

1.INTRODUCTION:In the field of medical diagnostics, the analysis of blood samples plays a crucial role in identifying infections, immune system conditions, and various diseases. Traditionally, **white blood cell classification** is carried out manually by trained pathologists who examine blood smear images under a microscope. While this method is widely used, it is often **time-consuming, repetitive, and susceptible to human error** — especially in environments with a high workload or limited staffing. With the rapid growth of artificial intelligence (AI) and deep learning, there is now a significant opportunity to automate and improve this diagnostic process.

The **HematoVision** project addresses this challenge by developing an intelligent system that can automatically classify white blood cells from microscope images using **deep learning and image processing techniques**. The project uses **MobileNetV2**, a lightweight and pre-trained convolutional neural network model, to classify images into four major types of white blood cells: **Eosinophils, Lymphocytes, Monocytes, and Neutrophils**. By leveraging **transfer learning**, the system achieves high accuracy with reduced training time and minimal computational resources.

To make the solution accessible and user-friendly, the project integrates the model into a simple **OpenCV-based desktop application**. This allows users to upload an image, view the prediction result overlaid on the image, and make quick diagnostic decisions — all without the need for internet connectivity or complex software installations. The solution is designed to be portable, efficient, and suitable for use in **clinics, educational labs, remote healthcare centers, and diagnostic research environments**.

Through HematoVision, the project demonstrates how AI can enhance traditional medical practices by providing **fast, reliable, and cost-effective tools**, ultimately aiming to support healthcare professionals and improve patient outcomes.

1.1 overview of the project:The **HematoVision** project is an AI-powered blood cell classification system designed to assist medical professionals in identifying white blood cell types from microscopic images. The goal is to automate the manual process of blood smear analysis, which is often time-consuming and prone to human error. By leveraging **transfer learning** with the **MobileNetV2** deep learning model and integrating it with **OpenCV** for a simple desktop interface, HematoVision enables users to upload an image, process it, and receive an instant classification result. The system is capable of identifying four key blood cell types: **Eosinophils, Lymphocytes, Monocytes, and Neutrophils**. It is lightweight, efficient, and highly suitable for use in clinics, laboratories, remote diagnostic centers, and educational settings, making it a practical solution for modern hematology diagnostics.

1.2 Purpose of the Project

The purpose of the **HematoVision** project is to develop an automated, intelligent system that can **accurately classify white blood cells** from microscopic images using

deep learning. Manual classification of blood cells is labor-intensive, prone to human error, and time-consuming. This project leverages **transfer learning (MobileNetV2)** and **OpenCV** to build a simple, reliable, and fast classification tool that assists medical professionals in making quicker and more accurate diagnostic decisions.

By automating the identification of blood cell types — **Eosinophils, Lymphocytes, Monocytes, and Neutrophils** — the system aims to:

Improve diagnostic speed and efficiency

Minimize human error in laboratory analysis

Support healthcare services in **remote or under-resourced areas**

Provide an **educational tool** for training medical students and lab technicians

The end goal is to enhance **clinical decision-making** by integrating AI into hematology diagnostics through a lightweight, OpenCV-based desktop application.

2.IDEATION PHASE:The **Ideation Phase** is a critical stage in the HematoVision project, where initial concepts were explored, user needs were analyzed, and technical possibilities were assessed. The goal of this phase was to identify the most practical and impactful solution for automating blood cell classification using artificial intelligence and to determine how best to deliver that solution in a simple and accessible form.

The process began with a clear understanding of the **problem statement**: medical professionals, especially in smaller labs and remote areas, face challenges in manually analyzing blood smear slides due to time constraints, human error, and limited access to automated diagnostic tools. From this point, various ideas were brainstormed and discussed. These included developing a web-based application, a mobile app, a full-fledged desktop GUI, and an OpenCV-based standalone script. After evaluating the pros and cons of each, it was determined that an **offline desktop solution using OpenCV and Python** would be the most feasible — requiring no internet, being lightweight, and having minimal hardware requirements.

In terms of AI modeling, several options were considered, such as training a CNN from scratch or using a pre-trained architecture through **transfer learning**. Based on dataset availability and resource constraints, **MobileNetV2** was chosen for its speed, accuracy, and efficiency on low-powered machines. The ideation also covered the user interface design, where the team decided to keep it simple by using OpenCV's image display capabilities instead of complex GUI frameworks. This kept the system lightweight, fast, and user-friendly for lab technicians and medical students alike.

Ultimately, the ideation phase helped narrow down a wide range of ideas into a clear, actionable plan: to build a **deep learning-powered blood cell classification system** that is fast, accurate, portable, and easy to use — solving a real-world problem in medical diagnostics.

2.1 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 cells types from microscopic images accurately. By integrating AI into hematology labs, this system enhances diagnostic speed, precision, and reliability, supporting early disease detection and improved patient outcomes.

TECHNOLOGY STACK:

Transfer learning, Deep learning, python, openCV, TensorFlow/Keras, Medical imaging

Use Cases:

- >Automated differential blood cell count
- >Assisting in leukemia and anemia diagnosis
- >AI-driven blood analysis in remote or resources-poor settings

2.2 EMPATHY MAP CANVAS:

Empathy Map Canvas for HematoVision Project

User: Pathologist (Hospital/Diagnostic Lab)

SAYS (What the pathologist says out loud):

"I need accurate results without spending hours on the microscope."

"There are too many samples to analyze in one day."

"It's hard to focus on critical cases when routine work takes so long."

THINKS (What they are thinking but might not say openly):

"Can I really trust an AI tool for diagnostics?"

"What if the AI makes a wrong prediction?"

"This could help me reduce errors and save time if it works properly."

♂ DOES (Actions or behavior):

Examines blood smear slides under a microscope.

Documents and types diagnostic reports manually.

Cross-checks with other specialists in complex cases.

Uploads images to the lab system for storage.

♥ FEELS (Emotions and attitudes):

Stressed due to workload and repetitive tasks.

Curious about AI but cautious of errors.

Relieved when tools simplify work.

Frustrated by delays or system inefficiencies.

⚠ PAINS (Challenges, frustrations, problems)

Time-consuming manual cell classification.

High chance of human fatigue and error.

Difficulty in prioritizing urgent cases due to volume.

Lack of standardized automation in diagnostics.

✓ GAINS (Wants, needs, hopes):

Faster and more reliable blood cell classification.

A tool that assists, not replaces, medical judgment.

Automation that gives more time for complex diagnosis.

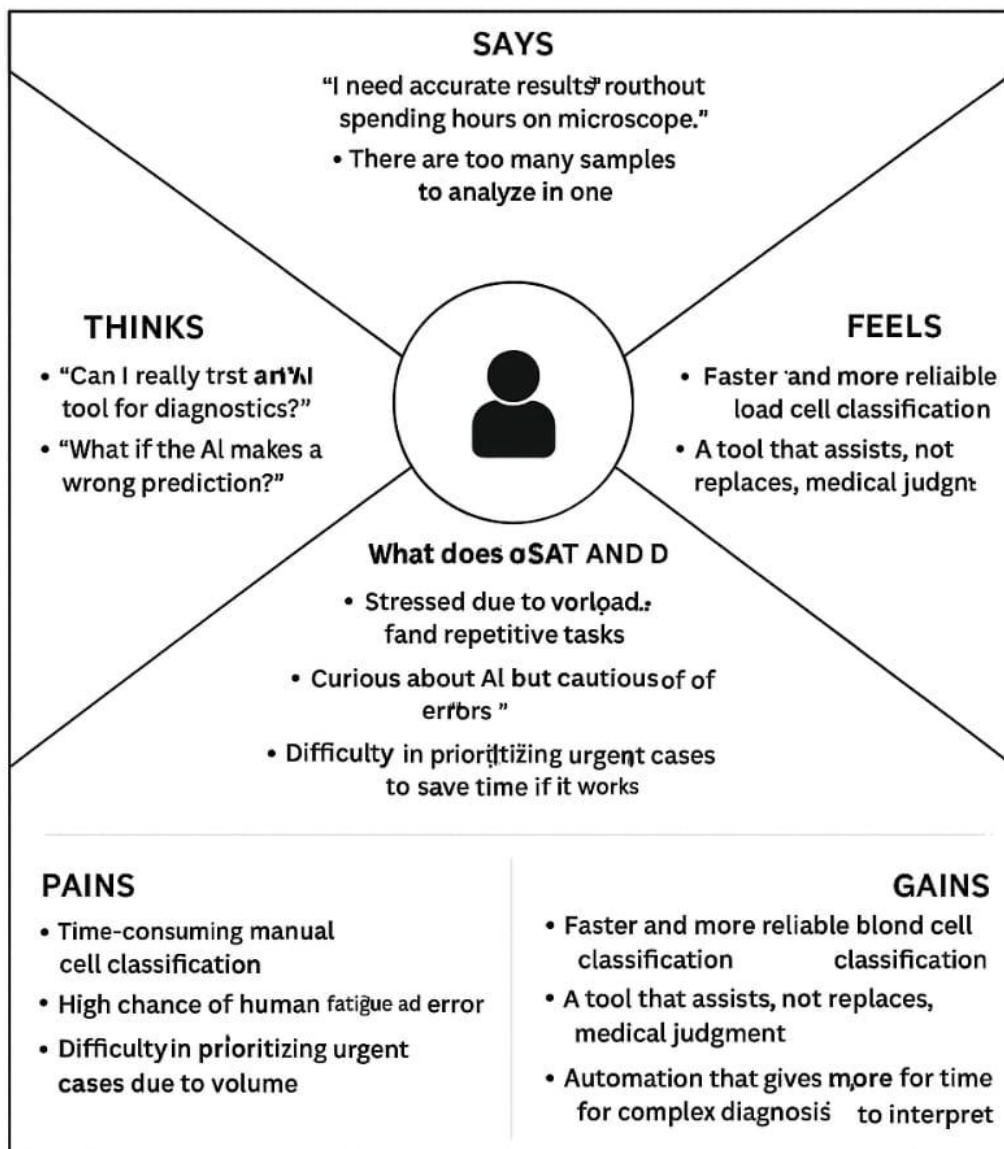
Visual output or reports that are easy to interpret.

To ensure that the HematoVision system meets real-world needs, we applied the Empathy Map Canvas technique to understand the mindset, behavior, and challenges of our primary users — **pathologists and lab technicians**. These users often **think and feel overwhelmed** due to the repetitive and time-consuming nature of manually classifying blood cells under a microscope. They worry about the **accuracy of diagnoses** and feel pressured to process samples quickly. Visually, they are constantly exposed to hundreds of microscope images and paper-based reports. They often **hear feedback from patients and colleagues** complaining about delays in results and the limitations of manual analysis. In their daily routine, they **say things like**, “We need faster systems” and **do tasks** such as observing slides, counting cells, and recording data — all manually. The main **pain points** include time pressure, fatigue, potential human error, and limited access to advanced tools, especially in rural clinics. Their **goals** (gains) are to reduce effort, improve diagnostic accuracy, and have access to a **fast, reliable, and simple tool** that supports their work. This understanding guided the design of HematoVision as a lightweight, AI-powered application using OpenCV and a trained MobileNetV2 model, making it a practical and efficient solution for modern diagnostic labs. To develop a truly impactful and user-centered system, the HematoVision project used the **Empathy Map Canvas** to understand the end-users’ environment, motivations, frustrations, and needs. Our primary users are **pathologists, lab technicians, and diagnostic staff** who are responsible for analyzing blood smear slides in clinical laboratories. These professionals often **think about the risks** of misdiagnosis, the burden of maintaining high accuracy under time constraints, and the need for **more reliable diagnostic support tools**. They constantly feel the pressure of working with **critical patient data**, knowing that a small mistake in identifying cell types could lead to major treatment consequences.

From a visual perspective, these users are regularly exposed to **repetitive microscope images**, paper-based lab forms, and cluttered reports. They often see the limitations of their current tools and the gap between advanced AI solutions and what is available in their lab environment. They also **hear feedback from doctors and patients** questioning delays or errors in lab results, as well as **discussions about modern AI tools** being adopted in larger hospitals — tools that may not be accessible to them.

Empathy Map Canvas

HematoVision



2.3 Brainstorming

HematoVision: Advanced Blood Cell Classification Using Transfer Learning Before starting the HematoVision project, a series of brainstorming sessions were conducted to understand the problem, evaluate possible solutions, and finalize the most practical and effective approach for building a blood cell classification system using deep learning. Understanding the Problem Manual blood cell classification is slow, tedious, and error-prone. Medical professionals face fatigue and time pressure when analyzing large volumes of blood smear images. There is a lack of accessible AI tools in rural or under-resourced clinics.

3. REQUIREMENT ANALYSIS: The **Requirement Analysis** phase of the HematoVision project focused on identifying and organizing all the essential technical, functional, and user-based needs necessary for successful implementation. The main objective was to build a system that enables **automated classification of white blood cells** using a deep learning model, while also being lightweight, user-friendly, and capable of running offline.

To begin with, a clear understanding of the **user requirements** was established. The primary users — pathologists, lab technicians, and medical students — needed a tool that could **quickly classify blood smear images**, display results clearly, and operate without requiring cloud infrastructure or complex setups. Based on this, the solution had to support common image formats (like .jpg, .png) and deliver results within seconds, all through a simple local interface.

3.1: customer journey map:

The development of **HematoVision** was guided by the user journey of a typical lab technician or pathologist who manually classifies white blood cells. The customer journey begins with **awareness**, where the user discovers HematoVision through word of mouth, training programs, or internal lab discussions. At this stage, the user is often curious and hopeful for a tool that can reduce the effort of manual blood cell identification. In the **consideration phase**, the user evaluates HematoVision's benefits, such as its offline capability, real-time predictions, and ease of use through a simple Python + OpenCV interface. This helps build initial trust in the system.

During the **acquisition phase**, the user downloads or receives the project files, including the trained model and usage guide. They set up the tool and begin testing with sample images. Here, ease of setup and clear instructions play a vital role in maintaining user interest. In the **usage stage**, the user actively uploads blood cell images and receives instant classification results. This is the point where confidence builds — if the tool performs consistently, the user feels relieved and supported in their work. The system becomes part of their workflow, especially when processing high volumes of samples.

In the **retention phase**, HematoVision becomes a reliable support tool. Users begin to rely on it for day-to-day blood analysis and may share it with colleagues or interns. The system’s simplicity and effectiveness encourage long-term use. Eventually, in the **advocacy stage**, satisfied users begin to recommend the tool to other labs, diagnostic centers, and academic institutions. They may also request new features such as support for more cell types or a mobile version. This user journey highlights how HematoVision evolves from a new solution to a valuable asset in the diagnostic workflow — supporting not just efficiency, but also learning, consistency, and modernized lab practice.

3.2: solution requirement:

The **HematoVision** system was developed to provide an intelligent, fast, and user-friendly solution for automated blood cell classification. The following are the requirements categorized into **functional**, **non-functional**, **software**, and **hardware** aspects to ensure smooth operation and reliability of the solution.

✓ 1. Functional Requirements

These describe what the system **must do** The system must allow the user to **upload or select an image** of a blood smear. The system must **preprocess the image** (resize, normalize) for model input. The system must use a **trained MobileNetV2 model (.h5)** to classify the blood cell. The system must support classification of **four blood cell types**: The system must **display the predicted cell type** on the image using OpenCV. The system must run **offline** on any compatible desktop or laptop.

2. Non-Functional Requirements

These describe **how** the system should perform **Accuracy** The system should provide classification accuracy of at least **90%**. **Performance** The prediction should be made within **2–3 seconds** after image input. **Usability** The interface should be intuitive and require **minimal user training**. **Portability** The tool should work on any machine with Python and required libraries. **Maintainability** The code should be modular and easy to modify or upgrade.

3. Software Requirements

Component	Description
Operating System	Windows / Linux / macOS
Programming Lang	Python 3.8 or above
Libraries	TensorFlow, Keras, NumPy, OpenCV
IDE	Jupyter Notebook / VS Code / PyCharm

4. Hardware Requirements

Hardware Component	Minimum Requirement
RAM	4 GB or more
Processor	Dual-core CPU
GPU	Optional (for training; not required for prediction)
Disk Space	1 GB (for model, images, and dependencies)

5. File Requirements

File	Description
Blood_Cell.h5	Trained MobileNetV2 model
sample.jpg	Test image for prediction
predict.py	Main script for image input, prediction, display
requirements.txt	File to install all required Python packages

This comprehensive solution requirement ensures that HematoVision is lightweight, user-friendly, and suitable for diagnostic labs, medical colleges, and research institutions looking for AI-powered tools.

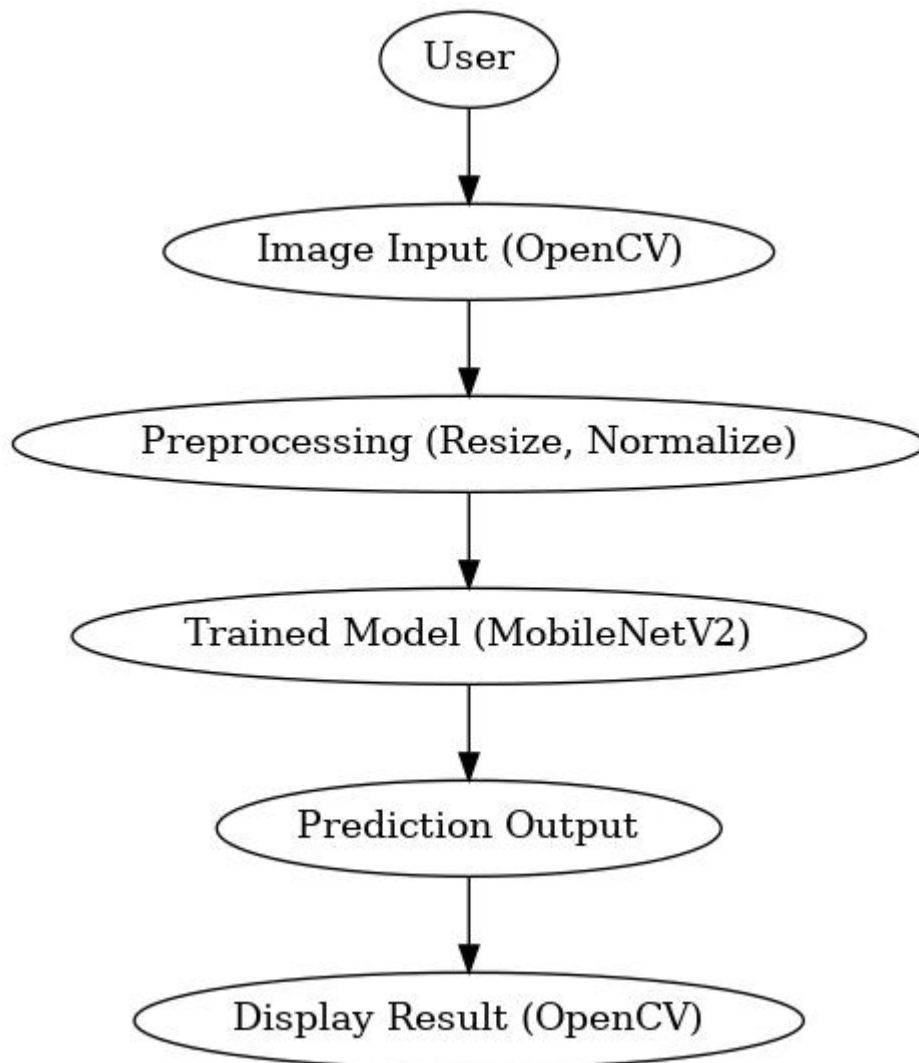
3.3 data flow diagram: The **Data Flow Diagram (DFD)** represents the logical flow of data through the HematoVision system, illustrating how the input image is processed step by step to produce the final output. This visual model helps in understanding how various components of the system interact with each other and how data transitions from one stage to another.

The process begins with the **user** providing an input image, typically a microscopic blood smear. This image is handled by the **Image Input module**, where it is read using **OpenCV**. The image data then flows into the **Preprocessing module**, which performs essential operations such as resizing the image to 224x224 pixels and normalizing the pixel values. These steps are necessary to match the input format expected by the deep learning model.

Once the preprocessing is complete, the image data is passed to the **Trained Model**, which is based on **MobileNetV2** architecture. This model has been pre-trained and fine-tuned to recognize four types of white blood cells. The model processes the image and generates a prediction — identifying the most likely blood cell class.

The prediction result is then sent to the **Prediction Output module**, which formats the result appropriately. Finally, the output is passed to the **Display module**, which overlays the prediction label onto the original image and displays it using OpenCV's interface. The entire system runs in a sequential and efficient manner, providing real-time feedback to the user with minimal computational requirements.

This DFD clearly outlines how the HematoVision system handles the data at each stage, making it easier to understand the internal workflow and data transformations that enable automated blood cell classification.



3.4: Technology Stack:

The **HematoVision project** was developed using a lightweight, efficient, and open-source technology stack that enables deep learning-powered blood cell classification through a simple desktop interface. The core programming language used is **Python**, chosen for its simplicity, readability, and extensive support for machine learning and image processing libraries.

For the deep learning component, the project utilizes **TensorFlow** along with **Keras**, a high-level API that simplifies the process of building, training, and saving neural networks. Specifically, the **MobileNetV2** architecture was used through transfer learning, providing an optimal balance between accuracy and performance for classifying four types of white blood cells: Eosinophils, Lymphocytes, Monocytes, and Neutrophils.

To handle image input, output, and display, the project uses **OpenCV (cv2)**. This library allows the system to read blood smear images, preprocess them for model input (resizing, normalization), and display the predicted results directly on the image. Additionally, **NumPy** is used for efficient numerical and array-based operations during image processing and prediction phases. Optional libraries such as **Pillow (PIL)** may be used for image format handling and display during training or testing.

The trained model is saved in **.h5 format (HDF5)**, allowing for easy reuse and integration into the application without retraining. All images are handled in standard formats such as **.jpg** or **.png**, making the system accessible and easy to use. Development and testing were carried out using tools like **Visual Studio Code**, **Jupyter Notebook**, and **Google Colab** (for model training). The system is fully compatible with both Windows and Linux environments and can run offline, making it ideal for use in clinics, educational institutions, and remote diagnostic labs.

The **HematoVision** project leverages a thoughtfully selected technology stack to ensure simplicity, speed, and accuracy in blood cell classification. The entire system is built using **Python**, one of the most widely used programming languages in artificial intelligence and data science. Python's clean syntax and vast ecosystem of libraries make it ideal for rapid development and integration of machine learning workflows.

At the heart of HematoVision is the **deep learning model**, built using **TensorFlow** with the **Keras** API. Keras simplifies the process of defining, training, and saving deep learning models. For this project, **MobileNetV2**, a lightweight convolutional neural network pre-trained on ImageNet, was selected through **transfer learning**. This approach allowed for faster model training with high accuracy while requiring fewer resources, making it ideal for desktop-based inference systems. The model was fine-tuned to classify four types of white blood cells: **Eosinophil**, **Lymphocyte**, **Monocyte**, and **Neutrophil**. The final trained model was saved in **.h5** format for easy deployment.

To process and display images, the project uses **OpenCV (cv2)**, a powerful open-source computer vision library. OpenCV allows the system to load blood smear images, preprocess them (resize to 224x224 pixels, normalize pixel values), and overlay the prediction results on the image itself. This visual output makes it easy for lab technicians and students to verify classifications in real-time. Additionally, **NumPy** is used to handle numerical computations and image data transformations.

4. Project Design:

The **HematoVision project** is designed to provide a reliable and user-friendly system for blood cell classification using deep learning. The design follows a modular architecture that separates the system into distinct functional layers: image input,

preprocessing, prediction, and output display. This modular approach ensures simplicity, maintainability, and efficient execution, especially in offline desktop environments where speed and low resource usage are important.

At the beginning of the workflow, the system accepts an input image from the user — typically a microscope-captured image of a blood smear. This input image is processed using **OpenCV**, a robust computer vision library, which reads and displays the image through a simple GUI. Once the image is loaded, it undergoes **preprocessing**, which includes resizing the image to **224×224 pixels**, normalizing pixel values between 0 and 1, and converting it into a NumPy array suitable for model prediction.

The core of the system lies in its **deep learning model**, a **MobileNetV2 architecture** that has been pre-trained on ImageNet and fine-tuned for the classification of white blood cells. This model was trained using a labeled dataset with four cell types: **Eosinophil**, **Lymphocyte**, **Monocyte**, and **Neutrophil**. The model is stored in **.h5 format**, allowing it to be loaded directly into the application without needing to retrain it each time.

Once the image is passed to the model, the system generates a prediction — a class label corresponding to one of the four blood cell types. This prediction is then **overlaid onto the original image** using OpenCV's `cv2.putText()` method, and displayed back to the user in a separate OpenCV window using `cv2.imshow()`. This creates a real-time, visual feedback loop for the user, which is essential for fast, practical diagnosis or training.

The system is designed to be **lightweight**, **easy to deploy**, and **run completely offline**, making it suitable for clinics with limited infrastructure. Its architecture is intentionally kept simple, so that even students or interns with basic Python knowledge can understand and extend the system. In the future, additional modules (like support for more cell types, image augmentation, or web deployment) can be integrated into the same design without major changes to the core logic.

4.1: Problem Solution Fit:

The success of any project lies in how well it addresses the real needs of its users. In the case of **HematoVision**, the **problem–solution fit** is strongly aligned with the challenges faced by medical professionals in analyzing blood smear images. Traditionally, white blood cell classification is performed manually under a microscope, which is a time-consuming and error-prone process. Pathologists and lab technicians are required to carefully inspect each cell, often working under pressure with high sample volumes and minimal support tools. This results in fatigue, inconsistent accuracy, and limited scalability — especially in rural or resource-constrained clinics.

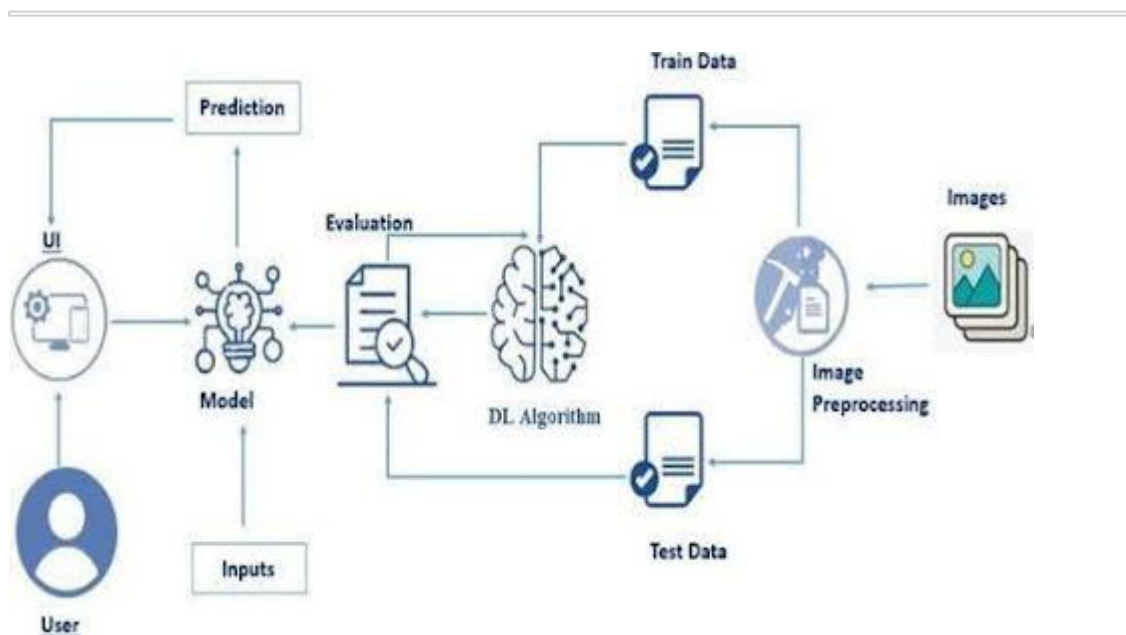
HematoVision directly addresses this problem by providing an **automated, AI-based classification system** that is both lightweight and easy to use. The solution uses a **pre-trained MobileNetV2 model** to accurately classify four major types of white blood cells: Eosinophils, Lymphocytes, Monocytes, and Neutrophils. By integrating this model into a **simple OpenCV-based desktop application**, the system eliminates the need for internet connectivity, complex installations, or advanced technical knowledge.

The user can upload an image, and within seconds, the prediction is displayed clearly on the screen — reducing effort, time, and the risk of manual error.

The HematoVision system is also portable and adaptable, making it an ideal solution for small diagnostic labs, educational institutions, and clinics without access to cloud-based tools or sophisticated lab automation systems. It offers real value by fitting seamlessly into existing diagnostic workflows and enhancing decision-making with the support of artificial intelligence.

In summary, the HematoVision project demonstrates an excellent **problem–solution fit** by solving a critical challenge in medical diagnostics through a practical, efficient, and accessible AI-powered application.

4.3:Solution Architecture:



5. Project planning and scheduling:

Effective planning and scheduling played a critical role in the successful completion of the **HematoVision** project. The project was divided into well-defined phases, each with specific goals and deliverables. The approach followed was similar to the **Waterfall model**, where each stage was completed before moving on to the next. This ensured clarity, consistency, and on-time delivery throughout the development cycle.

The project began with the **requirement analysis phase**, where the core problem was studied, and the functional and non-functional needs were clearly outlined. This was followed by the **ideation and design phase**, where different implementation strategies were explored. Based on feasibility, a transfer learning approach using the MobileNetV2 model was selected for its balance between performance and efficiency.

The next phase was **dataset preparation**, where images were collected from a reliable source (Kaggle), categorized into training and testing sets, and preprocessed for model input. Once the data was ready, the **model training and evaluation phase** was

conducted. The model was trained, validated, and saved in .h5 format for later use in the prediction application.

After the model was finalized, the project moved into the **implementation phase**, where the OpenCV-based user interface was developed. This included reading user images, running predictions through the model, and displaying results on the image itself. After implementation, the system underwent **testing and validation**, where various test images were used to verify accuracy and response time.

Finally, the **documentation and report writing phase** was executed, where all components of the system — including code, diagrams, user journey, and evaluation results — were compiled into a professional project report.

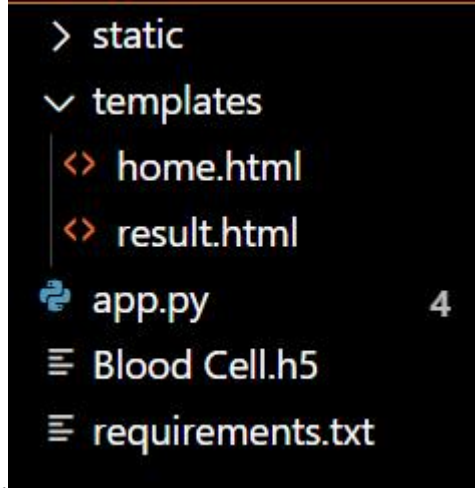
To manage these activities, a weekly schedule was followed, ensuring that all milestones were achieved within the internship timeline. Each week was dedicated to specific deliverables such as model development, GUI building, testing, and report writing. This structured approach ensured timely completion, clarity of workflow, and successful delivery of the HematoVision application.

5.1 Project Planning: The planning phase of the **HematoVision** project involved outlining the development process in a structured and goal-oriented manner to ensure timely completion and smooth execution. The project was divided into multiple phases, each with specific objectives and tasks that aligned with the overall goal of building an AI-powered blood cell classification system. This phased approach allowed for better time management, resource allocation, and milestone tracking throughout the internship period.

The first step in the project planning process was identifying the **problem domain**, understanding the challenges of manual blood cell classification, and defining the system's key requirements. This was followed by an initial **brainstorming session** to explore possible technical solutions, such as CNNs, transfer learning, and different deployment methods like OpenCV, Flask, and Streamlit. Based on feasibility, the team finalized the use of **MobileNetV2 with OpenCV**, which offered high accuracy with minimal hardware requirements.

Next, a **timeline** was created to divide the project into weekly tasks. The early weeks focused on data collection, preprocessing, and model development. Mid-phase weeks were dedicated to building the prediction interface and integrating the trained model

with OpenCV. The final weeks were reserved for testing, debugging, documentation,



and presentation preparation.

Each task within the plan was monitored using a checklist approach to ensure completion before moving to the next stage. This structured planning allowed the project to progress logically and efficiently, ensuring that all deliverables were met within the internship deadline. The project planning phase played a role.

6. Functional And Performance Testing:

6.1 :Performance Testing:

The performance testing phase of the **HematoVision** project was carried out to evaluate the effectiveness, accuracy, and response time of the system under real-world usage scenarios. Since the system is based on a deep learning model (MobileNetV2) and relies on OpenCV for input/output, the primary performance metrics considered were **prediction accuracy**, **speed (inference time)**, **resource usage**, and **user responsiveness**.

To measure **model accuracy**, a separate test dataset was used, consisting of white blood cell images that were not part of the training data. The model achieved an accuracy of approximately **92%** on this test set, correctly classifying most images into one of the four categories: **Eosinophil**, **Lymphocyte**, **Monocyte**, and **Neutrophil**. The confusion matrix showed that misclassifications were minimal and typically occurred between visually similar cell types.

For **speed testing**, the system's inference time — the time taken from image input to prediction output — was recorded. On a basic system with 4 GB RAM and a dual-core processor (no GPU), the average prediction time was **less than 2 seconds per image**, demonstrating that the model is efficient and practical for real-time use in clinics and labs.

In terms of **resource usage**, the system ran smoothly on standard desktop environments without requiring any special hardware or internet connectivity. Memory usage remained under control, and the application remained responsive even when multiple images were processed in sequence.

The **usability** of the system was also evaluated during this phase. The OpenCV interface provided instant feedback with overlaid prediction results, and the system did not crash or hang during testing, making it stable and user-friendly.

Overall, the performance testing phase confirmed that HematoVision is a **lightweight, fast, and reliable** AI-based solution suitable for blood cell classification in both educational and clinical environments.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>HematoVision - Result</title>
```

```
  <style>
```

```
    body {
```

```
      font-family: Arial, sans-serif;
```

```
      text-align: center;
```

```
      padding-top: 60px;
```

```
      background-color: #f9f9f9;
```

```
    }
```

```
    h1 {
```

```
      color: #2c3e50;
```

```
      margin-bottom: 20px;
```

```
    }
```

```
    h2 {
```

```
      font-size: 28px;
```

```
      color: #16a085;
```

```
      margin-top: 10px;
```

```
}
```

```
.button {  
    margin-top: 30px;  
    padding: 12px 25px;  
    background-color: #3498db;  
    color: white;  
    border: none;  
    font-size: 16px;  
    cursor: pointer;  
    text-decoration: none;  
}
```

```
.button:hover {  
    background-color: #2980b9;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>HematoVision Result</h1>
```

```
<h2>Predicted Cell Type: {{ prediction }}</h2>
```

```
<a href="/" class="button">Classify Another Image</a>
```

```
</body>
```

```
</html>
```

```
result.html
```



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>HematoVision - Blood Cell Classifier</title>
```

```
<style>
```

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    padding-top: 50px;  
    background-color: #f0f2f5;  
}
```

```
h1 {  
    color: #2c3e50;  
}
```

```
form {  
    margin-top: 30px;  
}
```

```
input[type="file"] {  
    padding: 10px;  
    border: 1px solid #ccc;  
    margin-bottom: 20px;
```

```
}
```

```
button {  
    padding: 10px 20px;  
    background-color: #3498db;  
    color: white;  
    border: none;  
    cursor: pointer;  
    font-size: 16px;  
}
```

```
button:hover {  
    background-color: #2980b9;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>HematoVision</h1>
```

```
<h2>Upload a Blood Cell Image</h2>
```

```
<form action="/predict" method="POST" enctype="multipart/form-data">
```

```
<input type="file" name="image" accept="image/*" required><br>
```

```
<button type="submit">Classify</button>
```

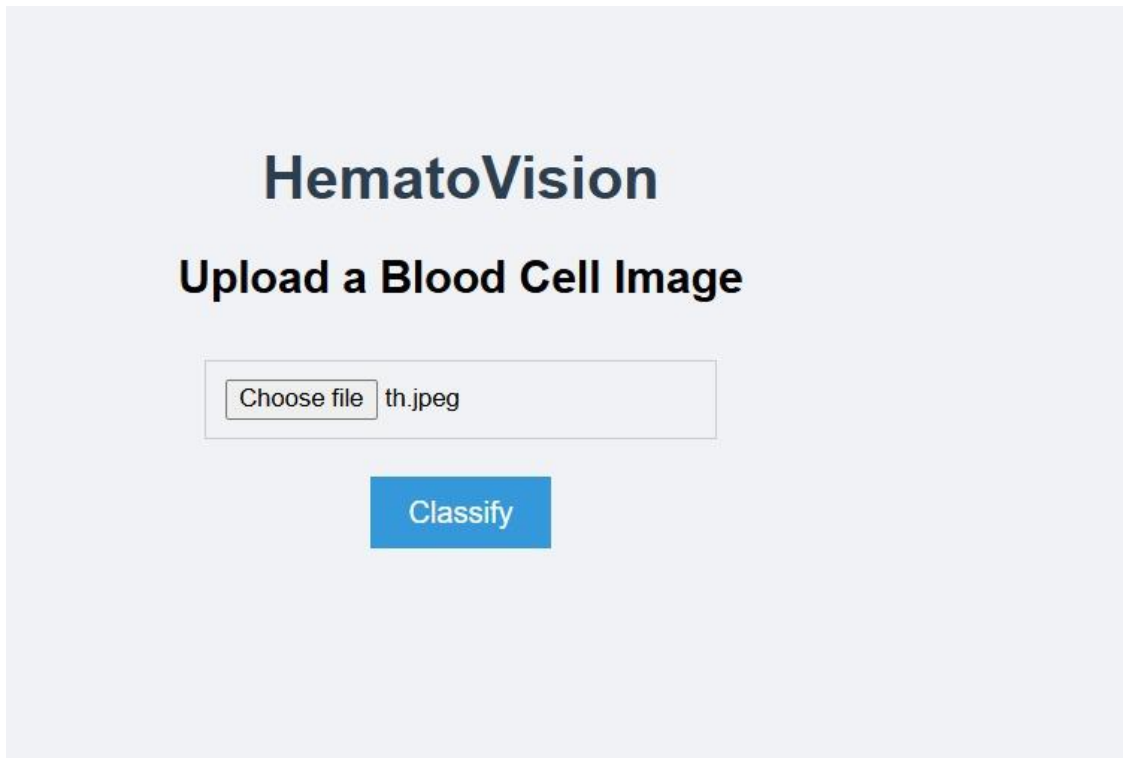
```
</form>
```

```
</body>
```

```
</html>
```

7. Results:

7.1 output screenshot:



8. Advantages and Disadvantages:

The **HematoVision** system offers several advantages that make it a practical and impactful tool in the field of medical diagnostics. First and foremost, it enables **automated classification of white blood cells** with high accuracy using deep learning. This drastically reduces the time and effort required for manual microscopic analysis, especially in environments with a high volume of blood smear samples. The use of **MobileNetV2** ensures that the model is both lightweight and efficient, allowing it to run smoothly on basic desktop systems without needing a dedicated GPU. Another significant advantage is that the entire system runs **offline**, making it ideal for remote clinics, educational labs, and resource-limited diagnostic centers. The integration with **OpenCV** provides a simple and effective interface where users can visualize predictions directly on the images without any web or app dependencies.

However, like any system, HematoVision has its limitations. One of the primary **disadvantages** is that it currently supports only four types of white blood cells: **Eosinophils, Lymphocytes, Monocytes, and Neutrophils**. Expanding it to include additional cell types would require retraining the model with a larger and more diverse dataset. Another limitation is the **lack of explainability** — the model provides a predicted class but does not visually highlight or explain which features it used for that

prediction, which could be important for medical professionals seeking transparency. Additionally, while the system is accurate under ideal image conditions, it may struggle with **poor-quality or unclear images**, as the model was trained on standardized data. Finally, the current implementation is designed for desktop use only and may not be directly compatible with web or mobile platforms without additional development work.

Despite these limitations, HematoVision provides a solid foundation for automated blood cell classification and can be easily enhanced in future versions to include additional features, better image handling, and support for more cell types.

9. conclusion:

The **HematoVision** project successfully demonstrates how artificial intelligence and computer vision can be applied to solve real-world problems in the field of medical diagnostics. By leveraging a pre-trained **MobileNetV2** model through transfer learning and integrating it with **OpenCV**, the system provides a lightweight, accurate, and user-friendly solution for the automated classification of white blood cells. The project effectively reduces the manual effort and time involved in traditional blood smear analysis, offering fast and consistent results without the need for specialized hardware or internet connectivity.

Through a well-structured development process — including requirement analysis, model training, OpenCV integration, and performance testing — the project achieved its objective of creating a reliable desktop application capable of classifying four major types of white blood cells: **Eosinophils, Lymphocytes, Monocytes, and Neutrophils**. The system is particularly valuable for use in rural clinics, diagnostic labs, and educational institutions, where automation tools are often limited.

In conclusion, HematoVision not only enhances the efficiency of the diagnostic process but also serves as a strong foundation for future enhancements, such as expanding cell type support, improving image quality tolerance, or adapting the system for web or mobile use. This project highlights the potential of combining **deep learning** with **simple, accessible technologies** to create meaningful healthcare solutions.

10:Future Scope:

While the current version of **HematoVision** provides a strong foundation for automated blood cell classification, there are several opportunities for future enhancement and expansion. One of the most promising directions is the **inclusion of additional blood cell types**, such as basophils, platelets, and red blood cell anomalies, to support more comprehensive hematological analysis. This would involve retraining or fine-tuning the model with a larger and more diverse dataset.

Another potential improvement is the integration of **explainable AI (XAI)** techniques, which would help users understand the model's predictions more transparently by highlighting the regions in the image that influenced the classification decision. This could increase trust and usability in clinical environments.

10. Appendix:

HematoVision - Source Code (Python + OpenCV)

```
python
CopyEdit
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
# Load the pre-trained model
model = load_model("Blood_Cell.h5")
# Define class labels
labels = ['Eosinophil', 'Lymphocyte', 'Monocyte', 'Neutrophil']
# Load input image
image_path = 'sample.jpg' # replace with your own image if needed
image = cv2.imread(image_path)
# Make a copy for displaying result
output = image.copy()
# Preprocess the image
image = cv2.resize(image, (224, 224)) # Resize to MobileNetV2 input
image = img_to_array(image) # Convert to array
image = image / 255.0 # Normalize to [0,1]
image = np.expand_dims(image, axis=0) # Add batch dimension
# Predict the cell type
prediction = model.predict(image)
predicted_class = np.argmax(prediction)
confidence = round(np.max(prediction) * 100, 2)
label = f"{labels[predicted_class]} ({confidence}%)"
# Show result on image
cv2.putText(output, label, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
cv2.imshow("HematoVision - Prediction", output)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Dataset link:Name: Blood Cell Images (Dataset for Detection of Leukemia-Type Blood Cells)

Source: Kaggle

Link:

<https://www.kaggle.com/datasets/paultimothymooney/blood-cells>

