Optimized Deep Ensemble Model for Early Detection of Leukemia Detection

Rahila[1], Mohammad Nihal[2], Harshitha H Hegde[3], Ganesh M.S[4]
[1]*Assistant Professor, Department of Artificial Intelligence and Machine Learning, Srinivas Institute of Technology (S.I.T), Mangalore, Karnataka, India*
[2,3,4]*Students, Department of Artificial Intelligence and Machine Learning, Srinivas Institute of Technology (S.I.T), Mangalore, Karnataka, India*
doi.org/10.64643/IJIRTV12I12-206680-459

***Abstract*—A fully automated pipeline for blood cell image recognition and categorization is introduced in this work, built around contemporary deep learning methods. Drawing from several independent repositories, 13,149 microscopic images were assembled into a dataset intentionally diverse enough to challenge the model and improve its real-world generalizability. Three neural architectures VGG16, ResNet50, and a bespoke convolutional design work in concert under an ensemble arrangement, each contributing its own analytical perspective to arrive at predictions more reliable than any single model could produce alone. Raw images are conditioned, meaningful representations are extracted, and a final label is assigned through a sequential processing chain. Fifty training epochs were completed using well-separated training and test partitions. Across all three-evaluation metrics accuracy, precision, and recall the results confirmed the model's ability to reliably differentiate between normal and abnormal cell samples. From a practical standpoint, the framework shortens the path to diagnosis, lightens the analytical workload on laboratory staff, and illustrates how thoughtfully applied artificial intelligence can elevate the standard of care in modern medicine.**

***Index Terms*—Artificial Intelligence, Blood Cell Classification, Biomedical Image Analysis, Computer Vision, Deep Learning, Ensemble Learning, Convolutional Neural Networks, VGG16, ResNet50.**

I. INTRODUCTION

Computational tools have been quietly reshaping clinical medicine for well over a decade, and the shift has accelerated dramatically in recent years. What began as experimental research into machine perception has matured into working diagnostic infrastructure. Practitioners now routinely rely on AI-assisted systems to flag anomalies, prioritize urgent cases, and support complex decisions. Blood cancers sit near the top of the list of conditions that stand to gain most from this shift, precisely because they can deteriorate so quickly and because early identification is so tightly linked to survival. Acute Lymphoblastic Leukemia ALL exemplifies both of those characteristics. Though it strikes disproportionately in childhood, adults are not exempt, and the prognosis worsens sharply when diagnosis comes late. The pathophysiology of ALL centers on a breakdown in normal cell production. Immature lymphoid progenitor cells replicate without restraint inside the bone marrow, eventually overwhelming the healthy cell population. The consequences are predictable: falling red cell counts produce anemia, depleted white cells leave the patient vulnerable to infection, and platelet deficits impair the body's ability to stop bleeding. The frustrating irony is that catching the disease at an early stage when treatment is most effective is extremely difficult. Malignant blasts and their benign counterparts look almost identical under a light microscope, even to an experienced eye.

Conventional hematological workups involve manual slide review by trained pathologists combined with a battery of laboratory tests. The process is thorough but slow, and it cannot function without access to scarce specialist expertise. In settings where such expertise is unevenly distributed which describes most of the world outside major urban centers diagnostic delays are not the exception but the norm. Those delays translate directly into worse outcomes. There is a clear and pressing need for diagnostic tools that can operate with speed and consistency regardless of where the patient happens to be.

Among the computational approaches that have shown promise in pathology, Convolutional Neural Networks stand out for their capacity to learn discriminative features directly from pixel data.

They require no hand-engineered descriptors and generalize well across imaging conditions when trained on sufficiently varied data. Ensemble methods extend this advantage: by combining several architectures in this case VGG16, ResNet50, and a purpose-designed CNN the final prediction reflects a consensus that smooths out the individual biases and failure modes of each constituent model.

Clinicians interact with the platform through a browser-based interface. Uploading a blood smear image triggers the full analysis pipeline, and a labeled result is returned within a matter of seconds. The entire experience was designed to require no technical background from the end user.

II. PROPOSED SYSTEM

This platform was built as a web-deployable screening aid for ALL, using an ensemble of deep learning models applied to microscopic blood smear images. Rather than bundling everything into one fragile mechanism, the system was divided into four distinct functional stages image conditioning, feature learning, model inference, and output rendering. Each stage owns its work completely before passing to the next, producing a workflow that can actually be traced, audited, and trusted.

Clinicians open the application to a clean upload panel. No orientation required. A slide goes in dragged across or browsed manually and the conditioning stage immediately takes over. Spatial dimensions are aligned, intensity values normalized, and background noise cleared away before anything consequential begins. These corrections seem minor until you consider what skipping them costs: a poorly conditioned image doesn't just underperform, it misinforms. Once a slide clears that standard, it moves simultaneously to VGG16, ResNet50, and a purpose-built CNN.

The three models work in deliberate isolation, each forming its own interpretation of the cellular structures without awareness of what the others are concluding. When the ensemble stage brings those independent readings together through voting or confidence-weighted blending the result carries a robustness no single model could achieve alone. Ambiguous cases, the ones closest to diagnostic boundaries, are precisely where that collective stability matters most.

Building that reliability took 13,149 microscopic images drawn from multiple varied datasets, trained across repeated cycles of refinement until accuracy, precision, and recall reached clinically defensible levels. Results return within seconds.

The platform's broader purpose is straightforward: faster, more dependable hematological screening particularly where specialist capacity is limited and the need is greatest.

III. SYSTEM DESIGN

The Three requirements dominated the architectural planning for this system: capacity to handle increasing user load without degrading performance, a component structure loose enough that any single layer can be updated without cascading effects through the rest of the application, and response times fast enough that inference feels instantaneous to the clinical user. Every major design choice can be traced back to at least one of these priorities. The external-facing layer is a React.js application that renders the upload interface and displays diagnostic results. Designed with clinical environments in mind, it maintains usability across device types from a fixed workstation monitor to a tablet used on a ward round. The interface keeps the interaction simple: upload, submit, read the result.

Requests from the frontend are received by a Node.js / Express backend that functions as the system's coordination layer. It authenticates users, validates incoming data, manages the sequence of operations, and relays information between the interface and the AI module. Patient data is handled with appropriate security controls, and error conditions are caught and communicated back to the user in plain language rather than raw stack traces. Requests from the frontend are received by a Node.js / Express backend that functions as the system's coordination layer. It authenticates users, validates incoming data, manages the sequence of operations, and relays information between the interface and the AI module. Patient data is handled with appropriate security controls, and error conditions are caught and communicated back to the user in plain language rather than raw stack traces.

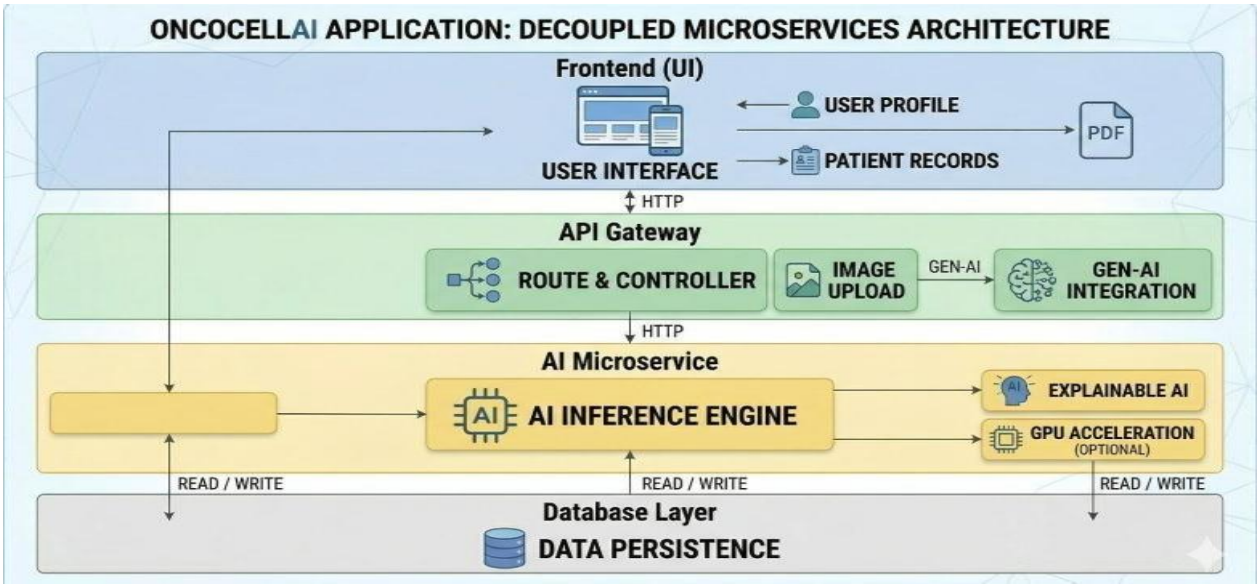

Figure 1. System Architecture

Requests from the frontend are received by a Node.js / Express backend that functions as the system's coordination layer. It authenticates users, validates incoming data, manages the sequence of operations, and relays information between the interface and the AI module. Patient data is handled with appropriate security controls, and error conditions are caught and communicated back to the user in plain language rather than raw stack traces.

The analytical core is the AI inference module. It receives the conditioned image, runs it through the ensemble stack, and produces a labeled prediction accompanied by a confidence percentage. Behind it sits a storage layer that retains uploaded images, diagnostic histories, and associated patient records in a form optimized for both structured lookups and large-file retrieval.

Data moves through the system along a single, clearly defined path. The frontend sends a request, the backend validates and forwards it, the AI module produces a result, the result flows back through the backend, and the frontend renders it for the user. This linear, layered flow is what makes the system easy to test, easy to maintain, and easy to extend.

IV. IMPLEMENTATION

The session begins the moment a clinician opens the application. Before any image is touched, the system establishes something essential: who this analysis belongs to. A patient record must be linked to the session retrieved if one exists, created if it doesn't. Without that anchor, a diagnostic result is just an isolated number. With it, there's a traceable thread connecting every finding to a named individual something clinicians can return to, build upon, and defend in an audit.

Submitting an image is intentionally frictionless. A blood smear in JPEG or PNG format can be dragged onto the upload zone or selected manually. An optional notes field sits alongside small enough to overlook, valuable enough to include giving clinicians space to attach a clinical observation before the image moves forward.

Pressing submit triggers an HTTP POST request carrying both the image and session metadata to the FastAPI backend. The server extracts the image, writes it to disk under a unique identifier, and runs a format and dimension check before anything else proceeds. Only clean files enter the preprocessing sequence.

Preprocessing converts the raw image into model-ready input: resized to 224 × 224 pixels, channel values normalized to a zero-to-one range, and arranged into a PyTorch-compatible tensor. That tensor is dispatched simultaneously to all three model branches.

VGG16, ResNet50, and the custom CNN each evaluate the image independently, producing a seven-class probability vector. The ensemble averages those

vectors and selects the highest-scoring class as the final prediction. Confidence is derived from the leading per-model scores

a meaningful indicator of how decisively the models agreed.

Grad-CAM then back-propagates gradient signals to VGG16's final convolutional layer, weighting the activation maps to produce a spatial heatmap of the regions that drove the prediction. Composited onto the original image at 60% opacity, this overlay makes the model's reasoning visible rather than assumed. The backend packages the prediction label, confidence score, heatmap, and AI-generated clinical guidance into a JSON response. The interface transitions to the results view diagnosis, overlay, confidence reading, and recommendations laid out together, with PDF export and a comparison panel available alongside

V. RESULTS

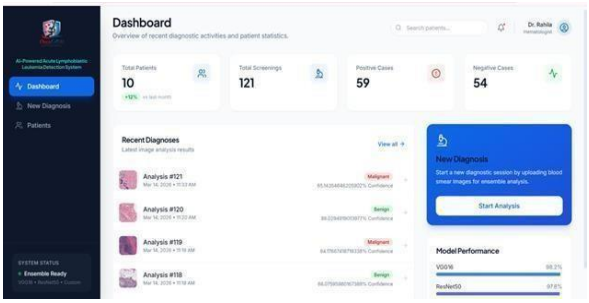

Figure 2: Oncocell AI Dashboard

Fig. 2. Dashboard Logging in deposits the user directly onto the Dashboard, which serves as the system's central overview panel. The layout was organized around a single governing question: what does a clinician need to see first thing? The answer shapes every element on the screen.

A row of metric cards occupies the upper portion of the view. Each card summarizes one operational dimension of the system the patient headcount, the number of completed screenings, and the split between malignant and benign findings. The values behind these cards are fetched fresh from the database on each page load, so what the user sees always matches the live state of the system rather than a cached snapshot

Just below the summary cards, the dashboard keeps a running memory of everything the system has seen. Each entry in the log tells a small story: a thumbnail of the submitted slide, the exact date and time the analysis ran, the cell class the ensemble landed on, and the confidence figure that came with it. Nothing is buried or requires clicking through to find. The color coding on the prediction badges does a lot of the quiet heavy lifting a clinician skimming the list after a long shift doesn't need to read every line carefully. The colors speak first, flagging the cases that might deserve a second pair of eyes before anything gets signed off on.

The bottom of the dashboard is built around the reality that clinical environments move fast and patience for navigation is thin. A clearly labeled button sits ready to launch a new analysis session without asking the user to leave the page they're already on because the extra step of navigating away, small as it sounds, is the kind of friction that compounds over a busy day. Right beside it, live accuracy readouts for each model in the ensemble are visible at a glance. These aren't decorative. Knowing how each model is performing in real time gives the person reading the predictions something important: a reason to trust them, or a reason to pause. That ongoing transparency is what separates a tool that clinicians rely on from one they merely tolerate

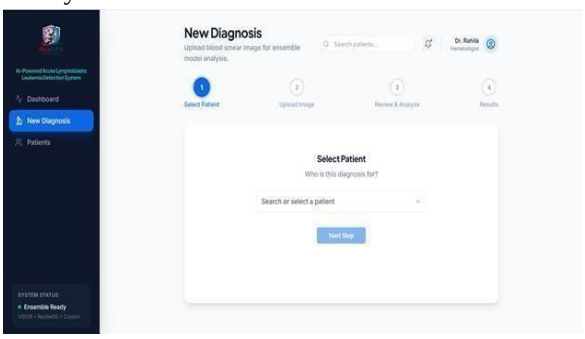

Figure 3: New Diagnosis Interface

Fig. 3. New Diagnosis interface showing the Clicking through to the New Diagnosis module initiates the structured intake flow. The design philosophy here is deliberate guidance rather than open-ended input: a four-step wizard leads the user through every required piece of information in a fixed sequence, closing off the possibility of skipping steps or submitting an incomplete record.

The opening step prompts the user to identify the patient whose sample is being analyzed. A dropdown list with live search filtering makes it straightforward to locate an existing record; arriving here from a patient's profile page auto-populates the field entirely. Subsequent steps cover image submission, a

confirmation review, and the final trigger to initiate the AI analysis with each step locked behind successful completion of the one before it.

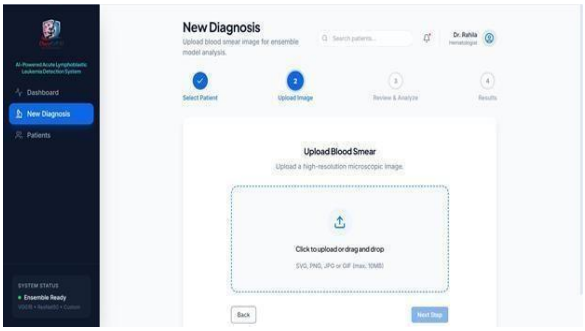

Figure 4: Blood Smear Upload Interface

Fig.4.Blood smear image Step two is entirely devoted to one thing: getting the image into the system the right way. The upload panel takes up most of the screen, and deliberately so. There's no clutter competing for attention, no reason to hunt for the right field. A blood smear slide PNG or JPEG can be dragged straight onto the panel or located through the standard file browser, whichever feels more natural in the moment. The moment a file is selected, a preview renders immediately. That instant visual confirmation matters more than it might seem.

It gives the submitting clinician a beat to pause and actually look to make sure the right slide made it in before anything moves forward. A small checkpoint, but one that catches the kind of quiet mistake that's easy to make and harder to catch later. Format and size checks happen right there in the browser, without touching the server. If something is wrong a file type that won't work, dimensions that fall outside what the pipeline expects the user finds out immediately, with a clear inline message explaining what needs to change. There's no waiting, no round-trip to a server and back, no ambiguous delay. The feedback is instant and specific.

This isn't just about convenience. Letting a malformed file slip through the front door and travel deeper into the pipeline is a real risk not because the system would crash, but because a corrupted or misformatted input can quietly skew the inference process and return a result that looks confident but isn't grounded in clean data. Catching the problem here, at the earliest possible moment, keeps that from ever becoming an issue.

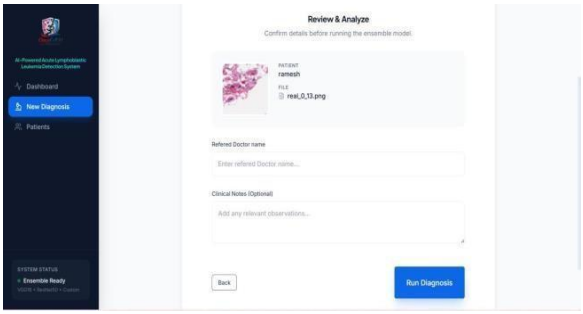

Fig 5: Review and Analyze Interface

Fig.5 Before the computationally expensive inference process begins, the system presents a summary screen showing the selected patient and the uploaded image side by side, giving the user one final opportunity to catch errors before committing. Two optional fields are available at this stage: • Referred Doctor Name: A text field allows the user to specify who requested the scan (e.g., Dr. Smith). • Clinical Notes: A text area allows the technician to add any relevant observations or symptoms for the AI to consider. Run Diagnosis Button: Submitting these triggers the upload to the server and initiates the VGG16, ResNet50, and Custom CNN inference scripts in the background.

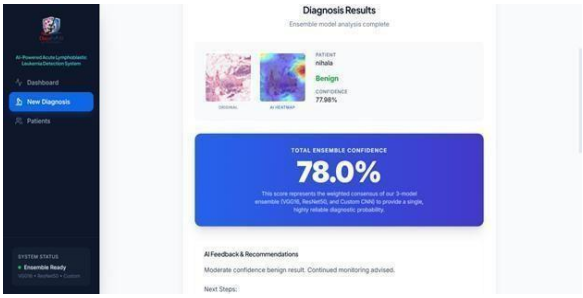

Fig 6: Diagnosis Results Interface

Fig. 6. *Results interface displaying the final diagnosis, confidence score, Grad-CAM heatmap overlay, and AI-generated clinical recommendation.*

Seconds after submission, the interface transitions to the results screen. Visual, numerical, and textual evidence are laid out together so the clinician can form a complete picture without toggling between views.

The Grad-CAM overlay sits beside the original slide image, marking in color the cellular subregions that most strongly drove the ensemble's decision. For a clinician reviewing the result, this annotation provides a starting point for visual cross-checking it shows not just what the model concluded but roughly where in the image that conclusion was anchored.

The predicted label malignant or benign is rendered in large, color-differentiated text positioned for immediate legibility. The accompanying confidence percentage quantifies the degree of agreement across the ensemble's component models. The Groq language model generates a set of tailored clinical recommendations beneath the core result, offering interpretive context and suggesting concrete next steps calibrated to the specific finding.


opp srinivas college,valachila,manglore -574143

MEDICAL REPORT

Visit Info

Refered Doctor name: Dr. AI Assistant **Visit Date:** 16.03.2026

Specialization: Hematology/Oncology

Patient Info

Full Name: nihala **Birth Date:** 1992.01.01

Med. Number: P-7855 **Gender:** Male

Assessment

The patient's submitted blood smear has been thoroughly evaluated by our ensemble of deep learning models(VGG16, ResNet50, and Custom CNN).The analysis focuses on detecting cellular anomalies indicative of malignant or benign conditions.

Diagnosis

The primary prediction indicates a **Benign** classification with an overall ensemble confidence of **77.97628343105316%**. Your diagnostic result is based on an **Ensemble Evaluation**, where multiple expert AI models assess the sample simultaneously to ensure the highest possible precision.

AI Recommendation

Moderate confidence benign result. Continued monitoring advised.

Next Steps:
- Follow-up appointment in 3-6 months
- Regular health check-ups
- Consider additional screening if risk factors present

Precautions:
- Maintain healthy lifestyle
- Be aware of family history and risk factors
- Report any concerning symptoms

Fig 7: Medical Report Interface

Fig. 7 Medical report generated by the system showing patient details, diagnosis, confidence score, and AI-based recommendations

Of everything the system produces, the Medical Report is the most complete. It doesn't summarize the diagnostic session it preserves it, in full, in a format that works equally well as a clinical record, a referral document, or an audit trail. The guiding principle behind its design is self-containment: someone picking up this report weeks or months later, with no access to the application and no memory of the original session, should be able to read it from top to bottom and walk away with everything they need. Important information shouldn't be hostage to a running application. Whatever the system finds must live somewhere permanent something a clinician, auditor, or specialist can pick up and read long after the software has moved on. That commitment quietly governed every structural choice made in building this report.

The header wastes no time. The referring clinician's name, specialty, and the exact timestamp of the analysis appear before anything else. These aren't courtesy details they're what give the document its accountability. Below them, the patient's full name, medical record number, date of birth, and biological sex complete the picture. By the final line of the header, every foundational question has already been answered: who ordered this, for whom, and precisely when.

The body follows a deliberate sequence methodology, then findings, then guidance. The methodology section establishes upfront that three independent models contributed to the result, not one algorithm making an unchecked call. That framing matters before the findings land. The findings section then delivers the classification plainly malignant or benign alongside the confidence score that reflects how strongly all three models agreed. Nothing is softened into ambiguity or hidden inside technical phrasing. The result is stated, and the evidence behind it is visible.

The recommendations section is where the report earns its real value. It takes the classification and converts it into direction a care team can actually use follow-up timelines, symptoms worth monitoring, flags for further workup where the findings suggest it. None of it is templated or generic; it responds directly to what was found. A report that only records a result informs. One that points toward the next step serves the patient and that is the standard this section holds itself to.

VI. CONCLUSION

This project set out to build a web-based diagnostic platform capable of detecting ALL from blood smear imagery and returning findings in a form clinicians could actually use. It delivered and then some. What emerged isn't just functional software; it's tangible proof that AI-assisted medical imaging can operate outside controlled research settings and contribute meaningfully to real clinical decisions.

The ensemble framework combining VGG16,

ResNet50, and a custom CNN proved to be the most important architectural choice made. Each individual model carries inherent limitations: patterns it handles well, edge cases it doesn't. Bringing three together doesn't eliminate those weaknesses; it ensures none of them gets the final word alone. The payoff was most visible on ambiguous slides, the diagnostically uncertain cases where individual models waver. The ensemble held steady precisely there where clinical consequences are heaviest.

Usability received equal standing alongside accuracy throughout development. First-time users navigated the workflow without guidance, documentation, or adjustment time. The system simply met them where they were. Speed carried the same weight in time-pressured clinical environments, a slow tool gets quietly abandoned. Keeping responses within seconds wasn't a performance goal; it was an adoption requirement.

Grad-CAM overlays and AI-generated recommendations addressed the trust problem that shadows every diagnostic AI: can a clinician understand why the system concluded what it did? By surfacing the visual evidence and translating it into actionable clinical language, the system earns its place as a collaborator rather than an authority one that shows its reasoning and respects the clinician's judgment.

Limitations exist and warrant honesty. Model behavior on images from underrepresented populations or unfamiliar imaging conditions remains incompletely characterized. Prospective validation under real-world conditions is not optional it is the necessary next step before any responsible large-scale deployment.
The foundation is solid. Broader datasets, clinical trials, and deployment pilots will determine how far it carries.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Prof. Ganesh M S, Department of Artificial Intelligence and Machine Learning, Srinivas Institute of Technology, for his valuable guidance, support, and encouragement throughout the development of this research work

REFERENCES

[1] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[2] T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.
[3] K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: MIT Press, 2012.
[4] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Birmingham, U.K.: Packt Publishing, 2019.
[5] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
[6] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 1999, pp. 1150–1157.
[7] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, vol. 1, pp. 269–271, 1959.
[8] F. Chollet, Deep Learning with Python. Shelter Island, NY, USA: Manning Publications, 2018.
[9] J. Brownlee, Machine Learning Mastery with Python. Machine Learning Mastery, 2016.
[10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[11] M. Ghaderzadeh et al., "Machine Learning in Leukemia Detection: A Review," Computational Intelligence and Neuroscience, vol. 2021, Art. no. 9910301, 2021.
[12] N. Sampathila et al., "Customized Deep Learning Classifier for Acute Lymphoblastic Leukemia Detection," Healthcare, vol. 10, no. 11, 2022.
[13] B. Elsayed et al., "Deep Learning Approaches for Leukemia Diagnosis," Diagnostics, vol. 13, 2023.
[14] F. Talaat et al., "Machine Learning-Based Leukemia Detection System," Multimedia Tools and Applications. Springer, 2024.
[15] A. Kasim et al., "Hybrid Deep Learning Model for Leukemia Classification," Scientific Reports, vol. 15, 2025.
[16] D. Muduli et al., "Comparison of Deep Learning Models for Leukemia Detection," Biomedical Signal Processing and Control, vol. 96, 2025.
[17] A. Abhishek et al., "Automated Leukemia

Detection Using Transfer Learning," Biomedical Signal Processing and Control, vol. 82, 2023.
[18] Scikit-learn Developers, Scikit-learn Documentation: Machine Learning in Python.
[19] TensorFlow Developers, TensorFlow Documentation: Deep Learning Framework.
[20] Pallets Projects, Flask Documentation: Flask Web Framework.