International Journal of Engineering, Science & InformationTechnology (IJESTY)
Volume 2, No. 3 (2022) pp. 64-72
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v1i4.291
Research Paper, Short Communication, Review, Technical Paper

Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO V4 as Object Detector

Marsa Mahasin, Irma Amelia Dewi

Department of Informatics, National Institute of Technology, Bandung, Indonesia
**Corresponding author E-mail: mahasingrad@mhs.itenas.ac.id*

Manuscript received 15 April 2022; revised 1 May 2022; accepted 15 June 2022. Date of publication 25 July 2022

Abstract

YOLO v4 has a structure consisting of 3 parts: backbone, neck, and head. The backbone is a part of the YOLO v4 structure that serves as a feature extractor from the image; the backbone is also a convolutional neural network that can be replaced with another convolutional neural network. Many backbones are recommended by previous research, such as CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0. Therefore, research needs to be done to determine the effect of different backbones on the YOLO v4 model. One of the research objects that can be used is a microfossil. Research on the detection of microfossils is fundamental to assist paleontologists in knowing the species of microfossils as a determinant of rock age and distinguishing between similar microfossils. In this research three backbones consisting of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 were used to train and detect image sets of 5 species of foraminiferal microfossils and the results were evaluated to determine the advantages of each backbone. There are a few metrics are that being used for evaluation, namely precision, recall, f1-score, average precision (AP), mean average precision (mAP), frames per second (FPS), and model size. As a result, the mean average precision (mAP) of the CSPDarkNet53 model reached 83.41%, the highest compared to CSPResNeXt-50 and EfficientNet-B0 which get a value of 81,00% and 81,76%. CSPResNeXt-50 model has a precision of 75.60%, recall of 81.10%, and f1-score of 78%. CSPDarkNet53 model also got the highest FPS value of 33.4FPS. However, the YOLO v4 model with the EfficientNet-B0 backbone is the lightest model with only 156.8 MB.

***Keywords**: YOLO, CSPDarkNet53, CSPResNeXt-50, EfficientNet-B0, Microfossil*

1. Introduction

You Only Look Once (YOLO) is an algorithm based on convolutional neural network that is often used for object detection and object classification. The structure of this algorithm consists of several parts such as backbone, neck, and head, each of which has a different function. The YOLO algorithm has advantages over other one-stage detectors such as RetinaNet and Single Shot Multibox Detector (SSD) when used in real-time conditions where the YOLO algorithm produces larger frames per second (FPS) and a lighter model size so that the detection capability is better [1]. The backbone of the YOLO algorithm acts as a feature extractor from the input image. The types of backbones are CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0. These are backbones that can be used in the YOLO algorithm and produce up to 70% accuracy while still detecting objects up to 40 FPS [2].
Microfossil objects are closely related to biostratigraphy, namely the science of determining the age of rocks using the fossils contained therein. The complex morphology of microfossils requires the use of specialists for correct systematics, especially to produce detailed and accurate biostratigraphic correlations [3]. Education and training in identifying microfossils is dwindling but technological developments allow the possibility of accelerating and standardizing the characterization and identification of fossils by machine learning [4]. the latest research on microfossil images classified using the Convolutional Neural Network (CNN) with 7 different models, namely VGG-19, Inception-ResNetV2, MobileNetV2, ResNet50, Xception, NASNetMobile, and DenseNet121. This study concludes that CNN with the ResNet50 model has the greatest accuracy of 81.8%, 76.7% precision, and 71.4% recall [5].
The backbone of the YOLO algorithm acts as a feature extractor from the input image. In research on YOLO v4 entitled "YOLOv4: Optimal Speed and Accuracy of Object Detection", RetinaNet, EfficientDet-D0, RFBNet, NAS-FPN, ATSS, RDSNet, CenterMask, LRF, Faster R-CNN, M2det, SSD, and TridentNet tested with YOLO v4. The result, RetinaNet and EfficientDet-D0 achieved FPS and AP values closer to YOLO v4 than other detectors. YOLO v4 with CSPDarkNet53 backbone scored 96 FPS and 41.2% AP. EfficientDet-D0 scored 62.5 FPS and 33.8% AP. While RetinaNet scored 37 FPS and an AP of 37%. The CSPResNeXt-50 backbone is used on RetinaNet and EfficientNet-B0 is used on EfficientDet-D0 [2].
Based on previous research, further research is needed to validate the hypotheses from previous researches and determine the effect of using different backbones on the YOLO v4. The effect of this backbone can be evaluated by mean average precision (mAP), average

precision (AP) at several intersection over union (IoU) scales, f1-score, and frames per second (FPS). Therefore, this study examines the effect of the CSPResNeXt-50, CSPDarkNet53, and EfficientNet-B0 backbones on the YOLO v4 model as object detectors. With this research, the influence of the backbone on the YOLO v4 configuration can be seen clearly through the evaluation of the research results.

2. Literature Review

2.1. You Only Look Once (YOLO)

YOLO or You Only Look Once is one of the single-stage object detectors besides SSD (Single Shot Detector). YOLO is a new approach in the realm of object detection. With YOLO, object detection is done by viewing the problem as a regression problem to spatially separate the bounding box and the probability classes associated with the bounding box. A neural network predicts the bounding box and prediction class directly from the entire image from a single evaluation [6].

The fourth version of YOLO adds several additions to its network to improve accuracy and efficiency, namely:

a. Weighted Residual Connection which learns to combine residues from neural network layers effectively and efficiently [7].
b. Cross Stage Partial Connections (CSP) integrates feature maps from the initial stage to the final stage of the network. CSP implementation reduces compute by as much as 20% and outperforms other state-of-the-art backbone architectures [8].
c. Cross Mini-Batch Normalization overcomes the problem where the statistics generated when normalization is defined cannot be estimated properly [9].
d. Self-adversarial Training is a new data augmentation technique that operates in 2 stages forward and backward on the network [10].
e. Mish-activation is a non-monotone activation function that is self-regulating [11].
f. Mosaic Data Augmentation is a data augmentation technique by combining several images in a dataset into one. DropBlock regularization is used as a new regularization method for CNN [12].
g. CIoU loss, a loss function that achieves faster convergence and better accuracy in bounding box regression problems [13].

In addition, YOLO v4 also uses Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN) on the neck. Spatial Pyramid Pooling serves to eliminate fixed network size limitations [14]. While the Path Aggregation Network serves to improve the flow of information in the segmentation instance. In particular, the feature hierarchy is enhanced by accurate localization signals in the lower layers with bottom-up path aggregation, which shortens the information path between the lower layers and the top features [15]. For the head, it still uses the head from YOLO v3 which uses 3 detectors of different sizes [16].

YOLO v4 is also based on Darknet and has obtained an AP value of 43.5 percent on the COCO dataset along with a real-time speed of 65 FPS on the Tesla V100, beating the fastest and most accurate detectors in terms of speed and accuracy and when compared to YOLO v3, AP and FPS increased by 10 percent and 12 percent, respectively [2]. The structure of YOLO v4 were shown in Fig. 1.

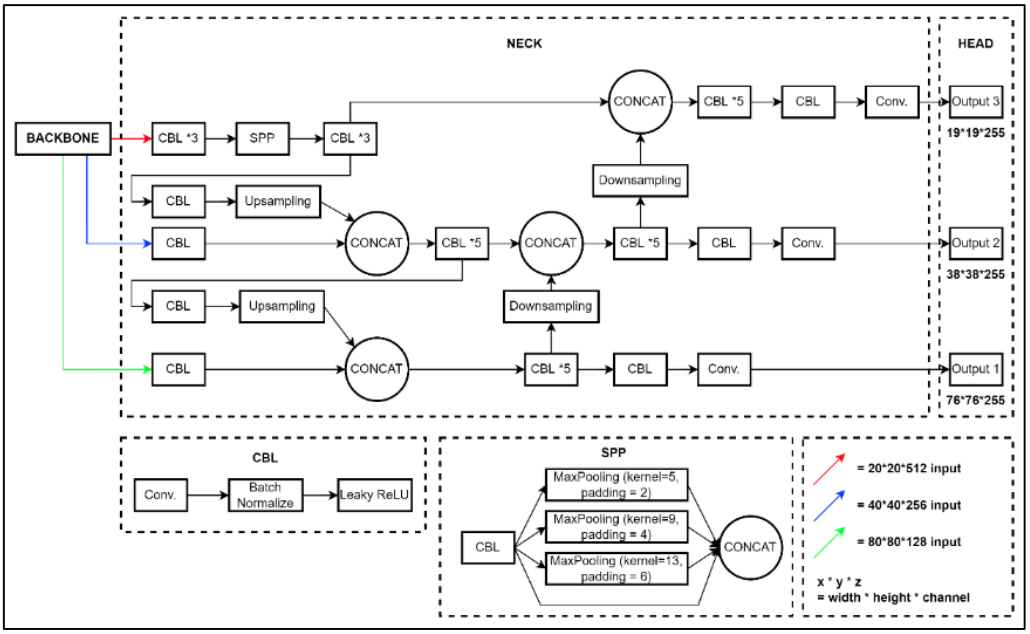

Fig 1. YOLOv4 structure

2.2. CSPDarkNet53

CSPDarkNet53 is the default backbone of the YOLO v4. This backbone is a development of the backbone in YOLO v3, namely DarkNet53 which has added the Cross Stage Partial Connections (CSP) feature which integrates feature maps from the initial stage to the final stage of the network. CSP implementation reduces computation by as much as 20% thereby outperforming other state-of-the-art backbone architectures [2]. The structure of CSPDarkNet53 is shown in Figure 2.

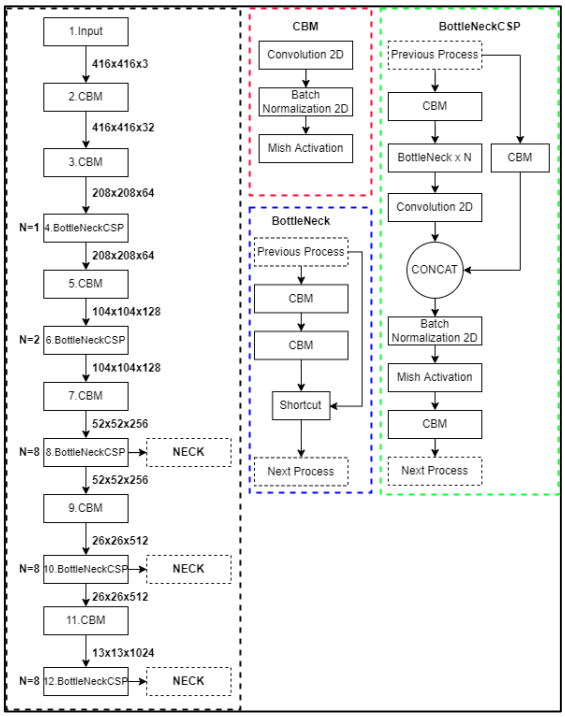

Fig 2. CSPDarkNet53 structure

2.3. CSPResNeXt-50

CSPResNeXt-50 is a Convolutional Neural Network algorithm that applies the Cross Stage Partial Network (CSPNet) approach to ResNeXt. ResNeXt itself is an algorithm that repeats building blocks that combine a series of transformations with the same topology [7]. Compared to ResNet, ResNeXt exhibits a new dimension, cardinality (size of the transformation set), as an important factor besides depth and width dimensions [8]. CSPResNeXt-50 uses Leaky-ReLU Activation, which is an extension of ReLU Activation [17]. The structure of CSPResNeXt-50 is shown in Figure 3.

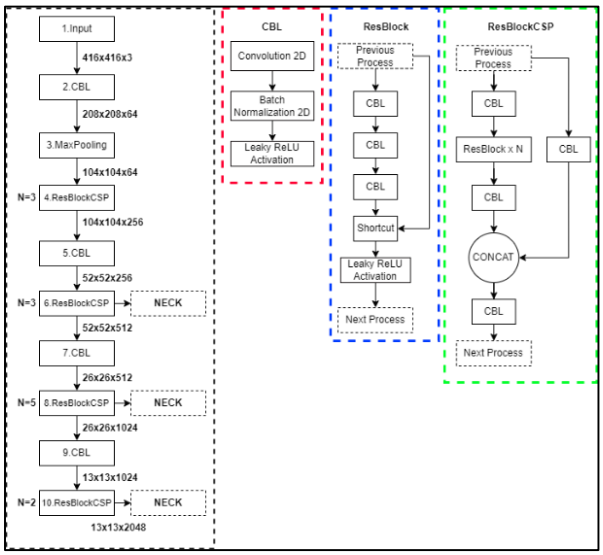

Fig 3. CSPResNeXt-50 structure

2.4. EfficientNet-B0

EfficientNet is a convolutional neural network architecture and a scaling method that scales all depth/width/resolution dimensions uniformly using combined coefficients. The EfficientNet scaling method uniformly scales the network width, depth, and resolution with a fixed set of scaling coefficients [18]. The network on EfficientNet is taken from the results of the MobileNetV2 inversion with the addition of squeeze-and-excitation [19]. EfficientNet-B0 uses swish activation in the process [20]. The structure of EfficientNet-B0 is shown in Figure 4.

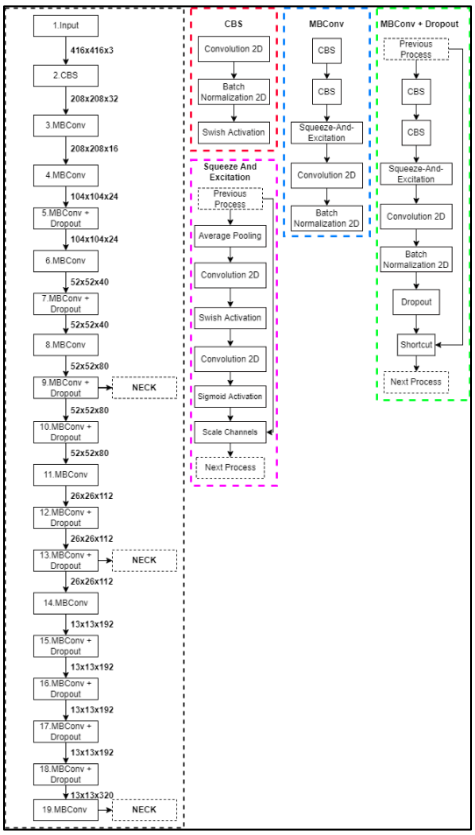

Fig 4. EfficientNet-B0 structure

2.5. Evaluation Metrics

This study uses metrics commonly used to evaluate the model as a reference for comparison of each backbone. There are 4 categories that can be generated from the comparison of the detection results with the actual label, namely True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Furthermore, the results will be presented through standard measurements in the form of precision, recall, f1-score, average precision (AP), mean average precision (mAP), and frames per second (FPS). The formulas of precision, recall, AP, mAP, and f1-score are presented in Equation (1), (2), (3), (4), and (5).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F_1 = \frac{2PR}{P+R} \quad (3)$$

$$AP = \int_0^1 p(r)dr \quad (4)$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} APi \quad (5)$$

Definition:
TP = True Positive
FP = False Positive
FN = False Negative
P = Precision
R = Recall
AP = Average Precision

3. Methods

There are two stages in this research, the first stage is the data training stage. At this stage, the training data will be trained through a series of backbone, neck, and head on YOLOv4 3 times using different backbones so as to produce a YOLO model that has different weights. The stage is shown in Figure 5.

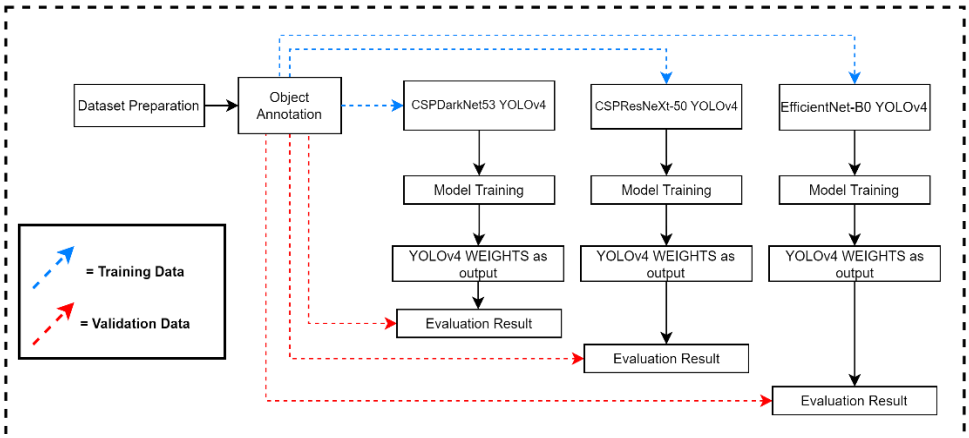

Fig 5. Training Stage

The second stage is the data testing stage where the validation data will be compared with the weights obtained from the data training stage, the result is a comparison between the bounding box ground truth and the bounding box detection results. The stage is shown in Figure 6.

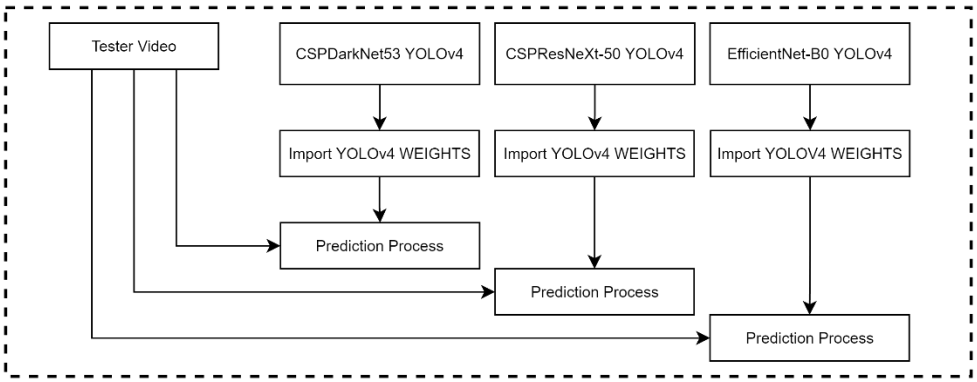

Fig 6. Testing Stage

3.1. Dataset Preparation

At this stage, the dataset uses 5 species of foraminiferal microfossils, each of which has its own unique shape. Microfossil images are taken after going through the preparation process first. Preparation is the process of separating fossils from rocks and other impurities. This process generally aims to separate the microfossils contained in the rock from the clay material (matrix) that surrounds it. The dataset was downloaded from endlessforams.org with a total of 1082 training data and 226 validation data, as shown in Table 1.

Table 1. Microfossils Dataset

No	Microfossil Species	Training Data	Validation Data
1.	Bulimina tenuata	200	40
2.	Takagamiya delicata	290	58
3.	Uvigerina peregrina	200	50
4.	Bolivina argentea	192	38
5.	Globigerinella siphonifera	200	40
	Total	1082	226

3.2. Object Annotation

Before the dataset can be used, the dataset needs to be labelled so that the species of the microfossil and its location in the image can be identified. The labelling process is carried out using the LabelImg application. Labelling through this application uses the YOLO format so as to produce a "txt" format file that has the same name as each image. The contents of this "txt" file consist of image categories arranged by number sequence as well as 4 coordinate points which are the boundaries of the object in the image. As seen in Figure 7.

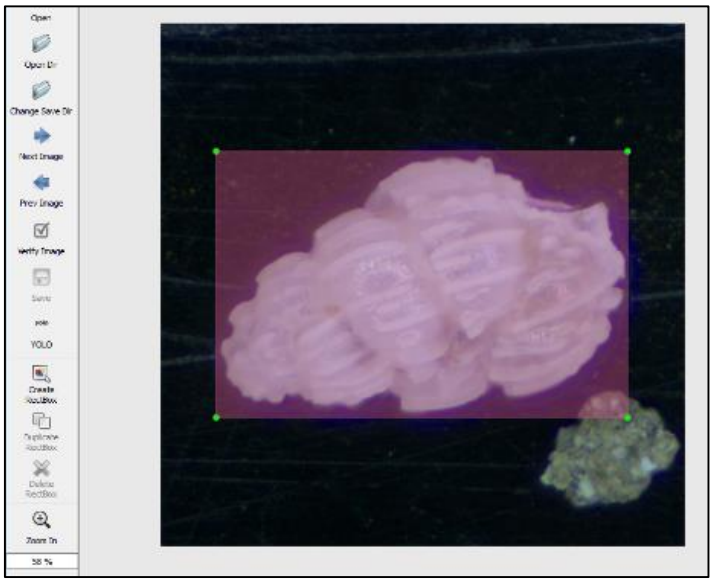

Fig 7. Image Labelling

3.3. Setting the Configuration File

The next step is to create a configuration file that is used to train the dataset and generate the YOLO v4 model. At this stage the 3 configuration files each have a different backbone. The differences between each backbone are described in Table 2.

Table 2. Backbone Comparison

	CSPDarkNet53	CSPResNeXt-50	EfficientNet-B0
Activation Type	Mish Activation	Leaky ReLU Activation	Swish Activation
Cross Stage Partial	√	√	
Squeeze-and-Excitation			√

Each backbone has a different number of stages, the number of these stages affects the time needed to carry out training. In addition, the types of activation that the three backbones have are also different. The activation function is a non-linear function that functions to convert linear inputs to non-linear ones after convolution. Activation used on each backbone has different functions. Then the CSPDarkNet53 and CSPResNeXt-50 backbones use Cross Stage Partial (CSP) which functions to relate problems with duplicate gradient information in network optimization, complexity can be reduced while maintaining accuracy. However, the EfficientNet-B0 backbone does not use CSP and only uses Squeeze-and-Excitation. Squeeze-and-Excitation serves to increase the strength of network representation by recalibrating dynamic channels. Each channel is squeezed into a single numeric value using average pooling, then convolution followed by Swish adding non-linearity and reducing the complexity of the output channel and followed by sigmoid which gives each channel a smooth gating function.

The configuration is managed through a "cfg" format file used during data training and data validation. This configuration file contains a series consisting of hyperparameters, backbone, neck, and head.

3.4. Training Models

The device used has the following specifications: OS, Windows 10; GPU, Nvidia Tesla P100. The training is carried out through the Google Colab with the Darknet framework. The parameters used were adjusted to the optimal parameters used in the initial research on YOLO v4 [2]. All trainings were carried out in the same environment with the parameter specifications described in Table 3.

Table 3. Parameter Configuration

Batch Size	64
Max Batches	10000
Subdivision	16
Width x Height	416 x 416
Data Augmentation	Mosaic Augmentation
Learning Rate	0,001

4. Results

In accordance with the max batches used, data training is carried out for 10000 iterations. However, YOLO v4 can perform early stopping training model automatically by taking the training results in certain iterations that produce the largest mean average precision (mAP), but training is continued to get the full graph. The graph on Darknet takes calculation data starting from the 1000th iteration so that the graph also starts from the 1000th iteration. Early stopping model training serves to stop model training and prevent model overfitting. The results of the training in the form of graphs of average loss and mAP for 10000 iterations can be seen in Figure 8, 9, and 10.

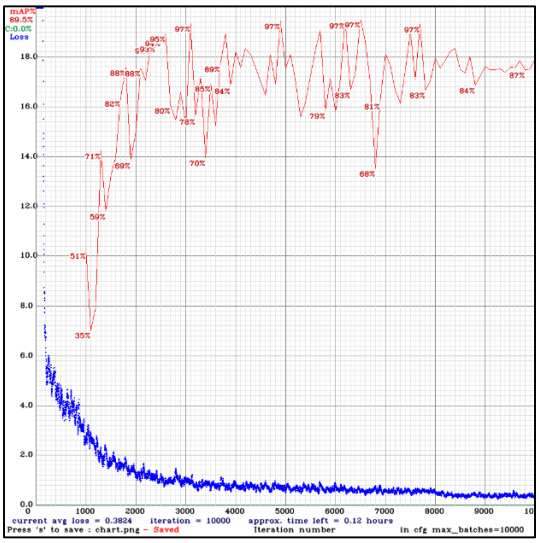

Fig 8. CSPDarkNet53 model result

YOLO v4 with CSPDarkNet53 backbone has been trained for 10000 iterations with an average loss of 0.3824. In the 4800th iteration, early stopping was carried out because the mAP value had shown a large number (97%) and the average loss had not decreased significantly.

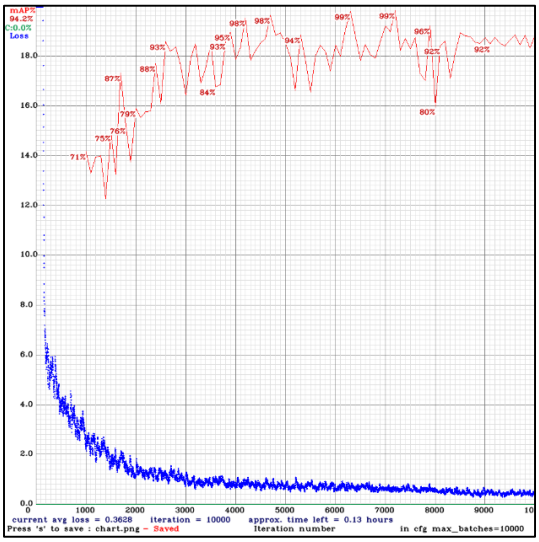

Fig 9. CSPResNeXt-50 model result

YOLO v4 with a CSPResNeXt-50 backbone has been trained for 10000 iterations with an average loss of 0.3628. In the 6500th iteration, early stopping was carried out because the mAP value had shown a large number (99%) and the average loss had not decreased significantly.

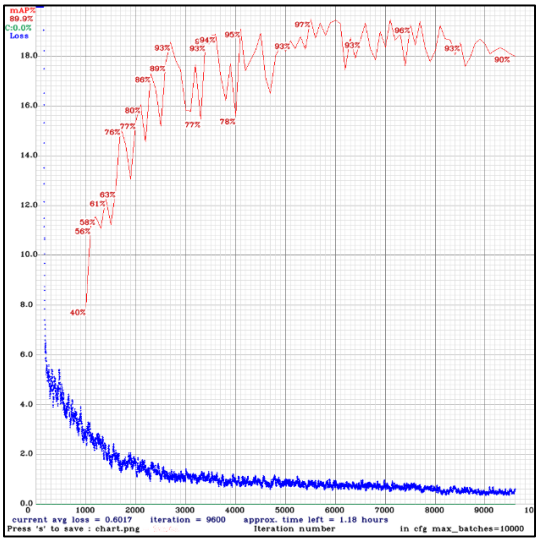

Fig 10. EfficientNet-B0 model result

YOLO version 4 with EfficientNet-B0 backbone has been trained for 10000 iterations with an average loss of 0.6017. In the 5400th iteration, early stopping was carried out because the mAP value had shown a large number (97%) and the average loss had not decreased significantly.
After the test was successfully carried out, the values of precision, recall, f1-score, average precision (AP), frames per second (FPS) and mean average precision (mAP) were obtained from each test. The test results for each backbone can be seen in Table 4, 5, and 6.

Table 4. CSPDarkNet53 Result

IoU(%)	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95
Precision (%)	72	72	72	72	72	71	70	67	54	11
Recall (%)	90	90	90	90	90	90	88	85	68	14
F1-Score (%)	80	80	80	80	80	79	78	75	60	12
AP (%)	97,48	97,48	97,48	97,48	97,48	96,61	95,07	88,12	62,65	4,26

Table 5. CSPResNeXt-50 Result

IoU	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95
Precision (%)	89	89	89	89	89	89	87	80	50	5
Recall (%)	96	96	96	96	96	95	93	85	53	5
F1-Score (%)	92	92	92	92	92	92	90	82	51	5
AP (%)	99,08	99,08	99,08	99,08	99,08	98,25	94,77	82,45	38,28	0,91

Table 6. EfficientNet-B0 Result

IoU	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95
Precision (%)	73	73	73	73	73	73	71	68	48	13
Recall (%)	92	92	92	92	92	92	90	86	60	16
F1-Score (%)	82	82	82	82	82	82	80	76	53	14
AP (%)	97,43	97,43	97,43	97,43	97,43	97,43	93,83	85,62	48,74	4,84

Based on 3 tests using 3 different backbones, several values can be used as a comparison for each backbone. One of the values that can be analysed and compared is the average precision (AP) value at a certain intersection over union (IoU). The IoU value of 0.0 has no intersection between the bounding box and ground truth, while the IoU value of 0.95 has almost complete intersection. The comparison can be seen in Table 7.

Table 7. Average Precision Comparison

Backbone	AP^{50}	AP^{55}	AP^{60}	AP^{65}	AP^{70}	AP^{75}	AP^{80}	AP^{85}	AP^{90}	AP^{95}
CSPDarkNet53	97,48%	97,48%	97,48%	97,48%	97,48%	96,61%	**95,07%***	**88,12%***	**62,65%***	4,26%
CSPResNeXt-50	**99,08%***	**99,08%***	**99,08%***	**99,08%***	**99,08%***	**98,25%***	94,77%	82,45%	38,28%	0,91%
EfficientNet-B0	97,43%	97,43%	97,43%	97,43%	97,43%	97,43%	93,83%	85,62%	48,74%	**4,84%***

The prediction of YOLO v4 with a CSPResNeXt-50 backbone gets the highest AP value in the IoU value range of 0.5 (50%) to 0.75 (75%). Meanwhile, YOLO v4 with a CSPDarkNet53 backbone got the highest AP value in the IoU value range of 0.8 (80%) to 0.9 (90%) and the EfficientNet-B0 backbone got the highest AP value at an IoU value of 0, 95 (95%). Overall, the mAP (Mean Average Precision), Precision, Recall, and F1-Score values of the three backbones can be seen in Table 8.

Table 8. Overall Comparison

Backbone	Precision (%)	Recall (%)	F1-Score (%)	mAP (%)
CSPDarkNet53	63,33%	79,50%	70,40%	**83,41%***
CSPResNeXt-50	**75,60%***	**81,10%***	**78%***	81,00%
EfficientNet-B0	63,80%	80,40%	71,50%	81,76%

CSPDarkNet53 backbone has a higher mAP (Mean Average Precision) value than the other backbones, this value is caused by the AP value of CSPDarkNet53 which has 62% predictive data at an IoU of 0.9. This is different from CSPResNext50 and EfficientNet-B0 which only have 38% and 48% predictive data on the IoU, respectively. However, the largest Precision, Recall, and F1-Score values belong to the CSPResNext50 backbone. This value is obtained from the large Precision, Recall, and F1-Score values in the IoU range of 0.5 to 0.85. Also, the frames per second (FPS) value of the video had been processed using the three YOLO v4 models with different backbones. The test results can be seen in Table 9.

Table 9: Video Testing

Backbone	Model Size	FPS
CSPDarkNet53	244,2MB	**32***
CSPResNeXt-50	216MB	31,6
EfficientNet_b0	**156,8MB***	30,5

The highest FPS was obtained from video that were processed through the YOLO v4 model with a CSPDarkNet53 backbone. However, the difference between the highest and lowest FPS is only 1.5 FPS, so there is no significant difference between the three models. The absence of significant differences is caused by the structure of the three models which are still similar in the arrangement of the backbone, neck, and head. It is different from other detectors which have different structures, as in previous studies regarding the comparison of

RetinaNet, SSD, and YOLO v3 which RetinaNet was only able to achieve an FPS of 22 FPS, while YOLO v3 was able to achieve an FPS of 69 FPS [1].

5. Conclusion

Research began by looking for a dataset in the form of foraminiferal microfossil images, then continued with labelling the image using the LabelImg application, then the dataset was trained using the YOLO v4 algorithm with 3 different backbones, namely CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0. By comparing the three models of the backbone, each model has its own advantages and disadvantages.

YOLO v4 with CSPDarkNet53 backbone has the largest mAP of 83.41% (0.83411) compared to the YOLO v4 with CSPResNeXt-50 and EfficientNet-B0 backbones which received mAP of 81,01% (0.81005) and 81,76% (0.81761). These values were tested on 10 different IoU ranges, starting from IoU 0.5 to IoU 0.95. YOLO v4 with CSPResNeXt-50 backbone scored the highest Precision, Recall, and F1-Score compared to the other 2 YOLO v4 models with scores of 75.6%, 81.1%, and 78%. YOLO v4 model with CSPResNeXt-50 backbone gets the highest AP value in the IoU range of 0.5 to 0.75. YOLO v4 with CSPDarkNet53 backbone got the highest AP value in the IoU range of 0.8 to 0.9. The YOLO v4 with EfficientNet-B0 backbone got the highest AP value at IoU 0.95. Then through testing using video, YOLO v4 with a CSPDarkNet53 backbone got the highest frames per second (FPS) of 32 FPS.

Acknowledgement

The author would like to thank his parents for always believing on their child in ups and downs. Also, thanks to all the author friends which is impossible to write all of them here. And lastly, the author would like to thank to Informatics Study Program at National Institute of Technology, Bandung, for all the supports that have been given to the author.

References

[1] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification.," *BMC Med. Inform. Decis. Mak.*, vol. 21, no. 1, p. 324, Nov. 2021, doi: 10.1186/s12911-021-01691-8.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *CoRR*, vol. abs/2004.1, 2020.

[3] C. Gaucher and D. Poiré, "Chapter 4.3 Biostratigraphy," in *Neoproterozoic-Cambrian Tectonics, Global Change And Evolution: A Focus On South Western Gondwana*, vol. 16, C. Gaucher, A. N. Sial, H. E. Frimmel, and G. P. Halverson, Eds. Elsevier, 2009, pp. 103–114.

[4] R. P. de Lima *et al.*, "Convolutional Neural Networks as an Aid to Biostratigraphy and Micropaleontology: A Test On Late Paleozoic Microfossils," *Palaios*, vol. 35, pp. 391–402, 2020.

[5] R. Marchant, M. Tetard, A. Pratiwi, M. Adebayo, and T. De Garidel-Thoron, "Automated analysis of foraminifera fossil records by image classification using a convolutional neural network," *J. Micropalaeontology*, vol. 39, no. 2, pp. 183–202, 2020, doi: 10.5194/jm-39-183-2020.

[6] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *CoRR*, vol. abs/1506.0, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.0, 2015.

[8] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "CSPNet: {A} New Backbone that can Enhance Learning Capability of {CNN}," *CoRR*, vol. abs/1911.1, 2019.

[9] Z. Yao, Y. Cao, S. Zheng, G. Huang, and S. Lin, "Cross-Iteration Batch Normalization," *CoRR*, vol. abs/2002.0, 2020.

[10] C.-J. Chou, J.-T. Chien, and H.-T. Chen, "Self Adversarial Training for Human Pose Estimation," *CoRR*, vol. abs/1707.02439, 2017, [Online]. Available: http://arxiv.org/abs/1707.02439.

[11] D. Misra, "Mish: {A} Self Regularized Non-Monotonic Neural Activation Function," *CoRR*, vol. abs/1908.0, 2019.

[12] G. Ghiasi, T.-Y. Lin, and Q. V Le, "DropBlock: {A} regularization method for convolutional networks," *CoRR*, vol. abs/1810.1, 2018.

[13] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *CoRR*, vol. abs/1406.4, 2014.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," *CoRR*, vol. abs/1803.0, 2018.

[16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.0, 2018.

[17] J. Park, J. Baek, J. Kim, K. You, and K. Kim, "Deep Learning-Based Algal Detection Model Development Considering Field Application," *Water*, vol. 14, no. 8, 2022, doi: 10.3390/w14081275.

[18] M. Tan and Q. V Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *CoRR*, vol. abs/1905.1, 2019.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," *CoRR*, vol. abs/1709.0, 2017, [Online]. Available: http://arxiv.org/abs/1709.01507.

[20] P. Ramachandran, B. Zoph, and Q. V Le, "Searching for Activation Functions," *CoRR*, vol. abs/1710.0, 2017, [Online]. Available: http://arxiv.org/abs/1710.05941.