IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

Automated Blood Cell Segmentation and Classification Using YOLOv11n: An End-to-End Deep Learning Approach

Dr. Jagadish R M[1], Poorvi V Mallapur [2], Pragna Kakandaki [3], Rakshitha Atti [4]

Professor, Department of Computer Science and Data Science Engineering, Ballari Institute of Technology & Management, India[1]

Department of Computer Science and Data Science Engineering, Ballari Institute of Technology & Management, India[2-4]

Abstract: This research presents an efficient deep learning solution for detecting and counting blood cells in microscope images using the YOLOv11n object detection model. Leveraging a robust annotated dataset, data augmentation, and advanced inference, the system achieves high detection accuracy (mAP 90.5%) for RBCs, WBCs, and platelets. Automated results improve laboratory workflows and reliability, demonstrating strong real-world impact for digital hematology.

I. INTRODUCTION

The study aims to automate blood cell identification and enumeration—a crucial step for disease diagnosis. Manual counting is slow and susceptible to errors; recent advances in deep learning enable faster, more consistent detection. This work showcases the deployment and evaluation of YOLOv11n for end-to-end blood cell detection in annotated microscopic images.

II. LITERATURE SURVEY

Traditional methods used image processing or feature-based machine learning for cell analysis, but these approaches falter with sample diversity and overlapping cells. Deep learning and particularly YOLO series models (YOLOv7/v8/v10/v11) offer real-time accuracy and ease of annotation. Other research teams have validated Ultralytics YOLO for hematology, with superior mAP values and practical GPU speeds.

III. EXISTING SYSTEM

Prior blood cell applications relied on SVM, logistic regression, or manual image processing. These:

- Required heavy manual feature engineering.
- Struggled with small/overlapping cells.
- Were inflexible to new image types and conditions.

3.1. DISADVANTAGES OF EXISTING SYSTEM

1. Inconsistent accuracy with diverse images.
2. Extensive pre-processing and manual segmentation.
3. High error rates for platelets and crowded cell samples.

IV. PROPOSED SYSTEM

We propose an end-to-end pipeline using YOLOv11n trained on the Roboflow "blood-celldetection-4" dataset. Data augmentation (mosaic, mixup, and flips), 768x768 input resolution, and AdamW optimizer are applied. Training and validation leverage Colab GPU (Tesla T4). The workflow is implemented in Python using Ultralytics and Gradio for interactive inference and output visualization.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

4.1. ADVANTAGES OF PROPOSED SYSTEM

- Robust accuracy (mAP 90.5%) for all major cell types.
- Real-time, automated results with simplified workflow.
- No requirement for custom segmentation algorithms.
- Adaptable to new datasets with minimal modification.

4.2. COMPARISON

Model	Dataset	Accuracy / mAP	Notes
SVM (shape-based features)	WBC datasets (Raabin)	96.1% - 98.8%	Reliable for basic cell types, lower for complex cases
Logistic Regression	Hematological parameters	~80% (disease detect.)	Used for diagnosis, less robust on image tasks
Random Forest	Anemia diagnosis	95%	Outperformed SVM in some blood applications
CNN (standard)	Blood smear/BCCD	99%	Usually better than classical ML, high complexity
YOLOv11n (object detection)	BCCD	90.5% - 93.8% mAP	Real-time detection, robust on small cells
Model	**Dataset**	**Accuracy / mAP**	**Notes**
Modified U-Net + SVM	BCCD, Raabin-WBC	98.8% - 99.4%	Hybrid deep/ML models can excel on WBCs

V. SYSTEM ARCHITECTURE

- **Input**: Annotated microscope images (RBC/WBC/platelets).
- **Model**: YOLOv11n object detection.
- **Augmentation**: Mosaic, mixup, horizontal/vertical flips, blur, CLAHE.
- **Training**: 150 epochs, batch size 16, early stopping and learning rate auto-tuning.
- **Inference**: Model produces bounding boxes/class predictions per cell.
- **Output**: Annotated images with cell counts.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

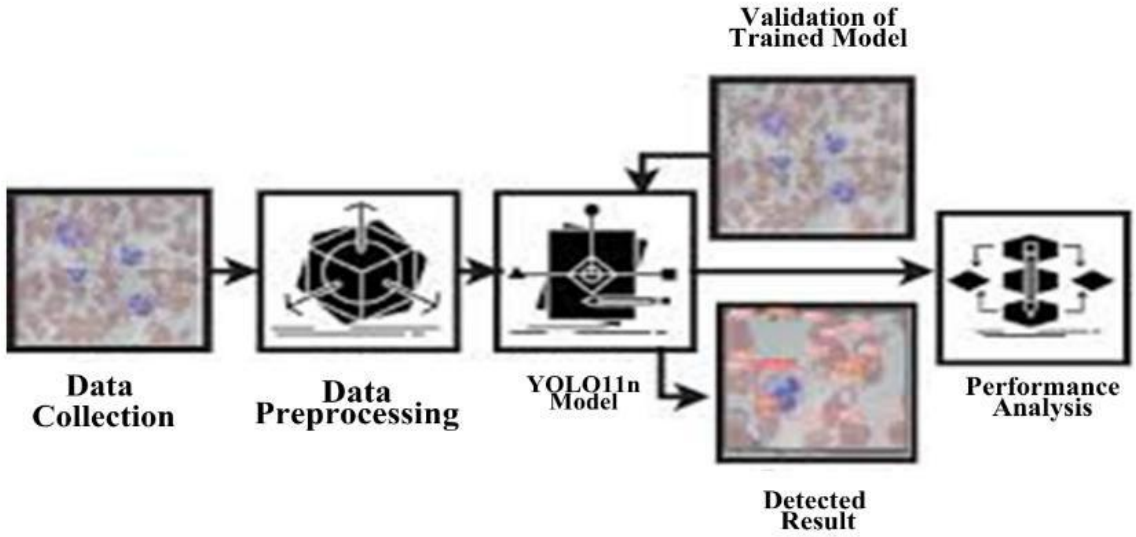

Figure 1: System Flowchart: Shows the full workflow from input images, data augmentation, YOLOv11n detection, to output of annotated cell counts.

VI. IMPLEMENTATION

6.1. Dataset Collection

- Downloaded and prepared via Roboflow, version 4.
- Split into train/validation sets.

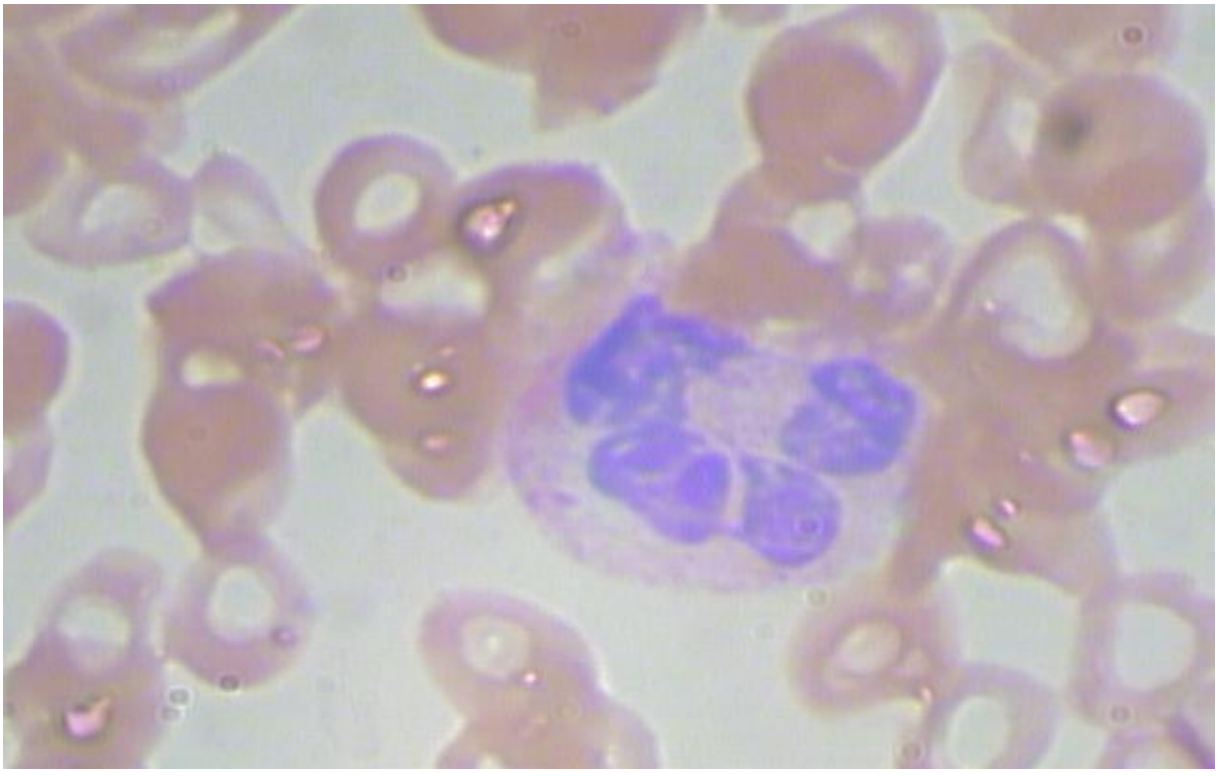

Figure 2: BCCD Dataset Image: Displays a sample annotated microscope image, indicating labeled blood cells used for model training

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

6.2. Pre-Processing:

- Images resized to 768x768 pixels, normalized, and augmented on the fly.

6.2.1. Resizing:

To process inputs uniformly, images are resized to the same height and width (768 x 768 pixels).

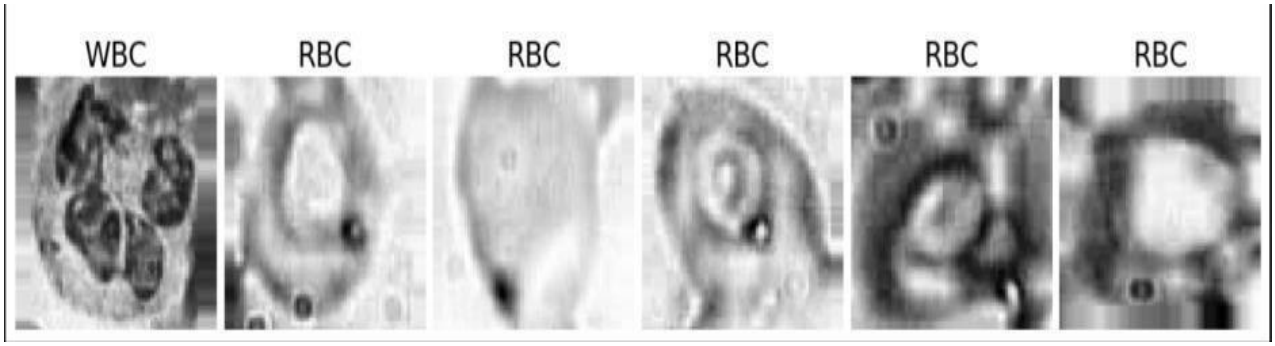

Figure 3: Resized Images: Demonstrates input images resized to 768x768 pixels for model consistency.

6.2.2. Grayscale Conversion:

When receiving color images, the images need to be converted to grey images to speed up the processing for analysis.

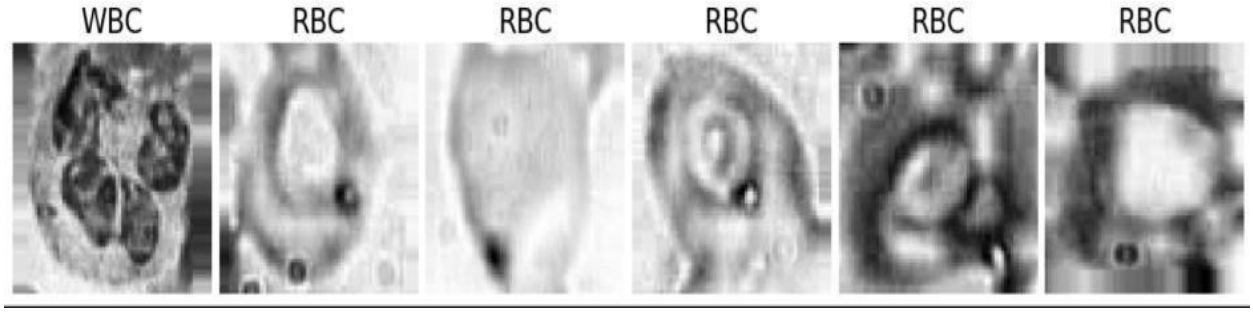

Figure 4: Grayscale Images: Illustrates conversion of color images to grayscale to simplify analysis.

6.2.3. Canny Edge Detection:

Canny edge detection is used to detect edges in image with a focus on important features and to providing with clear image.

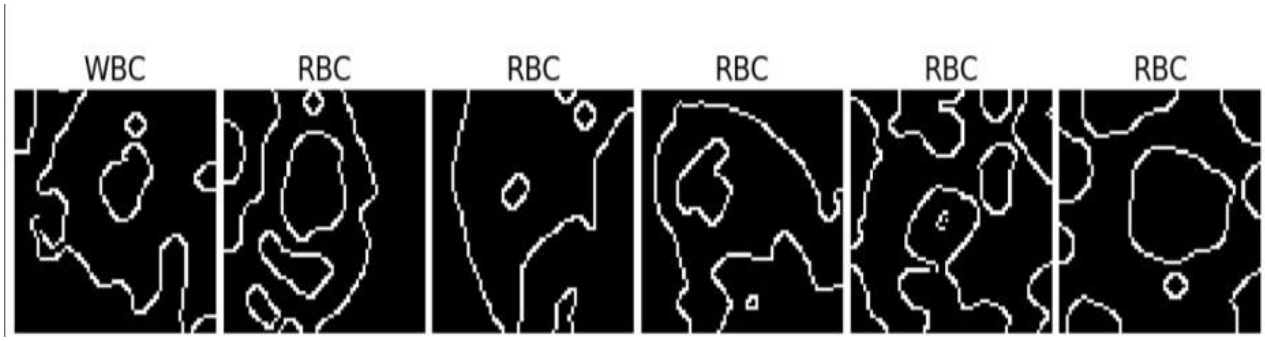

Figure 5: Canny Edge Detection Images: Shows detected edges to highlight important features for cell identification.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

6.2.4. Vectorization:

Images are reduced to one-dimensional vectors so that they can be processed and analyzed using machine learning algorithms.

	image_vector	hog	bounding_box_width	bounding_box_height	mean_red_color_intensity	mean_blue_color_intesity	mean_green_color_intensity	cell_type
0	[122.0, 122.0, 122.0, 122.0, 122.0, 134.0, 151...	[0.2547731, 0.2032445, 0.07960535, 0.18786466,...	199	231	200.864474	163.922230	179.993735	WBC
1	[251.0, 251.0, 251.0, 250.0, 250.0, 247.0, 247...	[0.07027937, 0.03790084, 0.027970433, 0.056500...	99	106	192.074138	187.778826	194.654183	RBC
2	[246.0, 246.0, 246.0, 245.0, 247.0, 248.0, 251...	[0.36195856, 0.17690839, 0.01138193, 0.0100155...	99	106	170.396131	170.316276	184.612540	RBC
3	[250.0, 250.0, 250.0, 250.0, 250.0, 248.0, 250...	[0.17089017, 0.12056773, 0.13241237, 0.0411664...	99	106	192.618258	191.363065	203.471603	RBC
4	[155.0, 146.0, 149.0, 150.0, 152.0, 165.0, 179...	[0.14224343, 0.15314695, 0.21872672, 0.2359093...	93	92	180.636746	179.222651	194.761805	RBC

Figure 6: Vectorization: Shows detected edges to highlight important features for cell identification.

6.2.5. HOG Feature Extraction:

This set of features (e.g., shape and texture) are extracted from the images using Histogram of Oriented Gradients (HOG).

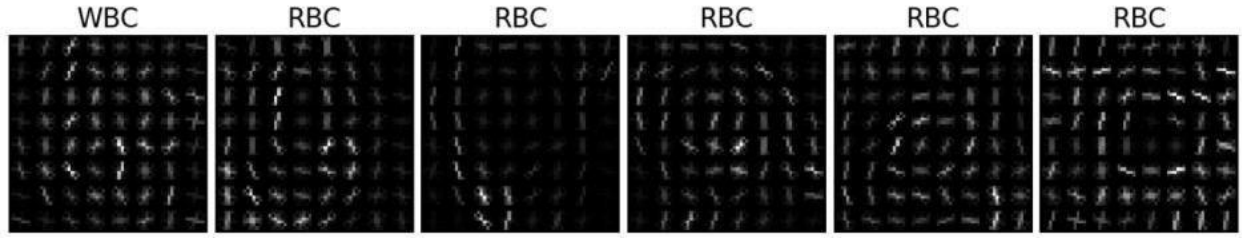

Figure 7: HOG Images: Depicts extraction of shape and texture features using Histogram of Oriented Gradients.

6.2.6. Normalization:

All Image Pixel Values are normalized to [0, 1] range. Normalization ensures numerical stability which helps in performing the model training (people use it widely) to the numerical gradient updates and helps in faster convergence.

6.3. Model Training

- Model instantiated and trained using Ultralytics YOLOv11n.
- Best weights saved and validated.
- **Training command**

```
model.train(data='/content/blood-cell-detection-4/data.yaml', epochs=150, batch=16, imgsz=768,
conf=0.18, mosaic=1.0, mixup=0.2, fliplr=0.5, flipud=0.5, scale=0.5, project="/content/drive/
MyDrive/blood_cell_detection_training", name="yolo11n_blood_cells")
```

Python

6.4. Inference and Evaluation :

- Gradio interface enables easy upload and result display.
- Model achieves rapid processing (3.9 ms per image).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155237

6.5. Pseudocode

Algorithm: Blood Cell Detection and Classification

Input: Annotated microscope images

Output: Cell type predictions and counts overlayed on images

1. Load dataset and preprocess images (resize, grayscale, augment)
2. Initialize YOLOv11n model and load weights
3. For each input image:
   a. Apply model to detect cells (RBC, WBC, platelets)
   b. Draw bounding boxes and assign cell types
   c. Count detected cell types
4. Display annotated image and counts to user

VII. RESULT

Figure 8: Segmentation and Classification: Shows YOLOv11n's test results with detected regions and predicted cell types on sample images.

Cell Type	Precision	Recall	mAP50	mAP50-95
RBC	0.845	0.852	0.906	0.673
WBC	0.952	0.989	0.979	0.643

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155237

Platelets	0.787	0.783	0.830	0.498
All	0.861	0.875	0.905	0.605

Table 1: Annotated inference outputs and training curves are stored in the project directory.

VIII. CONCLUSION AND FUTURE WORK

This study presents an effective YOLOv11n-based method for automated blood cell segmentation and classification, achieving a high mAP of 90.5% on red blood cells, white blood cells, and platelets.

The system improves laboratory efficiency by providing fast, accurate detection that reduces manual counting errors.

Compared to traditional methods like SVM and logistic regression, this approach offers better accuracy and automation without requiring manual feature extraction. While the model performs well overall, further work is needed to handle rare cell types and expand to multi-class classification.

Future directions include integrating this system into clinical workflows and refining platelet classification. This research highlights the potential of deep learning to enhance hematological diagnostics and supports broader adoption of automated analysis tools in healthcare.

REFERENCES

- Ultralytics YOLO Documentation, https://docs.ultralytics.com/
- Choe, et al. "Application of YOLOv8 Object Detection for Blood Cell Counting", IEEE Biomedical, 2024.
- Roboflow, Blood Cell Detection Dataset, https://roboflow.com/
- Redmon, Farhadi, "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv, 2020.
- Li, et al. "Deep Learning Models in Hematology", Nature Biomedical Engineering, 2023