Indonesian Journal of Physics
Vol. 35. No.1, 23 August 2024
Journal homepage: http://ijphysics.com

Application of YOLOv5s Algorithm for Real-Time Object Detection in Mobile Robot for Volcano Monitoring System

Maria Evita[1*], Sekar Tanjung Mustikawati1[1], Mitra Djamal[1]
[1]Department of Physics, Faculty of Mathematics and Natural Sciences, Jalan Ganesha No. 10, Bandung 40116, Indonesia

(Received: 18 January 2024, Revised: 21 June 2024, Accepted: 25 June 2024)

Abstract

Indonesia, a country with 172 volcanoes and second after Japan for the most eruption events, should monitor and predict the volcano eruption to prevent the effect of this natural disaster. Therefore, we have developed a 4-wheeled mobile robot equipped with monitoring sensors and a Logitech camera for this purpose. The robot should have the ability to detect objects in this extreme environment to avoid collision while moving and monitoring the volcano's physical parameters. It has been designed a deep machine learning of YOLOv5s algorithm for two objects mostly found at volcanoes such as trees and stones. After the training steps (object identification; dataset downloading (Google Chrome Extension and Open Images v6); image labeling (LabeImg); augmentation process (blur and rotation)) had been carried out, the images of the object then trained in three model variation which resulted in: mAP_0.5 = 51.9%, mAP_0.5:0.95 = 28.6%, 58% of precision and 50% recall with 12 minutes and 33 seconds of training time for the first model (batch=16 and epochs=100); mAP_0.5 = 59.7%, mAP_0.5:0.95 = 36.3%, 74% of precision and 54% recall with 36 minutes and 4 seconds of training time for the second model (batch=16 and epochs=300); mAP_0.5 = 59.9%, mAP_0.5:0.95 = 37.6%, 80% of precision and 55% recall with one hour and 25 seconds of training time for the last one (batch=16 and epochs=500) as the best model of these variations. Furthermore, these results were displayed for all test images for the best model.

Keywords: mobile robot, object detection, volcano monitoring system, YOLO

INTRODUCTION

Indonesia which has 172 volcanoes and is the second most eruption events after Japan, should monitor and predict the volcano eruption to prevent the effect of this natural disaster. Therefore, we have developed a 4-wheeled mobile robot equipped with monitoring sensors and a Logitech camera for this purpose in a volcano monitoring system called MONICA [1-10]. The robot should have the ability to detect objects in this extreme environment to avoid collision while moving and monitoring the volcano's physical parameters [11-15]. We have applied a YOLOv5s (the latest and the best version of YOLO [16,17] algorithm (open source) for this purpose by training 4 objects: persons, trees, stones and stairs. The best result for this training was batch 16 and epoch 500. Therefore, in this research, we have trained the datasets of two main objects usually found in volcanoes (trees and stones) [18] in batch 16 and different epochs (100, 300 and 500 [10]).

The choice of YOLOv5s for object detection is driven by several compelling reasons. Its high frame rate makes it ideal for real-time applications where rapid detection is crucial [19]. The algorithm's advanced detection capabilities ensure high precision, reducing false positives and enhancing the reliability of navigation and surveillance systems [19]. The adaptability of YOLOv5s across various domains—from robotics to environmental monitoring—demonstrates its robustness and suitability for diverse use cases [13]. Additionally, YOLOv5s's efficiency on low-power devices, such as the Raspberry Pi, allows for its deployment in resource-constrained environments, making it a practical choice for edge computing applications [20]. Therefore, YOLOv5s stands out due to its advanced detection capabilities, making it the preferred choice for applications demanding both

1* Corresponding author. E-mail address: maria@fi.itb.ac.id

speed and accuracy. Its integration into diverse systems—from mobile robots to smart surveillance—highlights its transformative impact on real-time object detection technologies.

Some works have been reported for YOLO application. Conrad and DeSouza's work on mobile robot navigation relied on a modified Expectation Maximization (EM) algorithm to segment object images from the ground plane, highlighting the importance of accurate object classification in navigation tasks such as obstacle avoidance and path planning [21]. This research established a foundation for the need to improve classification methods, which YOLOv5s addresses with its enhanced detection capabilities.

Moreover, advancements in Unmanned Surface Vehicles (USVs) have been documented by Garcia et al., emphasizing the critical role of obstacle avoidance in achieving optimal performance in environmental missions [13]. This research underscores the importance of robust object detection algorithms like YOLOv5s in enhancing navigation capabilities in various environments.

Additionally, Adam Gunnarsson's 2019 study compared object detection methods, including SSDLite and YOLOv3-tiny, using the COCO dataset to assess the feasibility of real-time object detection on a Raspberry Pi [20]. This benchmark study provided insights into the performance metrics of different algorithms, demonstrating the need for an improved solution like YOLOv5s that offers both high speed and accuracy.

Layek et al.'s development of a cloud-based smart surveillance system using Raspberry Pi and YOLO-based object recognition also serves as a benchmark [22]. This study showcased the integration of basic motion analysis on edge devices and detailed object detection in the cloud, highlighting the practical applications and benefits of using advanced detection algorithms such as YOLOv5s.

METHOD

Labelling

Some robots have been developed to have obstacle avoidance mechanisms for static objects such as persons or dynamical objects such as trees, light poles, trash cans, stones, etc. [23] in an urban area. Meanwhile, the robot in this research should have the ability to detect the object in front of it, to avoid obstacles in the volcano area especially when the volcano erupts. Therefore, it should be only trees and stones found in the area to be avoided by the robot, not including humans as mentioned in our previous research [10].

Thereafter, the object images were set as the dataset for the training process in YOLOv5s – a better algorithm than Fast R-CNN in background patch error, accuracy and ease to use in low-specification hardware [24-26]. Labeling as the first process (Fig. 1) where the images were highlighted by adding a label and object bounding box segmented [27] was carried out by LabelImg [10,28] (Fig. 2). The datasets were higher resolution images than our previous work to improve the quality of the training result [10].

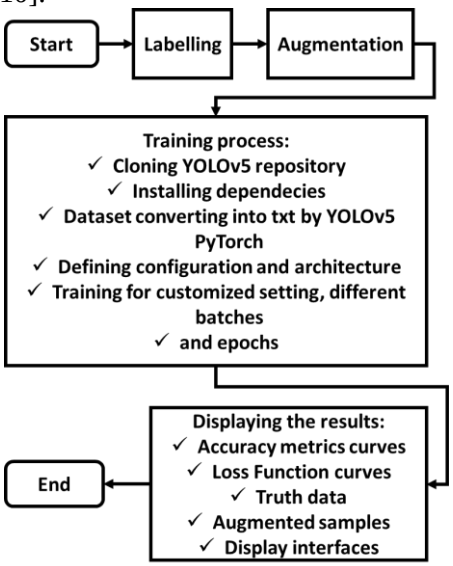

Fig. 1. Flowchart of the experimental design.

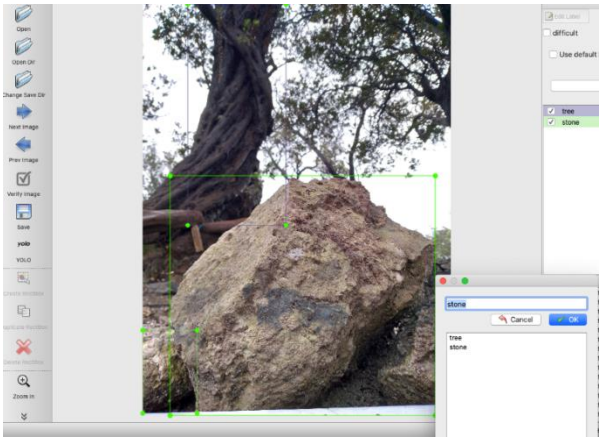

Fig. 2. Labelling of tree and stone in LabelImg.

Augmentation

Model performance could be improved by augmenting the dataset to produce a real dataset by flipping, rotating, cropping, adding noise, occluding portions of the image, etc. [29]. Hence, the model could face more situations by learning more datasets produced during augmentation. In this research, datasets were blurred and rotated for the robot to detect the object in a foggy environment and move on uneven terrain [10].

Training

The dataset was online trained in an open-source platform Google Colab using Phyton versions 2 and 3, with no installation and configuration, free access to GPU and TPU (for a faster training process) and an environment for Jupyter Notebook [10]. After the data had been divided into 3 parts: 70% for training, 20% for validation and 10% for testing [30], the yolo5 model (including yolov5s.yaml, yolov5m.yaml, yolov5l.yaml, yolov5x.yaml, etc.), utils (for analysis and graphic plotting), weights were cloned to YOLOv5 repository in PyTorch. Some dependencies for programming the (unified) detection [31] training and inference command environment were installed before the dataset was converted into .txt of YOLOv5 PyTorch. Furthermore, the data were trained after the YOLOv5s architecture had been configured [10]. The results are presented in accuracy metrics graphs (mean Average Precision for 0.5 Intersection over Union (IoU) of the neuron network cells, between 0.5 and 0.95 of IoU; precision and recall) and loss function (associated with IoU loss [32-34] graphs (bounding box, classification and object losses) [10]. Some trained samples are also presented in this paper: the result of the truth data to show the real object boundary, augmented images and display interface of tested images.

Average Precision (AP) is a common metric for the accuracy of the detection process by computing the precision average of the recall number between 0 to 1 [35].

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

where p(r) is the precision as the function of the recall. Precision quantifies the training accuracy (in percent) by

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

where TP (True Positive) is the result of the true positive class prediction of the model and FP (False Positive) is the failed one. Meanwhile, recall is a parameter of how well the model finds all the positive value data

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

where FN (False Negative) shows how the model fails to detect the negative class.

Furthermore, the loss function without a sub-optimal solution is performed with the SGD (Stochastic Gradient Descent) method [36, 37] for its weight (w) and bias (b)

$$w := w - \alpha \frac{\partial}{\partial w} J(w) \quad (4)$$

and

$$b := b - \alpha \frac{\partial}{\partial b} J(b) \quad (5)$$

where α is the learning rate to control the iteration and

$$J(w,b) = -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(\hat{y} - y)^2 \quad (6)$$

where i is single training data, m is the number of the data, $\hat{y}$ is different samples in the dataset, and y is the label of data [35]. Moreover, the Sigmoid (a non-linear active function) and Binary Cross-Entropy (BCE) class were combined in the same layer of YOLOv5s PyTorch for numerical stability reasons.

RESULTS AND DISCUSSION

Model 1 (batch=16 and epochs=100)

The actual data of this model in 753 seconds are presented in thin graphs while the mean values (on a bigger scale) [10] are in the smoothed thick graphs [10] (Fig. 3).

The Mean Average Precision of mAP_0.5 (0.5 Intersection of Union) after 500 times of training reaches 51,9% (Fig. 3(A)) training quality, while mAP_0.5:0.95 reaches 28.6% (Fig. 3(B)). These results are fit for YOLOv5s compared with the pre-trained checkpoint table of the COCO (Microsoft Common Object in Context – dataset which is usually used for the YOLO dataset (the average of some IoUs) with 55.4% mAP_.5 and 36.7% mAP_.5:.95 and Bochkovskiy statement for the smallest YOLO version (YOLOv5s) for mAP_0.5 (between 26%-36%) [38]. Moreover, this model's precision fluctuates and progressively reaches 8% (Fig. 3(C)) as the recall shows 50% training quality of 14.8 MB last and best weights.

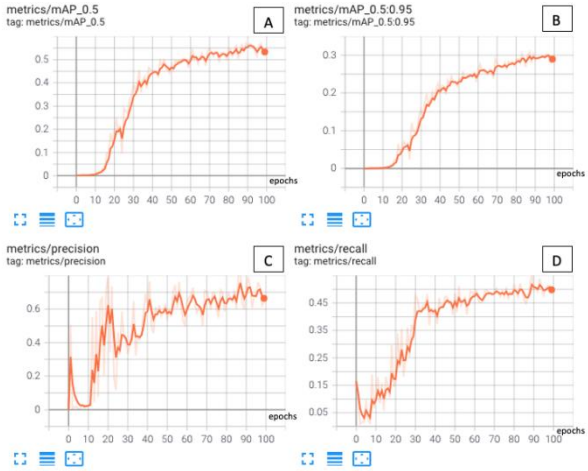

Fig. 3. The accuracy metrics of model 1: mAP_0.5 (A), mAP_0.5:0.95 (B), precession (C), and recall (D).

The losses for the bounding box, classification and object tend to zero in this model (Fig. 4 (A), (B) and (C)). Therefore, this model has a bigger probability of true events for each increasing epoch.

Furthermore, the labeling process was carried out with the numbers 0 for stones and 1 for trees the objects usually found at volcanoes. All the objects that could be precisely detected by the model (indicated by the truth data) are shown in Fig 5: a single stone (Fig. A, E, F, H, K and M), a bunch of rocks (Fig. D, I and N), a single tree (C, G and P) and a group of trees (C, G and P).

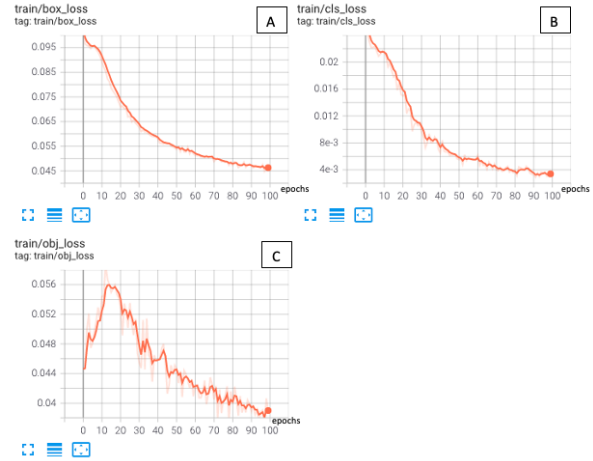

Fig. 4. The loss functions of model 1: bounding box loss (A), classification loss (B), and object loss (C).

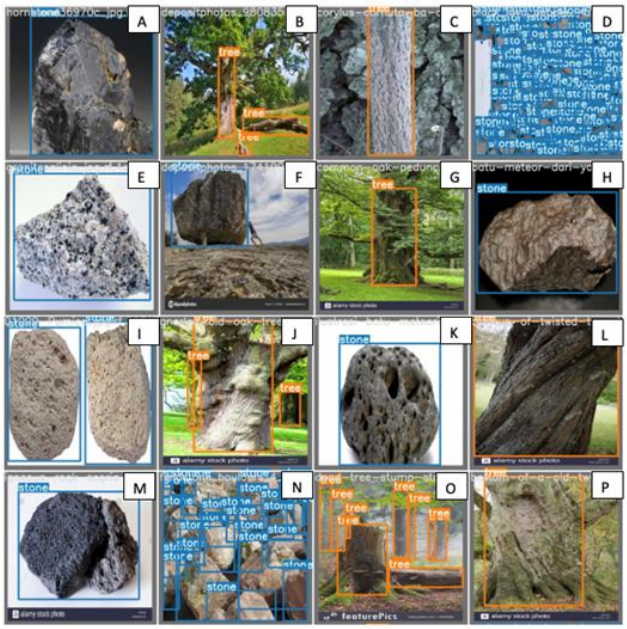

Fig. 5. The result of truth data for model 1: a single stone (A, E, F, H, K and M), a bunch of stones (D, I and M), a single tree (C, G and P), and a group of trees (B, J and O).

Furthermore, During the training data were augmented by rotating and blurring the pictures to know the reliability of the model (indicated by its weight) to detect the objects in such real volcano situations. In these situations, all objects could also be detected perfectly in Fig. 6.: a single tree (C, E, and I), a single stone (B, F, G, H, and N) and a combination of trees and stones (A, D, J, K, L, M, O and P).

However, the display interfaces show different results (Fig. 7), where objects failed to be detected. Some stones failed to be detected in Fig. C, while only one tree could be detected in Fig. D indicated by a no-precise-position of the purple bounding box as in Fig. A where a stone bounded by a less-precise green box. Moreover, a tree could not be detected entirely in Fig. B. These results were expected because of the dataset's uneven distribution and the quality of the low-resolution pictures [10].

Fig. 6. Augmented samples for model 1: a single tree (C, E, and I), a single stone (B, F, G, H, and N) and a combination of trees and stones (A, D, J, K, L, M, O and P).

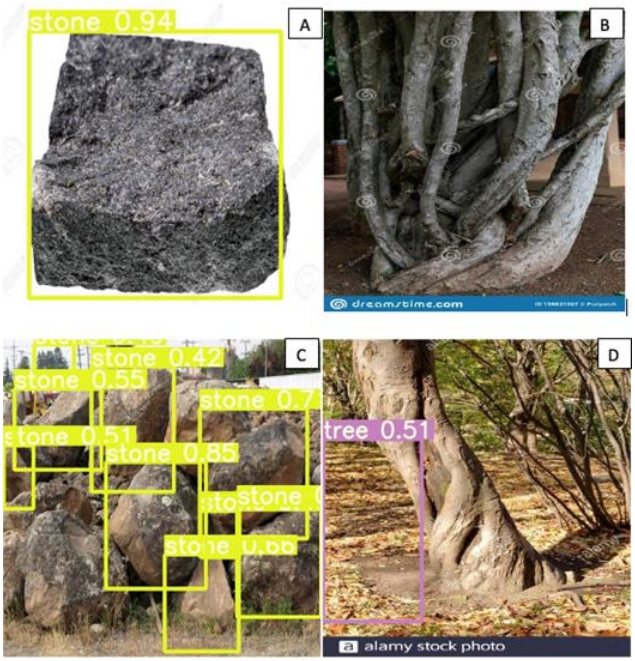

Fig. 7. The display interface of test images for model 1: a group of trees (D), a single stone (A), a bunch of stones (C) and a single tree (B).

Model 2 (batch=16 and epochs=300)

In this model, datasets were trained in the same batch, last and best weights yet three times epoch as Ultralytics default experiment for COCO dataset for 36 minutes and 4 seconds. The results are shown in Fig. 8. After 300 times of training, the mAP_0.5 of mean Average Precision (mAP) reaches 59,7% (Fig. A) and mAP_0.5:0.95 reaches 36.3%. These results are also consistent with Bochkovskiy's for mAP_0.5:0.95. This model has also been estimated by a pre-trained checkpoint table from Ultralytics using COCO datasets, which resulted in 55.4% of mAP_.5 YOLOv5s and 36.7% of mAP_.5:.95. There were 4,3% higher quality of the detection model using better quality images dataset [10]. Meanwhile, the precision (reaches 74% in Fig. C) and recall (reaches 54% in Fig. D) rise progressively in their epochs. The less precise bounding boxes could be one of the reasons for this fluctuating cycle.

These results are also confirmed by the loss graphs in Fig. 9 where all graphs (bounding box loss (A), classification loss (B), and object loss (C)) reach zero loss. The graphs describe how close the training result is to the true probability for each epoch increment. The less the loss the more the prediction represents the truth.

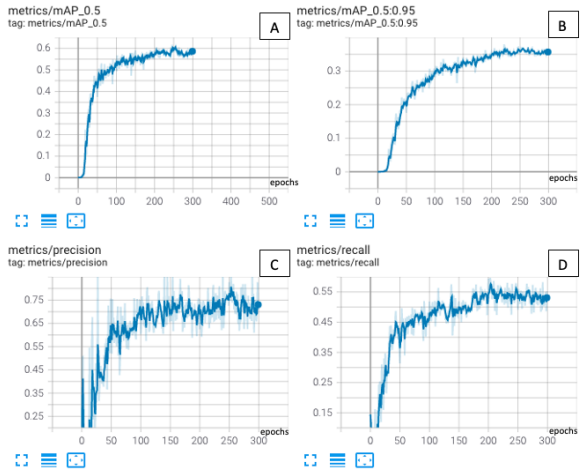

Fig. 8. The accuracy metrics of model 2: mAP_0.5 (A), mAP_0.5:0.95 (B), precession (C), and recall (D).

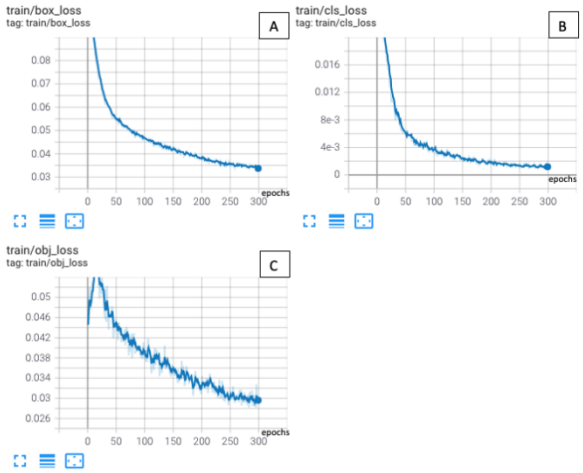

Fig. 9. The loss functions of model 2: bounding box loss (A), classification loss (B), and object loss (C).

Furthermore, the objects are successfully detected by this model (Fig.10): a single stone (C, E, I, M, N and P), a bunch of stones (A, F and L), a single tree (D, H, K and O), and a group of trees (B, G and J).

Moreover, the model could detect the objects when the dataset was also varied with two different augmentations (rotation and blur) (Fig. 11): a single tree (A, K and M), a single stone (F, J, N, O and P) and combination of trees and stones (B, C, D, E, G, H, I, and L).

Hereinafter, the display interface of the model shows that more objects could be detected than in the first model (Fig.12): a group of trees (D), a single stone (A), a bunch of stones (C) and a single tree (B). Some stones failed to be detected because of the quality of the images [10] and potentially because all images were converted into .jpg.

Fig. 10. The result of truth data for model 2: a single stone (C, E, I, M, N and P), a bunch of stones (A, F and L), a single tree (D, H, K and O), and a group of trees (B, G and J).

Model 3 (batch=16 and epochs=500)

The last model (batch=16 and epochs=500) was trained in one hour and 25 seconds and has the same last and best weight as the first and second models (14.8 MB). The mean Average Precisions are higher than the previous finding: 59,9% for mAP_0.5 and 37.6% for mAP_0.5:0.95 (Fig. 13 A and B). These results were also higher than the COCO dataset pre-trained checkpoint for the same model with 55.4% mAP_.5 YOLOv5s and 36.7% mAP_.5:.95. The precision (reaches 80%) and recall (reaches 55%) are also progressively as the previous results (Fig. C and D). therefore, the more epoch the higher the precision and recall.

Fig. 11. The augmented samples for model 3: a single tree (A, K and M), a single stone (F, J, N, O and P) and a combination of trees and stones (B, C, D, E, G, H, I, and L).

Fig. 12. The display interface of test images for model 2: a group of trees (D), a single stone (A), a bunch of stones (C) and a single tree (B).

The last loss graphs show the best result among the three variance models (Fig. 14). The bounding box reaches 0.029 (Fig. 14 A), classification loss reaches 0.0005 (Fig. 14 B), while object loss reaches 0.024 loss (Fig. 14 C).

In this training, the model could detect all the objects precisely as well as the previous ones, indicated by the result of truth data in Fig 15: a single stone (C, E, I, M, N and P), a bunch of stones (A, F and L), a single tree (D, H, K and O), and a group of trees (B, G and J).

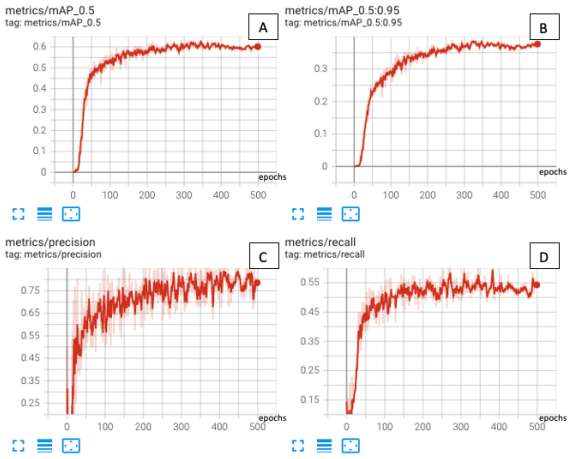

Fig. 13. The accuracy metrics of model 3: mAP_0.5 (A), mAP_0.5:0.95 (B), precession (C), and recall (D).

Furthermore, after the datasets were augmented by rotating and blurring, the result for the last model shows that the object could be detected and bounded by the box perfectly (Fig. 16) as the previous ones: a single tree (A, K and M), a single stone (F, J, N, O and P) and combination of trees and stones (B, C, D, E, G, H, I, and L).

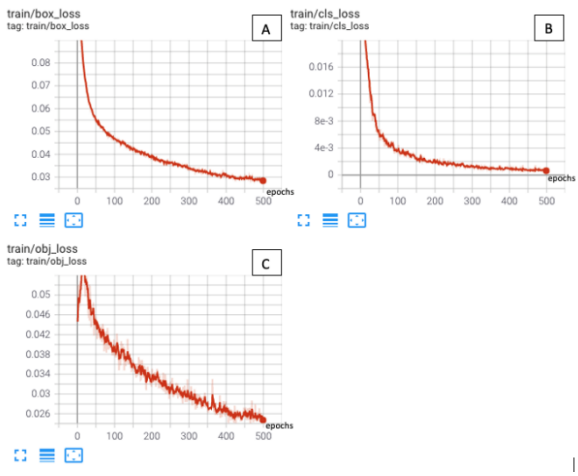

Fig. 14. The loss functions of model 3: bounding box loss (A),

classification loss (B), and object loss (C).

However, the display interface shows a different result for the tree(s), such as a single tree that could not be detected (Fig. 17 B) and a less precise bounding box for a group of trees (Fig. 17 D), while a single stone and a bunch of stones are successfully detected and bounded by a precise box (Fig 17 A and C).

Fig. 15. The result of truth data for model 3: a single stone (C, E, I, M, N and P), a bunch of stones (A, F and L), a single tree (D, H, K and O), and a group of trees (B, G and J).

Fig. 16. The augmented samples for model 2: a single tree (A, K and M), a single stone (F, J, N, O and P) and a combination of trees and stones (B, C, D, E, G, H, I, and L).

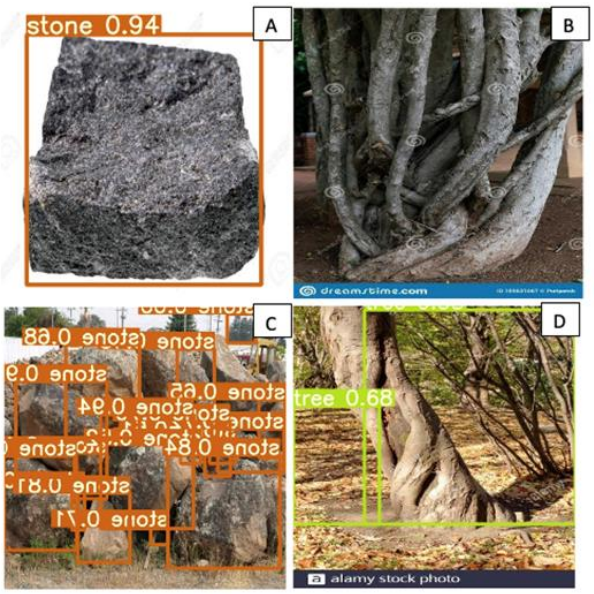

Fig. 17. The display interface of test images for model 3: a group of trees (D), a single stone (A), a bunch of stones (C) and a single tree (B).

CONCLUSION

A deep machine learning suitable for mini-hardware such as a microcontroller called YOLOv5s has been applied for object detection in a 4-wheeled mobile robot for the volcano monitoring system. The training process was carried out in three different epochs (100, 300 and 500) for the same batch (16). The more epoch, the higher the mean Average Precision (51.9% to 59.9% for 0.5 mAP and 28.6% to 37.6% for 0.5:0.95 mAP), precision (58% to 80%) and recall (50% to 55%); and the lower the losses (bounding box's (0.0046 to 0.029), classification's (0.0034 to 0.0005) and object's (0.039 to 0.024). Therefore, the last variant shows the best result for accuracy metrics and loss function. However, the second variant shows the optimum result indicated by no-fail object detection and precise bounding boxes. Accuracy metrics could be upgraded (5-10% improvement in mAP, as suggested by comparative research, a concrete and achievable target for enhancing the performance of YOLOv5 in computer vision applications) by improving performance with data: getting more data, inventing more data, rescaling data, transforming data, featuring selection and using higher image resolution data. Moreover, the part of the tree object could be detailed to get a more specific dataset.

ACKNOWLEDGMENT

We would like to say thank you to the Faculty of Mathematics and Natural Sciences of ITB for the PPMI and BKM research funds.

REFERENCES

[1] M. Evita, *et al.*, (Mobile Monitoring System for Indonesian Volcano), Proc. 4th Int. Conf. on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), (Nov. 2-3, 2015, Bandung, Indonesia), 278, 2015.

[2] M. Djamal, *et al.*, Development of a Low-cost Mobile Volcano Early Warning System, *J. Tech. Sci.*, **1**, 84, 2017.

[3] M. Evita, *et al.*, (Fixed-mode of mobile monitoring system for Indonesian volcano), Proc. 4th Int. Conf. on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), (Nov. 2-3, 2015, Bandung, Indonesia), 278, 2015.

[4] M. Evita, *et al.*, Development of Volcano Early Warning System for Kelud Volcano, *JETS ITB*, **53**, 2021.

[5] M. Evita, *et al.*, (Mobile robot deployment experiment for mobile mode of mobile monitoring system for Indonesian volcano), Proc. of Int. Conf. on Technology and Social Science, (May, 10-11, 2017, Kiryu, Japan), Keynote Lecture, 2017.

[6] M. Evita, *et al.*, (Development of a robust mobile robot for volcano monitoring application), Proc. of the 9th Int. Conf. on Theoretical and Applied Physics (ICTAP), (Sep., 26-28, 2019, Lampung, Indonesia), 1572, 2019.

[7] M. Evita, *et al.*, Photogrammetry using Intelligent-Battery UAV in Different Weather for Volcano Early Warning System Application, *J. Phys.: Conf. Ser.*, **1772**, 012017, 2021

[8] A. Zakiyyatuddin, *et al.*, Geospatial Survey Analysis for 3D Field and Building Mapping using DJI Drone and Intelligent Flight Battery, *J. Phys.: Conf. Ser.*, **1772**, 012015, 2021.

[9] V. F. Amaliya, *et al.*, Development of IoT-Based Volcano Early Warning System, *J. Phys.: Conf. Ser.*, **1772**, 012009, 2021.

[10] M. Evita, S. T. Mustikawati, and M. Djamal, Design of Real-Time Object Detection in Mobile Robot for Volcano Monitoring Application, *J. Phys.: Conf. Ser.*, **2243**, 012038, 2022.

[11] A. Ohya, A. Kosaka and A. Kak, (Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing), Proc. Of the 1997 IEEE/RSJ Int. Conf. on Intelligent Robot and Systems, Innovative Robotics for Real-World Applications IROS '97, (Sep. 7-11, 1997, Grenoble, France), 704, 1997.

[12] T. Xinchi, *et al.*, (A Research on Intelligent Obstacle Avoidance for Unmanned Surface Vehicles), 2018 Chinese Automation Congress (CAC), (Nov. 30 – Dec. 2, 2018, Xi'an, China), 1431, 2018.

[13] A. Gonzalez-Garcia, *et al.*, (A 3D Vision Based Obstacle Avoidance Methodology for Unmanned Surface Vehicles), XXI Congreso Mexicano de Robótica, (Nov. 19, 2019, Colima, Mexico), 2019.

[14] D. Fridovich-Keil, *et al.*, (Probabilistically Safe Robot Planning with Confidence-Based Human Predictions), 2018 IEEE International Conference on Robotic and Automation ICRA, (May 21-26, 2018, Brisbane, Australia), 387, 2018.

[15] Y. Peng, *et al.*, (Obstacle detection and obstacle avoidance algorithm based on. 2-d lidar), 2015 IEEE International Conference on Information and Automation, (Aug. 8-10, 2015, Yunnan, China), 1648, 2015.

[16] J. Yan, *et al.*, YOLOv5-Ytiny: A Miniature Aggregate Detection and Classification Model, *Electronics*, **10**, 1711, 2021.

[17] F. Yang, *et al.*, Deep Learning for smart manufacturing: Methods and applications, *Appl. Sci.*, **10**, 2361, 2020.

[18] R. Devnita, Melanic and Fulvic Andisols in Volcanic Soils derived from some Volcanoes in West Java, *Indonesian Journal of Geology*, **7**, 227, 2012.

[19] Z. Guan, Real time object recognition based on YOLO model, Proc. Of the 2023 International Conference on Mathematical Physics and Computational Simulation, (August, August 2023, Oxford, United Kingdom), 2023.

[20] A. Gunnarsson, Real time object detection on a Raspberry Pi, Bachelor Degree Project, Linnaeus University, 2019.

[21] D. Conrad, and G. N. DeSouza, Homography-Based Ground Plane Detection for Mobile Robot Navigation Using a Modified EM Algorithm, 2010 IEEE International Conference on Robotics and Automation, (May 3-8, 2010, Anchorage, Alaska, USA), 910-915, 2010.

[22] M. A. Layek, *et al.*, Cloud-based Smart Surveillance System using Raspberry Pi and YOLO, 2018 Korea Software Congress, (19 December 2018, Pyeongchang, Jeju, Korea), 2018.

[23] D. Castells, M. F. Rodrigues and M. H. du Buf, (Obstacle Detection and Avoidance on Sidewalk), Proc. of the International Conference on Computer Vision Theory and Applications, (May 17-21, 2010, Angers, France), 235, 2010.

[24] R. Girshick, *et al.*, (Rich feature hierarchies for accurate object detection and semantic segmentation), Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, (Jun. 23-28, 2014, OH, USA), 580, 2014.

[25] S. Ren, *et al.*, Faster R-CNN: Towards real-time object detection with region proposal networks,

Advances in neural information processing system, 91, 2015.
[26] J. Du, Understanding of Object Detection Based on CNN Family and YOLO, *J. Phys.: Conf.Ser.*, **1004**, 1, 2018.
[27] A. Kuznetsova, *et al.*, The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale, *Int. J. of Computer Vision*, **128**, 2020.
[28] I. Namatevs, K. Sudars and I. Polaka, Automatic data labeling by neural networks for the counting of objects in videos, *Procedia Computer Science*, **149**, 151, 2019.
[29] C. Shorten and T. G. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning. J. *of Big Data*, **6**, 60, 2019.
[30] Y. Xu and R. Goodacre, On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning, *J. of Analysis and Testing*, **2**, 2018.
[31] J. Redmon, *et al.*, (You Only Look Once: Unified, Real-Time Object Detection), 2016 IEEE Conf. on Computer Vision and Patern Recognition (CVPR), (Jun. 27-30, 2016, NV, USA), 779, 2016.
[32] J. Yu, *et al.*, (UnitBox: An advanced object detection network), Proceedings of the 24th ACM international conference on Multimedia, (Oct. 15-19, 2016, Amsterdam, Netherlands), pp. 516, 2016.
[33] H. Rezatofighi, *et al.*, (Generalized in- tersection over union: A metric and a loss for bounding box regression), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (June 15-20, 2019, CA, USA), 658, 2019.
[34] Z. Zheng, *et al.*, (Distance-IoU Loss: Faster and better learning for bounding box regression), Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), (Feb. 7-12, 2020, New York, USA), 12993, 2020.
[35] M. Andrychowicz, *et al.*, (Learning to learn by gradient descent by gradient descent), Proc. Of the 30th Int. Conf. on Neural Information Processing System, (Dec. 5-10, 2016, Barcelona, Spain), 3988, 2016.
[36] P. Netrapalli, Stochastic Gradient Descent and Its Variants in Machine Learning, *Journal of the Indian Institute of Science*, **99**, 2019.
[37] N. S. Keskar, *et al.*, (On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima), 5th Int. Conf. on Learning Representation, ICLR, (Apr. 24-26, 2017, Toulon, France), 149804, 2017.
[38] A. Bochkovskiy, C. Y. Wang and H. Y. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv 2004.10934v1, 2020.