

# project

## HematoVision: Advanced Blood Cell Classification Using Transfer Learning

**Team ID:** LTVIP2025TMID47286

### **Team Members:**

- Shaik Sheema Firdose (Team Lead)
- Shaista Mulla
- Srinivas
- Baibare Mohammed Affan

### **Phase 1: Brainstorming and Ideation**

**Objective:** To conceptualize a practical and efficient solution for automated classification of blood cells using deep learning methodologies, with a focus on medical relevance and real-time deployment potential.

#### **Key Points:**

- **Problem Statement:** Manual identification of blood cells is time-consuming, prone to errors, and requires expert pathologists.
- **Proposed Solution:** Develop a deep learning model using transfer learning to accurately classify different types of blood cells from microscopic images.
- **Target Users:** Medical professionals, diagnostic labs, healthcare researchers, and AI in healthcare developers.
- **Expected Outcome:** A web-based application capable of accurately classifying blood cells using an uploaded image, improving diagnostic accuracy and efficiency.

### **Phase 2: Requirement Analysis**

**Objective:** To identify all technical and functional needs essential for the development and deployment of the blood cell classification system.

#### **Key Points:**

- **Technical Requirements:**
  - Python 3.9 or 3.10
  - TensorFlow >= 2.10
  - Flask (for web app integration)
  - NumPy, Pandas, Matplotlib, OpenCV, Seaborn
  - Anaconda / Google Colab / VS Code (for development)

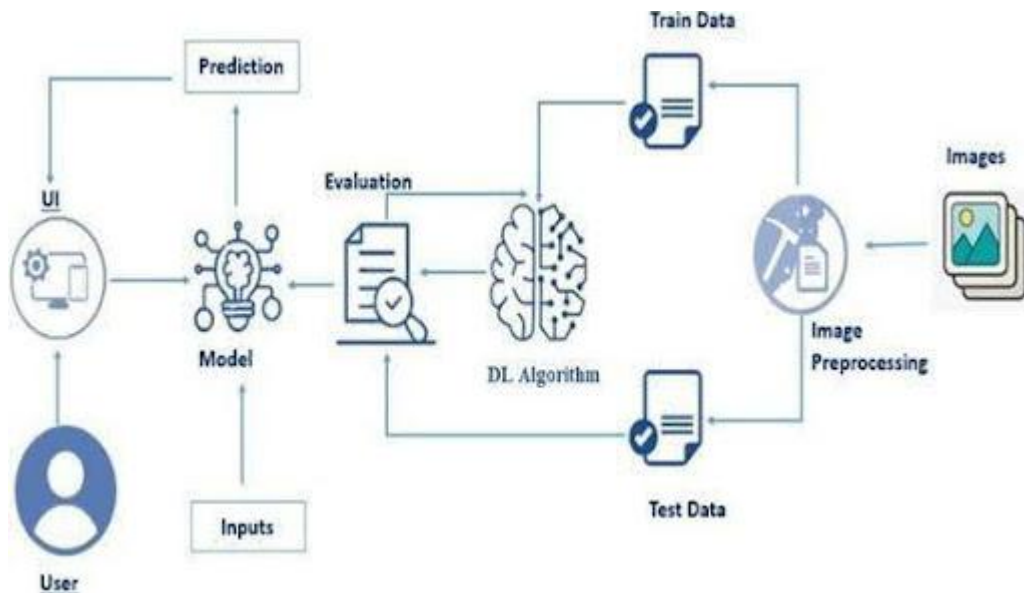
- **Functional Requirements:**
  - User uploads a blood cell image
  - The system preprocesses the image
  - The trained model predicts the cell type
  - Result is displayed with classification confidence
- **Constraints & Challenges:**
  - Model performance depends on dataset quality
  - File compatibility for .h5 in different environments
  - Limitations on Colab runtime and file storage
  - Hardware limitations on local machines for training

### Phase 3: Project Design

**Objective:** To establish a clear and scalable system architecture and define user flow for seamless application interaction.

#### Key Points:







- **System Architecture:**



- **User Flow:**
  - User accesses the application (locally or via browser)
  - Uploads image of blood cell
  - Backend model processes and classifies the image
  - Result is shown on the frontend

## Phase 4: Project Planning

**Objective:** To outline the project timeline, task distribution, and dependencies.

<i><b>Sprint</b></i>	<i><b>Task</b></i>	<i><b>Priority</b></i>	<i><b>Duration</b></i>	<i><b>Deadline</b></i>	<i><b>Assigned to</b></i>	<i><b>Dependencies</b></i>	<i><b>Expected outcome</b></i>
<i>Sprint 1</i>	<i>Environment Setup &amp; Package Installation</i>	 <i>High</i>	<i>3 hours</i>	<i>Day 1</i>	<i>Member 1</i>	<i>Anaconda, Python</i>	<i>Project environment ready</i>
<i>Sprint 1</i>	<i>Dataset Collection &amp; Preprocessing</i>	 <i>High</i>	<i>4 hours</i>	<i>Day 1</i>	<i>Member 2</i>	<i>Dataset access</i>	<i>Clean, prepared image dataset</i>
<i>Sprint 2</i>	<i>Model Building using Transfer Learning</i>	 <i>High</i>	<i>5 hours</i>	<i>Day 2</i>	<i>Member 3</i>	<i>Preprocessed data, TensorFlow</i>	<i>Trained classification model</i>
<i>Sprint 2</i>	<i>Flask Web App Integration</i>	 <i>Medium</i>	<i>3 hours</i>	<i>Day 2</i>	<i>Member 1 &amp; 4</i>	<i>Trained Model, Flask installed</i>	<i>Working web interface</i>
<i>Sprint 3</i>	<i>Testing &amp; Debugging</i>	 <i>Medium</i>	<i>2 hours</i>	<i>Day 2</i>	<i>Member 2 &amp; 3</i>	<i>Complete System</i>	<i>Bug-free and responsive system</i>
<i>Sprint 3</i>	<i>Final Presentation &amp; Deployment</i>	 <i>Low</i>	<i>1 hour</i>	<i>End of Day 2</i>	<i>Entire Team</i>	<i>Working application</i>	<i>Project deployed and demo-ready</i>

## Phase 5: Project Development

**Objective:** To build and test the system iteratively, addressing challenges and refining model performance.

### Key Points:

- **Technology Stack Used:**
  - Language: Python
  - Frameworks: TensorFlow, Flask
  - Tools: Google Colab, Anaconda, Jupyter Notebook, VS Code
- **Development Process:**
  1. Data preprocessing and augmentation
  2. Building and training model using transfer learning (e.g., MobileNetV2, ResNet50)

3. Model evaluation and tuning
4. Web application integration using Flask
5. Deployment and testing

- **Challenges and Fixes:**

- Issue: .h5 file not opening in VS Code
  - Fix: Used correct Python environment with TensorFlow installed (Python 3.9/3.10)
- Issue: Jupyter notebook not launching from virtual environment
  - Fix: Installed Jupyter inside the specific conda environment and added kernel
- Issue: TensorFlow install errors
  - Fix: Switched to supported Python version (3.10) and used clean virtual environment