MineDL: Class-Rarity Adaptive Ensemble Learning for Blood Cell Detection Using YOLO and Faster R-CNN

Dr. M. K. Jayanthi Kannan[1], Gowri Nair[2]
[1]*Professor, School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114.*
[2]*PG Students, School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114.*

***Abstract*—Automated blood cells detection helps with the consistent analysis of blood smear images. Manual examination is quite time-consuming. In this project, we propose a deep learning ensemble framework that combines three object detection models for detecting mainly three cells – White Blood Cells (WBC), Red Blood Cells (RBC), and Platelets. The images used here are from the TXL-PBC dataset of 1260 images that are already annotated. The models in the ensemble are YOLOv8n, YOLOv11n, and Faster R-CNN. The predictions of these models were combined using Weighted Boxes Fusion (WBF) as a baseline and a Class Rarity Adaptive Weighted Boxes Fusion (CRA-WBF). The CRA-WBFcomputes fusion weights per model and per class. The models have achieved validation mAP50 of 98.85%, 98.51%, and 96.34%. On the test set, WBF achieved an overall mAP50 of 97.409% and CRA-WBF 97.411%.**

***Index Terms*—Blood Cell Detection, Deep Learning, Object Detection, YOLO, Faster R-CNN, Weighted Boxes Fusion, Ensemble Learning, Medical Image Analysis.**

I. INTRODUCTION

Blood cell analysis is an important diagnostic procedure used to detect various types of infections and diseases like anemia, leukemia and immune disorders. The counts of the different types of cells give essential information about the patients' health. Traditionally this task is done by hematologists who examine blood smears under a microscope, which is time consuming and requires great expertise and precision. Deep learning object detectors can locate and classify many cells in one image, be it single-stage detectors such as YOLO and two stage-detectors such as Faster R-CNN. Blood smear images are also very imbalanced as red blood cells far outnumber white blood cells and platelets. Combining several detectors using WBF can improve reliability, but standard WBF only applies one fixed weight per model across all classes.In this work the three detectors are evaluated on TXL-PBC dataset and uses a class aware variant which computes the weights per model and per class.

II. LITERATURE REVIEW

Kouzehkanan et al. introduced Raabin-WBC, a large, free-access dataset of white blood cells constructed from peripheral blood films, containing bounding box annotations for approximately 40,000 white blood cells and artifacts, verified through dual-expert labeling. The dataset provides an important resource for training detection models on real blood smear data. However, the dataset's free-detection and cropped classification subsets are easily confused, and the severe imbalance present across WBC subtypes remains unaddressed by the dataset itself. [1] TXL-PBC was developed to correct annotation errors present in several existing blood cell datasets by merging BCCD, BCDD, PBC, and Raabin-WBC into a single, standardized detection dataset of 1,260 images with 18,143 bounding box annotations. The dataset was benchmarked against six detector architectures, providing a reliable, ready-to-use resource for detection research. A limitation of this dataset is that white blood cells are represented as a single class, reducing the subtype granularity available for finer-grained imbalance analysis. [2] Mao et al. proposed DWS-YOLO, a lightweight blood cell detector incorporating an enhanced attention mechanism, depth wise separable convolutions, and improved Soft-NMS to better handle overlapping

cells, a known limitation of standard non-maximum suppression. The model was evaluated on BCCD and demonstrated generalization to Raabin-WBC. However, the study considered only a single detection model and did not address class imbalance directly. [3]

A two-stage detection framework combining an improved YOLOv7 detector with an EfficientNetv2 classifier was proposed for blood cell detection and classification, incorporating a balanced cross-entropy loss specifically to address sample imbalance. The approach was evaluated across multiple datasets, including BCCD, LDWBC, LISC, and Raabin. While effective at addressing imbalance during training, the method handles imbalance only at the training stage and does not consider imbalance during any subsequent ensembling or fusion process. [4] Tarimo et al. proposed WBC YOLO-ViT, combining YOLOv5 for detection with a Vision Transformer for fine-grained classification across up to nineteen WBC subtypes. The study demonstrated strong performance on fine-grained subtype discrimination. However, the inclusion of a transformer component increases computational cost, and the approach relies on a single detection pipeline without ensembling [5]. A comprehensive benchmarking study of the YOLOv11 architecture family evaluated model variants ranging from Nano to XLarge for peripheral blood cell detection using transfer learning from COCO-pretrained weights. The study provides a systematic comparison of accuracy against model size and computational cost. However, only single models were benchmarked, without consideration of ensembling or class imbalance handling [6].

Momenian and Shojaedini proposed a hybrid ConvNextV2-Swin Transformer architecture combined with R3GAN-based synthetic data balancing to address severe class imbalance in WBC classification, evaluated on the Raabin dataset, where certain classes, such as basophils, are represented by as few as 301 samples relative to substantially larger majority classes. The approach directly targets imbalance through GAN-based oversampling and achieved strong classification accuracy. However, the study addresses classification rather than detection, and synthetic oversampling introduces a risk that generated samples may not fully reflect real cellular morphology [7]. WBCBench 2026 was proposed as a standardized benchmark jointly evaluating fine-grained WBC morphology, severe class imbalance, and robustness to acquisition variability, explicitly identifying the absence of a benchmark that addresses these dimensions together in existing literature. The benchmark establishes a valuable evaluation protocol for future research. As a benchmark and challenge paper, however, it does not itself propose a novel detection or fusion method [8]. A saliency-guided attention framework, SG-CLDFF, was proposed for WBC classification and segmentation, combining saliency maps with CNN feature fusion to improve localization precision. The approach demonstrates the value of attention-guided feature extraction for blood cell analysis. The study is focused on classification and segmentation rather than multi-model detection ensembling, and the added attention mechanisms increase computational overhead [9]. Weighted Boxes Fusion was introduced by Solovyev et al. as a method for ensembling bounding box predictions from multiple object detection models, using confidence-weighted averaging rather than discarding overlapping boxes as in conventional non-maximum suppression. The method has been widely adopted, including in winning solutions for the COCO and Open Images detection challenges, and is model-agnostic. A key limitation, directly relevant to the present study, is that the method assigns a single scalar weight to each contributing model, applied uniformly across all object classes, without accounting for class-specific differences in model reliability [10].Terzi evaluated an ensemble of nine independently trained object detection models on brain MRI, applying bounding-box fusion across four different ensembling strategies and reporting gains of up to ten percent in overall mAP and eighteen percent in per-class AP compared with the best individual model. This study provides strong empirical evidence for the value of detector ensembling in medical imaging. However, the work was not applied to blood cell detection, and fusion weighting remained per-model rather than per-class [11]. An ensemble strategy developed for the RIVA cervical cytology challenge combined three independently trained YOLO models using Weighted Boxes Fusion, with each model weighted according to its own validation mAP. The approach achieved a substantial improvement over the best individual model on the challenge test set. This study represents a close methodological analog to the present work,

applied within a related microscopy-based cell detection domain, though it similarly retains uniform per-model weighting without class-specific adaptation [12].

Table 1. Literature Survey

Authors	Year	Methodology	Major Contribution	Limitation
Kouzehkanan et al. [1]	2021/ 2022	Dual-expert bounding box annotation of blood films	Large-scale, expert-verified blood cell detection dataset	Severe WBC subtype imbalance unaddressed; subsets easily confused
TXL-PBC [2]	2025	Semi-automatic annotation, multi-dataset merging	Curated, standardized detection dataset with 6 published baselines	WBC collapsed into a single class, reduced subtype granularity
Mao et al. [3]	2024	YOLO with attention, Soft-NMS	Lightweight detector with improved overlap handling	Single-model only, no imbalance handling
Improved YOLOv7 + EfficientNetv2 [4]	2025	Two-stage detection with balanced cross-entropy	Explicit imbalance handling via training-time loss weighting	Imbalance addressed only at training, not fusion
Tarimo et al. [5]	2024	YOLOv5 + Vision Transformer	Fine-grained classification of up to 19 WBC subtypes	High computational cost, no ensembling
YOLOv11 Benchmark [6]	2025	Transfer learning across YOLOv11 variants	Systematic size-vs-performance benchmark	No ensembling or imbalance handling
Momenian & Shojaedini [7]	2025	ConvNextV2-Swin + R3GAN balancing	Directly targets severe imbalance via GAN oversampling	Classification only, not detection
WBCBench 2026 [8]	2026	Benchmark suite with robustness testing	Identifies gap in joint imbalance/robustness benchmarking	Benchmark paper, not a novel method
SG-CLDFF [9]	2025	Saliency-guided attention + CNN fusion	Improved localization via saliency guidance	Classification/segmentation only, added compute cost
Solovyev et al. [10]	2021	Weighted Boxes Fusion	Confidence-weighted box averaging ensembling method	Single scalar weight per model, uniform across all classes
Terzi [11]	2023	9-detector ensemble, bounding-box fusion	Up to 18% per-class AP gain from ensembling in medical imaging	Not applied to blood cells; per-model, not per-class, weighting
RIVA Cervical Cytology Ensemble [12]	2026	3-model YOLO ensemble, mAP-weighted WBF	29% improvement over best individual model	Still per-model-only weighting, no class adaptation
Tertytchny et al. [13]	2024/ 2025	MIP-based per-classifier, per-class weighting	First precedent for per-class ensemble weighting	Classification only, not applied to detection/fusion
Masood & Taj [14]	2026	Weighted ensemble loss with diversity enhancement	Joint imbalance and diversity handling at training time	Training-time only, no inference-stage fusion mechanism
Lin et al. [15]	2021	Importance-aware balanced group softmax	Addresses severe imbalance in blood smear detection	Softmax-level only, no multi-detector ensembling

An optimal ensemble weighting approach based on Mixed-Integer Programming was proposed for rare event detection in imbalanced multi-class datasets, optimizing a distinct weight for each classifier-class pair rather than a single weight per classifier. This work establishes an important precedent for class-specific ensemble weighting. However, the method was developed for classification tasks and has not been applied to bounding-box fusion in object detection [13]. WELDE proposed a unified, weighted ensemble loss function with diversity enhancement, jointly addressing class imbalance and model diversity during the training of ensemble members for imbalanced medical image classification. The approach demonstrates the value of addressing imbalance and diversity together rather than separately. As a training-time loss modification, however, it does not provide a mechanism for adapting predictions during the inference-time fusion stage [14]. Lin et al. addressed severe class imbalance in malaria cell detection from blood smear microscopy using an importance-aware balanced group softmax, modifying the classification head to balance predictions between rare and common classes. This work shares the same imaging domain

and core imbalance problem addressed by the present study. The approach, however, operates at the classification-head level within a single model and does not involve ensembling or fusion of multiple detectors [15].

III. PROJECT FUNCTIONAL MODULES IMPLEMENTATION

The project is structured around several functional modules designed to ensure a seamless and engaging user experience:

- Dataset Collection and Preparation: The TXL-PBC dataset of 1,260 blood smear images with WBC, RBC, and Platelet bounding boxes is used, pre-divided into 882 training, 252 validation, and 126 test images.
- Data Preprocessing: YOLO-format annotations are used directly for the YOLO models and converted to absolute [x1, y1, x2, y2] coordinates for Faster R-CNN.
- Individual Model Training: YOLOv8n, YOLOv11n, and Faster R-CNN (ResNet-50 FPN) are trained independently for 50 epochs with pretrained weights.
- Prediction Extraction: Boxes, confidence scores, and labels from all three models are generated on the test set and normalized to a common format.
- Standard Weighted Boxes Fusion: The predictions are fused with a single fixed weight per model and an IoU threshold of 0.5.
- CRA-WBF: WBF is applied separately for each class using weights computed per model and per class from class-specific validation AP50 and class frequency.
- Evaluation: Per-class AP50 and mAP50, bootstrap resampling, and prediction agreement analysis are computed.

IV. METHODOLOGY

Three detectors were trained independently using a transfer learning mechanism. YOLOv8n and YOLOv11n were fine-tuned from COCO for 50 epochs, at 640x640 resolution. Faster R-CNN with a ResNet – 50 FPN was initialised from COCO-pretrained TorchVision weights. It was trained for 50 epochs with SGD, where the learning rate was 0.005, momentum was 0.9, and weight decay was 0.0005; a step scheduler was used to reduce the learning rate every 10 epochs. The predictions from all three models on the test set were combined in two ways. The Standard Weighted Boxes Fusion (WBF) will pool all the boxes, cluster them by IoU, and will fuse each cluster by confidence-weighted averaging, with one fixed weight per model for all classes. The CRA-WBF, a class variant, replaces the per-model weight with a weight for each model-class pair.

$$w(i, c) = AP(i, c) \times (1 + \alpha \times ((f_max / f_c) - 1) / f_max)$$

where AP (i, c) is the validation AP50 of model i for class c, f_c is the number of validation instances of class c (257 WBC, 3,368 RBC, 112 Platelets), f_max is the count of the most frequent class, and α (set to 1.0) controls the strength of the rarity adjustment. WBF is run separately for each class with its class-specific weights and the results are merged. The performance was measured using a per class AP50 and mAP50, AP50 for the yolo models came from the Ultralytics validation routine, while Faster R-CNN and both ensembles are evaluated with torchmetrics at IoU 0.50. Bootstrap resampling on test set accessed the variability and prediction agreement measured diversity between detectors.

V. CODE IMPLEMENTATION

The system is implemented in Python using PyTorch, Ultralytics, TorchVision, torch metrics, and the ensemble-boxes library in Google Colab Pro with a Tesla T4 GPU. YOLOv8n and YOLOv11n are trained through the Ultralytics API, and Faster R-CNN is trained with a custom PyTorch dataset class that converts YOLO labels to pixel coordinates. Predictions from all models are saved, fused with Standard WBF and CRA-WBF, and evaluated with torch metrics.

Fig 5.1 Import Libraries and Dataset

```
!git clone https://github.com/lugan113/TXL-PBC_Dataset.git

Cloning into 'TXL-PBC_Dataset'...
remote: Enumerating objects: 4317, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 4317 (delta 4), reused 2 (delta 2), pack-reused 4308 (from 2)
Receiving objects: 100% (4317/4317), 89.39 MiB | 36.12 MiB/s, done.
Resolving deltas: 100% (28/28), done.

from google.colab import drive
drive.mount('/content/drive')
!cp -r /content/TXL-PBC_Dataset /content/drive/MyDrive/TXL-PBC_Dataset

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Fig. 1. Yolov8n Training

```
!pip install ultralytics -q
from ultralytics import YOLO

model = YOLO('yolov8n.pt')
results = model.train(
    data='/content/TXL-PBC_Dataset/TXL-PBC/data.yaml',
    epochs=50,
    imgsz=640,
    project='/content/drive/MyDrive/blood_cell_project',
    name='yolov8n_baseline'
)
```

Fig. 2. Yolov8n Results

```
metrics = model1.val(data='/content/drive/MyDrive/TXL-PBC_Dataset/TXL-PBC/data.yaml')
print(metrics.box.map50)

Ultralytics 8.4.172 🚀 Python-3.13.15 torch-2.11.0+cu130 CUDA:0 (Tesla T4, 14913MiB)
Model summary (fused): 73 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
WARNING ⚠️ val: Slow image access detected (ping: 1.6±1.9 ms, read: 9.3±8.1 MB/s, size: 23.6 KB). Use loca
val: Scanning /content/drive/MyDrive/TXL-PBC_Dataset/TXL-PBC/labels/val.cache... 252 images, 0 backgrounds
val: /content/drive/MyDrive/TXL-PBC_Dataset/TXL-PBC/images/val/0ac132c270355979b686c0801e558849.png: 15 du
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━
                   all        252       3737      0.967      0.969      0.988       0.86
                   WBC        250        257      0.994      0.996      0.994      0.901
                   RBC        252       3368      0.947      0.974       0.99      0.902
             Platelets         64        112      0.959      0.938      0.982      0.778
Speed: 5.6ms preprocess, 6.8ms inference, 0.0ms loss, 5.3ms postprocess per image
Results saved to /content/runs/detect/val-6
0.988482935905374
```

Fig. 3. Yolov11n Training

```
from ultralytics import YOLO

model2 = YOLO('yolo11n.pt')
results2 = model2.train(
    data='/content/drive/MyDrive/TXL-PBC_Dataset/TXL-PBC/data.yaml',
    epochs=50,
    imgsz=640,
    project='/content/drive/MyDrive/blood_cell_project',
    name='yolo11n_baseline'
)
```

Fig. 4. Yolov11n Results

```
metrics2 = model2.val(data='/content/drive/MyDrive/TXL-PBC_Dataset/TXL-PBC/data.yaml')

class_names = ['WBC', 'RBC', 'Platelets']
for i, name in enumerate(class_names):
    print(f"{name}: AP50 = {metrics2.box.ap50[i]:.4f}, mAP50-95 = {metrics2.box.maps[i]:.4f}")
```

Fig. 5. Faster R-CNN Training

```
import os

NUM_EPOCHS = 50  # Faster R-CNN typically converges in fewer epochs than YOLO

for epoch in range(NUM_EPOCHS):
    model.train()
    epoch_loss = 0.0

    for images, targets in train_loader:
        images = [img.to(device) for img in images]
        targets = [{k: v.to(device) for k, v in t.items()} for t in targets]

        loss_dict = model(images, targets)
        losses = sum(loss for loss in loss_dict.values())

        optimizer.zero_grad()
        losses.backward()
        optimizer.step()

        epoch_loss += losses.item()

    lr_scheduler.step()
```

Fig. 6 Standard wbf

```
def run_standard_wbf(image_name, all_preds, model_weights):
    boxes_list, scores_list, labels_list = [], [], []
    for model_name in ['yolov8n', 'yolov11n', 'fasterrcnn']:
        p = all_preds[model_name].get(image_name, {'boxes': [], 'scores': [], 'labels': []})
        boxes_list.append(p['boxes'])
        scores_list.append(p['scores'])
        labels_list.append(p['labels'])

    if not any(len(b) > 0 for b in boxes_list):
        return [], [], []

    boxes, scores, labels = weighted_boxes_fusion(
        boxes_list, scores_list, labels_list,
        weights=model_weights, iou_thr=0.5, skip_box_thr=0.0001
    )
    return boxes, scores, labels

ap_matrix = {
    'yolov8n':     {0: 0.9943, 1: 0.9897, 2: 0.9815},    # WBC, RBC, Platelets — AP50
    'yolov11n':    {0: 0.9936, 1: 0.9903, 2: 0.9715},    # AP50
    'fasterrcnn': {0: 0.9884, 1: 0.9660, 2: 0.9359},      # <-- fill with real AP50, see note below
}
```

Fig. 7. CRA – WBF

```
def run_cra_wbf(image_name, all_preds):
    fused_boxes, fused_scores, fused_labels = [], [], []
    for class_id in [0, 1, 2]:
        boxes_list, scores_list, labels_list, weights = [], [], [], []
        for model_name in ['yolov8n', 'yolov11n', 'fasterrcnn']:
            p = all_preds[model_name].get(image_name, {'boxes': [], 'scores': [], 'labels': []})
            idxs = [i for i, l in enumerate(p['labels']) if l == class_id]
            boxes_list.append([p['boxes'][i] for i in idxs])
            scores_list.append([p['scores'][i] for i in idxs])
            labels_list.append([p['labels'][i] for i in idxs])
            weights.append(compute_class_weight(model_name, class_id))

        if any(len(b) > 0 for b in boxes_list):
            b, s, l = weighted_boxes_fusion(boxes_list, scores_list, labels_list,
                                            weights=weights, iou_thr=0.5, skip_box_thr=0.0001)
            fused_boxes.extend(b); fused_scores.extend(s); fused_labels.extend(l)

    return fused_boxes, fused_scores, fused_labels
```

Fig. 8 Bootstrap

```
def bootstrap_eval(fused_fn, image_names, labels_dir, images_dir, n_bootstrap=10):
    class_aps = {0: [], 1: [], 2: []}
    all_fused = {img: fused_fn(img) for img in image_names}
    for _ in range(n_bootstrap):
        sample = list(np.random.choice(image_names, size=len(image_names), replace=True))
        result = evaluate_fused_predictions(all_fused, sample, labels_dir, images_dir)
        for i, cid in enumerate(result['classes'].tolist()):
            class_aps[cid].append(result['map_per_class'][i].item())
    return {cls: (np.mean(v), np.std(v)) for cls, v in class_aps.items() if v}

print("\nBootstrap — standard WBF:")
print(bootstrap_eval(lambda img: run_standard_wbf(img, all_preds, model_weights_standard),
                     test_image_names, labels_dir, images_dir_test))

print("\nBootstrap — CRA-WBF:")
print(bootstrap_eval(lambda img: run_cra_wbf(img, all_preds),
                     test_image_names, labels_dir, images_dir_test))
```

```
standard_wbf:
Overall mAP50: 0.9740949273109436
  class 0: AP = 0.9802
  class 1: AP = 0.9748
  class 2: AP = 0.9672

cra_wbf:
Overall mAP50: 0.9741061925888062
  class 0: AP = 0.9802
  class 1: AP = 0.9749
  class 2: AP = 0.9672

Bootstrap — standard WBF:
{0: (np.float64(0.9851485192775726), np.float64(0.013466798470476587)), 1: (np.float64(0.980109053850174),

Bootstrap — CRA-WBF:
{0: (np.float64(0.9891089141368866), np.float64(0.011245359709611288)), 1: (np.float64(0.980328494310379),
```

Fig. 9 Final Numbers

VI. RESULTS

The three detectors all achieved high validation AP50 (Table I). YOLOv8n gave the best mean AP50, YOLOv11n was comparable, and Faster R-CNN was lower, particularly for RBC and Platelets. Platelets were the weakest class for every model. Faster R-CNN also reached a COCO-style mAP (0.50:0.95) of 0.8040.

Table II. Validation AP50 of Individual Models

Model	WBC AP50	RBC AP50	Platelets AP50	Mean AP50
YOLOv8n	0.9943	0.9897	0.9815	0.9885
YOLOv11n	0.9936	0.9903	0.9715	0.9851
Faster R-CNN	0.9884	0.9660	0.9359	0.9634

On the 126 test images, Standard WBF and CRA-WBF performed essentially identically (Table II). The difference in overall mAP50 was 0.0000113, so the results do not support claiming an advantage for CRA-WBF.

Table III. Ensemble Performance on the Test Set

Method	WBC AP50	RBC AP50	Platelets AP50	Overall mAP50
Standard WBF	0.9802	0.9748	0.9672	0.97409
CRA-WBF	0.9802	0.9749	0.9672	0.97411

Prediction agreement was 0.9597 between YOLOv8n and YOLOv11n and 0.8065 between YOLOv8n and Faster R-CNN. The two YOLO models behave very similarly, while Faster R-CNN provides more diverse predictions, which supports combining architecturally different detectors.The bootstrap results show higher mean AP values for CRA-WBF for all three classes. The mean AP increased from 0.9713 to 0.9772 for WBC, from 0.9751 to 0.9767 for RBC, and from 0.9807 to 0.9847 for Platelets.These bootstrap

differences are more noticeable than the single overall test-set mAP50 difference and suggest that the class-aware weighting may provide some stability across resampled test subsets.

Table IV. Bootstrap AP50 Results

Class	Standard WBF	CRA-WBF
WBC	0.9713 ± 0.0223	0.9772 ± 0.0213
RBC	0.9751 ± 0.0095	0.9767 ± 0.0075
Platelets	0.9807 ± 0.0154	0.9847 ± 0.0149

VII. CONTRIBUTION AND FINMDINGS

The framework offers several advantages. Transfer learning from pretrained weights gives strong detection with a small dataset, and the three detectors achieve high AP50 across all classes. Combining a two-stage detector with single-stage YOLO models adds measurable prediction diversity, and the fusion step is model-agnostic and requires no retraining. The per-class weighting is simple to implement on top of existing WBF. The limitations are that CRA-WBF did not show a meaningful improvement over standard WBF, and bootstrap differences lie within resampling variability, with only ten resamples used. The test set has just 126 images, which makes platelet estimates noisy, and the models were evaluated on a single dataset with only three classes. The CRA-WBF weight parameter α was not tuned. The ensemble also requires running three detectors, which increases inference cost.

VIII. CONCLUSION

This paper evaluated an ensemble framework for automated blood cell detection using YOLOv8n, YOLOv11n, and Faster R-CNN on the TXL – PBC dataset. All three detectors achieved high validation AP50, and Platelets were consistently the weakest class. The WBF combined their predictions into a working ensemble, while the proposed CRA-WBF produced a little improvement. Agreement analysis also confirms that Faster R-CNN adds diversity among the two similar YOLO models. This provides a very strong foundation for further study of CRA-WBF on a much larger dataset. In future work, we aim to use larger datasets with more blood cell categories like Raabin-WBC, where class imbalance may be more severe, and Class Aware RBF may matter way more. Other parameters like the weighting parameter α and the weight formulation can be tuned, and a larger, paired bootstrap can be used to test significance. Other architectures, including transformer-based detectors and other ensemble methods, can be compared. Finally, the system can also be validated on clinical data that is collected individually and deployed as an assistant-based application for specialists like haematologists for first-level diagnosis

REFERENCES

[1] Mousavi Kouzehkanan, S.-Z., et al. (2021). Raabin-WBC: A large free access dataset of white blood cells from normal peripheral blood. bioRxiv. https://doi.org/10.1101/2021.05.02.442287

[2] Gan, L., Li, X., & Wang, X. (2025). A curated and re-annotated peripheral blood cell dataset integrating four public resources. Scientific Data. https://doi.org/10.1038/s41597-025-05980-z

[3] G. D. K., Singh, M. K., & Jayanthi, M. (Eds.). (2016). Network security attacks and countermeasures. IGI Global. https://doi.org/10.4018/978-1-4666-8761-5

[4] Mao, Y., Zhang, H., Wu, W., Gao, X., Lin, Z., & Lin, J. (2024). DWS-YOLO: A lightweight detector for blood cell detection. Applied Artificial Intelligence, 38. https://doi.org/10.1080/08839514.2024.2318673

[5] Kannan, J. K., Aaijaz, N., Mani, K. G., & Tewari, V. (2025). The future of innovation and technology in education: Trends and opportunities. S&M Publications.

[6] R. M., B., & M. K., J. K. (2023). Intrusion detection on AWS cloud through hybrid deep learning algorithm. Electronics, 12, Article 1423. https://doi.org/10.3390/electronics12061423

[7] Wang, X., Pan, G., Hu, Z., & Ge, A. (2025). A two-stage blood cell detection and classification algorithm based on improved YOLOv7 and EfficientNetV2. Scientific Reports, 15. https://doi.org/10.1038/s41598-025-91720-7

[8] Tarimo, S. A., Jang, M.-A., Ngasa, E. E., Shin, H. B., Shin, H., & Woo, J. (2024). WBC YOLO-ViT: Two-way, two-stage white blood

cell detection and classification with a combination of YOLOv5 and vision transformer. Computers in Biology and Medicine, 169, Article 107875. https://doi.org/10.1016/j.compbiomed.2023.107875

[9] Kannan, J. K., Kumar, S., Kalaivaani, P. T., & Tripathi, G. (2025). Artificial intelligence and blockchain technology for human resource management. Scientific International Publishing House.

[10] Jayanthi, M. K. (2017). Strategic planning for information security—DID mechanism to befriend the cyber criminals to assure cyber freedom. In Proceedings of the 2017 2nd International Conference on Anti-Cyber Crimes (ICACC) (pp. 142–147). IEEE. https://doi.org/10.1109/Anti-Cybercrime.2017.7905280

[11] Abou Ali, M., et al. (2025). Comprehensive benchmarking of YOLOv11 architectures for scalable and granular peripheral blood cell detection. arXiv. arXiv:2509.24595.

[12] Shukla, S. K., Dwivedi, U., Kannan, J. K., & Sarvani, C. (2024). Python for data analytics: Practical techniques and applications. JSR Publications.

[13] Jain, P., Rajvaidya, I., Sah, K. K., & Kannan, J. (2022). Machine learning techniques for malware detection—A research review. In Proceedings of the 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1–6). IEEE. https://doi.org/10.1109/SCEECS54111.2022.9740918

[14] Momenian, M., & Shojaedini, S. V. (2025). Advancing WBC classification: A hybrid ConvNeXtV2-Swin Transformer framework with R3GAN data balancing and CLAHE preprocessing. Journal of Imaging Informatics in Medicine. https://doi.org/10.1007/s10278-025-01740-y

[15] B. R. M., M. M. V., & J. K. M. K. (2021). Performance analysis of bag of password authentication using Python, Java and PHP implementation. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES) (pp. 1032–1039). IEEE. https://doi.org/10.1109/ICCES51350.2021.9489233

[16] Kannan, M. K. J., & Shree Nee, T. R. (2025). Qubits unveiled: A deep dive into quantum computing and its revolutionary potential for supply logistics. In P. Gaba, A. Panwar, V. Jain, & R. Kannan (Eds.), Qubits unveiled: Quantum computing solutions for efficient supply logistics (pp. 273–293). Nova Science Publishers. https://doi.org/10.52305/WSXW8884

[17] Tian, X., et al. (2026). WBCBench 2026: A challenge for robust white blood cell classification under class imbalance. In Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI).

[18] Gashti, M. Z., Mohammadpour, M., & Farjamnia, G. (2025). SG-CLDFF: A novel framework for automated white blood cell classification and segmentation. arXiv. arXiv:2510.17278.

[19] Naik, H., & Kannan, M. K. J. (2020, December 1). A survey on protecting confidential data over distributed storage in cloud. SSRN. https://ssrn.com/abstract=3740465

[20] Kavitha, E., Tamilarasan, R., Baladhandapani, A., & Kannan, M. K. J. (2022). A novel soft clustering approach for gene expression data. Computer Systems Science and Engineering, 43(3), 871–886. https://doi.org/10.32604/csse.2022.021215

[21] Solovyev, R., Wang, W., & Gabruseva, T. (2021). Weighted boxes fusion: Ensembling boxes from different object detection models. Image and Vision Computing, 107, Article 104117. https://doi.org/10.1016/j.imavis.2021.104117

[22] Terzi, R. (2023). An ensemble of deep learning object detection models for anatomical and pathological regions in brain MRI. Diagnostics, 13(8), Article 1494. https://doi.org/10.3390/diagnostics13081494

[23] Shree Nee, T. R., Kannan, M. K. J., & Mariyappan, K. (2025). Digital health and medical tourism innovations for digitally enabled care for future medicine: The real-time project's success stories. In Navigating innovations and challenges in travel medicine

and digital health (pp. 325–344). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-8774-0.ch016

[24] Kogan, L., & Ríos, M. V. (2026). Detection and classification of (pre)cancerous cells in Pap smears: An ensemble strategy for the RIVA Cervical Cytology Challenge. arXiv. arXiv:2603.23742.

[25] Tertytchny, G., Stavrinides, G. L., & Michael, M. K. (2025). Rare event detection in imbalanced multi-class datasets using an optimal MIP-based ensemble weighting approach. Proceedings of the AAAI Conference on Artificial Intelligence, 39(19). https://doi.org/10.1609/aaai.v39i19.34300

[26] Masood, R. F., & Taj, I. A. (2026). WELDE: A weighted ensemble loss with diversity enhancement for imbalanced object classification in medical imaging. Network Modeling Analysis in Health Informatics and Bioinformatics, 15. https://doi.org/10.1007/s13721-026-00797-1

[27] Lin, C., Wu, H., Wen, Z., & Qin, J. (2021). Automated malaria cells detection from blood smears under severe class imbalance via importance-aware balanced group softmax. In Medical Image Computing and Computer Assisted Intervention (MICCAI 2021). Springer. https://doi.org/10.1007/978-3-030-87237-3_44