



Combination of Cross Stage Partial Network and GhostNet with Spatial Pyramid Pooling on YOLOv4 for Detection of Acute Lymphoblastic Leukemia Subtypes in Multi-Cell Blood Microscopic Image

Tanzilal Mustaqim¹, Chastine Fatichah^{2*}, Nanik Suciati³

^{1,2,3}Department of Informatics, Faculty of Electrical Technology and Intelligent Information Technology, Institut Teknologi Sepuluh Nopember, Indonesia

Abstract.

Purpose: Acute Lymphoblastic Leukemia (ALL) Detection with microscopic blood images can use a deep learning-based object detection model to localize and classify ALL cell subtypes. Previous studies only performed single cell-based detection objects or binary classification with leukemia and normal classes. Detection of ALL subtypes is crucial to support early diagnosis and treatment. Therefore, an object detection model is needed to detect ALL subtypes in multi-cell blood microscopic images.

Methods: This study focuses on detecting the ALL subtype using YOLOv4 with a modified neck using Cross Stage Partial Network (CSPNet) and GhostNet. CSPNet is combined with Spatial Pyramid Pooling (SPP) to become SPPCSP to get various features map before the YOLOv4 final layer. Ghostnet was used to reduce the computation time of the modified YOLOv4 neck.

Result: Experimental results show that YOLOv4 SPPCSP outperformed the recall value of 14.6%, the value of mAP@.5 0.8%, and reduced the computation time by 4.7 ms compared to the original YOLOv4.

Novelty: The combination of CSPNet and GhostNet for YOLOV4 neck modification can increase the variety of features map and reduce computing time compared to the Original YOLOv4.

Keywords: Acute Lymphoblastic Leukemia, YOLOv4, Spatial Pyramid Pooling, GhostNet, Blood Microscopic Image

Received June 2022 / **Revised** July 2022 / **Accepted** October 2022

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



INTRODUCTION

Automatic analysis of microscopic images for disease detection and diagnosis is becoming very important and challenging in the medical field of computer vision. Microscopic images of blood contain essential information for early diagnosis and treatment of diseases, one of which is leukemia. Leukemia is a hematological disease of the bone marrow characterized by abnormal white blood cell production [1]. Leukemia is divided into two infected cell types: myeloid and lymphocyte. Based on the speed of spread of infection and aggressiveness, leukemia is divided into 2, namely acute and chronic. Acute Lymphoblastic Leukemia (ALL) is characterized by abnormal lymphocyte cell development and overproduction [2]. ALL can cause a decrease in the immune system and disease complications in crucial organs of the body.

Analysis of computer-aided blood cell images from previous research by [3] uses stages such as preprocessing, segmentation, feature extraction, and classification. Each stage affects the quality of subsequent analysis. The preprocessing stage aims to improve the quality of the cell image through filtering, followed by the segmentation stage, which separates the white blood cell image from other images such as erythrocytes, platelets, and background. The segmentation results are extracted with morphological characteristics followed by a classification process using machine learning or deep learning methods.

Previous researchers have used several methods, namely k-means clustering in the segmentation process, Local Directional Pattern (LDP) for feature extraction, and Support Vector Machine (SVM) as a classifier process [2]. Other researchers demonstrated the segmentation process by k-means clustering and feature

*Corresponding author.

Email addresses: derekln14@gmail.com (Mustaqim), chastine@if.its.ac.id (Fatichah), nanik@if.its.ac.id (Suciati)

DOI: [10.15294/sji.v9i2.37350](https://doi.org/10.15294/sji.v9i2.37350)

extraction with a hidden Markov model that accentuated the core and cytoplasm of contour features [4]. Another method uses watershed transformation for the segmentation process and Gaussian Mixed Modeling (GMM) for the classification process [5]. Another method uses geometric and texture features in cells and nuclei to analyze the cells in contact across ALL subtypes, which are finally classified using SVM [6]. The conventional method has several disadvantages, namely, it requires several separate steps, is not effective in touching cells, and cannot localize ALL subtypes heterogeneously.

Another method used by previous researchers is developing a Convolutional Neural Network (CNN) model. CNN can perform feature extraction and classification in one learning framework to make architecture performance more efficient. The literature from previous studies that performed leukemia analysis with CNN outperformed machine learning methods in terms of accuracy, precision, and recall [7]. Other literature compares CNN with machine learning models for classifying white blood cell types [8]. The same thing also uses CNN as the basis of the blood cell image analysis method for AML classification [9]. Previous studies using CNN only performed single cell-based object detection or binary classification with leukemia and normal classes.

The method for obtaining the location of the ALL subtype in multi-cell microscopic blood images can use the object detection method. Object detection methods from the previous literature have been used for cancer analysis and can also classify and detect cancer cell locations [10], [11]. Detection of leukemia cells using the original YOLOv4 by [12] only performed localization and classification on multi-cell images with normal and leukemia classes. However, detecting ALL subtypes is crucial to support early diagnosis and treatment. Therefore, this study uses object detection based on You Only Look Once (YOLO) to classify and localize ALL subtypes on multicellular blood microscopic images.

The development of the YOLO object detection model has reached YOLOv5. Previous literature reported that the YOLOv4 model could detect apple images with higher mAP values than YOLOv5 [13]. Another study that detected and localized melanoma lesions using YOLOv4 achieved a more robust and optimal bounding box than YOLOv5 [14]. Although YOLOv4 outperforms YOLOv5, YOLOv4 is still ineffective at detecting small and touching objects [15]. YOLOv4 is also prone to object class misclassification if the object class characteristics are similar [16].

Following the development of the object detection model, CSPNet proposed by Wang and his colleagues in 2020 improves feature map analysis by integrating the initial features map from the base layer through a cross-stage hierarchy to the resulting CNN feature map layer [17]. CSPNet has been used on the YOLOv4 backbone but not on the YOLOv4 neck. Even though CSPNet makes the YOLOv4 model more robust, it still has a higher computation time and a heavier memory [18].

Several popular light detection models for extracting feature maps in images have been proposed, one of which is GhostNet. GhostNet has the advantage of extracting feature maps by reducing computation time while maintaining the quality of the analysis results [19]. Previous studies used GhostNet for colon disease classification, where GhostNet outperformed Residual Network (ResNet) and Mobile Neural Architecture Search (MnasNet) [20].

In this study, we propose a neck-modified YOLOv4 model using CSPNet and GhostNet for ALL subtype detection. CSPNet is combined with Spatial Pyramid Pooling (SPP) into SPPCSP to get various features mapped before the YOLOv4 final layer. Ghostnet was used to reduce the computation time of the modified YOLOv4 neck.

To summarize, this paper has two main contributions:

1. Enhancing the variety of feature maps with CSPNet
2. Reducing computation time of the YOLOv4 SPPCSP model using GhostNet

This paper is arranged in the following order: Section 2 describes the methods and models used, section 3 describes and compares the results obtained, and section 4 concludes this study.

METHODS

YOLOv4

YOLOv4 improves the object detection model to improve accuracy and efficiency [21]. Effective and efficient Weighted Residual Connection combines residual neural networks [22]. Cross Stage Partial Connections (CSPNet) integrates the feature map from the initial to the final stage of the network. CSPNet implementation reduces computing by up to twenty percent and outperforms other backbone architectures [22]. Cross Mini-Batch Normalization solves a problem in statistics where the result of defining normalization cannot be estimated accurately [23]. Self Adversarial Training is a technique for adding new data to a network that operates in two phases, forward and backward. Mish activation is a function of self-regulating non-monotone activation [24]. Mosaic Data Augmentation is a technique for adding data by combining several input image attributes into one image data set. CNN uses DropBlock regularization as a new regularization method [24]. Lastly is CIoU loss, a loss function that achieves faster convergence and greater accuracy for bounding box regression problems [25].

The YOLOv4 architecture leverages the CSPDarknet53 backbone network, SPP (Spatial Pyramid Pooling), PANet for Neck, and 3 YOLO heads, each generating a feature map of S^2 size ($3(5+C)$). S represents the grid size, 3 represents the number of bounding boxes generated per feature map, ($5+C$) represents the bounding box attributes, which include the center, width, and height coordinates and their objectivity scores, and C represents the confidence score of the bounding box for each class. Details of the YOLOv4 architecture are shown in Figure 1.

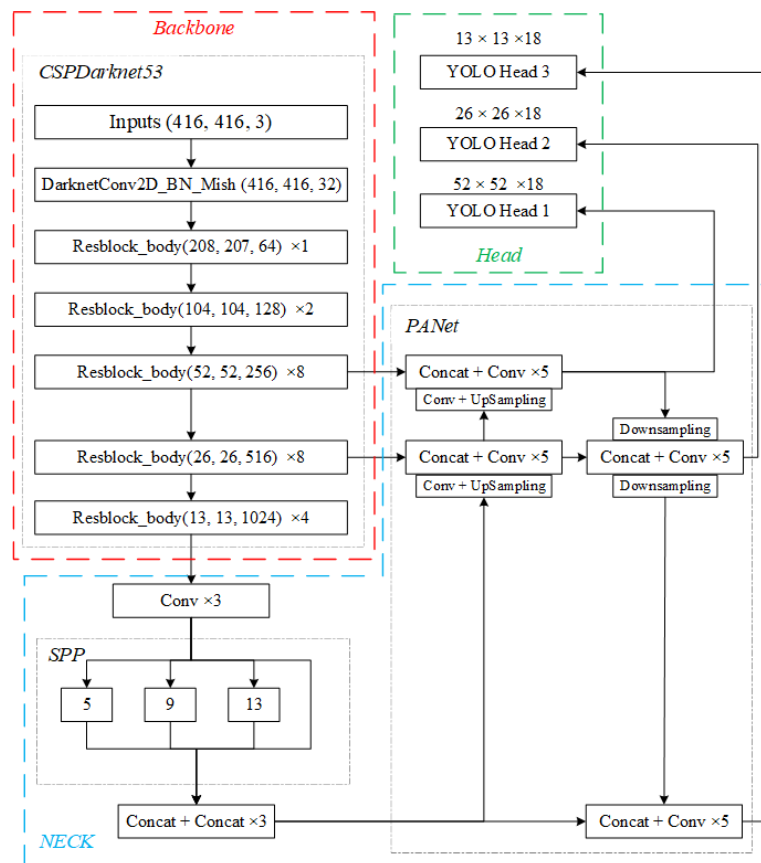


Figure 1. YOLOv4 architecture [12]

Architectural components such as the backbone, neck, and head in YOLOv4 perform different tasks. The CSPdarknet53 backbone functions to extract features map from the image. On the neck, an SPP performs max pooling with four different pool sizes (5×5 , 9×9 , 13×13 , 1×1), and the output of these four processes will be combined and processed PANet. PANet works to improve the quality of feature maps by upsampling and downsampling using low-level and high-level feature maps. Object detection is carried out in the head section using three different sizes. The head section of the first YOLOv4 detects small objects, the second

YOLOv4 head section detects medium-sized objects, and the third YOLOv4 head section detects large objects.

Cross Stage Partial Network (CSPNet)

Cross Stage Partial Network (CSPNet) attempts to solve the vanishing gradient problem and extensive computational processes of a CNN-based architecture to be used on mobile CPUs and GPUs with limited computing power [17]. Deep learning architecture acquires knowledge through the backpropagation of gradients in the network. CSPNet aims to achieve more diverse gradient combinations while minimizing computations. CSPNet accomplishes this by splitting the gradient stream into separate network paths.

The base layer feature map is divided into two parts using the channel $x_0 = [x_0', x_0'']$. Merging the different feature maps x_0' and x_0'' will produce the original feature map x_0 . The former is directly connected to the last layer, while the latter will be analyzed for its feature map with the CNN architectural block, and in this study, the SPP block is used. Details of the CSPNet architecture used in this study are shown in Figure 2.

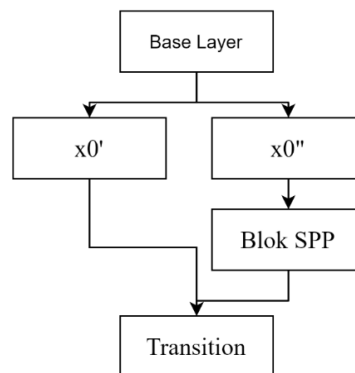


Figure 2. CSPNet architecture

GhostNet

GhostNet aims to provide a better model performance alternative in computing for convolution layers used in convolutional neural networks with reduced computational time [26]. GhostNet generates a portion of the total output feature map using standard convolution, while the rest uses a more computationally efficient linear operation, depthwise convolution.

The output tensor of each convolution layer is analyzed by serializing two operations. First, it analyzes a portion of the total channel for the output tensor using a sequential stack of three layers: standard convolution, batch normalization, and non-linear activation function, which by default is Rectified Linear Unit (ReLU). Output is sent to a secondary block, a sequential stack of three layers: depthwise convolution, batch normalization, and ReLU. The output tensor is provided by stacking the tensor of the first sequential block with the output of the second sequential block. The detailed architecture of GhostNet is shown in Figure 3.

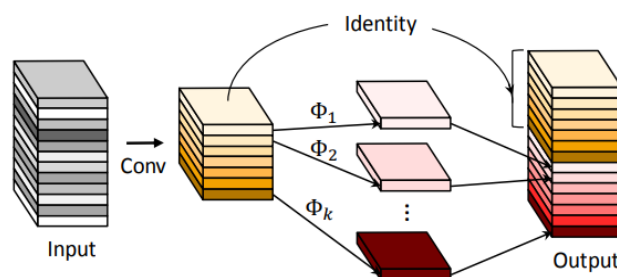


Figure 3. GhostNet architecture

Evaluation

The experiment results from the model carried out an evaluation stage to determine the performance in detecting the location and classification of ALL subtypes on multi-cell blood microscopic images. The evaluation metrics used in this study are precision, recall, F1, Mean Average Precision (mAP), and

computation time. The precision measurement process aims to compare the True Positive prediction to the overall results of the positive predicted model [27]. Recall provides information regarding comparing True Positive predictions to the comprehensive data predicted True Positive [28]. F1 can measure the proportion of the average comparison of recall and precision values weighted [29] by Equation (1) to produce precision values, Equation (2) to create recall, and Equation (3) to create F1.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1 = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (3)$$

Average precision is a measurement result that combines recall and precision [30]. In other words, it compares the prediction accuracy of all documents or output data with the actual data. True Positive indicates that an object with the correct classification has been detected, and the IoU value is more significant than specified, for example, $IoU \geq 0.5$ or 50%. False Positive refers to detecting an object that does not conform to the classification and has an IoU that is less than the specified threshold. False Negative represents the number of things that are not detected based on ground truth data.

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k + 1)] * Precisions(k) \quad (4)$$

$$Recalls(n) = 0, Precisions(n) = 1$$

$$n = \text{Number of treshold}$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

Equation (4) shows the Average precision formula. The calculation of precision and recall will calculate the Average Precision (AP) score, which calculates the average precision value of the total recall where r indicates the recall value and p is the precision value. The mAP is calculated by finding the AP for each class and then the average across several classes, as shown in Equation (5). The thresholds used in this study were 0.5 and 0.95.

The evaluation of computing time is the process of searching for one bounding box from 4 criteria: preprocessing, inference, Non-Maximum Suppression (NMS), and total. Preprocessing and inference computation time calculates the time from the start of the image to the preprocessing stage and detection of the ALL subtype. Inference computation time is longer than other variables because the detection process tries many possible object combinations. The NMS selects each bounding box with the highest filter confidence and an overlapping bounding box.

Proposed Method

The analysis process aims to extract feature map variations to detect touching objects and small objects in multi-cell ALL images. At this stage, the original YOLOv4 neck was modified using CSPNet as an additional block for SPP. The work process of the SPPCSP Block is divided into 2, the first going through SPP processing and the second going straight to the final block through a standard convolution process. CSPNet Blocks are composed of 3 Ghost Convolution Blocks before SPP and 2 Convolution Blocks after SPP. The results of the two paths are carried out in the concatenation stage to unify the resulting feature map.

Ghost Convolution is used in this study to make the performance of the SPPCSP neck modification model faster in computing time while maintaining a high accuracy value. Ghost convolution comprises 2 convolution processes, namely standard convolution and Depthwise convolution. Depthwise convolution performs the convolution process of each channel on each filter channel so that the computational process is more efficient. The kernel sizes used in the SPP block model are 5 x 5, 9 x 9, and 13 x 13, with the

padding size value obtained from the variation in kernel size divided by 2. The design of the YOLOv4 neck modification is shown in Figure 4.

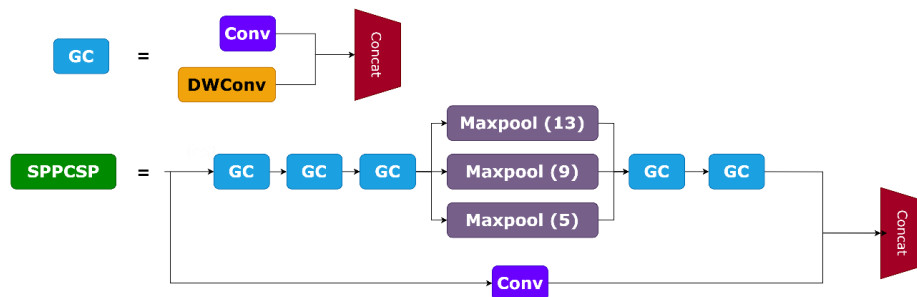


Figure 4. YOLOv4 neck modification

RESULT AND DISCUSSION

The dataset used is a microscopic image of blood cells subtype ALL from the paper by Fatonah et al., which has a multi-cell image [6]. The dataset consists of 3 classes, namely L1, L2, and L3. The literature states that the dataset source was obtained from a hospital with detailed images taken with a 1000x scale and the same staining process. The total image data is 301 multi-cell subtype ALL consisting of 128 L1 images, 63 L2 images and 110 L3 images.

Preprocessing the data in this study conducted an initial analysis of the dataset by manual labelling using a roboflow annotator, which can be accessed online, and resizing the image size from 591 x 443 pixels to 416 x 416 pixels. The process of selecting the resize size to 416 x 416 pixels is based on the standard size often used as image input in the YOLOv4 detection object model. The labelling process provides a bounding box for each white blood cell seen in the image according to ALL subtypes, such as L1, L2 and L3. An example of the labelling process is shown in Figure 5.

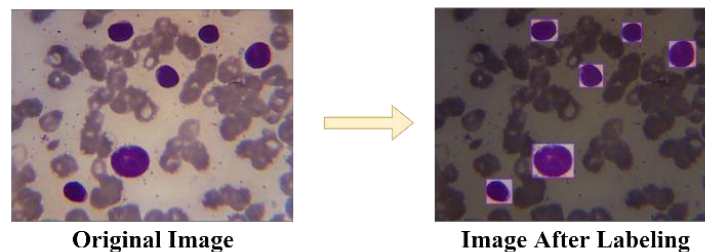


Figure 5. Example of a labelled image

The process of sharing the data resulting from the preprocessing is divided into 3 types: training, validation, and testing data. Each proportion is 75% on training data, 15% on validation data, and 10% on testing data. Details of the results of the division of 200 images on the training data, 39 images on the validation data, and 27 images on the testing data. The division process is carried out evenly from each class so that the model can find class representations at the training, validation, and testing stages. The testing process is divided into 2 experiment scenarios: the training process with the YOLOv4 model without modification and the training process with the modification of YOLOv4 on the neck with SPPCSP. Details of the implementation and experiment stages are shown in Figure 6.

The first scenario experiment extracted the location and classification of ALL subtypes on multi-cell blood microscopic images using the YOLOv4 model without modification. The testing process aims to determine the strength shown from the results of the overall analysis and each subtype of L1, L2, and L3. The experiment in the second scenario adds the implementation of the CSPNet model to the SPP so that the model gets a wider variety of features map. CSPNet extracts the features map from the base layer features map combined with the features map resulting from the SPP convolution process. The merging process produces a more varied features map to improve the quality of the analysis of subtypes of small and touching ALL cells in microscopic multi-cell blood images.

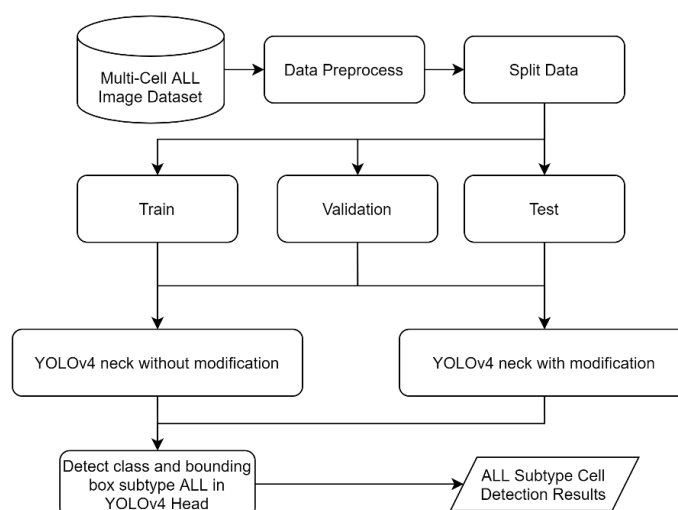


Figure 6. Flowchart of research experiment stages

The experiment results from 2 scenarios were evaluated using mAP values, recall, precision and computation time. The threshold used for the mAP metric is 0.5 (mAP@.5) and 0.95 (mAP@.95) to evaluate the model's existence from the confidence bounding box resulting from the analysis. The evaluation process is carried out as a whole from the image in the dataset and each subtype ALL. The evaluation was carried out by testing data on as many as 27 images to determine the model's strength in generalizing to new data that was not included in the training process.

Details of the L1 experiment results are shown in Table 1. The modified YOLOv4 SPPCSP model can outperform all aspects of the evaluation compared to the original YOLOv4. The slightest difference is found in the value of mAP@.5 by 0.1%. The most significant difference was found in the recall metric, with 4.7%. A high recall value indicates that the YOLOv4 SPPCSP model can detect True Positive data accurately. The L1 object detection results are compared in Figure 7, where the YOLOv4 SPPCSP model can detect more L1 than the original YOLOv4 model.

Table 1. Evaluation results in subtype L1

No	Model	L1			
		Precision	Recall	mAP@.5	mAP@.95
1	YOLOv4 Original	98.9%	86.8%	98.6%	70.9%
2	YOLOv4 SPPCSP	99.4%	91.5%	98.7%	71.9%

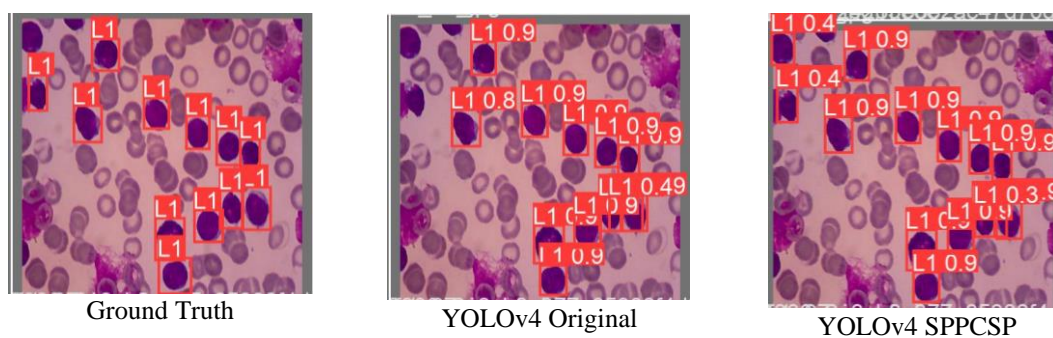


Figure 7. Comparison of L1 object detection results with ground truth

Table 2. Evaluation results in the L2 subtype

No	Model	L2			
		Precision	Recall	mAP@.5	mAP@.95
1	YOLOv4 Original	66.6%	66.7%	63.1%	47.5%
2	YOLOv4 SPPCSP	50.0%	100%	66.5%	49.9%



Figure 8. Comparison of L2 object detection results with ground truth

Details of the L2 experiment results are shown in Table 2. The original YOLOv4 model excels in evaluation precision by 16.6% but is still lower than other metrics. The YOLOv4 SPPCSP model has the same high precision evaluation value as the result of L1 object detection, with a difference of 33.3%. Figure 8 shows that the original YOLOv4 model cannot detect L2 according to ground truth, in contrast to the YOLOv4 SPPCSP model, which can detect L2 according to ground truth and has a high IoU confidence value of 0.9.

Table 3. Evaluation results in the L3 subtype

No	Model	L3			
		Precision	Recall	mAP@.5	mAP@.95
1	YOLOv4 Original	99.8%	88.9%	98.6%	70.0%
2	YOLOv4 SPPCSP	97.7%	94.7%	97.7%	66.4%

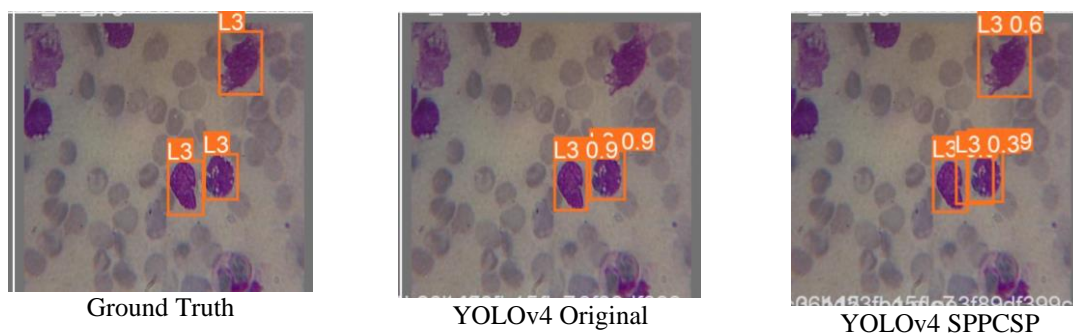


Figure 9. Comparison of L3 object detection results with ground truth

Details of the L3 experiment results are shown in Table 3. The original YOLOv4 model outperformed the three evaluation metric values except for recall. The result can also be seen in Figure 9, where the original YOLOv4 model can detect L3 well without any bounding box redundancy compared to the YOLOv4 SPPCSP model. However, according to the ground truth, the YOLOv4 model cannot detect L3 objects entirely. The YOLOv4 model can detect L3 objects according to ground truth, characterized by a high recall value.

Table 4. Evaluation results for all subtypes of ALL

No	Model	All subtype ALL			
		Precision	Recall	mAP@.5	mAP@.95
1	YOLOv4 Original	88.4%	80.8%	86.8%	62.8%
2	YOLOv4 SPPCSP	82.4%	95.4%	87.6%	62.8%

Apart from evaluating each subtype of ALL, an overall evaluation was also carried out to determine the model's performance against the generalization of all subtypes of ALL. From Table 4, the original YOLOv4 model has the advantage of a precision value of 6% and the same evaluation value at mAP@.95. The YOLOv4 SPPCSP model still has the advantage of recall values such as from the evaluation of each ALL subtype with a difference of 14.6%. The result shows that the YOLOv4 SPPCSP model can detect true positive data more accurately than the original YOLOv4 model.

Table 5. The results of the evaluation of the bounding box computation time

No	Model	Computing time (ms)			
		Pre-process	Inference	NMS	Total
1	YOLOv4 Original	2.3	11.2	4.7	18.2
2	YOLOv4 SPPCSP	0.9	9.8	2.8	13.5

The evaluation of the computational time is shown in Table 5. The evaluation process was carried out with 4 variables: preprocessing, inference, Non-Maximum Suppression (NMS), and the total of all computational time. The computational time of the preprocess and inference calculates the time from the beginning of the image through the preprocessing stage and detection of ALL subtypes. The inference computation time is longer than other variables because the detection process tries many combinations of possible objects found. The NMS process selects each bounding box with the highest filter confidence and overlaps bounding boxes. From Table 5, the YOLOv4 SPPCSP model can outperform all evaluation variables for computing time with a total difference of 4.7 ms. Our experiment in Table 5 indices YOLOv4 SPPCSP best in computing time, than YOLOv4 original [21], especially in this dataset [6].

CONCLUSION

In this study, location extraction and classification of ALL subtypes were carried out using microscopic multi-cell blood images and modified YOLOv4 neck using CSPNet and GhostNet. CSPNet is combined with Spatial Pyramid Pooling (SPP) into SPPCSP to get various features mapped before the YOLOv4 final layer. Ghostnet was used to reduce the computation time of the modified YOLOv4 neck. The Experimental results show that YOLOv4 SPPCSP outperformed the recall value of 14.6%, the value of mAP@.5 0.8%, and reduced the computation time by 4.7 ms compared to the original YOLOv4. The results showed that the YOLOv4 SPPCSP model could perform location extraction and classification on multi-cell blood microscopic images better than the original YOLOv4 model.

ACKNOWLEDGMENTS

This research was funded by the Ministry of Education and Culture of the Republic of Indonesia with a Master's Education Towards a Superior Undergraduate Doctorate (PMDSU) scholarship program under the Doctoral Dissertation Research (PDD) scheme.

REFERENCES

- [1] L. Boldú, A. Merino, A. Acevedo, A. Molina, and J. Rodellar, "A Deep Learning Model (Alnet) for The Diagnosis of Acute Leukaemia Lineage Using Peripheral Blood Cell Images," *Comput. Methods Programs Biomed.*, vol. 202, 2021.
- [2] D. Goutam and S. Sailaja, "Classification of Acute Myelogenous Leukemia in Blood Microscopic Images Using Supervised Classifier," in *2015 IEEE Int. Conf. Eng. Technol.*, 2015, pp. 1–5.
- [3] J. Su, J. Han, and J. Song, "A Benchmark Bone Marrow Aspirate Smear Dataset and a Multi-Scale Cell Detection Model for The Diagnosis of Hematological Disorders," *Comput. Med. Imaging Graph.*, vol. 90, p. 101912, 2021.
- [4] J. Su, S. Liu, and J. Song, "A Segmentation Method Based on HMRf for The Aided Diagnosis of Acute Myeloid Leukemia," *Comput. Methods Programs Biomed.*, vol. 152, pp. 115–123, Dec. 2017.
- [5] P. Ghosh, D. Bhattacharjee, M. Nasipuri, and D. K. Basu, "Automatic White Blood Cell Measuring Aid for Medical Diagnosis," *Proc. 2011 Int. Conf. Process Autom. Control Comput. PACC 2011*, 2011, pp. 1-6.
- [6] N. S. Fatonah, H. Tjandrasa, and C. Fatichah, "Identification of Acute Lymphoblastic Leukemia Subtypes in Touching Cells Based on Enhanced Edge Detection," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 4, pp. 204–215, 2020.
- [7] A. Khan, A. Eker, A. Chefranov, and H. Demirel, "White Blood Cell Type Identification Using Multi-Layer Convolutional Features with an Extreme-Learning Machine," *Biomed. Signal Process. Control*, vol. 69, p. 102932, 2021.
- [8] K. K. Anilkumar, V. J. Manoj, and T. M. Sagi, "Automated Detection of B Cell and T Cell Acute Lymphoblastic Leukaemia Using Deep Learning," *Irbm*, vol. 1, pp. 1–9, 2021.
- [9] C. Matek, S. Krappe, C. Münzenmayer, T. Haferlach, and C. Marr, "Highly Accurate Differentiation of Bone Marrow Cell Morphologies Using Deep Neural Networks on a Large Image Data Set," *Blood*, vol. 138, no. 20, pp. 1917–1927, 2021.
- [10] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving Small Object Detection in YOLOv5 for Autonomous Vehicles" 2021.

- [11] E. K. Wang *et al.*, “Multi-Path Dilated Residual Network for Nuclei Segmentation and Detection,” *Cells*, vol. 8, no. 5, p. 499, 2019.
- [12] R. Khandekar, P. Shastry, S. Jaishankar, O. Faust, and N. Sampathila, “Automated Blast Cell Detection for Acute Lymphoblastic Leukemia Diagnosis,” *Biomed. Signal Process. Control*, vol. 68, no. April, p. 102690, 2021.
- [13] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, “A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5,” *Remote Sens.*, vol. 13, no. 9, pp. 1–23, 2021.
- [14] S. Albahli, N. Nida, A. Irtaza, M. H. Yousaf, and M. T. Mahmood, “Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour,” *IEEE Access*, vol. 8, pp. 198403–198414, 2020.
- [15] E. Prasetyo, N. Suciati, and C. Fatichah, “YOLOv4-tiny with Wing Convolution Layer for Detecting Fish Body Part,” *Comput. Electron. Agric.*, vol. 198, 2022.
- [16] B. Richey and M. V. Shirvaikar, “Deep Learning Based Real-Time Detection of Northern Corn Leaf Blight Crop Disease Using YOLOv4,” p. 5, 2021.
- [17] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, “CSPNet: A New Backbone That Can Enhance Learning Capability of CNN,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, 2020, pp. 1571–1580.
- [18] I. Shim, J. H. Lim, Y. W. Jang, J. H. You, S. T. Oh, and Y. K. Kim, “Developing a Compressed Object Detection Model based on YOLOv4 for Deployment on Embedded GPU Platform of Autonomous System,” *Trans. Korean Soc. Automot. Eng.*, vol. 29, no. 10, pp. 959–966, 2021.
- [19] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More Features from Cheap Operations,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1577–1586.
- [20] N. M. Mhaidat *et al.*, “Study The Epigenetic Down-Regulation of Bim on Colorectal Cancer Chemotherapy Response,” *J. King Saud Univ. - Sci.*, vol. 31, no. 3, pp. 308–313, 2019.
- [21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv*, 2020.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016, 2016, pp. 770–778.
- [23] Z. Yao, Y. Cao, S. Zheng, G. Huang, and S. Lin, “Cross-Iteration Batch Normalization,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12326–12335.
- [24] D. Misra, “Mish: A Self Regularized Non-Monotonic Activation Function,” *arXiv*, 2019.
- [25] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, “CIA-SSD: Confident IoU-Aware Single-Stage Object Detector from Point Cloud,” *arXiv*, 2020.
- [26] C. Zhao, Y. Feng, R. Liu, and W. Zheng, “Application of Lightweight Convolution Neural Network in Cancer Diagnosis,” in *Proc. 2020 Conf. Artif. Intell. Healthc.*, 2020, pp. 249–253.
- [27] H. T. Ismet, T. Mustaqim, and D. Purwitasari, “Aspect Based Sentiment Analysis of Product Review Using Memory Network,” *Sci. J. Informatics*, vol. 9, no. 1, pp. 73–83, 2022.
- [28] A. Adimas and S. Y. Irianto, “Image Sketch Based Criminal Face Recognition Using Content Based Image Retrieval,” *Sci. J. Informatics*, vol. 8, no. 2, pp. 176–182, 2021.
- [29] E. Pirdia Wanti, A. Pariyandani, S. Zulkarnain Syed Idrus, and A. Hasudungan Lubis, “Utilization of SVM Method and GLCM Feature Extraction in Classifying Fish Images with Formalin,” *Sci. J. Informatics*, vol. 8, no. 1, 2021.
- [30] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv*, 2018.