epoch |
+---+-----------------+----------------+--------------+----------+------------+
| 0 |    ResNet-18    |      0.85      |     0.87     |   0.27   |     5      |
| 1 |  DenseNet-121   |      0.93      |     0.94     |   0.12   |     9      |
| 2 | EfficientNet-B0 |      0.93      |     0.94     |   0.11   |     9      |
| 3 |   Custom CNN    |      0.82      |     0.87     |   0.27   |     15     |
+---+-----------------+----------------+--------------+----------+------------+
```

Fig.14 Image Comparing different Models' Efficiency

The figure below(Fig-15) shows the percentage of accuracy achieved by each model(i.e. ResNet-18, DenseNet-121, EfficientNet-B0 and Custom CNN) in a easy visualization of a bar graph. Custom CNN showed by far the worst performance.

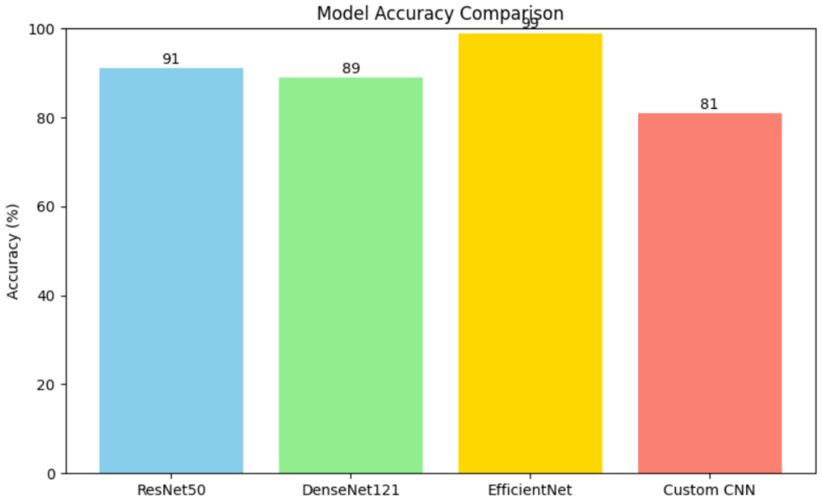
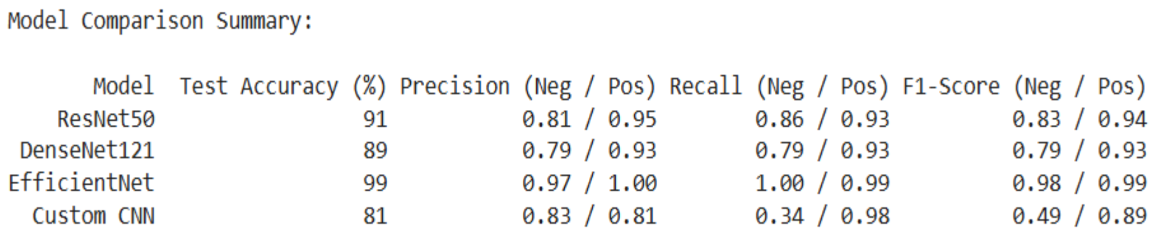


Fig.15 Model Accuracy Comparison

The models were also compared on various performance metrics like Test Accuracy, Precision, Recall and F1-Score to evaluate the effectiveness of classification models and for understanding how well each model performed in the classification.

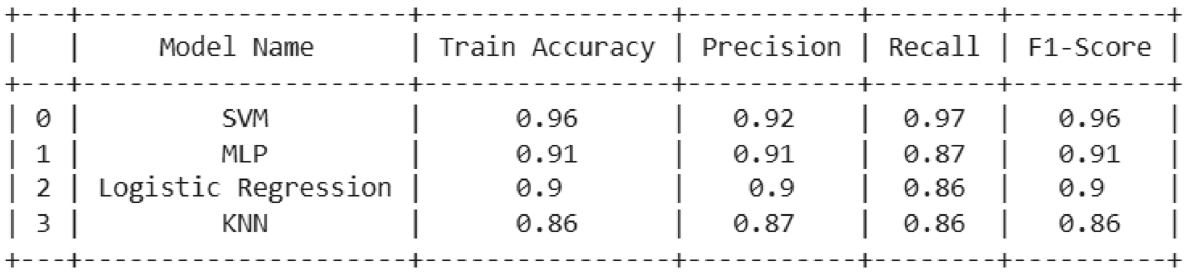


```
Model Comparison Summary:

         Model  Test Accuracy (%) Precision (Neg / Pos) Recall (Neg / Pos) F1-Score (Neg / Pos)
      ResNet50                 91           0.81 / 0.95        0.86 / 0.93          0.83 / 0.94
   DenseNet121                 89           0.79 / 0.93        0.79 / 0.93          0.79 / 0.93
  EfficientNet                 99           0.97 / 1.00        1.00 / 0.99          0.98 / 0.99
    Custom CNN                 81           0.83 / 0.81        0.34 / 0.98          0.49 / 0.89
```

Fig.16 Model Comparison Summary

As we can see EfficientNet-B0 gave an almost perfect result with accuracy of 99% and therefore, was decided as the final model choice.

This table compares four machine learning models — SVM, MLP, Logistic Regression, and KNN — based on their training accuracy and evaluation metrics: precision, recall, and F1-score.



```
+---+---------------------+----------------+-----------+--------+----------+
|   |     Model Name      | Train Accuracy | Precision | Recall | F1-Score |
+---+---------------------+----------------+-----------+--------+----------+
| 0 |         SVM         |      0.96      |   0.92    |  0.97  |   0.96   |
| 1 |         MLP         |      0.91      |   0.91    |  0.87  |   0.91   |
| 2 | Logistic Regression |      0.9       |    0.9    |  0.86  |   0.9    |
| 3 |         KNN         |      0.86      |   0.87    |  0.86  |   0.86   |
+---+---------------------+----------------+-----------+--------+----------+
```

Fig.17 Performance comparison of models

Based on the evaluation, SVM is the most reliable model, providing high accuracy and a strong balance of precision and recall, making it the best choice for deployment. MLP also performs well and may be considered depending on computational or architectural preferences. Logistic Regression and KNN offer acceptable but comparatively weaker performance.

7.1.2 User Interface Case 1: Numerical Data Only (SVM Classifier)

In situations where only the patient's blood report values are available, the system uses a Support Vector Machine (SVM) classifier. The model was trained on structured data using clinically relevant hematological parameters such as:

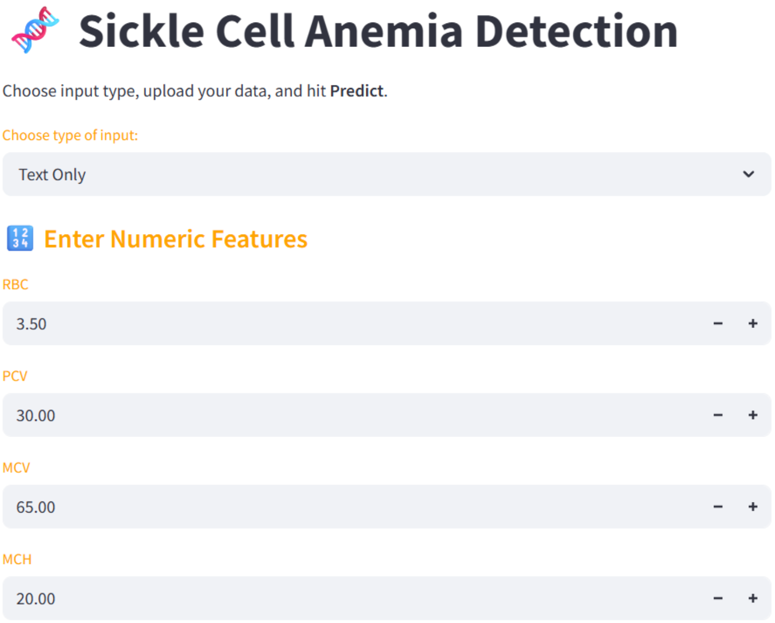
- RDW, PCV, MCV, MCH, RBC, HGB, TLC, MCHC, PLT

Results:

- The SVM model achieved strong accuracy on cross-validation.

- Integration with LIME allowed for transparent feature attribution, showing how individual features contributed to the classification (e.g., low hemoglobin or abnormal MCV influencing a positive prediction).

This user interface(Fig-18.1, 18.2) for **Sickle Cell Anemia Detection** allows users to input data, currently showing fields for numeric blood test features (RBC, PCV, MCV, MCH) with example values. The "Choose type of input" dropdown is set to "Text Only," suggesting the system can also analyze textual data for detection. Users would input their data and then click a "Predict" button to get the diagnosis.



Sickle Cell Anemia Detection

Choose input type, upload your data, and hit **Predict**.

Choose type of input: Text Only

Enter Numeric Features

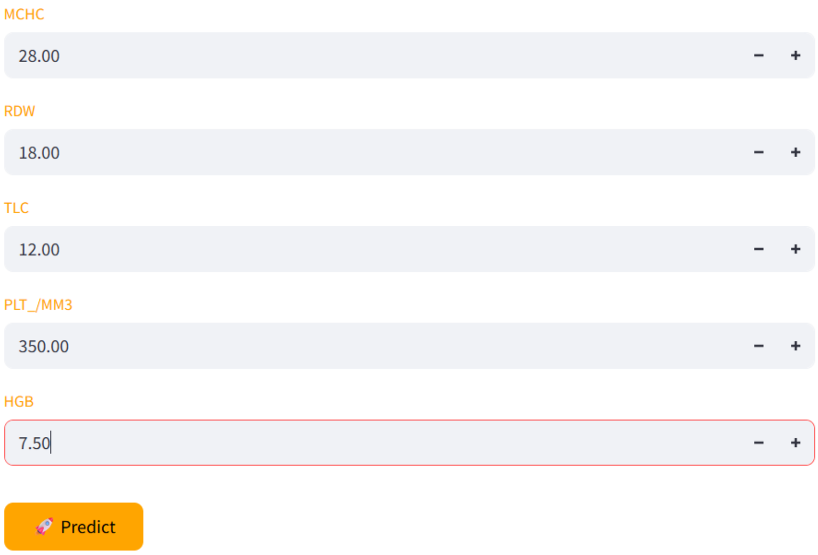
RBC: 3.50

PCV: 30.00

MCV: 65.00

MCH: 20.00

Fig.18.1 Text-only UI Input



MCHC: 28.00

RDW: 18.00

TLC: 12.00

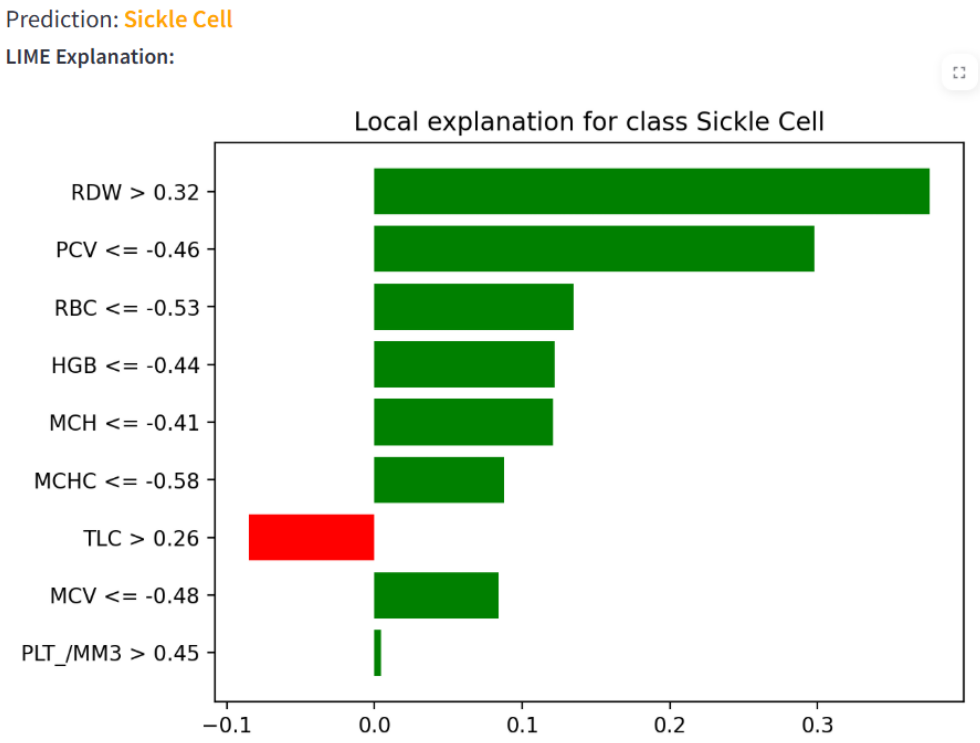
PLT_/MM3: 350.00

HGB: 7.50

Fig.18.2 Text-only UI Input

Output

Based on the numeric inputs (RBC: 3.50, PCV: 30.00, MCV: 65.00, MCH: 20.00) from Fig. 18, the model has predicted Sickle Cell. The LIME explanation highlights the factors that most influenced this prediction. Higher RDW, lower PCV, RBC, HGB, MCH, MCHC, and MCV strongly support the "Sickle Cell" prediction (green bars), while a lower TLC slightly counteracted it (red bar). Platelet count (PLT_/MM3) had a minimal positive influence.



Prediction: Sickle Cell

LIME Explanation:

Fig.19 Text based UI Output

7.1.3 User Interface Case 2: Image Data Only (EfficientNet-B0 CNN)

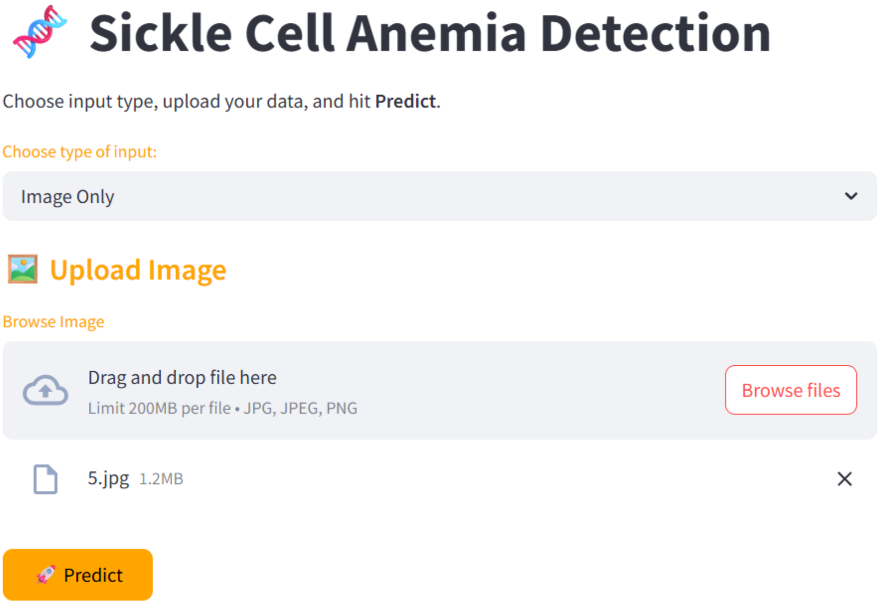
When blood smear images are available but textual data is missing, the EfficientNet-B0 model is used. This model was selected after a comparative evaluation of four CNN architectures, and it showed the best performance with 99% accuracy.

Results:

- EfficientNet-B0 was able to identify sickle-shaped RBCs with high reliability.
- The model was pretrained on the AneRBC dataset (12,000 images) and fine-tuned on a smaller sickle cell-specific dataset (422 positive and 127 negative images).
- Data augmentation techniques enhanced generalization.
- Grad-CAM visualizations were integrated to highlight which parts of the image influenced the prediction, ensuring explainability.

Input: 1

This interface is set to "Image Only" input for Sickle Cell Anemia Detection. The user has uploaded an image file ("5.jpg") by dragging and dropping or browsing. Clicking the "Predict" button will initiate the model's analysis of this image to determine the likelihood of Sickle Cell Anemia.



🧬 Sickle Cell Anemia Detection

Choose input type, upload your data, and hit **Predict**.

Choose type of input:

Image Only ⌄

🖼️ Upload Image

Browse Image

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

5.jpg 1.2MB ✕

🚀 Predict

Fig.20 Image only UI Input 1

The uploaded blood smear image was analyzed by the Sickle Cell Anemia detection model. The model's prediction is **Sickle Cell**. To provide insight into the decision-making process, a Grad-CAM (Gradient-weighted Class Activation Mapping) visualization was generated. The highlighted areas in the Grad-CAM output indicate the regions within the blood smear image that most strongly contributed to the "Sickle Cell" prediction, suggesting the model identified morphological features characteristic of the condition, such as the presence of sickled red blood cells.

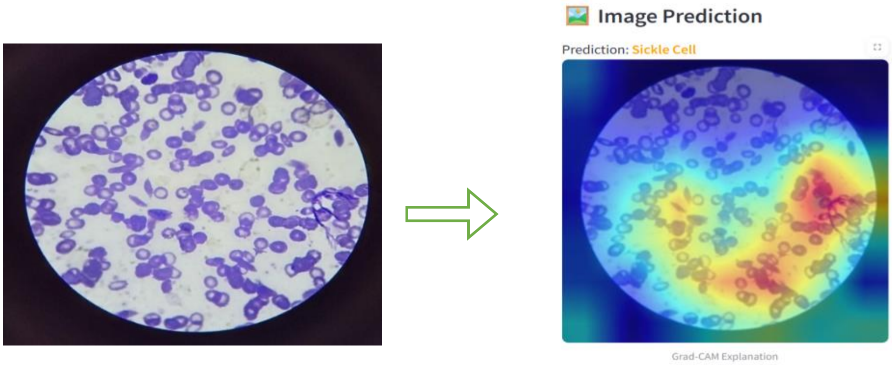
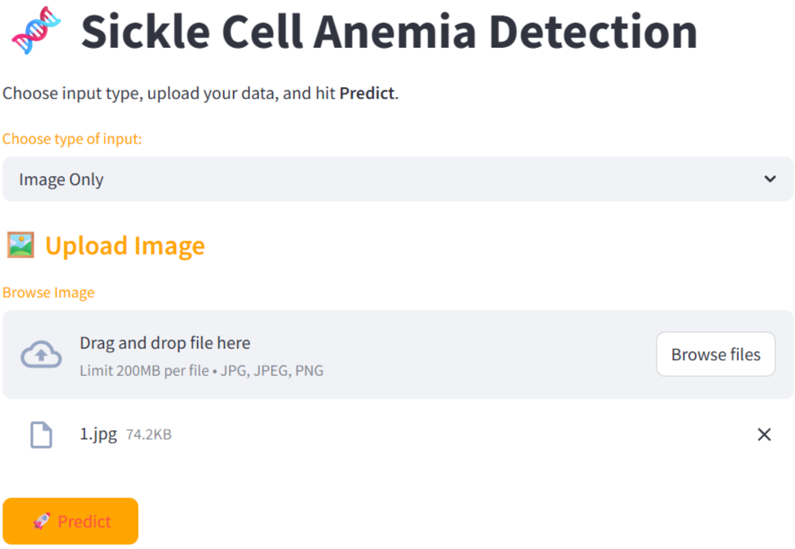


Fig.21 Image 1 Based UI Output

Input:2

The user interface(Fig-22) shows the system configured for "Image Only" input. An image file named "1.jpg" (74.2KB) has been uploaded. Clicking the "Predict" button will initiate the analysis of this new blood smear image to determine the likelihood of Sickle Cell Anemia based on its visual characteristics.



🧬 Sickle Cell Anemia Detection

Choose input type, upload your data, and hit **Predict**.

Choose type of input:

Image Only

🖼️ Upload Image

Browse Image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

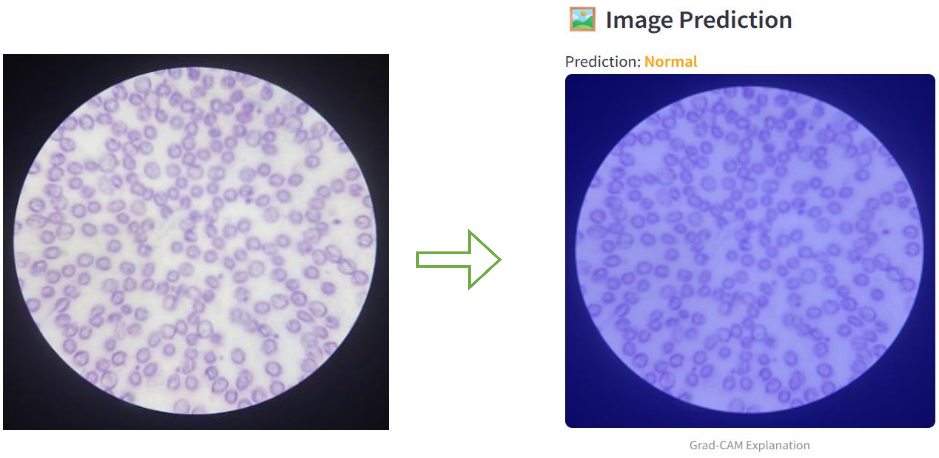
Browse files

1.jpg 74.2KB ×

Predict

Fig.22 Image only UI Input 2

The uploaded blood smear image was analyzed by the Sickle Cell Anemia detection model. The model's prediction is Normal. The Grad-CAM visualization indicates that no specific regions within the blood smear strongly activated the "Sickle Cell" classification. This suggests that the model did not detect the morphological features typically associated with Sickle Cell Anemia in the provided image.



🖼️ Image Prediction

Prediction: Normal

Fig.23 Image 2 Based UI Output

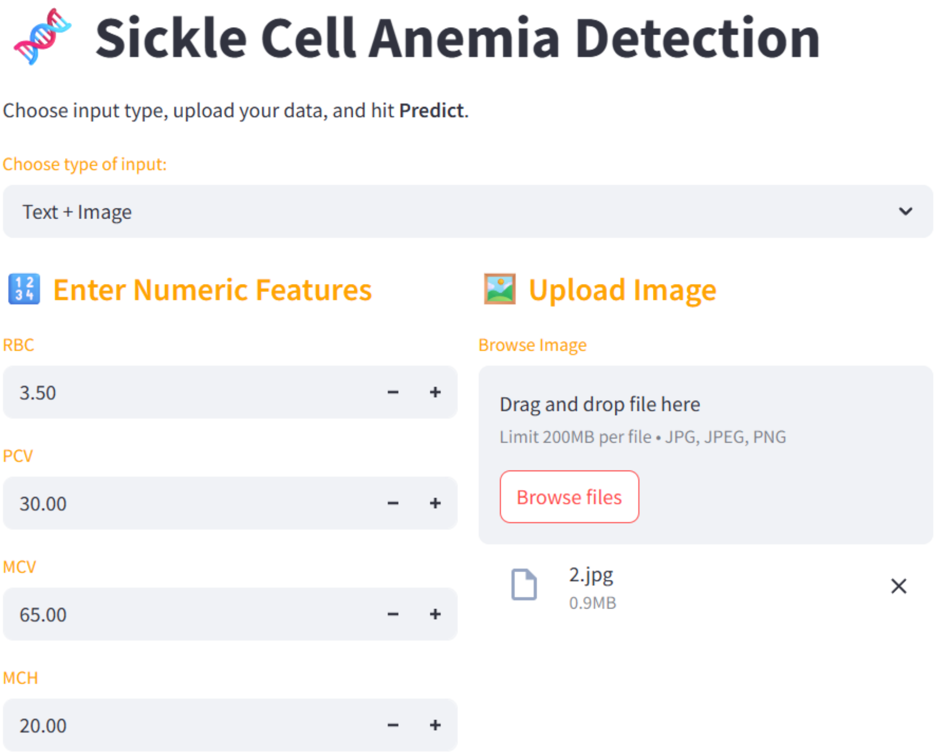
7.1.4 User Interface Case 3: Both Numerical and Image Data Available (Ensemble Decision)

When both image and numerical data are provided, the system combines predictions from the SVM and EfficientNet models using a decision logic (e.g., weighted ensemble or conditional activation based on confidence thresholds).

Results:

- Ensemble predictions improved overall diagnostic robustness.
- The combined system benefits from both statistical trends in blood metrics and visual characteristics from images.
- This approach is most reflective of real clinical workflows where both data types are available for evaluation.

This interface allows for a combined input approach for Sickle Cell Anemia detection, utilizing both numeric blood test features and an uploaded blood smear image. The user has entered values for RBC (3.50), PCV (30.00), MCV (65.00), and MCH (20.00). Additionally, an image file named "2.jpg" (0.9MB) has been uploaded, suggesting the model will analyze both the quantitative blood parameters and the visual information from the blood sample.



Sickle Cell Anemia Detection

Choose input type, upload your data, and hit **Predict**.

Choose type of input:

Text + Image

🔢 Enter Numeric Features

RBC

3.50

PCV

30.00

MCV

65.00

MCH

20.00

Upload Image

Browse Image

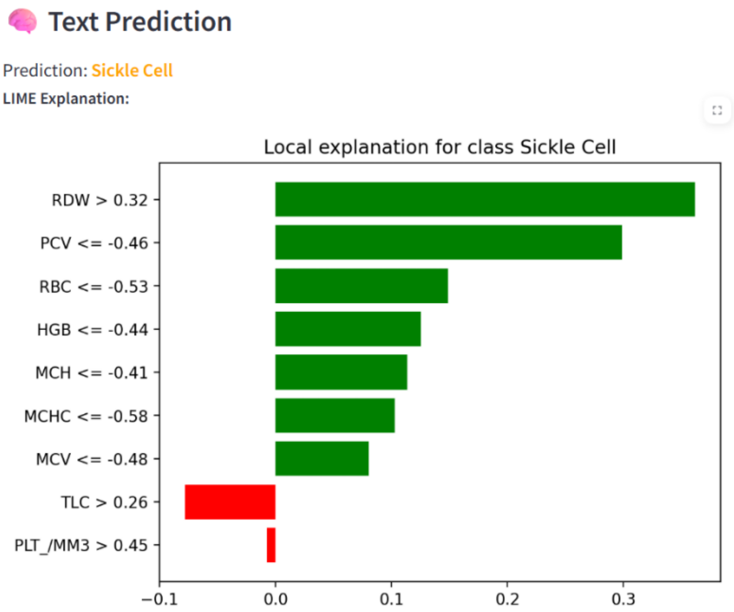
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

2.jpg
0.9MB

Fig.24 Text+Image UI Input

The text prediction output indicates a "Sickle Cell" diagnosis based on the provided numeric blood features. The LIME explanation highlights that lower values for RDW, PCV, RBC, HGB, MCH, MCHC, and MCV strongly support this prediction (green bars). A lower TLC slightly counteracts this (red bar), while platelet count has a minimal positive influence, collectively leading to the "Sickle Cell" classification from the numerical data.



Text Prediction

Prediction: Sickle Cell

LIME Explanation:

Fig.25 Text Prediction UI Output

The image prediction output also indicates a "Sickle Cell" diagnosis based on the analysis of the uploaded blood smear image. The Grad-CAM visualization highlights a central region within the image as most influential for this prediction. This suggests the model identified visual patterns, likely the presence of sickled red blood cells, in that specific area as key indicators of Sickle Cell Anemia.

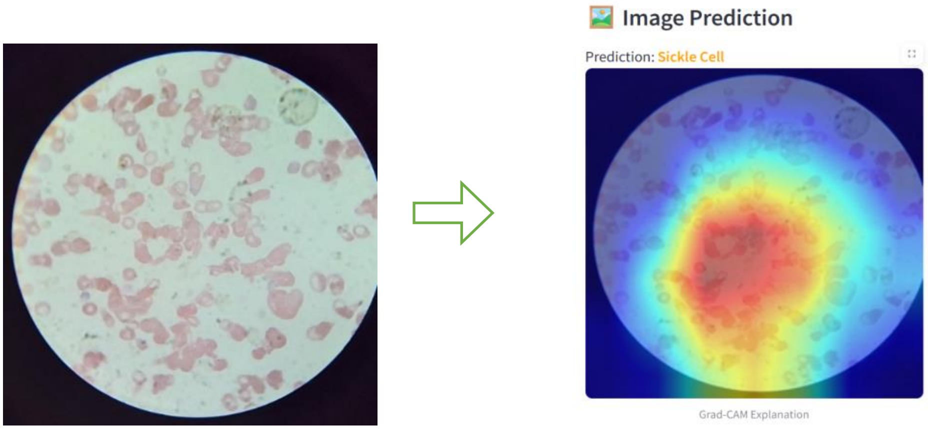
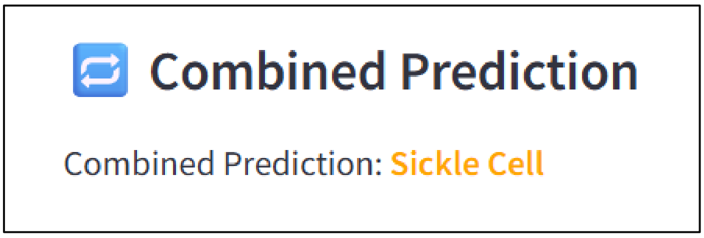


Image Prediction

Prediction: Sickle Cell

Grad-CAM Explanation

Fig.26 Image Prediction UI Output



🔁 Combined Prediction

Combined Prediction: Sickle Cell

Fig.27 Combined UI Output

7.2 Discussion

- The system provides flexibility for use in a variety of practical situations depending on available data.
- The use of explainable AI builds user trust, especially important in medical diagnostics.
- Fine-tuning on disease-specific data helped models generalize better to sickle cell cases.
- The pipeline is modular, enabling future upgrades or model replacements with minimal changes.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The project successfully developed an AI-driven system for the detection of sickle cell anemia by integrating image-based and clinical text data analysis. Through the use of transfer learning with pre-trained CNN models—particularly EfficientNet-B0—and classical machine learning via Support Vector Machine (SVM), the system achieved high accuracy in both image and numerical input cases. The application of data augmentation techniques such as random rotations, flips, and brightness/contrast adjustments significantly enhanced model robustness by simulating real-world variations in blood smear samples.

The project also integrated Explainable AI (XAI) tools like **Grad-CAM** and **LIME**, allowing healthcare professionals to understand the reasoning behind model predictions. This transparency is critical for clinical applications, as it builds trust and offers insight into the model’s focus areas (in images) and decision logic (in text data).

In comparison to traditional methods of disease diagnosis—which can be time-consuming, subject to human error, and require extensive expertise—this system offers an automated, scalable, and accurate alternative that can be deployed in real-world healthcare environments. Moreover, the user-friendly interface bridges the gap between complex AI models and end-users, making the system practical and accessible for clinical decision-making.

8.2 Future Work

While the current system demonstrates high performance, there is still scope for improvement and expansion. Future work may include:

1. Incorporation of Larger and More Diverse Datasets: To improve generalization across different populations and imaging conditions, more annotated data from varied sources should be collected and used for training and validation.
2. Real-time Detection and Deployment: Developing a lightweight version of the model for real-time inference on mobile or embedded devices would make the system more portable and useful in remote or resource-limited settings.
3. Integration with Hospital Databases and EHRs: Linking the system to electronic health records (EHR) can automate patient monitoring and offer holistic insights combining historical trends and current predictions.
4. Enhanced Explainability: Expanding the use of explainability tools by integrating attention mechanisms or SHAP values to further increase trust and interpretability in clinical environments.
5. Multi-modal Diagnosis Extension: Combining other forms of medical data, such as genomics or patient history, could further increase diagnostic accuracy and offer comprehensive insights for rare variants of anemia.

References

[1] Ette, T. N., Nagalla, H. N., Addanki, V., & Kumar, G. K. (2024). Geometric Feature-Based Classification of Anemia in Blood Smear Images using Decision Tree. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–6. https://doi.org/10.1109/icccnt61001.2024.10724527

[2] Singh, B. K., & Thakkar, H. (2021). Hydroxyurea Dosage Classification for Sickle Cell Disease Patients. Hydroxyurea Dosage Classification for Sickle Cell Disease Patients, 1289–1292. https://doi.org/10.1109/icict50816.2021.9358788

[3] Das, P. K., Dash, A., & Meher, S. (2024). ACDSSNet: Atrous Convolution-based Deep Semantic Segmentation Network for Efficient Detection of Sickle Cell Anemia. IEEE Journal of Biomedical and Health Informatics, 28(10), 5676–5684. https://doi.org/10.1109/jbhi.2024.3362843

[4] Mohamad, A. S., Nordin, M. N., & Senu, A. C. (2023). Convolutional Neural Network Modeling for Classification of Human Red Blood Cells in Sickle Cell Disease Diagnosis. Convolutional Neural Network Modeling for Classification of Human Red Blood Cells in Sickle Cell Disease Diagnosis, 17, 241–246. https://doi.org/10.1109/ice2t58637.2023.10540528

[5] Sen, B., Ganesh, A., Bhan, A., Dixit, S., & Goyal, A. (2021). Machine learning based Diagnosis and Classification Of Sickle Cell Anemia in Human RBC. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 753–758. https://doi.org/10.1109/icicv50876.2021.9388610

[6] Tengshe, R., N, A. U., Raj, A., K, A., Pattanshetty, A. A., & Fatimah, B. (2021). Sickle cell anemia detection using convolutional neural network. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 9, 1–6. https://doi.org/10.1109/icccnt51525.2021.9580165

[7] Goswami, N. G., Goswami, A., Sampathila, N., Bairy, M. G., Chadaga, K., & Belurkar, S. (2024). Detection of sickle cell disease using deep neural networks and explainable artificial intelligence. Journal of Intelligent Systems, 33(1), Article 20230179. https://doi.org/10.1515/jisys-2023-0179

[8] Deng, H., Zhou, Y., Wang, L., & Zhang, C. (2021). Ensemble learning for the early prediction of neonatal jaundice with genetic features. BMC Medical Informatics and Decision Making, 21, 338. https://doi.org/10.1186/s12911-021-01701-9

[9] Abdulrazzak, A. Y., Mohammed, S. L., Al-Naji, A., & Chahl, J. (2024). Real-Time Jaundice Detection in Neonates Based on Machine Learning Models. BioMedInformatics, 4(1), 623-637. https://doi.org/10.3390/biomedinformatics4010034

[10] Mukkapati, N., & Anbarasi, M. S. (2022). Brain tumor classification based on enhanced CNN model. Revue d'Intelligence Artificielle, 3(1), 125-130. https://doi.org/10.18280/ria.360114

[11] Khamparia, A., Singh, P. K., Rani, P., Samanta, D., Khanna, A., & Bhushan, B. (2021). An internet of health things-driven deep learning framework for detection and classification of skin cancer

using transfer learning. Transactions on Emerging Telecommunications Technologies, 32(7), e3963. https://doi.org/10.1002/ett.3963

[12] Inamori, G., Kamoto, U., Nakamura, F., Isoda, Y., Uozumi, A., Matsuda, R., Shimamura, M., Okubo, Y., Ito, S., & Ota, H. (2021). Neonatal wearable device for colorimetry-based real-time detection of jaundice with simultaneous sensing of vitals. Science Advances, 7(10), eabe3793. https://doi.org/10.1126/sciadv.abe3793

[13] Chakraborty, A., Goud, S., Shetty, V., & Bhattacharyya, B. (2020). Neonatal jaundice detection system using CNN algorithm and image processing. International Journal of Electrical Engineering and Technology, 11(3), 248-264. https://doi.org/10.34218/IJEET.11.3.2020.029

[14] Ette, T. N., Nagalla, H. N., Addanki, V., & Kumar, G. K. (2024). Geometric Feature-Based Classification of Anemia in Blood Smear Images using Decision Tree. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–6. https://doi.org/10.1109/icccnt61001.2024.10724527

[15] Singh, B. K., & Thakkar, H. (2021). Hydroxyurea Dosage Classification for Sickle Cell Disease Patients. 2021 6th International Conference on Inventive Computation Technologies (ICICT), 1289–1292. https://doi.org/10.1109/icict50816.2021.9358788

[16] Das, P. K., Dash, A., & Meher, S. (2024). ACDSSNet: Atrous Convolution-based Deep Semantic Segmentation Network for Efficient Detection of Sickle Cell Anemia. IEEE Journal of Biomedical and Health Informatics, 28(10), 5676–5684. https://doi.org/10.1109/jbhi.2024.3362843

[17] Mohamad, A. S., Nordin, M. N., & Senu, A. C. (2023). Convolutional Neural Network Modeling for Classification of Human Red Blood Cells in Sickle Cell Disease Diagnosis. 2023 International Conference on Engineering Technology and Technopreneurship (ICE2T), 17, 241–246. https://doi.org/10.1109/ice2t58637.2023.10540528

[18] Sen, B., Ganesh, A., Bhan, A., Dixit, S., & Goyal, A. (2021). Machine learning based Diagnosis and Classification Of Sickle Cell Anemia in Human RBC. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 753–758. https://doi.org/10.1109/icicv50876.2021.9388610

[19] Tengshe, R., N, A. U., Raj, A., K, A., Pattanshetty, A. A., & Fatimah, B. (2021). Sickle cell anemia detection using convolutional neural network. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 9, 1–6. https://doi.org/10.1109/icccnt51525.2021.9580165