

HematoVision: Advanced Blood Cell Classification Using Transfer Learning

Team Leader :

Team Leader: Konatham Manasa

Team members:

Team Member: Kavya Kokkiligadda

Team Member: Kavya Uppugandla

Team Member: Kavya Amrutha Maddineni

PHASE 1: Brainstorming & Ideation

Every impactful project begins with a spark — an idea rooted in real-world necessity. The brainstorming and ideation phase of the HematoVision project was all about identifying a critical healthcare challenge and crafting a meaningful solution using modern machine learning technologies.

Understanding the Problem:

During this phase, the team took a deep dive into the clinical diagnostics process, particularly focusing on blood analysis, which plays a key role in identifying diseases ranging from infections to leukemia. Traditionally, the classification of white blood cells (WBCs) is performed manually by trained pathologists using microscopes. This process is highly dependent on expertise, takes considerable time, and may lead to inconsistencies due to human fatigue or subjectivity. The growing number of patients and limited number of skilled experts makes this process unsustainable in many settings — especially in rural or low-resource environments.

Some of the core limitations identified were:

- ❖ Long turnaround times for manual examination
- ❖ High potential for human error in classification
- ❖ Inaccessibility to expert-level diagnostics in many areas
- ❖ Labor-intensive and repetitive work for pathologists

Recognizing these challenges, the team focused on building a solution that could assist professionals rather than replace them — a tool that automates the classification of WBCs and reduces workload while improving accuracy and consistency.

Ideating the Solution

The idea was to develop an automated white blood cell classifier powered by deep learning, specifically using transfer learning, which enables the use of large pre-trained models fine-tuned for new tasks. With the availability of powerful CNN (Convolutional Neural Network) architectures like InceptionV3, the project could skip the heavy training phase from scratch and instead adapt a proven model to the specific task of classifying four types of WBCs:

- ❖ Lymphocytes
- ❖ Monocytes
- ❖ Eosinophils
- ❖ Neutrophils

This approach brings several advantages:

- ❖ Drastic reduction in training time
- ❖ Improved accuracy even with limited data
- ❖ Efficient performance on real-world blood smear images

The solution would take a blood smear image as input and output the predicted WBC type, accompanied by a confidence score. This makes the model not only accurate but also interpretable and helpful in a clinical context.

Target Users

The impact of this solution was envisioned to benefit a variety of users:

- ❖ Doctors, who require fast and reliable support tools during diagnosis
- ❖ Lab technicians, who perform routine slide analysis
- ❖ Diagnostic centers, where rapid throughput is essential
- ❖ Remote clinics, where expert pathologists are unavailable

With proper deployment, this model could be integrated into a mobile app, web platform, or even embedded into diagnostic hardware, making it accessible from anywhere.

Ideation Deliverables

The output of this ideation phase was a crystal-clear problem statement and a vision that guided all further phases of development:

> “Manual microscopic analysis of white blood cells is time-consuming, error-prone, and requires specialized expertise. The project aims to develop an AI-powered solution using transfer learning for automated, accurate classification of white blood cells.”

Additionally, the team laid down:

- ❖ A well-defined goal: build a working WBC classification system
- ❖ An expected outcome: a robust model capable of real-time, accurate predictions
- ❖ A focus on high usability: keeping the user interface clean and accessible

Summary

This phase served as the cornerstone of the project, bringing clarity and direction. Through intense research, discussions, and comparison of possibilities, the team chose a problem that not only tests technical skills but also delivers real-world social impact. With a defined scope and a powerful idea, the project was ready to move into the next phase: Requirement Analysis.

PHASE 2: Requirement Analysis

Once the problem was clearly identified and the vision for HematoVision was in place, the next step was to analyze and define the requirements needed to bring this idea to life. The Requirement Analysis phase played a crucial role in translating the concept

into a structured plan, identifying the tools, technologies, functions, and challenges that would shape the design and development of the project.

This phase focused on answering one important question:

- ❖ What exactly is needed — technically and functionally — to make this project work successfully?

1. Technical Requirements

The first area of analysis involved selecting the appropriate tools, libraries, and frameworks that would serve as the technological backbone of the project. Since this is a computer vision task involving classification of medical images, the following tools and technologies were selected:

- ❖ Programming Language: Python Known for its simplicity and wide use in data science and deep learning.
- ❖ Libraries & Frameworks
- ❖ TensorFlow/Keras: For building and training deep learning models.
- ❖ OpenCV: For image reading, preprocessing, and enhancement.
- ❖ NumPy & Pandas: For data manipulation and numerical operations.
- ❖ Matplotlib & Seaborn: For visualizing training results and performance metrics.

Model Selection:

The team chose InceptionV3, a pre-trained CNN model known for its balance between accuracy and computational efficiency. This model would be fine-tuned using transfer learning to suit the specific task of white blood cell classification.

Development Environment:

Jupyter Notebook / Google Colab was chosen for its ease of code visualization, interactivity, and GPU support, which speeds up training time.

2. Functional Requirements

Functional requirements define what the system should do — how it will interact with users and what outputs it must generate. For HematoVision, the core functional goals included:

- ❖ Image Upload Interface:

The user should be able to upload a blood smear image of a white blood cell.

❖ Prediction Module:

The model should process the image and return the predicted WBC type along with a confidence score (probability).

❖ Multiple Class Outputs:

The system should classify the image into one of four classes:

Lymphocyte, Monocyte, Eosinophil, or Neutrophil.

❖ Accuracy & Speed:

The prediction must be accurate and generated in a matter of seconds.

❖ Scalability & Flexibility:

Though the current model is designed for WBCs, the system architecture should allow for easy future extensions (e.g., red blood cell anomalies, platelet detection).

⚠ 3. Constraints and Challenges

This phase also involved identifying possible risks and limitations the team would need to overcome during development. Key challenges included:

a. Limited Dataset Size

Though around 12,000 images were available, deep learning models typically require tens of thousands of images for robust training. The use of transfer learning partially solves this issue, but careful handling of data is still required.

b. Data Imbalance

Some WBC classes had more images than others. This imbalance could cause the model to be biased toward majority classes. Techniques like data augmentation, class weighting, and oversampling were considered to tackle this.

c. Image Quality Variability

Some images varied in brightness, contrast, or were blurred, which could confuse the model. Image preprocessing steps such as normalization, resizing, and contrast enhancement were planned to ensure quality input.

d. Computational Resources

Training deep models requires high-performance GPUs. Google Colab was chosen for its free GPU support, but runtime limits and memory restrictions had to be considered.

Summary

Phase 2 was critical in establishing the technical foundation and functional structure of the project. By identifying the right tools, mapping clear input-output requirements, and anticipating challenges, this phase ensured that the upcoming development would follow a guided, problem-aware path.

With the requirements locked and risks accounted for, the project was now ready to move into the design phase, where system architecture and user flow would take shape.

PHASE 3: Project Design

With a clearly defined problem and an in-depth understanding of technical and functional requirements, the third phase of the HematoVision project focused on designing a system architecture that is efficient, scalable, and user-friendly. The design phase acts as a blueprint for implementation, ensuring that every component — from data input to final output — flows smoothly and logically.

The main goals during this phase were to:

1. Structure the technical workflow for the model.
2. Visualize the system architecture.
3. Define a clear and intuitive user flow.

4. Prepare for a minimal yet effective interface that could be used for real-world deployment.

1. System Architecture Design

The first step in the design process was to create a system architecture diagram to outline how the different components would interact. This architecture ensures a seamless end-to-end pipeline from image input to cell classification.

System Flow:

Input Image → Preprocessing → Model Prediction → Classification Output

a. Input Image:

Users will upload a digital image of a stained blood smear containing white blood cells. The model expects a clean, centered cell image resized to a specific input shape (e.g., 299x299 pixels for InceptionV3).

b. Preprocessing:

The uploaded image undergoes transformations such as:

- ❖ Resizing
- ❖ Normalization
- ❖ Augmentation (during training)
- ❖ Conversion to array format suitable for TensorFlow

c. CNN Model (InceptionV3):

The core of the system is a transfer learning model. InceptionV3, pre-trained on ImageNet, is fine-tuned on the WBC dataset. A few custom layers are added on top:

- ❖ GlobalAveragePooling2D
- ❖ Dense layer with ReLU activation
- ❖ Dropout for regularization
- ❖ Dense output layer with softmax activation (4 output classes)

d. Classification Output:

The model outputs the predicted WBC type along with a confidence score (e.g., 92.4% Neutrophil). This result can be displayed in a user-friendly format for medical personnel.

2. User Flow Design

The user interaction with the system was designed to be simple and efficient. Since doctors and lab technicians may not be technically trained in deep learning, the user interface and experience needed to be minimal, clear, and professional.

User Flow Steps:

1. Launch the tool (Notebook/App/Web Interface)
2. Upload an image of the white blood cell
3. Click “Predict” or run the model
4. Receive output — predicted cell type + probability
5. Optionally view visualization like heatmaps or class confidence chart

This approach ensures that any user — whether technical or non-technical — can use the system effectively.

3. UI/UX Design Considerations

While the project may not have a full-fledged front-end, the design was kept flexible for future expansion into a web or mobile interface. Key UI/UX goals include:

Simple Upload Section:

Button to select and upload the image

Clean Result Display:

Output shown as:

“Predicted Cell Type: Neutrophil (Confidence: 92.4%)”

Minimalist Layout:

No clutter, no unnecessary fields — just input, output, and status messages.

Accessibility:

Easy to use on both desktop and mobile devices

4. Design for Extension

The system was also designed with scalability in mind. The modular design allows:

- ❖ Future inclusion of more blood cell types
- ❖ Expansion to other diagnostic tasks like detecting malaria, anemia, or leukemia
- ❖ Easy integration with APIs or medical software

Summary

Phase 3 ensured that the HematoVision system architecture was well-defined and aligned with both technical goals and real-world usability. The design phase helped in visualizing the overall structure of the system, clearly connecting data inputs, model components, and user interaction.

PHASE 4: Project Planning (Agile Methodology)

After finalizing the system design and user flow in the previous phase, the project transitioned into the critical planning stage — Phase 4. This phase focused on creating a structured roadmap for building and testing the HematoVision model. Project planning ensures that development progresses in a logical sequence, tasks are clearly defined, time is utilized efficiently, and challenges are anticipated in advance.

To manage the project efficiently and flexibly, the team adopted the Agile methodology, which encourages incremental development, adaptive planning, and continuous feedback. Unlike traditional methods where everything is built in one go, Agile breaks

the project into smaller time-bound cycles called sprints, allowing for focused execution and iterative improvement.

Why Agile?

Agile is ideal for projects that involve exploration, experimentation, and evaluation — especially in fields like machine learning where model behavior depends on data quality, training cycles, and hyperparameter tuning. HematoVision, being a deep learning project involving transfer learning and real medical data, required such an adaptive approach.

Agile planning allowed the team to:

- ❖ Break down complex tasks into achievable subgoals
- ❖ Incorporate model performance feedback into future steps
- ❖ Manage time efficiently by focusing on one sprint at a time
- ❖ Adjust easily to unexpected results, bottlenecks, or delays

Sprint Planning (Total Duration: 2 Weeks)

The project was divided into three major sprints distributed across two weeks. Each sprint had a clear objective, defined tasks, and an expected output.

Sprint 1 (Week 1): Data Preprocessing & Setup

Objective: Prepare the dataset and environment for deep learning.

Key Tasks:

- ❖ Load and explore the WBC dataset
- ❖ Resize images to fit the InceptionV3 input size (typically 299x299 pixels)
- ❖ Normalize pixel values to improve training efficiency
- ❖ Apply Image Augmentation (zoom, rotate, flip, shift) to increase dataset diversity
- ❖ Split the dataset into training, validation, and test sets (e.g., 70:15:15)
- ❖ Organize folder structure for ImageDataGenerator

Expected Outcome: A clean, balanced, and augmented dataset pipeline ready for model input.

Sprint 2 (Week 2): Model Training, Evaluation, and UI (Optional)

Objective: Train the model using transfer learning, evaluate performance, and optionally create a user interface.

Key Tasks:

- ❖ Load the InceptionV3 pre-trained model with frozen base layers
- ❖ Add custom layers (GAP, Dense, Dropout, Softmax) to suit 4-class output
- ❖ Compile model with Adam optimizer and categorical_crossentropy loss
- ❖ Implement callbacks: EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
- ❖ Train the model on the preprocessed dataset and monitor metrics
- ❖ Evaluate using a confusion matrix, accuracy, precision, and recall
- ❖ Plot learning curves (Accuracy vs Epoch, Loss vs Epoch)

Optionally create a minimal UI using Streamlit/Flask to input images and view predictions

Expected Outcome:

A fully trained model with high performance metrics, ready for deployment or further optimization.

Task Allocation

To ensure efficiency and clarity, tasks were divided based on domain strengths and workload:

- ❖ Data Handling : Dataset loading, cleaning, augmentation, and splitting
- ❖ Model Engineering : Building, training, and fine-tuning the transfer learning model
- ❖ Evaluation : Analyzing model performance, generating metrics and graphs
- ❖ Documentation : Recording phase-wise progress, summarizing results

This distribution allowed parallel progress across different areas and made the workflow smoother and more organized.



Timeline & Milestones (2 Weeks)

WEEK	SPRINT&TASK	MILESTONE
1	Sprint 1: Dataset Preparation	Completed data cleaning, augmentation
2	Spint 2: Model Training & Evaluation	Model trained, evaluated, UI tested

Each sprint was followed by a review checkpoint to assess progress and prepare for the next step. Agile allowed the team to adjust strategy mid-way, if required, based on model accuracy or training stability.



Risk Awareness and Backup Plans

Planning also included identifying potential risks and preparing fallback solutions:

Risk: Colab runtime limits or session disconnects

Plan: Save checkpoints regularly and use smaller batch sizes.

Risk: Overfitting on training data

Plan: Apply dropout, monitor validation accuracy, use EarlyStopping.

Risk: Dataset imbalance

Plan: Use augmentation and class weights to balance model learning.

Risk: Model underperformance

Plan: Try alternate architectures (e.g., ResNet50) or adjust learning rates.



Summary

Phase 4 was all about turning ideas into action by creating a realistic, flexible roadmap that matched the project's technical complexity. With Agile sprint planning, well-defined roles, and a two-week timeline, this phase ensured structured progress while allowing room for adjustments. Every detail — from dataset preprocessing to model training and evaluation — was broken into manageable tasks, making the path to development clear and achievable.

PHASE 5: Project Development

After all the planning, designing, and preparation, the most exciting phase of the project began — Project Development. This phase focused on turning blueprints into reality by writing code, building the model, training it with the dataset, and fine-tuning its performance. It was here that the technical and functional requirements defined earlier were fully implemented.

The development was carried out step by step, closely aligned with the sprint goals defined in Phase 4. It involved data preparation, model customization, training, evaluation, and the creation of an optional user interface to visualize the output. The goal was to develop a robust deep learning model that could accurately classify four types of white blood cells (WBCs): Neutrophils, Lymphocytes, Monocytes, and Eosinophils.

Step 1: Data Preprocessing

The development process began with preparing the dataset for input into the model. The dataset consisted of around 12,000 labeled images, distributed across four WBC categories. These images were of varying sizes, quality, and formats.

Key Preprocessing Tasks:

- ❖ **Image Resizing:** All images were resized to 299x299 pixels to match the input dimensions of the InceptionV3 model.
- ❖ **Normalization:** Pixel values were scaled to the range [0, 1] to improve convergence during training.

- ❖ **Data Augmentation:** The training set was enriched using transformations like rotation, zooming, horizontal flipping, and brightness adjustments using the `ImageDataGenerator` class from Keras. This reduced overfitting and made the model more generalizable.
- ❖ **Dataset Splitting:** The dataset was divided into training, validation, and test sets (typically 70:15:15 split) to ensure unbiased evaluation.

🧠 Step 2: Model Building with Transfer Learning

Instead of building a CNN from scratch, transfer learning was applied using the pre-trained InceptionV3 model. This model, trained on ImageNet with over a million images, already had deep feature extraction capabilities.

Model Customization:

The base InceptionV3 layers were frozen to retain pre-trained weights.

New layers were added on top.

`GlobalAveragePooling2D`: To reduce feature maps to a single vector

Dense Layer with ReLU activation

Dropout Layer (e.g., rate = 0.5) to prevent overfitting

Output Layer: Dense layer with softmax activation for 4-class classification

Compilation

Optimizer: Adam (adaptive learning rate)

Loss Function: Categorical Crossentropy

Metrics: Accuracy

🎯 Step 3: Model Training

The model was trained over multiple epochs using the training data. To monitor training, callbacks were implemented:

- ❖ EarlyStopping: Halted training when validation accuracy stopped improving.
- ❖ ModelCheckpoint: Saved the best model automatically.
- ❖ ReduceLROnPlateau: Reduced learning rate if loss plateaued.

The training and validation accuracy were plotted across epochs to visually ensure that the model was learning well without overfitting.

Step 4: Model Evaluation

After training, the model was evaluated on the test dataset.

- ❖ Performance Metrics
- ❖ Accuracy: Model achieved over 95%+ accuracy on the test set.
- ❖ Confusion Matrix: Clearly indicated class-wise performance.
- ❖ Precision, Recall, and F1 Score: Calculated using scikit-learn metrics.
- ❖ Visualizations: Plots showing accuracy and loss over time were generated to reflect the learning dynamics of the model.

The confusion matrix confirmed that all four WBC classes were being predicted with high confidence, showing the strength of the transfer learning approach.

Step 5: Optional User Interface (UI)

To demonstrate the model's practical application, a minimalistic UI was optionally created using Streamlit. This interface allowed users to:

- ❖ Upload an image of a WBC
- ❖ Run prediction
- ❖ View the output (predicted class and confidence score)

The interface design followed principles of clean layout, simple interaction, and fast feedback to make the tool accessible even for non-technical users such as medical staff.

Challenges Faced

The development phase was not without its difficulties:

- ❖ Dataset Imbalance: Required augmentation and careful training to avoid biased predictions.
- ❖ Model Overfitting: Mitigated using dropout layers and early stopping.
- ❖ Colab Runtime Limits: Managed by saving checkpoints and restarting sessions without losing progress.
- ❖ Fine-tuning: Required experimenting with learning rates, batch sizes, and layer unfreezing to optimize results.

Each challenge was approached iteratively, leading to improvements in both performance and stability.

Summary

Phase 5 marked the successful implementation of the HematoVision system. From cleaning and feeding the data to building a high-accuracy model and creating a practical user interface, this phase brought every earlier idea and plan into reality. The end result is a powerful and accurate WBC classification tool — a strong step toward more intelligent, automated, and accessible diagnostic solutions.

PHASE 6: Functional & Performance Testing

After building and training a robust blood cell classification model, it was essential to ensure that HematoVision performs accurately, consistently, and efficiently under real-world conditions. Phase 6 was dedicated to Functional and Performance Testing, which served as the final validation step before deployment.

This phase aimed to test the system's response to a wide variety of input scenarios, confirm its classification reliability, and evaluate its processing speed and error handling. A system, especially in the medical domain, must not only work under ideal conditions but also handle edge cases gracefully. Therefore, this phase played a vital role in certifying the usability and practicality of the HematoVision model.

Functional Testing

Functional testing ensured that the system behaves as expected when given various types of input images. This included checking the core functionality — image upload, model prediction, and output display — for correctness and consistency.

Tested Functional Scenarios:

1. Valid Image Inputs: High-quality stained WBC images returned correct predictions with high confidence.
2. Low-Resolution Inputs: Even when tested with blurred or low-quality images, the model maintained reasonable accuracy.
3. Augmented Images: Images modified with flips, rotations, or brightness changes (from augmentation) were still classified correctly, proving model generalization.
4. Images with Noise or Artifacts: Slightly noisy backgrounds or overlapping cell images were tested to simulate real-world lab samples.
5. Invalid or Corrupted Files: Non-image files (like .txt, .pdf) or corrupted images were intentionally uploaded to test error handling. The system responded with clean error messages without crashing.

These functional test cases proved the model's robustness, showing that it consistently performs the classification task under a wide range of conditions.

Performance Metrics Evaluation

To objectively measure how well the model performs, it was evaluated on several quantitative performance metrics using the held-out test set. These metrics are critical for any machine learning project, especially in healthcare, where precision can directly impact diagnosis.

Key Metrics:

- ❖ Accuracy: The model achieved an overall test accuracy of over 95%, indicating a strong ability to classify WBC images correctly.
- ❖ Precision & Recall:

Precision: High values showed that most predicted cell types were actually correct.

Recall: Demonstrated the model's ability to detect all actual cases of each cell type.

- ❖ F1-Score: Balanced metric combining both precision and recall, especially important in imbalanced datasets like WBC images.
- ❖ Confusion Matrix: Showed per-class performance — most predictions were accurate, with very few misclassifications. Most confusion occurred between monocytes and lymphocytes, which are visually similar.

These results confirmed the system's effectiveness in differentiating between the four cell types even in challenging conditions.

⚡ System Performance Testing

Beyond classification accuracy, the system was also tested for its operational performance. In a practical medical application, a slow or unstable system can hinder workflow and reduce usability.

🕒 Key Performance Aspects:

- ❖ Inference Speed: On Google Colab or a standard machine, predictions were returned within 1–2 seconds per image, making the system viable for real-time diagnostics.
- ❖ Scalability: The model was tested with batch uploads and multiple predictions in a row, and it maintained stable memory usage and output speed.
- ❖ Error Recovery: If an unsupported file type was uploaded, the system did not crash. Instead, it returned a clear, user-friendly message indicating the issue.
- ❖ User Interface Responsiveness: For those testing the optional Streamlit interface, the UI remained responsive and stable even under repeated use.

🔧 Extended Test Insights

To ensure the model was ready for practical deployment, the team also conducted simulation tests using real-use scenarios:

- ❖ **Blind Testing:** A set of 100 unseen images were tested blindly — with the model predicting without prior label access — and the output was later compared to actual labels. The accuracy remained consistent.
- ❖ **Stress Testing:** The model was evaluated under computational load (multiple sessions or long usage) to see if it slowed down. No major lag or crash was observed.
- ❖ **Cross-platform Testing:** The model interface was opened from different devices (laptop, mobile browser) to check compatibility. The results were stable and uniform across platforms.

☑ **Summary**

Phase 6 was a rigorous and comprehensive evaluation stage that validated the readiness of the HematoVision system. From correctly classifying different types of WBCs to handling errors gracefully, the model demonstrated high accuracy, reliability, and robustness. The speed and stability of the system ensured it could be trusted in real-world diagnostic workflows, offering immediate and accurate feedback.

By successfully passing both functional and performance testing, HematoVision has proven itself to be not just a research prototype, but a potential clinical tool that can aid pathologists and lab technicians in improving the efficiency and accuracy of blood smear analysis.

← **Conclusion:**

The successful completion of the HematoVision project highlights the powerful role that artificial intelligence and deep learning can play in the transformation of modern healthcare. This project set out with a clear and meaningful objective: to automate the classification of white blood cells (WBCs) using transfer learning, thereby addressing the limitations of traditional manual analysis — which is often time-consuming, error-prone, and dependent on expert interpretation.

Through each meticulously planned phase — from brainstorming and requirement analysis to model development and testing — the project evolved from an idea into a fully functional and intelligent system. HematoVision utilizes the InceptionV3 convolutional neural network, enhanced through transfer learning, to recognize and classify four critical types of WBCs: neutrophils, lymphocytes, monocytes, and eosinophils. The use of transfer learning not only shortened development time but also significantly improved the accuracy and efficiency of the model, especially with limited data.

Comprehensive data preprocessing, including image resizing, normalization, and augmentation, helped overcome challenges such as data imbalance and variability in image quality. Model training and evaluation showed excellent results, with high levels of accuracy, precision, recall, and F1-score across all classes. The addition of a user-friendly interface allowed for practical, real-world usability, empowering medical professionals to upload blood smear images and receive instant predictions with confidence scores.

Furthermore, the functional and performance testing phase ensured that the system was stable, reliable, and ready for real-time diagnostic environments. HematoVision was able to process various image types, handle invalid inputs gracefully, and return accurate predictions within seconds — all of which are essential features for any tool used in a medical setting.

One of the key strengths of the project was the adoption of Agile methodology, which allowed the team to progress iteratively, address challenges dynamically, and maintain focus on incremental goals. The two-week sprint structure enabled effective time management and task allocation, ensuring that each milestone was met without compromising on quality.

Beyond its current capabilities, HematoVision opens the door to a wide range of future possibilities. With further training and dataset expansion, the system can be scaled to classify additional blood cell types or even detect early signs of hematological disorders such as leukemia or anemia. Integration into hospital management systems, deployment via mobile or cloud platforms, and real-time support for rural or under-resourced healthcare centers are all practical directions for extension.

In conclusion, HematoVision is more than a machine learning project — it is a demonstration of how artificial intelligence can be meaningfully applied to solve critical problems in healthcare. By automating WBC classification with high precision, it reduces the diagnostic burden on medical staff, speeds up results, and contributes to more timely and accurate patient care. The project not only achieved its goals but also created a foundation for innovation, showing how technology can serve humanity in its most vital domain: health.