Journal of
Information Systems Engineering
and Business Intelligence

Vol.11, No.1, February 2025
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

BloodCell-YOLO: Efficient Detection of Blood Cell Types Using Modified YOLOv8 with GhostBottleneck and C3Ghost Modules

Mohammad Farid Naufal [1)*], **Selvia Ferdiana Kusuma** [2)]

[1)] *Universitas Surabaya, Surabaya, Indonesia*
[1)] faridnaufal@staff.ubaya.ac.id

[2)] *Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia*
[2)] selvia@pens.ac.id

Abstract
Background: Diagnosing many medical ailments, including infections, immunological problems, and hematological diseases, is a process that depends on precise as well as quick identification of blood cell. Conventional methods for blood cell identification may include skilled pathologists visually inspecting the cell under a microscope, which is a time-consuming choreography. This method is not appropriate for processing vast amounts of data, because the process is time-consuming and is prone to human mistakes.
Objective: This study aimed to improve YOLOv8 architecture, offering a more efficient and simplified model for blood cell identification. In addition, the main objective of the analysis was to reduce computational load as well as amount of parameters and still maintaining or improving detection performance.
Methods: GhostBottleneck and C3Ghost modules used in the study were included in the head and backbone of YOLOv8 architecture for improvement. All versions of YOLOv8 was subjected to the changes including n, s, m, l, and x. During the analysis, the efficacy of the recommended method was evaluated using a dataset of seven kinds of blood, namely basophil, eosinophil, lymphocyte, monocyte, neutrophil, platelets, and red blood cells (RBCs). The analysis also tested the proposed method on the well-known Blood Cell Count and Detection (BCCD) dataset, which was a common benchmark in this field, for comparing the performance. Performance of the model relating to past studies was assessed through this process.
Results: The investigation used GhostBottleneck and C3Ghost modules to reduce GFLOPS by 45.56% and the number of parameters by 76.55%. Mean average precision (mAP50) of 0.984 was achieved using recommended method. Additionally, on BCCD, the method scored 0.94 on New Cell Dataset.
Conclusion: Modifications performed to YOLOv8 design significantly increased its blood cell detection efficiency and effectiveness. The improvements showed that the changed model was suitable for real-time use in settings with constrained resources.

Keywords: Blood Cell Detection, C3Ghost, Ghostbottleneck, YOLOv8

Article history: Received 1 August 2024, first decision 17 January 2025, accepted 14 February 2025, available online 28 March 2025

I. INTRODUCTION

The classification of blood cell correctly and rapidly is a process that helps identify and treat illnesses, immune issues, and hematology diseases [1]. Quick and accurate blood cell types identification affects medical choices and patient health. Moreover, traditional methods require experts to observe by hand, whichs takes a lot of time and can cause mistakes[2], [3]. Automation and application of machine learning as well as computer vision is being studied to solve the limits [4]. These models require significant computing power in GFLOPs, which poses challenges, especially in resources-limited areas where heavy task demand more efficient processing.

Fast and reliable object identification models, particularly those based on You Only Look Once (YOLO) method, perform well in image recognition tests [5], [6], [7]. Among the models, YOLOv8 model outperforms the others, accurately identifying objects with high precision [8], [9]. Difficulties implementing this method are caused by high processing needs and large parameter counts in resource-constrained situations such as mobile devices and low-power medical equipment[10], [11], [12].

* Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print)
doi: http://dx.doi.org/10.20473/jisebi.11.1.41-52

YOLO models are great for identifying objects in image because the systems are both fast and accurate[5], [6], [7]. Among all the available types, YOLOv8 performs better in accurately finding objects using advanced technology [8], [9]. Additionally, the model needs high processing power and large number of parameters. The high demand make it hard to use the system in devices with limited resources, such as cell phones and low-power medical tools [10], [11], [12]. However, development of models that are more efficient, and still preserve significant level of detection performance is essential. Recent advances in model optimization methods have created opportunities to construct lightweight designs that may improve performance, greatly lowering computing needs. A method, based on GhostNet convolutional architecture [13] uses GhostBottleneck and C3Ghost modules to efficiently reduce the number of parameters as well as computational load in deep learning models. Combining these modules into YOLOv8 architecture will enable production of a highly modified blood cell identification model, specifically for real-time applications in resource-limited surroundings.

Previous work had shown individual potential in medical diagnostics by effectively implementing blood cell identification and changing YOLO object recognition methods. Studies on changing YOLOv5 architecture have focused on improving performance and computational economy [14], [15], [16]. In comparison, other analyses aimed at advancing YOLOv3 model to achieve comparable changes [17], [18], [19]. These analyses have led to various architectural modifications and optimization methods aimed at improving model efficiency and allowing it to be suitable for resource-limited environments. The main objectives of the changes are to shorten inference times and reduce processing requirements while maintaining detection accuracy. Following the discussion, all investigations have used BCCD dataset [20]. Three types of blood cell including red (RBCs), white (WBCs), and platelets, are identified in image from BCCD collection. Despite the dataset being beneficial, it was limited to only three object classes which signified the necessity of additional complete datasets with a greater range of blood cell types for more strong and generic detection models.

This study introduces BloodCell-YOLO, which is a modified YOLOv8 model incorporating GhostBottleneck and C3Ghost modules in its head as well as backbone to reduce computational burden and parameter count while maintaining or improving detection performance for blood cell detection. The model is evaluated on the publicly available New Cell Dataset [21], which includes seven blood cell types, namely basophil, eosinophil, lymphocyte, monocyte, neutrophil, platelets, and RBCs, using all YOLOv8 versions (n, s, m, l, x). Previous analyses using modified YOLOv5 and YOLOv3 models focused on detecting only the three cell types, improving computational efficiency as well as detection performance but limiting applicability due to the narrower range of cell types. Consequently, this analysis expands detection to a broader range of blood cell and excludes WBCs due to individual simpler detectability, addressing a critical need. BloodCell-YOLO outperforms the others because it can identify more kinds of cell, leading to better medical outcomes. This analysis evaluate the model using mean Average Precision at 50% Intersection over Union (IoU) (mAP50), along with precision, recall, parameters, and GFLOPs. These steps help the method work well in apps that have limited resources and require immediate responses. The analysis shows that BloodCell-YOLO is better for medical analysis because it accurately identifies blood cell kinds and requires less computer power.

II. Literature Review

Studies showed that Gu and Sun [15] improved YOLOv5 to AYOLOv5, increasing the spotting of blood cell in imaging. During these analyses, Transformer Encoder Block and CBAM YOLOv5 structure urged new ideas. The improvements allowed the system to identify nearby cell, even with background noise, where BCCD results showed significant increment. Moreover, mAP50 of the model score was 93.3%, and its recognition accuracy increased from 89 to 98%. The model performed better than older methods in difficult cell situations.

Mao et al. [14] upgraded YOLOv5-Nano using DWS by applying lighter blood cell monitor. During the investigation, ECAM was incorporated into DWS-YOLO model for major information processing, Depthwise Separable Convolutions (DSC) was used to reduce parameters, efficient Scylla Intersection over Union (SIoU) loss function was applied to increase learning, and improved Soft-NMS to aid accuracy. Additionally, DWS-YOLO performed well on BCCD and Raabin-WBCs datasets. The model was appropriate for real-time medical usage since it used less computer resources.

Xu et al. [17]used lightweight and efficient microscopic blood cell identification model TE-YOLOF. This YOLOF-based model collected major characteristics using EfficientNet. The model reduced parameters with DSC while maintaining accuracy. Following the process, Mish activation function increased FED model RBCs identification. In extensive trials on BCCD dataset, TE-YOLOF outperformed existing models in both accuracy and processing efficiency, allowing it to be suitable for portable devices as well as cross-domain applications.

Shakarami et al. [18] provided a fast and efficient YOLOv3 model for blood cell detection. Incorporating EfficientNet-B3 as the backbone, dilated convolutions, DSC, Swish activation function, and an improved Distance

Intersection over Union (DIoU) loss function helped this model to augment original YOLOv3. These few changes significantly reduced the parameter count and improved detection accuracy. The efficiency and low processing requirements of this paradigm fit embedded systems as well as portable devices.

Liu et al. [19] showed YOLOv3 deep-learning model improved blood cell identification. Global Average Pooling and Global Max Pooling combined in an improved Squeeze-and- Excitement (SE) modules offered ISE-YOLO model improved feature extraction and network discrimination of input features. On the BCCD dataset, the model signified increased performance with detection accuracy of 96.5% WBCs, 92.7% RBCs, and 89.6% platelets, respectively. This study provided important progress over current methods by showing the possibility of using sophisticated attention mechanisms in convolutional neural networks to improve medical picture processing.

Previous studies clearly showed that earlier investigations had significant limitation, as it relied on BCCD dataset, containing only three types of blood cell, namely WBCs, RBCs, and platelets. The low concentration of the models caused limited use in more comprehensive diagnostic settings. This work used training on a larger dataset including seven blood cell types, namely basophil, eosinophil, lymphocyte, monocyte, neutrophil, platelets, and RBCs which improved the accuracy as well as flexibility of the algorithm in detecting various cell. The analysis applied the pre-trained model, improved on BCCD dataset after BloodCell-YOLO model training on a large dataset. This method used the initial training on a diverse dataset to raise performance on the dataset, showing the potential of the model to handle specific and broad classification tasks.

III. METHODS

A. Dataset

This study used public dataset from Roboflow, more specifically "New Cell Dataset" [21]. The dataset offered a complete set of blood cell image labeled for different cell kinds, enabling study and development in automated blood cell identification as well as categorization. Building on the process, the dataset contained seven different types of blood cell including basophil, eosinophil, lymphocyte, monocyte, neutrophil, platelets, and RBCs. These blood cell types were important for the diagnosis and treatment of many medical issues such as infections, immunological problems, and hematological diseases.

Dataset used in this analysis, was divided into 75% for training and 25% for testing. The distribution guaranteed strong assessing performance of the model. In line with process, learning the patterns and properties of various blood cell types required some training, offering an objective assessment of generalizing capacity of the model. Table 1 showed the detailed distribution of New Cell Datasets, guaranteeing the experiments could be repeated easily. Figure 1 showed a sample from blood cell dataset, signifying how diverse and complex the image was for testing as well as training. This study also modified pre-trained model using BCCD public dataset, containing RBCs, WBCs, and platelet. Following the discussion, BCCD dataset was widely used in medical image analysis. As a standard benchmark for blood cell identification algorithms, this dataset enabled comparisons with existing methods. During the process, the analysis divided BCCD dataset into training, validation, and testing subsets to check the model carefully. The validation part was used for hyperparameter tuning and overfitting prevention. In addition, the training subset aided the model in learning from a varied group of instances, and testing subsection was used for last assessment. Table 2 showed the exact distribution of BCCD dataset, offering a clear perspective of image and instance count used in every step of the analysis. In the context of this investigation, strong and generalizable model was designed for blood cell identification that performed well across many datasets and situations by using both New Cell and BCCD datasets.

B. BloodCell-YOLO

The study derived BloodCell-YOLO derived from state-of- the-art YOLOv8 model. This model renowned for balancing speed and accuracy in object identification tasks [22], [23]. Figure 2 showed BloodCell-YOLO architecture during the process of this analysis. YOLOv8 was highly efficient for real-time applications, predicting bounding boxes and class probabilities in a specific evaluation by processing entire image through single neural network. Additionally, YOLOv8 had significant processing needs and numerous parameters, which restricted deployment in resource-constrained situations such as mobile devices as well as low-power medical equipment even when it performed better. GhostBottneck modules and C2f with C3Ghost replaced the conventional convolutional layers in YOLOv8, addressing these issues.

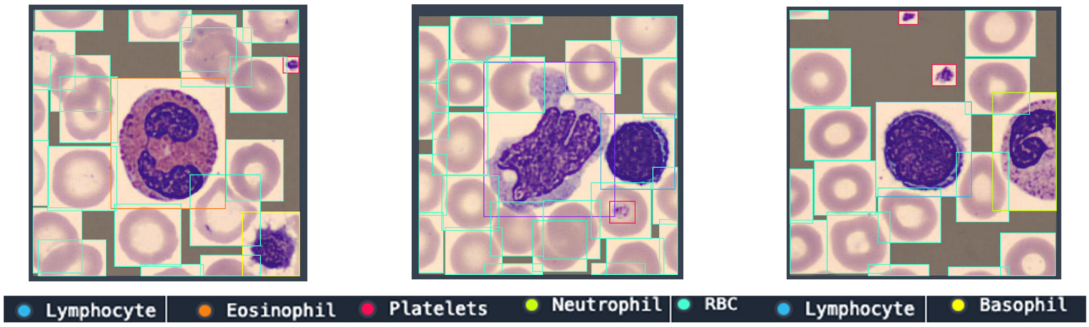

Fig. 1 The labeled categories of blood cells used in this study. Each label was color-coded to represent a specific type of blood cell, namely Basophil, Eosinophil, Lymphocyte, Monocyte, Neutrophil, Platelets, and RBC.

TABLE 1
THE DISTRIBUTION OF NEW CELLS DATASET

Class	Training		Testing	
	Images	Instances	Images	Instances
Basophil	406	412	102	103
Eosinophil	534	578	145	150
Lymphocyte	545	555	130	134
Monocyte	522	530	126	128
Neutrophil	524	544	126	130
Platelets	1329	2256	347	601
RBCs	2803	47552	708	12305

TABLE 2
THE DISTRIBUTION OF BCCD DATASET

Class	Instances		
	Train	Val	Test
RBCs	2938	819	398
WBCs	263	72	37
Platelets	249	76	36

1) GhostBottleneck

GhostBottleneck, an innovation introduced by GhostNet [13], generated more feature maps from fewer convolution operations, effectively reducing computational load and number of parameters. Figure 2 showed architecture of GhostBottleneck modules during the process of the study. The primary advantage of GhostBottleneck was its ability to maintain high representational power while significantly lowering computational complexity, which was achieved through a two-step process. The first step included generating intrinsic feature maps using standard convolutions. During the process, the initial input passed through GhostConv layer, which applied convolution operation to extract essential features from input data. The output of this layer retained the critical information needed for accurate representation. In the second step, the feature maps produced by initial convolution were further processed by a series of cheap linear operations to generate additional feature maps. These operations were less computationally intensive compared to standard convolutions. The processed feature maps then pass through another GhostConv layer during the process. The second GhostConv layer refined the features while using stride 1 (S=1) and maintaining efficiency. Finally, an addition operation brought the second output of GhostConv layer to the first feature maps. This stage combined the produced and intrinsic feature maps, improving representational capability of modules and maintaining a low computational cost. The method enabled BloodCell-YOLO to achieve effective feature extraction and representation, fitting for real-time blood cell identification in environments with restricted resources. Relating to the discussion, combining GhostBottleneck modules guaranteed the model rapidly and precisely analyzed image without demanding large processing resources.

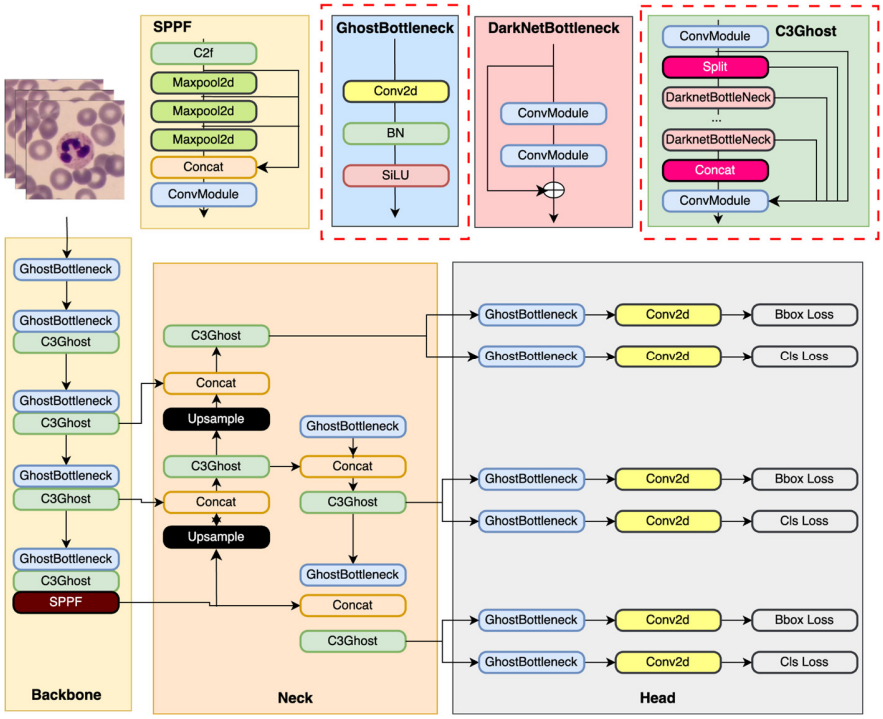

Fig. 2 Architecture of BloodCell-YOLO. The boxes with dashed lines showed modifications made to the original YOLOv8 architecture.

2) C3Ghost

BloodCell-YOLO substituted C2f modules in YOLOv8 with C3Ghost, which was built to improve network performance and efficiency. Modules used the ideas of GhostNet by combining standard as well as linear procedures to generate more feature maps with fewer parameters and reduced computing costs [24], [25]. Moreover, benefits of C3Ghost were reduced model size, higher inference speed, and better feature extraction capabilities. BloodCell-YOLO lowered number of parameters and GFLOPs by including C3Ghost, preserving or improving the accuracy of blood cell detection. This process was a crucial need for applications requiring real-time performance and great accuracy.

The architecture of C3Ghost modules began with two convolution layers processing the input characteristics, as shown in Figure 2. Following these first convolution layers, GhostBottleneck layers were then given individual output, where n represented the number of these levels. This process allowed modules to include many GhostBottleneck to improve feature extraction. The feature maps were concatenated with the output of the first convolution layers after passage through the sequence of GhostBottleneck segment. During this combining phase, C3Ghost improved the feature representation by combining features from GhostBottleneck layer with those from the first convolution. The combined feature maps passed through a convolution layer to produce the outcome of C3Ghost modules. This architecture allowed modules to create detailed and unique feature maps with fewer resources and computational expenses. Relating to the discussion, the process improved BloodCell-YOLO speed and performance. BloodCell-YOLO was lighter and more efficient with C3Ghost module, having the ability to detect blood cell rapidly, when resources were few.

BloodCell-YOLO combined GhostBottleneck and C3Ghost into structure and output of YOLOv8 model. This blend improved image analysis and blood cell recognition of the model, useful for hospitals as well as care settings with limited computer tools. During the study, BloodCell-YOLO performed well without abundant computing power, allowing it to be more viable for real-world applications.

C. Evaluation Metrics

The investigation evaluated the performance of BloodCell-YOLO model through important criteria often used in object identification applications. These calculations assessed the all-around accuracy, precision, recall, and computing cost of the model. The first major metric identified was mean average accuracy (mAP) at 50% IoU (mAP50). mAP was a frequently used metric to assess item identification quality. After computing the average accuracy of every class, the model generated individual mean of averages. Following the process, mAP50 evaluated

accuracy at a 50% IoU level. The outcome ensured that when a projected bounding box cross the ground truth bounding box by at least 50%, it was considered accurate. Subsequently, mAP came from averaging AP values for every class. Showing *N* as the number of classes and AP_i as the average accuracy for *i*-th class, mAP formula was presented as formula 1.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (1)$$

Using formula 2 for IoU helped the analysis to ascertain when a projected bounding box was accurate.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (2)$$

Precision was the true positive (TP) detection ratio to the total of false positives (FP) and TP, as the values showed how faithfully the model forecast relevant items. A high precision number signified that the model had a low FP rate. This outcome guaranteed that the discovered items were significantly valid and accurate, as 3 represented the formula for accuracy.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Another crucial statistic was recall, which showed TP detection ratio against the total of TP and false negatives (FN). The analysis used the parameter to measure how the model found every relevant item in the dataset. Additionally, a high recall value guaranteed that most relevant items were detected as it signified a low FN rate of the model, where recall had a formula shown by (4).

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Parameter count was an important statistic used to evaluate the total number of trainable parameters in the model. This value estimated how complex the model was, as well as computer capability required for training and inference. A model with many parameters typically required substantial processing power and memory, which limited its use in environments with fewer resources. Therefore, it was crucial to reduce the number of parameters and still preserving or improving performance was important. The reduction allowed the deployment of efficient and effective models on devices with limited processing capability, such as mobile devices as well as low-power medical equipment, improving the performance of the model in real-world applications.

Analytical computational complexity of BloodCell-YOLO model was evaluated using GFLOPs. The model showed the floating-point operations (FLOPs) required for a single forward pass over the network, which was scalable to billions. Following the discussion, GFLOPs had a formula shown by (5) in the analysis. This metric helped in understanding the processing requirements of the model as fewer GFLOPs point to a more efficient model needing minimum computational capability. Reducing the model-assisted BloodCell-YOLO in obtaining faster inference times and more acceptable for deployment in low-power medical equipment as well as mobile devices, which were two resource-constrained circumstances. Concerning the criteria, analyzing BloodCell-YOLO model permitted observing the performance in its totality, ensuring that the model was accurate and performed well for individual usage in different clinical as well as medical settings. Additionally, the review showed the strength of the model and areas for improvement, providing valuable guidance for future efforts in developing more efficient versions.

$$GFLOPs = \frac{2\ x\ \sum_{l=1}^{L}(IC\ x\ KH\ x\ KW\ x\ OC\ x\ OH\ x\ OW)}{10^9} \quad (5)$$

The variables in GFLOPs formula represented major aspects of convolutional layer computations. IC (Input Channels) referred to the number of channels in input feature map of this layer, while KH (Kernel Height) and KW (Kernel Width) represented the dimensions of convolutional filter. OC (Output Channels) signified the number of output feature map channels generated by convolution. In addition, OH (Output Height) and OW (Output Width) were the dimensions of output feature map, determined by input size, kernel size, stride, as well as padding applied during convolution. The total number of layers in the model was represented by L (Number of Layers), and Factor 2 accounts for the two operations (multiplication and addition) in each multiply-accumulate (MAC) operation. Finally, dividing by 10^9 converted total FLOPs into GFLOPs, providing a measure of computational complexity in billions of operations.

TABLE 3
DEFAULT PARAMETERS FOR TRAINING YOLO MODELS USING THE ULTRALYTICS FRAMEWORK

Argument	Value
epochs	100
confidence	0.001
optimizer	auto
batch	16
learning_rate	0.01
iou_threshold	0.5
nms_threshold	0.6
img_size	640

D. Training model

The training process of BloodCell-YOLO used New Cell Dataset, which included seven types of blood cell, namely basophil, eosinophil, lymphocyte, monocyte, neutrophil, platelets, and RBCs. The careful training on this dataset helped the model to learn various traits and representations of different blood cell types. Following the process, the learned model was applied as a pre-trained model for BCCD dataset, which focused on three classes, namely WBCs, RBCs, and platelets. Using the information learned from New Cell Dataset, transfer learning method helped the model perform better on BCCD dataset by allowing it to adapt adequately. This pre-trained model helped speed up training and improved accuracy in detecting the fewer types of classes in BCCD dataset. Table 3 showed the standard training settings used to keep attributes the same and reliable for both datasets.

IV. RESULTS

A. Result for New Cell Dataset

BloodCell-YOLO model trained on New Cell Dataset which comprised seven classes of blood mentioned earlier. Several important criteria, including mAP50, mAP at IoU thresholds between 50% and 95% (mAP50-95), the number of parameters, GFLOPs, precision, and recall were used to assess the model.

TABLE 4
PERFORMANCE METRICS OF BLOODCELL-YOLO ON NEW CELLS DATASET

Model	Number of Parameters	GFLOPs	Precision	Recall	mAP50	mAP50-95
YOLOv8n	3,007,013	8.1	0.964	0.949	0.982	0.931
YOLOv8s	11,128,293	28.5	**0.976**	0.946	0.983	0.938
YOLOv8m	25,843,813	78.7	0.962	0.959	0.983	0.94
YOLOv8l	43,612,005	164.8	0.974	0.952	0.983	0.94
YOLOv8x	68,130,309	257.4	0.972	0.95	0.982	0.938
BloodCell-YOLO-n	**1,547,209**	**4.9**	0.97	0.95	0.98	0.922
BloodCell-YOLO-s	5,223,725	15.2	0.967	0.962	0.983	0.933
BloodCell-YOLO-m	8,860,405	30.2	0.961	0.96	**0.984**	0.938
BloodCell-YOLO-l	12,003,797	49.4	0.954	**0.966**	0.983	0.939
BloodCell-YOLO-x	18,685,969	76.7	0.964	0.959	0.983	**0.942**

TABLE 5
PERFORMANCE METRICS OF BLOODCELL-YOLO ON BCCD DATASET

Model	Number of Parameters	GFLOPs	Precision	Recall	mAP50	mAP50-95
BloodCell-YOLO-n	1,547,209	**4.9**	0.845	0.885	0.92	0.632
BloodCell-YOLO-s	5,223,725	15.2	0.823	**0.902**	0.907	0.646
BloodCell-YOLO-m	8,860,405	30.2	0.86	0.893	0.927	0.651
BloodCell-YOLO-l	12,003,797	49.4	0.868	0.895	**0.94**	**0.676**
BloodCell-YOLO-x	18,685,969	76.7	**0.871**	0.899	0.936	0.667

Table 4 showed the measures for many BloodCell-YOLO model variants. Incorporation of GhostBottleneck and C3Ghost modules greatly improved efficiency of the model while maintaining good detection accuracy, according to the results. BloodCell-YOLO-variant was the altered form of YOLOv8 where every variation corresponded to a certain YOLOv8 version. For instance, BloodCell-YOLO-n was the altered form of YOLOv8-n, BloodCell-YOLO-s was YOLOv8-s, etc. This naming convention signified that while maintaining the original structure and size of the relevant YOLOv8 version, each BloodCell-YOLO variant combined the improvements of GhostBottleneck and C3Ghost modules.

Evaluating BloodCell-YOLO and original YOLOv8 models showed considerable increase in computing efficiency and model performance on New Cell Dataset. When GhostBottneck and C3Ghost modules were combined in BloodCell-YOLO models, GFLOPs and the number of parameters were significantly lowered relative to the original YOLOv8 variations. Moreover, BloodCell-YOLO variations reduced GFLOPs on average by 45.56% and the number of parameters by 76.55%. The models regularly showed good performance in many other criteria during the analysis. BloodCell-YOLO-l scored 0.966 on recall, 0.984 BloodCell-YOLO-m had the highest mAP50, and BloodCell-YOLO-x also improved in mAP50-95 with 0.942. This result signified how well BloodCell-YOLO offered effective and high-accuracy detection fit for real-time applications, particularly in resource-limited settings. The variants of the model showed better total performance even when YOLOv8s had the highest accuracy as the models provided a more balanced work across all measures. Figure 3 and 4 showed mAP50 graph of training output, as well as the detecting results on New Cell Dataset.

Table 4 showed that the adjusted models provided a good mix of efficiency and computing requirements. This outcome signified the models were effective for rapidly finding blood cell in areas with few resources. BloodCell-YOLO adequately detected different types of blood, which helped in finding and managing health problems. Moreover, the new models performed well, balancing individual speed with the amount of computer power required. The process allowed the models to detect the cell in areas with minimal resources. These results showed BloodCell-YOLO correctly detected blood cells, which was important for managing various health problems.

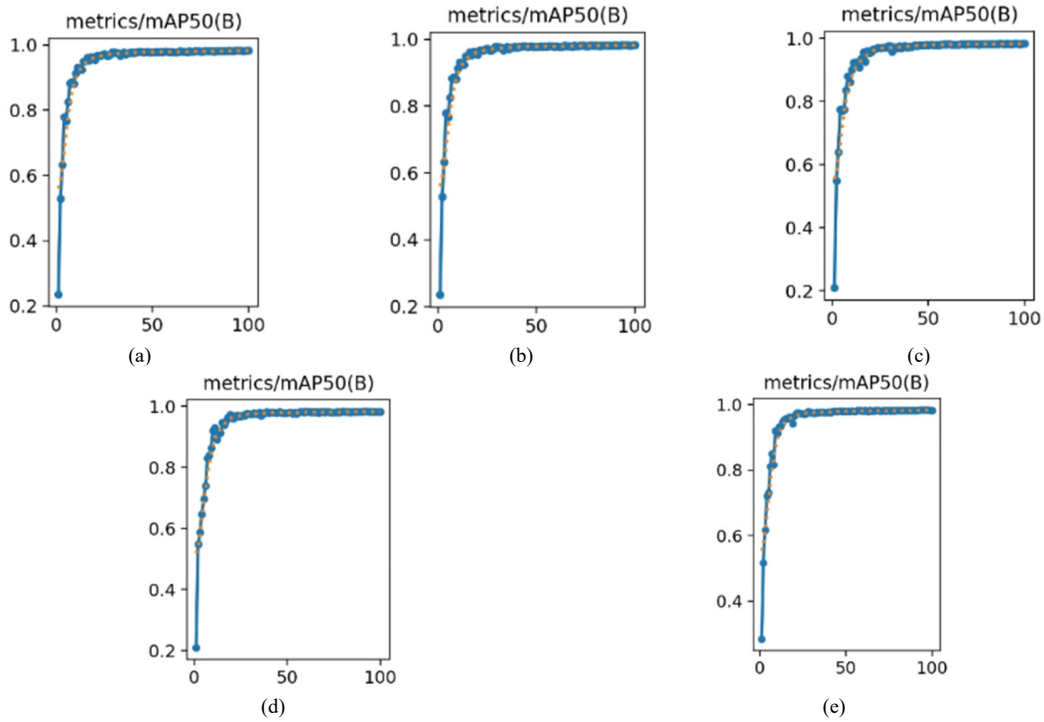

Fig. 3 mAP50 graph of training results for BloodCell-YOLO variant n (a), s (b), m (c), l (d), and x (e) in New Cells Dataset.

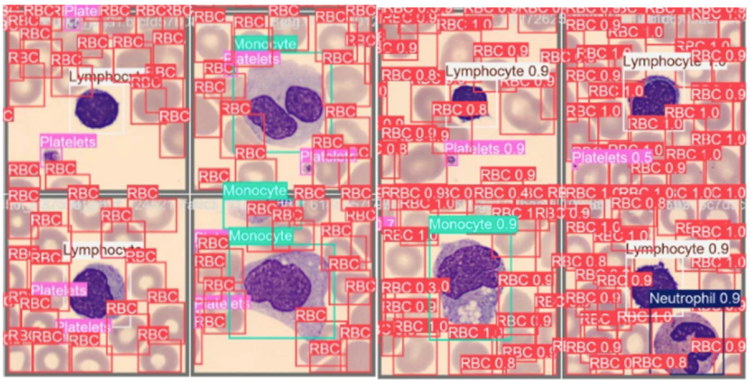

Fig. 4 Detection results of BloodCell-YOLO on the New Cells Dataset

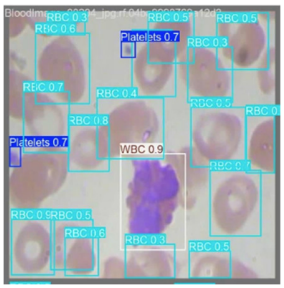

Fig. 5 Detection results of BloodCell-YOLO on the BCCD Dataset

B. Result for BCCD Dataset

After the first training with New Cell Dataset, BloodCell-YOLO model was improved using BCCD dataset. The collection contained annotated image of three blood cell types, namely RBCs, WBCs, and Platelets which was widely used in medical image analysis. In addition, the study used the Dataset from pre-trained model as a starting point to improve performance on BCCD dataset by using the features it learned.

The analysis used important measurements such as mAP50, mAP50-95, number of parameters, GFLOPs, accuracy, and recall to determine the success of the model on BCCD dataset. The pre-trained model greatly improved accuracy and speed in identification because it was trained on a wide variety of data from New Cell Dataset. The rating results were shown in Table 5, as Figure 5 showed the identification outcomes from BCCD datasets. Moreover, the results signified that the model was very accurate and efficient in detecting, allowing it to be useful in medical diagnosis after being improved.

V. DISCUSSION

The performance evaluation of BloodCell-YOLO model on New Cell Dataset showed significant improvements in efficiency and detection accuracy. Incorporating GhostBottleneck and C3Ghost modules into YOLOv8 architecture led to substantial reductions in GFLOPs and number of parameters. Averagely, BloodCell-YOLO variants achieved 45.56% reduction in GFLOPs and 76.55% reduction in the number of parameters compared to the original YOLOv8 models. The model was particularly suitable for deployment in resource-constrained environments, such as mobile devices and low-power medical equipment. Table 4 showed the major metrics, signifying that BloodCell-YOLO maintained high precision, recall, mAP50, and mAP50-95 performance. Additionally, BloodCell-YOLO-l achieved the highest recall at 0.966, BloodCell-YOLO-m reached the highest mAP50 at 0.984, and BloodCell-YOLO-x excelled in mAP50-95 with a score of 0.942. Despite YOLOv8s recording the best precision at 0.976, BloodCell-YOLO variants presented a more balanced performance across all metrics, showing individual superior effectiveness. These results signified BloodCell-YOLO was accurate and quick for blood cell identification, crucial for the diagnosis as well as treatment of many medical disorders. The model was a good option for real-time applications in medical diagnostics as the changes to YOLOv8 design provided a fair compromise between computing needs and speed.

BloodCell-YOLO, which was a recommended method was evaluated following an earlier study concerning BloodCell-YOLO-n and BloodCell-YOLO-l variations. BloodCell-YOLO-n had the greatest computational efficiency

TABLE 6
COMPARISON WITH PREVIOUS STUDIES ON BCCD DATASET

Method	Baseline	#Params (M)	GFLOPS	Precision	Recall	mAP50			
						WBCs	RBCs	Platelets	all
DWS-YOLO [14]	YOLOv5	1.56	**3.8**	-	-	**0.991**	**0.917**	0.906	0.938
AYOLOv5 [15]	YOLOv5	-	-	0.862	**0.915**	-	-	-	0.933
Advanced YOLOv5s [16]	YOLOv5	-	17	**0.874**	0.88	-	-	-	0.935
TE-YOLOF [17]	YOLOv3	16.76	6.6	-	-	0.987	0.873	0.898	0.919
Efficient YOLOv3 [18]	YOLOv3	13.6	-	-	-	0.989	0.804	0.902	0.898
ISE-YOLO [19]	YOLOv3	-	118.1	-	-	-	-	-	0.857
BloodCell-YOLO-n (proposed)	YOLOv8	**1.54**	4.9	0.845	0.885	0.975	0.868	0.918	0.920
BloodCell-YOLO-l (proposed)	YOLOv8	12	49.4	0.868	0.895	0.973	0.908	**0.924**	**0.94**

(GFLOPs), allowing these two models to be selected. Following the discussion, BloodCell-YOLO-l obtained the highest mAP, signifying exceptional performance. The comparison showed how a better number of parameters and mAP the recommended BloodCell-YOLO-n and BloodCell-YOLO-l models had than in other studies. Despite reaching a mAP of 0.938, the former method DWS-YOLO showed superior GFLOPs at 3.8 than 4.9 of BloodCell-YOLO-n. BloodCell-YOLO-l achieved a higher mAP of 0.94 with a greater GFLOPs of 49.4. This outcome signified how well the recommended methods balanced detection performance with computing cost. Incorporating GhostBottleneck and C3Ghost modules helped explain this better performance by lowering the computational burden and preserving excellent detection accuracy. Table 6 showed a careful comparison between the proposed method and past studies, signifying the developments in blood cell identification through the recommended changes.

The improved algorithms found potential use in mobile health devices, point-of-care diagnostics, and real-time monitoring systems where computational efficiency as well as accuracy were essential. Innovations of BloodCell-YOLO improved the performance of these systems by providing quicker and more accurate diagnosis, specifically in settings with limited processing resources. Moreover, faster and more consistent blood cell analysis helped in creating opportunities for more effective use of AI-driven diagnostic instruments in clinical practice.

The study showed the effectiveness of the proposed BloodCell-YOLO model in detecting blood cell types using New Cell Dataset and BCCD dataset. However, a limitation was that the model has not been evaluated on challenging imaging conditions, such as noisy image, occlusions, or rare cell types. These factors could affect detection accuracy and show areas where further improvements would be necessary.

The study focused on blood cell detection and did not evaluate the applicability of the model to other medical image data with different conditions. The analysis used public data for comparison with previous studies, but was limited to blood cell identification, without exploring broader medical applications. Future studies could test the model in more complex settings and observe how its performance on other medical imaging tasks to improve the usefulness. Additionally, further analysis should examine the effectiveness of various training methods. The process included testing new ways to improve data, using transfer learning, and forming optimization methods better. The result of analysis was to improve the model in performing better in different situations.

BloodCell-YOLO model was used to diagnose medical conditions by correctly identifying and classifying blood cell. However, there were significant issues that were needed to be addressed due to differences in colors, lighting, and background sounds, all of which could affect the performance of the model. The issue could be fixed by using different data that focused on specific areas. In addition, the systems also needed to connect properly with lab information systems (LIS), follow healthcare regulations and provide training for medical staff to effectively fit into the organization. Despite these challenges, the model performed effectively and reliably, allowing the model to be useful for automating evaluations, reducing workloads, as well as improving patient care.

VI. CONCLUSIONS

In conclusion, the study showed that BloodCell-YOLO model performed better in detecting blood cell in New Cell Dataset. The model was well-suited for deployment in resource-constrained environments, such as low-power medical imaging devices. Averagely, BloodCell-YOLO achieved a 45.56% reduction in computational cost and a 76.55% decrease in the number of parameters compared to the original YOLOv8 models.

The results signified that BloodCell-YOLO was effective across multiple evaluation metrics. During the analysis, BloodCell-YOLO-l achieved the highest recall of 0.966, while BloodCell-YOLO-m had the best mAP50 of 0.984.

BloodCell-YOLO-x had the highest mAP50-95 with a score of 0.942 during this study. Despite YOLOv8s achieving the highest precision at 0.976, BloodCell-YOLO variants showed a more balanced performance across all major metrics, allowing the models to be highly suitable for real-world applications.

BloodCell-YOLO-l and BloodCell-YOLO-n were particularly significant when compared to previous studies. BloodCell-YOLO-n offered the most efficient computational performance, while BloodCell-YOLO-l achieved the highest mAP50. Despite DWS-YOLO outperforming BloodCell-YOLO-n in GFLOPs (3.8 vs. 4.9), it achieved a slightly lower mAP of 0.938 compared to BloodCell-YOLO-l, which reached 0.94. This result showed the effectiveness of the proposed method in balancing computational efficiency and detection performance.

Future studies would focus on further improving BloodCell-YOLO framework to increase both performance and efficiency. Exploring new modules incorporations and advanced training methodologies could further optimize computational requirements. Additionally, applying BloodCell-YOLO to a broader range of medical imaging tasks and datasets would help validate its adaptability and strength. Investigating real-world implementation of the model in clinical workflows would also provide valuable insights into its practical impact on medical diagnostics..

Author Contributions: *[Mohammad Farid Naufal]*: Conceptualization, Methodology, Writing - Original Draft. *[Selvia Ferdiana Kusuma]*: Supervision, Software, Writing – review & editing.

Funding: This research received no specific grant from any funding agency.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The datasets used in this study are publicly available. The BCCD dataset can be accessed at https://public.roboflow.com/object-detection/bccd, and the New Cell dataset can be accessed at https://universe.roboflow.com/hhh-r05so/new-cells. For access to the code used in this study, requests can be made by contacting the first author via email for privacy reasons.

Informed Consent: There were no human subjects involved in this study. The data utilized were publicly available blood cell datasets.

Institutional Review Board Statement: Not applicable.

Animal Subjects: There were no animal subjects.

ORCID:
First Author: https://orcid.org/0000-0002-1194-2760
Second Author: https://orcid.org/0000-0002-7366-1542

REFERENCES

[1] A. Engert *et al.*, "The European Hematology Association Roadmap for European Hematology Research: a consensus document," *Haematologica*, vol. 101, no. 2, pp. 115–208, Feb. 2016, doi: 10.3324/haematol.2015.136739.
[2] A. Gupta *et al.*, "Deep Learning in Image Cytometry: A Review," *Cytometry A*, vol. 95, no. 4, pp. 366–380, Apr. 2019, doi: 10.1002/cyto.a.23701.
[3] K. Almezhghwi and S. Serte, "Improved Classification of White Blood Cells with the Generative Adversarial Network and Deep Convolutional Neural Network," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–12, Jul. 2020, doi: 10.1155/2020/6490479.
[4] D. Ryu *et al.*, "Label-Free White Blood Cell Classification Using Refractive Index Tomography and Deep Learning," *BME Front.*, vol. 2021, p. 9893804, Jan. 2021, doi: 10.34133/2021/9893804.
[5] B. Vinoth Kumar, S. Abirami, R. J. Bharathi Lakshmi, R. Lohitha, and R. B. Udhaya, "Detection and Content Retrieval of Object in an Image using YOLO," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 590, no. 1, p. 012062, Oct. 2019, doi: 10.1088/1757-899X/590/1/012062.
[6] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.
[7] A. Vijayakumar and S. Vairavasundaram, "YOLO-based Object Detection Models: A Review and its Applications," *Multimed. Tools Appl.*, Mar. 2024, doi: 10.1007/s11042-024-18872-y.
[8] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.
[9] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition," *Drones*, vol. 7, no. 5, p. 304, May 2023, doi: 10.3390/drones7050304.
[10] S. Liu, T. Fang, J. Han, and D. Xiao, "Approach state detection of lightweight steel sections based on improved YOLOv8-pose," in *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Zhuhai, China: IEEE, Apr. 2024, pp. 1248–1256. doi: 10.1109/CVIDL62147.2024.10603545.

[11] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention," *Agronomy*, vol. 13, no. 7, p. 1824, Jul. 2023, doi: 10.3390/agronomy13071824.
[12] Z. Liu, R. M. Rasika D. Abeyrathna, R. Mulya Sampurno, V. Massaki Nakaguchi, and T. Ahamed, "Faster-YOLO-AP: A lightweight apple detection algorithm based on improved YOLOv8 with a new efficient PDWConv in orchard," *Comput. Electron. Agric.*, vol. 223, p. 109118, Aug. 2024, doi: 10.1016/j.compag.2024.109118.
[13] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features from Cheap Operations," Mar. 13, 2020, *arXiv*: arXiv:1911.11907. Accessed: Jul. 07, 2024. [Online]. Available: http://arxiv.org/abs/1911.11907
[14] Y. Mao, H. Zhang, W. Wu, X. Gao, Z. Lin, and J. Lin, "DWS-YOLO: A Lightweight Detector for Blood Cell Detection," *Appl. Artif. Intell.*, vol. 38, no. 1, p. 2318673, Dec. 2024, doi: 10.1080/08839514.2024.2318673.
[15] W. Gu and K. Sun, "AYOLOv5: Improved YOLOv5 based on attention mechanism for blood cell detection," *Biomed. Signal Process. Control*, vol. 88, p. 105034, Feb. 2024, doi: 10.1016/j.bspc.2023.105034.
[16] Y. He, "Automatic Blood Cell Detection Based on Advanced YOLOv5s Network," *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3360142.
[17] F. Xu, X. Li, H. Yang, Y. Wang, and W. Xiang, "TE-YOLOF: Tiny and efficient YOLOF for blood cell detection," *Biomed. Signal Process. Control*, vol. 73, p. 103416, Mar. 2022, doi: 10.1016/j.bspc.2021.103416.
[18] A. Shakarami, M. B. Menhaj, A. Mahdavi-Hormat, and H. Tarrah, "A fast and yet efficient YOLOv3 for blood cell detection," *Biomed. Signal Process. Control*, vol. 66, p. 102495, Apr. 2021, doi: 10.1016/j.bspc.2021.102495.
[19] C. Liu, D. Li, and P. Huang, "ISE-YOLO: Improved Squeeze-and-Excitation Attention Module based YOLO for Blood Cells Detection," in *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA: IEEE, Dec. 2021, pp. 3911–3916. doi: 10.1109/BigData52589.2021.9672069.
[20] M. MIT, "Blood Cell Count and Detection." Accessed: Jan. 06, 2024. [Online]. Available: https://public.roboflow.com/object-detection/bccd
[21] roboflow roboflow, "New Cells Dataset." Accessed: Jul. 31, 2024. [Online]. Available: https://universe.roboflow.com/hhh-r05so/new-cells
[22] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., in Algorithms for Intelligent Systems. , Singapore: Springer Nature Singapore, 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2_39.
[23] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
[24] J. Xu, H. Yang, Z. Wan, H. Mu, D. Qi, and S. Han, "Wood Surface Defects Detection Based on the Improved YOLOv5-C3Ghost With SimAm Module," *IEEE Access*, vol. 11, pp. 105281–105287, 2023, doi: 10.1109/ACCESS.2023.3303890.
[25] C. Gao *et al.*, "A fast and lightweight detection model for wheat fusarium head blight spikes in natural environments," *Comput. Electron. Agric.*, vol. 216, p. 108484, Jan. 2024, doi: 10.1016/j.compag.2023.108484.