HematoVision: Advanced blood cell classification using transfer learning

Team Name:

Maddala Lazer

Team Members:

- Madrasu Pavithra
- Maddala Lazer

Phase-1: Brainstroming & Ideation

Objective:

- Problem Statement:

Microscopic analysis of blood cells plays a critical role in diagnosing conditions such as leukemia, anemia and infections. However, manual analysis is time-consuming and subject to human error. HematoVision utilizes transfer learning to classify blood cell types from microscopic images accurately. By integrating AI into hematology tabs, this system enhances diagnostic speed, precision and reliability, supporting early disease detection and improved patient outcomes.

- Purpose of the Project:

The purpose of hematovision is to develop an intelligent system that can automatically classify different types of blood cells using deep learning techniques, specifically transfer learning. This project aims to:

- **Assist medical professionals** in analyzing blood samples quickly and accurately.

- **Reduce manual errors** in microscopic blood cell classification.
- **Speed up diagnosis** of blood-related diseases like infections and leukemia.
- **Make diagnostics accessible** in remote or low-resource areas through automation.

- Impact of the Project:
  - **Early Disease Detection:** Helps in diagnosing diseases like leukemia or infections through accurate cell type identification.
  - **Faster Results:** Automates classification, reducing the time needed by pathologists.
  - **Smart Healthcare:** Integrates AI into medical labs for better efficiency.
  - **Useful in Telemedicine:** Enables remote diagnostics in rural or under-equipped areas.
- Key points:

1.Problem Statement:

- Manual blood cell classification is time-consuming, error-prone, and needs expert pathologists.
- Lack of access to diagnostic tools in rural or low-resource areas.

2.Proposed Solution:

- Use transfer learning with a pre-trained CNN model (e.g., VGG16) to automatically classify blood cells from microscopic images.

3. **Target Users:**

- Doctors and Pathologists
- Medical Laboratories

- Remote Healthcare Centers

4.**Expected Outcome:**

- Accurate and fast blood cell classification
- Support for early diagnosis
- Reduction in manual workload
- Enhanced accessibility in telemedicine and rural care

Phase-2: Requirements Analysis

- Objective**:**
- Functional Requirements:

Functional requirements define what the system should do. They describe the core functionalities, features, and behaviors of the system from the user's perspective.

- Technical Requirements:

Technical requirements define how the system will be built. They include the tools, technologies, platforms, and performance standards needed for implementation.

- Key Points:

1. Technical Requirements:

- Languages: Python, HTML, CSS, JavaScript
- Frameworks: TensorFlow/Keras, Flask
- Tools: Jupyter Notebook, OpenCV, VS Code

2.**Functional Requirements:**

- Upload and preview blood cell images
- Predict blood cell type (Neutrophil, Eosinophil, etc.)
- Display prediction result
- Show accuracy/performance graphs
- User-friendly web interface

3. **Constraints and Challenges:**
    - Limited dataset size (may affect accuracy)
    - Model overfitting risk
    - High-quality images required for good predictions
    - Deployment compatibility issues

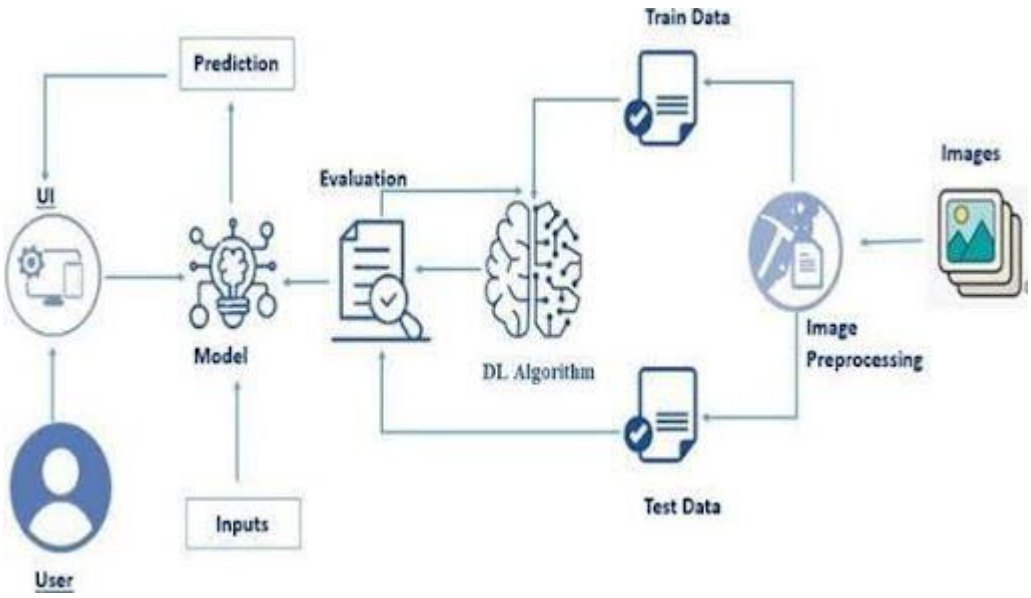
Phase 3: Project Design

- Objective:

  To create the system architecture and define the user flow for the HematoVision application.

- Key Points:

  1.System Architecture diagram:



  2.User Flow:
    - User opens the web app
    - Uploads a blood cell image
    - Clicks "Predict" button
    - System processes image and shows predicted cell type
    - We get an blood cell name.

3. UI/UX Considerations:

- Simple and clean interface
- Image preview before prediction
- Clear display of prediction result
- Responsive design for desktop and mobile

Phase 4: Project Planning (Agile Methodologies)

- Objective:
  Break down the project into smaller, manageable tasks using Agile methodology
- Key Points:

1.Sprint Planning:

- Data collection, preprocessing
- Model training (VGG16 using transfer learning)
- Web UI design (HTML/CSS)
- Flask backend + Model integration
- Testing & deployment
- Final report & presentation

2.**Task Allocation:**

- **Madrasu Pavithra:** Model development, documentation, flask integration
- **Maddala Lazer:** UI/UX design, testing, deployment

3.**Timeline & Milestones:**

- Dataset ready, preprocessing complete.
- Model trained with good accuracy.
- Web UI ready.
- Backend connected to model.

- Full testing & bug fixing.
- Project demo and documentation completed.

Phase 5: Project Development

- Objective:

To code the project and integrate all components (model, backend, and frontend) into a working application.

- Key Points:

1.Technology Stack Used:

- **Languages:** Python, HTML, CSS, JavaScript
- **Frameworks:** Flask, TensorFlow/Keras
- **Model File:** .h5 (trained VGG16 model)
- **Files Used:**
  - index.html – Upload page
  - result.html – Result display page
  - app.py – Flask backend
  - model.h5 – Pre -trained blood cell classification model
  - requirements.txt – List of dependencies

2. **Development Process:**

- Trained VGG16 model and saved it as model.h5
- Designed index.html for image upload
- Created result.html to show predictions
- Built app.py to connect frontend with model
- Listed required libraries in requirements.txt

3.**Challenges and Fixes:**

- Image not loading correctly**:**
  Fixed by checking file paths and using secure_filename

- Model prediction errors**:**
  Resolved with proper image resizing and preprocessing
- Flask not updating prediction:
  Handled with correct routing and form methods
- Deployment issues**:**
  Installed correct versions via requirements.txt

Phase 6: Functional & Performance Testing

- Objective:

  To ensure the project works as expected by testing all features and measuring performance.

- Key Points:

  1. **Test Cases Executed:**

     - Upload valid blood cell image → prediction displayed correctly
     - Upload unsupported file format → error message shown
     - Predict multiple times with different images → works correctly
     - Broken or missing image → handled gracefully
     - UI buttons (upload, predict) → functional and responsive
     - Page reload → system behaves correctly without crash

Welcome to the HematoVision

About Blood Cells

Blood cells play essential roles in immunity, oxygen transport, and clotting. Understanding different types of blood cells is crucial for diagnosis.

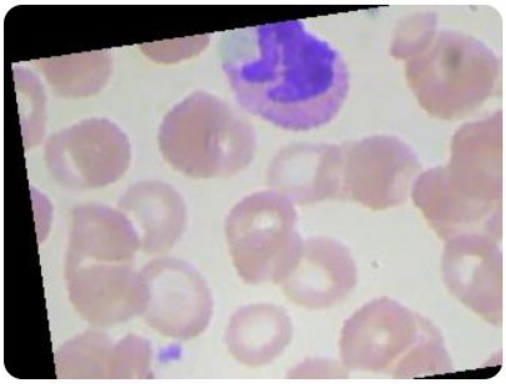
Predict Blood Cell Type

Choose File No file chosen

Predict

Prediction Result

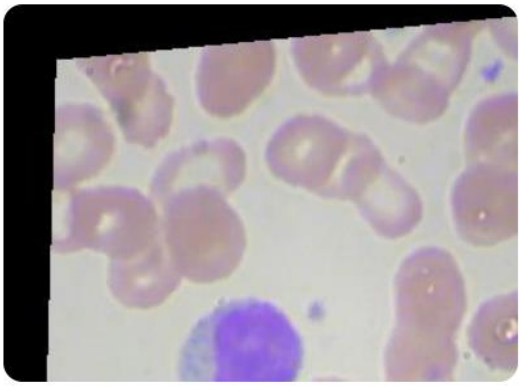
Predicted Class: eosinophil



Upload Another Image

Prediction Result

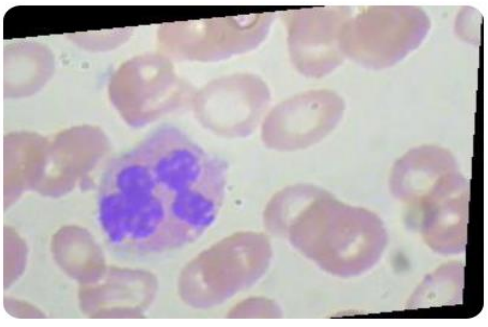
Predicted Class: monocyte



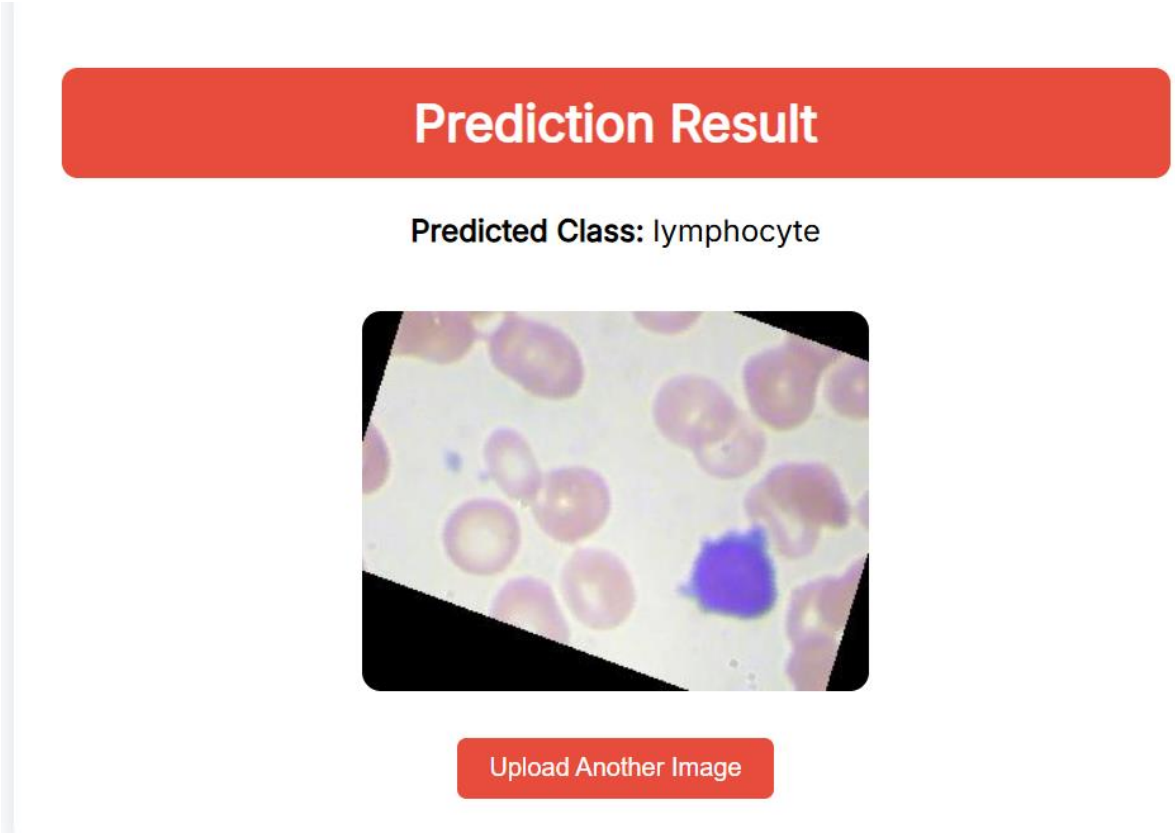
Upload Another Image

Prediction Result

Predicted Class: neutrophil



Upload Another Image



2.**Bug Fixes and Improvements:**

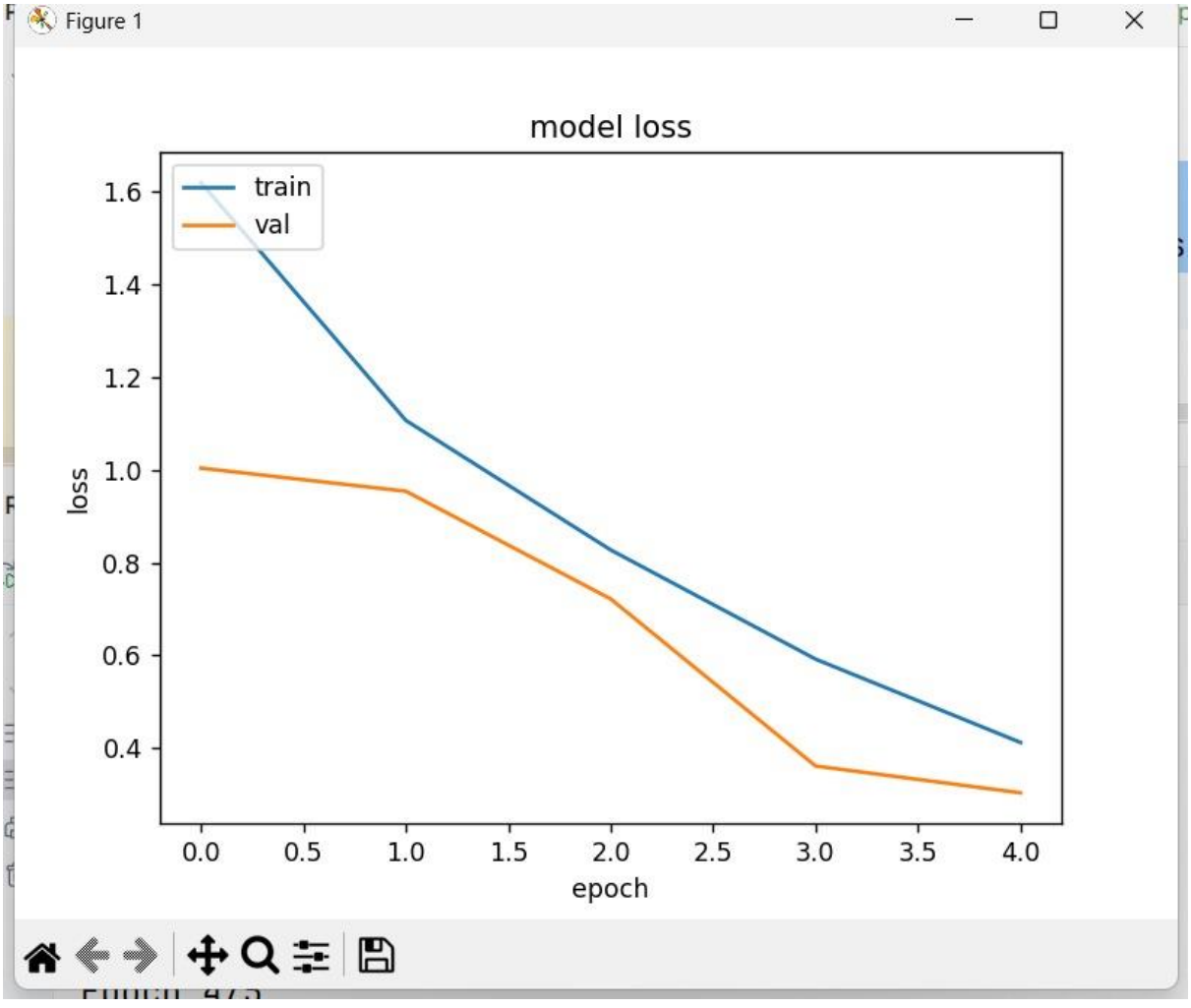
- Fixed image not displaying after upload → added proper HTML preview logic
- Prediction not updating on new upload → fixed Flask route refresh issue
- Improved UI layout for better user experience
- Optimized model loading to reduce prediction time
- Handled invalid inputs with clear error messages

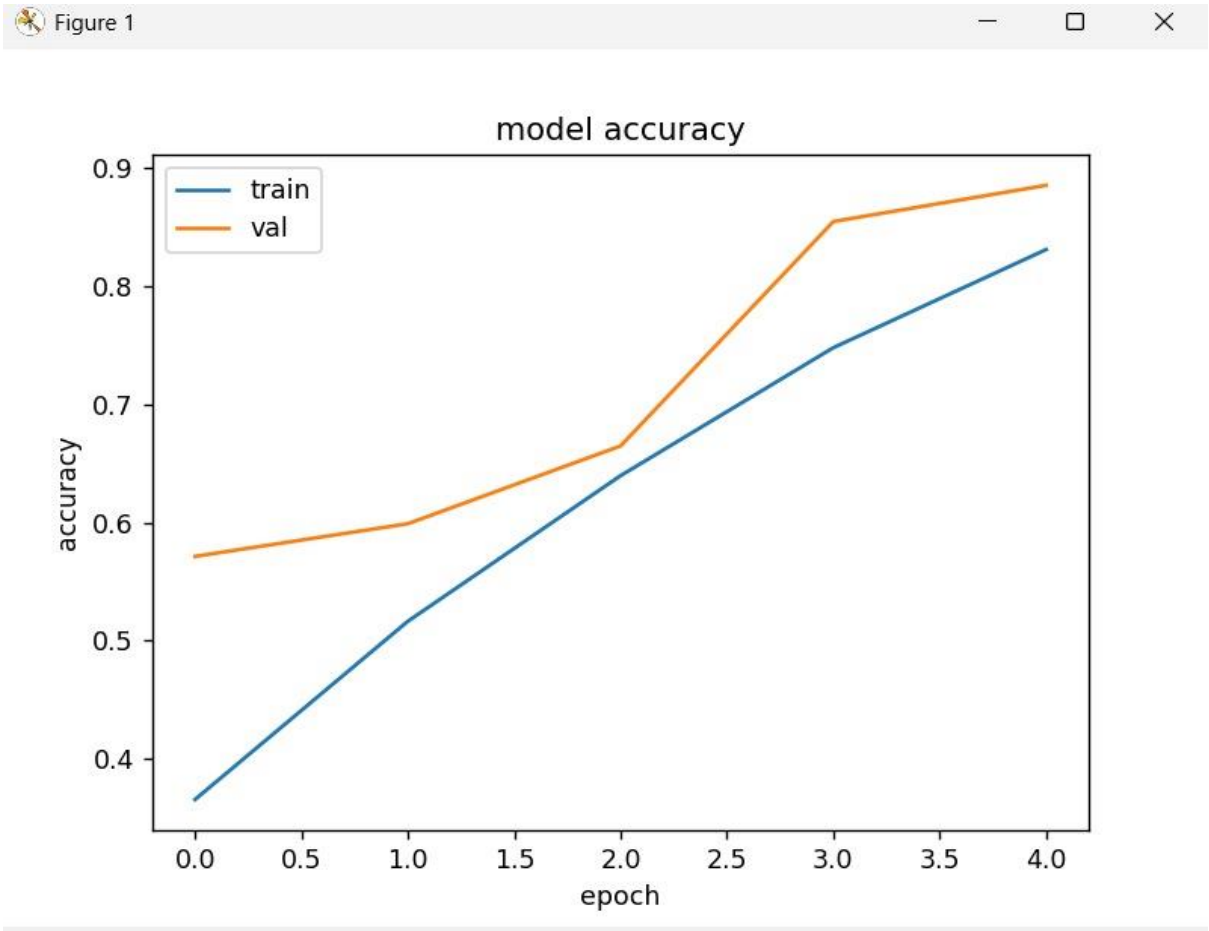
3. **Final Validation:**

- End-to-end testing of full workflow
- Verified prediction accuracy with test images
- Confirmed correct display of results on web interface
- Ensured responsiveness across devices (desktop/mobile)

4. Deployment:

- Platform: Deployed using Flask
- Hosted on local server or cloud platform
- All files organized: app.py, templates/, static/, model.h5, requirements.txt
- Final version tested post-deployment to ensure smooth functionality






```
996/996 ━━━━━━━━━━━━━━━━━━━━ 1426s 1s/step - accuracy: 0.6104 - loss: 0.8850 - val_accuracy: 0.6647 - val_loss: 0.7215
Epoch 4/5
996/996 ━━━━━━━━━━━━━━━━━━━━ 1402s 1s/step - accuracy: 0.7255 - loss: 0.6363 - val_accuracy: 0.8544 - val_loss: 0.3608
Epoch 5/5
996/996 ━━━━━━━━━━━━━━━━━━━━ 1855s 2s/step - accuracy: 0.8272 - loss: 0.4189 - val_accuracy: 0.8850 - val_loss: 0.3030
374/374 ━━━━━━━━━━━━━━━━━━━━ 109s 291ms/step
              precision    recall  f1-score   support

  eosinophil       0.81      0.82      0.82       739
  lymphocyte       0.93      0.99      0.96       753
    monocyte       0.99      0.99      0.99       766
  neutrophil       0.85      0.79      0.82       730

    accuracy                           0.90      2988
   macro avg       0.90      0.90      0.90      2988
weighted avg       0.90      0.90      0.90      2988

Accuracy of the Model: 89.9%
```