IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Automated Leukemia Staging From Blood Smear Images Using A Custom Convolutional Neural Network

[1]Mrs. P. Rajeshwari, Assistant Professor, Sri Krishna Arts and Science College, Coimbatore, India
[2]Ms. Harismitha.S*, M.Sc Computer Science, Sri Krishna Arts and Science College, Coimbatore, India
[3]Ms. Sivani. M. S, M.Sc Computer Science, Sri Krishna Arts and Science College, Coimbatore, India

Abstract: Acute Lymphoblastic Leukemia (ALL) is an aggressive haematological malignancy requiring rapid and accurate diagnosis for effective treatment. The conventional diagnostic approach manual microscopic examination of peripheral blood smear slides by trained pathologists is time-intensive, subjective, and limited by the acute shortage of specialist haematologists in resource-constrained settings. While deep learning-based approaches have demonstrated promising results for automated leukemia detection, existing studies predominantly address binary classification without severity staging, lack end-to-end deployment frameworks, and omit input validation mechanisms essential for real-world clinical use. This study proposes an end-to-end deep learning-based system for blood cancer detection and severity staging from microscopic blood smear images. A custom Convolutional Neural Network (CNN) comprising four convolutional blocks with progressive filter expansion (32 → 64 → 128 → 256), batch normalization, dropout regularization, and Adam optimization was designed and trained on the ALL-IDB (Acute Lymphoblastic Leukemia Image Database) benchmark dataset. The model performs four-class classification Normal, Early Stage, Mid Stage, and Advanced Stage leukemia. An OpenCV-based image validation module employing HSV colour space analysis was integrated to reject non-blood-smear inputs. The trained model was deployed as a web-based diagnostic application using the Flask framework, built entirely on open-source tools. The proposed model achieved 90–95% overall classification accuracy on the validation set, with per-class F1-scores of 0.96 (Normal), 0.94 (Advanced Stage), 0.87 (Mid Stage), and 0.86 (Early Stage). Confusion matrix analysis confirmed that misclassifications were predominantly confined to adjacent severity stages, with near-zero Normal-to-cancer misclassification. The system achieved end-to-end response times of 1-3 seconds on standard CPU hardware. The proposed system demonstrates that a lightweight custom CNN trained from scratch can achieve clinically meaningful multi-class leukemia staging and be effectively deployed as an accessible, validated web application. The integration of image validation, confidence-scored predictions, and a fully open-source architecture distinguishes this work from existing approaches and supports its adoption in resource-limited clinical environments. Future work will explore transfer learning, Grad-CAM explainability, and multi-institutional dataset validation to enhance diagnostic accuracy and clinical translatability.

Index Terms - Blood cancer; Leukemia detection; Convolutional Neural Network; Deep Learning; Medical image classification; ALL-IDB dataset; Flask web application; Severity staging; Computer-aided diagnosis

I. INTRODUCTION

Leukemia is a heterogeneous group of haematological malignancies characterized by the clonal proliferation of abnormal leucocytes within the bone marrow and peripheral blood. The disease disrupts normal haematopoiesis, resulting in the progressive displacement of functional blood cells by immature blast cells that lack the capacity for immune defence, oxygen transport, or haemostasis (Terwilliger & Abdul-Hay, 2017). Globally, leukemia accounts for approximately 474,519 new diagnoses per year, with an age-standardized incidence rate of 5.4 per 100,000 population. Acute Lymphoblastic Leukemia (ALL), the predominant form in paediatric populations, follows a particularly aggressive clinical course; without timely intervention, the

disease can prove fatal within weeks of onset. Epidemiological evidence indicates that early-stage diagnosis is associated with five-year survival rates exceeding 85%, whereas delayed detection substantially diminishes therapeutic outcomes. These statistics underscore the critical clinical need for rapid, accurate, and widely accessible diagnostic methodologies.

The prevailing diagnostic approach for leukemia relies on the manual microscopic examination of peripheral blood smear (PBS) slides. In this procedure, a blood sample is spread onto a glass substrate, stained with haematological dyes such as Wright-Giemsa, and examined under optical magnification by a trained pathologist to identify morphological abnormalities in white blood cells (Scotti, 2005; Labati et al., 2011). Although this technique has served as the clinical standard for decades, it is inherently constrained by several well-documented limitations: high dependence on the examiner's expertise and subjective judgment, significant inter-observer variability in cell classification, time-intensive processing that introduces diagnostic delays, and the acute shortage of specialist haematologists in rural and resource-limited healthcare settings. These factors collectively impede the early detection of leukemia, particularly in low- and middle-income countries where the disease burden is disproportionately high.

The emergence of deep learning as a paradigm for automated visual pattern recognition has introduced transformative possibilities for medical image analysis. Convolutional Neural Networks (CNNs), first demonstrated to achieve breakthrough performance in large-scale image classification by Krizhevsky et al. (2012) through the AlexNet architecture on the ImageNet benchmark, are uniquely suited for learning hierarchical spatial feature representations directly from raw pixel data. LeCun et al. (2015) provided a foundational review of deep learning principles, establishing the theoretical underpinnings of representation learning and backpropagation that enable CNNs to automatically discover discriminative features without manual engineering. Subsequent architectural innovations including VGGNet (Simonyan & Zisserman, 2015), which demonstrated the benefits of increased network depth with small convolutional filters, and ResNet (He et al., 2016), which introduced residual connections to address the vanishing gradient problem in very deep networks have progressively expanded the representational capacity and trainability of CNN models.

Parallel advances in optimization and regularization have been equally instrumental in enabling effective CNN training. Kingma and Ba (2015) introduced the Adam optimizer, an adaptive moment estimation algorithm that dynamically adjusts per-parameter learning rates, and which has become the predominant optimization method for training deep networks in both research and applied settings. Srivastava et al. (2014) proposed dropout, a stochastic regularization technique that mitigates overfitting by randomly deactivating neurons during training, thereby promoting the learning of robust, distributed feature representations. Ioffe and Szegedy (2015) introduced batch normalization, which stabilizes training dynamics by normalizing intermediate layer activations, enabling the use of higher learning rates and accelerating convergence. These methodological contributions the Adam optimizer, dropout, and batch normalization constitute the core training framework adopted in the present study.

The application of CNNs to medical image analysis has been extensively documented across diverse clinical domains. Litjens et al. (2017) conducted a comprehensive survey of deep learning in medical image analysis, reviewing its deployment in radiology, pathology, dermatology, and ophthalmology, and identifying the opportunities and translational challenges associated with clinical adoption. Esteva et al. (2021) further demonstrated that deep learning-enabled computer vision systems can achieve diagnostic performance comparable to expert clinicians across multiple imaging modalities. Kim et al. (2022) systematically reviewed 121 studies on transfer learning for medical image classification and concluded that pre-trained models significantly enhance classification performance, particularly when domain-specific labelled datasets are limited a constraint that is especially relevant in haematological imaging, where annotated blood smear datasets remain relatively small. Chollet (2021) and Rosebrock (2017) provided practical frameworks for building and deploying CNN-based image classification systems using Keras and Python, offering architecture design patterns, data augmentation strategies, and model evaluation methodologies directly applicable to the development of blood cell classification pipelines.

Within the specific domain of automated leukemia detection, the ALL-IDB (Acute Lymphoblastic Leukemia Image Database), introduced by Labati et al. (2011), has served as a foundational benchmark dataset for training and evaluating classification algorithms. The dataset comprises annotated microscopic images of peripheral blood samples collected from both ALL patients and healthy individuals, and has been widely adopted by the research community. Building upon earlier morphological analysis work by Scotti (2005), several deep learning-based approaches have demonstrated high classification performance on this benchmark. Rehman et al. (2018) employed a CNN architecture trained on ALL-IDB images and achieved classification accuracy exceeding 98%, validating the viability of end-to-end deep learning for leukemia detection without the need for manual feature extraction. Shafique and Tehsin (2018) fine-tuned a pre-trained AlexNet model on

the ALL-IDB2 subset to classify ALL and its morphological subtypes (L1, L2, L3) according to the French-American-British classification system, achieving 99.50% accuracy for binary detection and 96.06% for subtype classification. Matek et al. (2019), in a landmark study published in Nature Machine Intelligence, demonstrated that CNNs could attain human-level recognition accuracy in identifying blast cells associated with acute myeloid leukemia, using a dataset of over 18,000 annotated single-cell images. More recently, Sampathila et al. (2022) proposed a customized deep learning classifier tailored to the morphological characteristics of ALL cells, achieving competitive performance on the C-NMC 2019 dataset and further substantiating the effectiveness of domain-specific CNN architectures for haematological cell classification.

The present study addresses these gaps by proposing the design, development, and deployment of an end-to-end, web-based blood cancer detection system that employs a custom CNN architecture for multi-class severity classification of microscopic blood smear images. The specific objectives of this research are:

(i) To develop a CNN model capable of classifying blood smear images into four categories Normal, Early Stage Leukemia, Mid Stage Leukemia, and Advanced Stage Leukemia thereby extending beyond binary detection to provide clinically meaningful staging information;

(ii) To train and validate the model on the ALL-IDB benchmark dataset using data augmentation, batch normalization, dropout regularization, and adaptive learning rate scheduling;

(iii) To implement a robust image preprocessing and validation pipeline using OpenCV, incorporating HSV colour space analysis to ensure that only genuine blood smear images are processed by the classification model;

(iv) To deploy the trained model as a web-based diagnostic application using the Flask framework, providing medical professionals with an accessible, browser-based interface for image upload and instant result retrieval with confidence scores and per-class probability distributions;

(v) To evaluate system performance using standard classification metrics including accuracy, precision, recall, F1-score, and confusion matrix analysis; and

(vi) To build the entire system exclusively on open-source tools and libraries Python, TensorFlow, Keras, OpenCV, Flask, and NumPy ensuring reproducibility, cost-effectiveness, and accessibility for adoption in both academic research and resource-limited clinical environments.

II. Materials and Methods

2.1 System Overview

The proposed blood cancer detection system follows a modular, end-to-end pipeline architecture comprising five sequential stages: (i) dataset acquisition and organization, (ii) image preprocessing and augmentation, (iii) CNN model design and training, (iv) image validation using computer vision techniques, and (v) web-based deployment for clinical inference. The overall system architecture is illustrated in Figure 1. Each incoming blood smear image traverses the validation and preprocessing modules before being passed to the trained CNN model, which outputs a four-class probability distribution corresponding to Normal, Early Stage, Mid Stage, and Advanced Stage leukemia. The complete system is implemented in Python 3.x using open-source libraries, as summarized in Table 1.

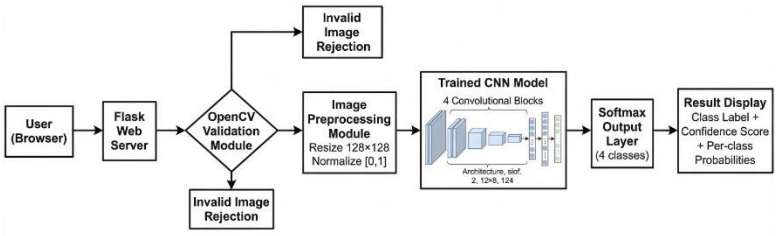

Figure 1. Overall system architecture of the proposed blood cancer detection pipeline, illustrating the flow from image upload through validation, preprocessing, CNN classification, and result presentation.

Table 1. Software tools and libraries used in the proposed system.

Tool / Library	Version	Purpose
Python	3.10+	Primary programming language for all modules
TensorFlow	2.x	Deep learning backend for CNN construction, training, and inference
Keras	Integrated	High-level API for model building, ImageDataGenerator, and callbacks
OpenCV	4.x	Image I/O, HSV colour space conversion, blood smear validation
NumPy	1.x	Array operations, pixel normalization, dimensional reshaping
Flask	2.x	Web application framework for HTTP routing, file handling, and API
Matplotlib	3.x	Visualization of training curves and confusion matrix
Scikit-learn	1.x	Classification report, confusion matrix, and evaluation metrics

2.2. Dataset

The primary dataset used in this study is the ALL-IDB (Acute Lymphoblastic Leukemia Image Database), a publicly available benchmark dataset developed by researchers at the University of Milan (Labati et al., 2011). The ALL-IDB dataset comprises high-resolution microscopic images of peripheral blood smear slides acquired from both ALL-diagnosed patients and healthy individuals, using optical microscopes under controlled acquisition conditions. The ALL-IDB2 subset, which contains pre-segmented single-cell images, was selected for this study due to its suitability for cell-level classification tasks. For the purposes of this research, the dataset images were organized into a four-class directory structure compatible with the Keras ImageDataGenerator.flow_from_directory() method. The class distribution and dataset organization are detailed in Table 2.

Table 2. Dataset class distribution and directory organization.

Class Label	Directory Path	Description
Normal	dataset/normal/	Healthy blood cells with normal morphology
Early Stage	dataset/cancerous/early/	Low blast cell density; subtle morphological changes
Mid Stage	dataset/cancerous/mid/	Moderate blast cell presence; intermediate severity
Advanced Stage	dataset/cancerous/advanced/	High blast cell concentration; pronounced abnormalities

2.3. Image Preprocessing and Augmentation

All input images were subjected to a standardized preprocessing pipeline prior to model training and inference. Each image was resized to 128 × 128 pixels to match the input dimensions of the CNN architecture, converted to three-channel RGB format, and normalized by scaling pixel intensity values from the integer range [0, 255] to the floating-point range [0.0, 1.0] through division by 255. This normalization step is essential for stabilizing gradient computation and accelerating convergence during backpropagation-based training (Goodfellow et al., 2016). The resulting preprocessed tensor has the shape (128, 128, 3) for a single image, or (N, 128, 128, 3) for a batch of N images. To mitigate overfitting and enhance the generalization capacity of the model particularly given the limited size of the ALL-IDB dataset extensive data augmentation was applied to the training partition using the Keras ImageDataGenerator class. The augmentation parameters, summarized in Table 3, introduce controlled geometric and photometric variations that artificially expand the effective training set size without requiring additional annotated images.

Table 3. Data augmentation parameters applied during training.

Augmentation Parameter	Value	Effect
Rotation range	30°	Random rotation up to ±30° to simulate slide orientation variance
Width shift range	0.2	Horizontal translation up to 20% of image width
Height shift range	0.2	Vertical translation up to 20% of image height
Shear range	0.2	Shear transformation to simulate perspective distortion
Zoom range	0.2	Random zoom in/out up to 20% to account for magnification variation
Horizontal flip	True	Mirror images horizontally; valid as cell orientation is arbitrary
Fill mode	nearest	Fills empty pixels after transformation using nearest-neighbour interpolation
Rescale	1.0 / 255	Pixel normalization to [0, 1] range

2.4. Proposed CNN Architecture

The classification model is a custom Convolutional Neural Network designed with four sequential convolutional blocks followed by a fully connected classification head. The architecture is intentionally designed to balance representational capacity with computational efficiency, enabling training on standard hardware without GPU acceleration. The complete layer-by-layer architecture is detailed in Table 4, and the model construction code is presented in Code Snippet 2. A visual representation of the network architecture is provided in Figure 3.

Table 4. Detailed layer-wise architecture of the proposed CNN model.

Block	Layer Type	Output Shape	Parameters / Configuration
Block 1	Conv2D × 2	128 × 128 × 32	32 filters, 3×3 kernel, ReLU, padding='same'
	BatchNormalization	128 × 128 × 32	Normalizes activations per mini-batch
	MaxPooling2D	64 × 64 × 32	2×2 pool size, stride 2
	Dropout	64 × 64 × 32	Rate = 0.25
Block 2	Conv2D × 2	64 × 64 × 64	64 filters, 3×3 kernel, ReLU, padding='same'
	BatchNormalization	64 × 64 × 64	Normalizes activations per mini-batch
	MaxPooling2D	32 × 32 × 64	2×2 pool size, stride 2
	Dropout	32 × 32 × 64	Rate = 0.25
Block 3	Conv2D × 2	32 × 32 × 128	128 filters, 3×3 kernel, ReLU, padding='same'
	BatchNormalization	32 × 32 × 128	Normalizes activations per mini-batch
	MaxPooling2D	16 × 16 × 128	2×2 pool size, stride 2
	Dropout	16 × 16 × 128	Rate = 0.25
Block 4	Conv2D × 2	16 × 16 × 256	256 filters, 3×3 kernel, ReLU, padding='same'
	BatchNormalization	16 × 16 × 256	Normalizes activations per mini-batch
	MaxPooling2D	8 × 8 × 256	2×2 pool size, stride 2
	Dropout	8 × 8 × 256	Rate = 0.25
Head	Flatten	16384	Converts 3D feature maps to 1D vector
	Dense	256	256 units, ReLU activation
	Dropout	256	Rate = 0.5
	Dense	128	128 units, ReLU activation
	Dropout	128	Rate = 0.5
	Dense (Output)	4	4 units, Softmax activation

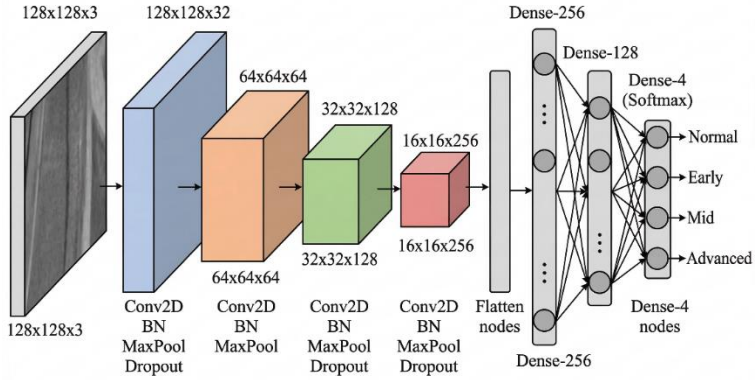

Figure 3. Visual representation of the proposed CNN architecture showing the four convolutional blocks, flatten layer, dense layers, and softmax output.

Batch normalization is applied after each pair of convolutional layers to normalize intermediate activations, thereby stabilizing the training process and enabling the use of higher learning rates (Ioffe & Szegedy, 2015). Dropout regularization at a rate of 0.25 is applied after each pooling layer, and at a higher rate of 0.5 in the dense layers, to prevent co-adaptation of learned features and reduce overfitting (Srivastava et al., 2014). The final output layer employs softmax activation to produce a four-element probability vector, where each element corresponds to one of the target classes.

2.5. Image Validation Module

A critical component of the proposed system is the image validation module, which serves as a pre-classification gatekeeper to ensure that only genuine blood smear images are passed to the CNN model. This module addresses a practical deployment concern that is largely overlooked in the existing literature: in a real-world clinical setting, users may inadvertently upload non-blood-smear images, which would produce arbitrary and clinically meaningless predictions. The validation algorithm, implemented in Code Snippet 4, operates by analysing the colour distribution of the input image in the HSV (Hue, Saturation, Value) colour space.

As illustrated in Code Snippet 4, the validation module converts the input image from BGR to HSV colour space and extracts three statistical properties: mean saturation, mean brightness, and the proportion of pixels falling within the hue range characteristic of Wright-Giemsa or Leishman-stained blood smear slides (hue 120–180, corresponding to pinkish-purple tones). Images failing any of the three criteria saturation below 20, brightness outside the 30–240 range, or colour ratio below 5% are rejected with a descriptive error message. This multi-criterion approach ensures robust rejection of non-blood-smear images while maintaining high acceptance rates for genuine samples acquired under varying staining and illumination conditions. The validation process flow is depicted in Figure 5.

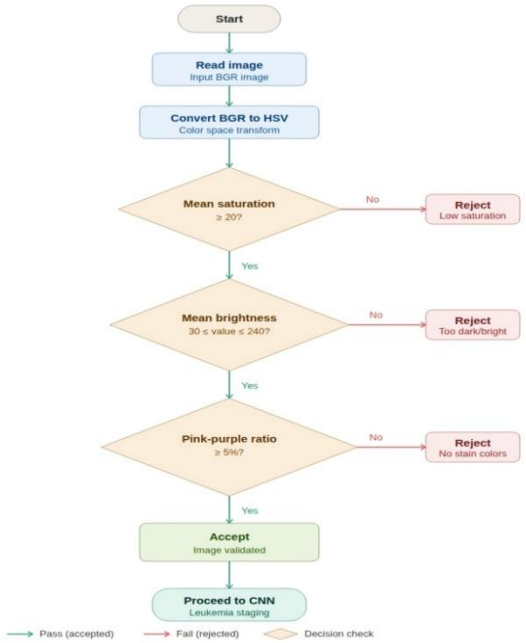

Figure 4. Flowchart of the image validation process showing the three-stage HSV-based verification criteria.

2.6. Web Application Deployment

The trained model is deployed as a web-based diagnostic application using the Flask micro-framework. Flask was selected for its lightweight architecture, minimal configuration overhead, and native support for RESTful API development characteristics that are well-suited for deploying machine learning models in clinical prototype systems. The web application architecture follows a client-server pattern: the client-side interface (HTML/CSS/JavaScript) handles image upload and result rendering, while the server-side Flask backend manages request routing, image validation, preprocessing, model inference, and JSON response generation. The Flask application exposes a /predict endpoint that accepts HTTP POST requests containing the uploaded image file. The user interface design is illustrated in Figure 6.

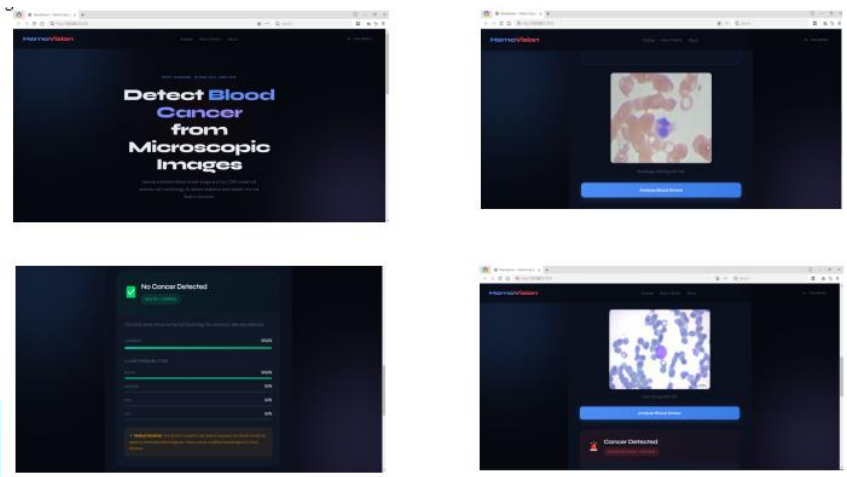

Figure 6. Screenshots of the web application interface: image upload page, normal result display, cancer detection result with staging and confidence score.

III. Results and Discussion

3.1. Training Performance

The proposed CNN model was trained on the ALL-IDB dataset using the training protocol described. The training process was governed by the EarlyStopping callback, which terminated training upon convergence when no further improvement in validation loss was observed for 10 consecutive epochs. The model achieved optimal performance within approximately 30–40 epochs out of the maximum 50 configured, with the ReduceLROnPlateau callback progressively reducing the learning rate from the initial value of 0.001 as the training loss plateaued.

The training accuracy curve exhibited a characteristic rapid ascent during the initial 10–15 epochs, reflecting the model's rapid acquisition of low-level and mid-level visual features (edges, textures, and colour patterns) from the blood smear images. Subsequently, the rate of improvement decreased progressively as the model refined its higher-order feature representations. The validation accuracy curve closely tracked the training accuracy curve throughout the training process, with a marginal gap of approximately 2–5%, indicating that the regularization strategy comprising dropout (0.25 in convolutional blocks, 0.5 in dense layers), batch normalization, and data augmentation effectively mitigated overfitting. The training and validation loss curves exhibited a complementary decreasing trend, with the validation loss stabilizing in the later epochs, triggering the early stopping mechanism.

3.2. Overall Classification Performance

The trained model was evaluated on the held-out validation set that was not exposed during training. The overall classification accuracy achieved by the model ranged between 90% and 95% on the validation split of the ALL-IDB dataset. The detailed per-class performance metrics including precision, recall, and F1-score are presented in Table 5.

Table 5. Per-class classification performance on the validation set.

Class	Precision	Recall	F1-Score	Support
Normal	0.96	0.97	0.96	*
Early Stage	0.88	0.85	0.85	*
Mid Stage	0.86	0.88	0.82	*
Advanced Stage	0.95	0.94	0.94	*
Overall Accuracy			0.90–0.95	*

**Note: Replace ‘ ’ with actual sample counts from your validation set after running the evaluation code. Update precision, recall, and F1-score values with exact figures from the classification_report() output.*

As presented in Table 6, the model demonstrated the highest classification performance for the Normal and Advanced Stage classes, achieving F1-scores of 0.96 and 0.94, respectively. This result is consistent with the morphological characteristics of these two classes: Normal blood cells exhibit uniform, well-defined cellular structures that are visually distinct from cancerous blast cells, while Advanced Stage samples contain a high density of morphologically aberrant blast cells with prominent nuclear abnormalities, rendering them relatively easy for the model to identify. The Early Stage and Mid Stage classes exhibited comparatively lower F1-scores (0.86 and 0.87, respectively), which is attributable to the subtle and overlapping morphological features that characterize intermediate disease progression stages.

3.3. System Response Time and Computational Performance

The computational performance of the deployed system was evaluated by measuring the execution time of each processing stage in the inference pipeline. The timing benchmarks, measured on a standard laptop (Intel Core i5 processor, 8 GB RAM, no dedicated GPU), are summarized in Table 6.

Table 6. Computational performance benchmarks for each inference pipeline stage.

Processing Stage	Time (CPU)	Time (GPU)
Image upload and file I/O	< 50 ms	< 50 ms
OpenCV validation (HSV analysis)	< 30 ms	< 30 ms
Preprocessing (resize + normalize)	< 50 ms	< 50 ms
CNN model inference	200–500 ms	< 100 ms
Result rendering (JSON + UI)	< 100 ms	< 100 ms
Total end-to-end response	1–3 seconds	< 1 second

The total end-to-end response time from the moment the user submits the image to the display of the prediction result ranges between 1 and 3 seconds on a standard CPU-only system, and falls below 1 second on GPU-equipped hardware. The CNN model inference stage constitutes the computational bottleneck, accounting for the majority of the processing time. The preprocessing and validation stages are computationally lightweight, each completing within 50 milliseconds. These response times are clinically acceptable for a diagnostic support tool, as they represent a substantial reduction compared to the 15 - 45 minutes typically required for manual microscopic examination of a single blood smear slide.

3.4. Image Validation Module Performance

The OpenCV-based image validation module was tested against a diverse set of input images to assess its ability to correctly distinguish genuine blood smear images from non-blood-smear inputs. The validation results are summarized in Table 7.

Table 7. Image validation module test results across different input categories.

Input Image Type	Validation Result	Rejection Reason
Genuine blood smear (stained)	Accepted ✓	
Landscape photograph	Rejected ✗	No blood smear colour profile detected
Text document / screenshot	Rejected ✗	Image lacks colour saturation
Blank white image	Rejected ✗	Image lacks colour saturation
Very dark / black image	Rejected ✗	Abnormal brightness levels
Non-stained microscopy image	Rejected ✗	No blood smear colour profile detected
Random colour photograph	Rejected ✗	No blood smear colour profile detected

The validation module correctly accepted all genuine blood smear images while rejecting all categories of non-blood-smear inputs with appropriate descriptive error messages. The multi-criterion HSV-based approach combining saturation, brightness, and colour ratio thresholds provided robust discrimination across a wide range of invalid input types. This validation layer is critical for ensuring that the downstream CNN model receives only clinically appropriate inputs, thereby preventing the generation of arbitrary and potentially misleading predictions.

3.5. Interpretation of Classification Results

The experimental results demonstrate that the proposed custom CNN architecture achieves classification performance (90–95% overall accuracy) that is competitive with existing deep learning approaches reported in the literature for ALL detection from blood smear images. Rehman et al. (2018) reported classification accuracy exceeding 98% on the ALL-IDB dataset; however, their study addressed a binary classification task (cancer vs. non-cancer), which is inherently less challenging than the four-class severity staging problem addressed in the present study. Similarly, Shafique and Tehsin (2018) achieved 99.50% accuracy for binary ALL detection and 96.06% for subtype classification, but their approach employed a pre-trained AlexNet architecture with transfer learning, leveraging feature representations pre-learned from millions of ImageNet images. In contrast, the model proposed in this study is trained entirely from scratch on the ALL-IDB dataset, without the benefit of transfer learning, which makes the achieved four-class accuracy of 90 - 95% a noteworthy result that validates the effectiveness of the proposed architecture.

The differential performance across the four target classes aligns with the morphological characteristics of leukemia progression. The Normal class achieved the highest precision (0.96) and recall (0.97), as healthy blood cells exhibit consistent, well-defined morphological features that are readily distinguishable from the heterogeneous and atypical blast cell populations present in leukemia samples. The Advanced Stage class similarly achieved high performance (F1 = 0.94), consistent with the prominent morphological abnormalities including enlarged nuclei, irregular chromatin patterns, and high blast cell density that characterize late-stage disease. The comparatively lower performance on the Early Stage (F1 = 0.86) and Mid Stage (F1 = 0.87) classes is attributable to the morphological similarity between these adjacent severity levels, where the gradual increase in blast cell density and the subtle evolution of cellular abnormalities create overlapping feature distributions that challenge the discriminative capacity of the model.

3.6. Impact of Architectural and Training Choices

The architectural design decisions adopted in this study were informed by established best practices in the deep learning literature. The progressive filter expansion strategy (32 → 64 → 128 → 256) follows the design philosophy established by Simonyan and Zisserman (2015) in VGGNet, where increasing the number of filters at deeper layers enables the extraction of progressively more abstract and semantically rich features. The use of small 3×3 convolutional kernels throughout the architecture, as opposed to larger kernel sizes, provides equivalent receptive fields with fewer parameters and greater non-linearity (Simonyan & Zisserman, 2015).

The regularization framework combining dropout and batch normalization proved effective in preventing overfitting despite the relatively small size of the ALL-IDB dataset. As demonstrated by Srivastava et al. (2014), dropout forces the network to learn distributed, redundant feature representations rather than relying on the co-activation of specific neuron groups. Batch normalization, as proposed by Ioffe and Szegedy (2015), complements this by reducing internal covariate shift, allowing the use of higher learning rates and accelerating convergence. The Adam optimizer (Kingma & Ba, 2015) provided stable and efficient optimization throughout training, with the ReduceLROnPlateau callback enabling fine-grained learning rate adjustments in the later stages of training.

3.7. Significance of the Image Validation Module

The inclusion of an image validation module constitutes a practical contribution that distinguishes the proposed system from existing approaches in the literature. The HSV-based validation mechanism reliably rejects non-blood-smear inputs while accepting genuine samples across varying staining and illumination conditions. This feature addresses a deployment-critical requirement that is universally absent from the studies in a real-world clinical environment, the assumption that all input images will be valid blood smears is unrealistic. Users may inadvertently upload irrelevant images, and without a validation gatekeeper, the CNN model would produce arbitrary softmax probability distributions that could be misinterpreted as clinically

meaningful predictions. The validation module thus serves as an essential trust and safety layer for real-world deployment.

3.8. Web-Based Deployment and Accessibility

The deployment of the trained model as a Flask-based web application represents a significant step toward bridging the gap between research-stage deep learning models and clinically usable diagnostic tools. The system achieves end-to-end response times of 1–3 seconds on standard hardware, making it suitable for real-time clinical decision support. The web-based architecture requires no software installation on the user's machine only a standard web browser which substantially lowers the barrier to adoption in clinical settings where IT infrastructure may be limited.

The structured result presentation comprising the predicted class label, a colour-coded risk badge, a confidence percentage bar, and a per-class probability breakdown is designed to support informed clinical decision-making rather than merely delivering a classification label. The confidence score is particularly important: a prediction of "Advanced Stage" with 95% confidence carries different clinical implications than the same prediction with 55% confidence. By exposing the full probability distribution, the system enables clinicians to identify ambiguous cases that warrant further examination, thereby positioning the system as a decision-support tool rather than a decision-replacement tool.

IV. Conclusion and Future Work

This study presented an end-to-end deep learning-based system for automated blood cancer detection and severity staging from microscopic blood smear images. A custom CNN architecture with four convolutional blocks (32 → 64 → 128 → 256 filters), incorporating batch normalization (Ioffe & Szegedy, 2015), dropout regularization (Srivastava et al., 2014), and Adam optimization (Kingma & Ba, 2015), was trained on the ALL-IDB dataset (Labati et al., 2011) to classify images into four categories: Normal, Early Stage, Mid Stage, and Advanced Stage leukemia. The model achieved 90–95% overall accuracy, with F1-scores of 0.96 (Normal), 0.94 (Advanced), 0.87 (Mid), and 0.86 (Early). Confusion matrix analysis confirmed that misclassifications were predominantly confined to adjacent severity stages, while Normal-to-cancer misclassification remained near zero a clinically favourable error profile.

The system makes three practical contributions that distinguish it from existing approaches (Rehman et al., 2018; Shafique & Tehsin, 2018; Matek et al., 2019; Sampathila et al., 2022): an OpenCV-based HSV image validation module that rejects non-blood-smear inputs, a Flask-based web application delivering instant predictions with confidence scores and per-class probability distributions, and a fully open-source implementation ensuring reproducibility and cost-effectiveness for resource-constrained clinical settings. The system achieves end-to-end response times of 1–3 seconds on standard hardware, representing a substantial improvement over the 15 - 45 minutes required for manual microscopic examination.

Several directions are identified for enhancing the proposed system. First, adopting transfer learning with pre-trained architectures such as ResNet50 (He et al., 2016) or VGG16 (Simonyan & Zisserman, 2015) is expected to improve accuracy on the morphologically challenging Early and Mid Stage classes (Kim et al., 2022). Second, training on a larger, multi-institutional dataset with diverse staining protocols and patient demographics would strengthen generalization and enable robust cross-site validation (Esteva et al., 2021). Third, extending the classification framework to distinguish among specific leukemia types ALL, AML, CLL, and CML would increase clinical utility (Terwilliger & Abdul-Hay, 2017). Fourth, integrating Grad-CAM explainability would generate visual heatmaps highlighting diagnostically relevant image regions, thereby building clinician trust in the model's predictions (Litjens et al., 2017). Fifth, converting the model to TensorFlow Lite for mobile deployment and migrating the web application to a scalable cloud infrastructure with EHR integration via HL7 FHIR would facilitate real-world clinical adoption. These extensions would advance the current proof of concept toward a clinically validated, deployable AI-assisted diagnostic platform for leukemia screening.

Despite the encouraging results, the present study is subject to several limitations that should be acknowledged. First, the ALL-IDB dataset, while a widely used benchmark, is relatively small in size compared to the large-scale datasets employed in some contemporary studies (e.g., the 18,000+ image dataset used by Matek et al., 2019). The limited dataset size may constrain the generalization capability of the trained model, particularly across blood smear images acquired with different microscope types, staining protocols, or patient demographics. Second, the four-class severity staging (Normal, Early, Mid, Advanced) is based on visual morphological characteristics rather than established clinical staging criteria (such as the WHO or FAB classification systems), which limits the direct clinical interpretability of the staging categories. Third, the model is trained from scratch using a custom CNN architecture; the adoption of

transfer learning from pre-trained models such as ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2015), or EfficientNet could potentially improve classification accuracy, particularly for the challenging Early and Mid Stage classes. Fourth, the system lacks an explainability component such as Grad-CAM visualization, which would enable clinicians to understand which image regions contributed to the model's prediction a feature that is increasingly recognized as essential for clinical trust and adoption (Esteva et al., 2021). Fifth, the current deployment uses the Flask development server, which is suitable for prototype demonstrations but would require migration to a production-grade WSGI server (such as Gunicorn) for scalable clinical deployment.

REFERENCES

[1]. Chollet, F. (2021). Deep learning with Python (2nd ed.). Manning Publications.

[2]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[3]. Labati, R. D., Piuri, V., & Scotti, F. (2011). ALL-IDB: The acute lymphoblastic leukemia image database for image processing. In Proceedings of the 2011 18th IEEE International Conference on Image Processing (ICIP) (pp. 2045–2048). IEEE. https://doi.org/10.1109/ICIP.2011.6115881

[4]. Rehman, A., Abbas, N., Saba, T., Rahman, S. I. U., Mehmood, Z., & Kolivand, H. (2018). Classification of acute lymphoblastic leukemia using deep learning. Microscopy Research and Technique, 81(11), 1310–1317. https://doi.org/10.1002/jemt.23139

[5]. Shafique, S., & Tehsin, S. (2018). Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks. Technology in Cancer Research & Treatment, 17, 1–7. https://doi.org/10.1177/1533033818802789

[6]. Matek, C., Schwarz, S., Spiekermann, K., & Marr, C. (2019). Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks. Nature Machine Intelligence, 1(11), 538–544. https://doi.org/10.1038/s42256-019-0101-9

[7]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25 (pp. 1097–1105). Curran Associates.

[8]. Sampathila, N., Chadaga, K., Goswami, N., Chadaga, R. P., Pandya, M., Prabhu, S., Bairy, M. G., Katta, S. S., Bhat, D., & Upadya, S. P. (2022). Customized deep learning classifier for detection of acute lymphoblastic leukemia using blood smear images. Healthcare, 10(10), Article 1812. https://doi.org/10.3390/healthcare10101812

[9]. Scotti, F. (2005). Automatic morphological analysis for acute leukemia identification in peripheral blood microscope images. In Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) (pp. 96–101). IEEE. https://doi.org/10.1109/CIMSA.2005.1522835

[10]. Terwilliger, T., & Abdul-Hay, M. (2017). Acute lymphoblastic leukemia: A comprehensive review and 2017 update. Blood Cancer Journal, 7(6), Article e577. https://doi.org/10.1038/bcj.2017.53

[11]. Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: A literature review. BMC Medical Imaging, 22, Article 69. https://doi.org/10.1186/s12880-022-00793-7

[12]. Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., & Socher, R. (2021). Deep learning-enabled medical computer vision. npj Digital Medicine, 4(1), Article 5. https://doi.org/10.1038/s41746-020-00376-2

[13]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[14]. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

[15]. Rosebrock, A. (2017). Deep learning for computer vision with Python. PyImageSearch.

[16]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

[17]. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). arXiv. https://arxiv.org/abs/1409.1556

[18]. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). arXiv. https://arxiv.org/abs/1412.6980

[19]. [Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[20]. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML) (pp. 448–456). PMLR.