Detection System of Cattle Foot and Mouth Disease (FMD) using Deep Learning

Moch. Zen Samsono Hadi, Rizky Bintang Fahreza, Dinda Dwimagfiroh, Aries Pratiarso, and Haniah Mahmudah

Department of Electrical Engineering
Electronic Engineering PolytechnicInstitute of Surabaya (EEPIS)
Surabaya, Jawa Timur, Indonesia
zenhadi@pens.ac.id, rizkybintangf@te.student.pens.ac.id,
dindadm@te.student.pens.ac.id, aries@pens.ac.id,
haniah@pens.ac.id

Abstract. Foot and Mouth Disease (FMD) is an extremely transmissible viral illness that specifically targets cloven-hoofedanimals, such as cattle. It is caused by the Foot and Mouth Disease Virus (FMDV). Traditionally, FMD detection involves manual observation by trained veterinarians, which is time- consuming and subjective. The proposed system leverages the power of deep learning algorithms to automate the detection process, allowing for faster and more accurate FMD identification in cattle. In this research, we contrast various approaches for applying deep learning to diagnose Foot and Mouth Disease (FMD) in cattle. YOLOv4 and YOLOv4-tiny are the two algorithms that we concentrate on. By utilizing the FMDdataset to train each system, we can compare how effectively it performs to detect FMD in cattle. From the study we have done, a better accuracy was obtained in YOLOv4 with an accurate value of 98%. However, the detection speed of YOLOv4-tiny is much faster compared to Yolov4, but with a lower accuracy than YOLOV4.

Keywords. c*attle, illness, Foot and Mouth Disease (FMD), deep learning, YOLO*

1 Introduction

Foot and Mouth Disease (FMD) is an extremely transmissible viral illness that specifically targets cloven- hoofed animals, such as cattle. The virus responsible for thisdisease is called Foot and Mouth Disease Virus (FMDV), which belongs to the Picornaviridae family. It causes an acutedisease with high morbidity and low mortality that is characterized by fever, lameness, and vesicular lesions on thefeet, tongue, snout, and teats[1]. Even within 300 kilometresof each other through the air, the disease can spread betweeninfected and healthy animals. [2].

Traditionally, FMD detection involves manual observation by trained veterinarians, which is time-consumingand subjective. The proposed system leverages the power of deep learning algorithms to automate the detection process, allowing for faster and more accurate FMD identification in cattle. By analysing large datasets of cow images, the systemcould study complex patterns and features associated with animals infected with FMD.

The proposed system employs a convolutional neural network (CNN) based on the

M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_32

YOLO architecture, which is a specialized deep learning model used for analysing images. This model is trained using labelled datasets so that it may learn to recognize recognizable traits and patterns that indicatethe presence of disease. After a number of training rounds, thesystem is able to accept the mouth's cattle photos as input andmake predictions about whether the animal is infected with Foot and Mouth Disease (FMD).

We utilize the You Only Look Once (YOLO) algorithm for our research. The dataset we employ comprises images ofcattle mouths, which serve as the foundation for detecting early signs of Foot and Mouth Disease (FMD) in cows. This dataset contains a combination of images depicting bothFMD-infected and healthy cattle, enabling us to conduct comprehensive investigations in this area.

According to previous research [3], to detect FMD disease in cattle, it does not include the YOLO architecture in it, where it could be better than other CNN architectures and get higheraccuracy values.

In this research, we aim to evaluate and compare the performance of YOLOv4 and YOLOv4-tiny models. Our objective is to use these models for the detection and classification of FMD-infected and healthy cattle. The datasetused in this study consists of two classes: FMD and healthy. Both models will be trained using identical parameters, and the results will be analysed to determine the algorithm with the highest effectiveness and optimization.

2 Related works

M. Rony, et al.[3] conducted research to classify externaldiseases in cattle using the Convolutional Neural Network (CNN) method. The study explored different CNN architectures, including conventional Deep CNN, Inception- V3, and VGG-16, within the domain of deep learning, to detect the prevalent external diseases. The study achieved a model accuracy of 95%. However, it should be noted that theYOLO architecture was not included as a comparative modelin this particular study.

M. Adamu Islam Mashuri, et al.[4] presented system to identify the best method or algorithm for detecting objects using a dataset of natural disasters. The models under comparison consist of YOLO algorithm using YOLOv5, along with CNN models employing MobileNet and VGG-16architectures. The research yields have strong-performing models, but their application is limited to specific use cases within the context of natural disasters.

P. Malhotra and E. Garg.[5] applied method to compare different deep learning algorithms for object detection. The study focuses on comparing the performance of three specific algorithms: YOLO, R-CNN, and Fast R-CNN. Variants of these algorithms will also be considered, including variationsin datasets and object detection methods. However, it shouldbe noted that the FMD datasets was not included as a comparative model in this particular study.

R. A. Asmara, et al.[6] focused research on utilizing the object detection method, specifically YOLO approach, to identify vehicles as objects that contribute to traffic congestion. The system design involved the utilization of various devices such as Raspberry Pi 3+, Intel NCS 2, and website applications. The objective was to implement and evaluate the effectiveness of the YOLO method in detecting vehicles

and analyzing their impact on traffic congestion. The proposed system can be compared to previous studiesand existing systems using the following criteria:

Table 1. Comparison with previous research

Previous research	Dataset FMD in cattle	Method
[3]	Yes	Conventional deep CNN, and VGG-16, and Inception-V3
[4]	No	CNN, MobileNet,, VGG16, and YOLOv5
[5]	No	Fast R-CNN, R-CNN, and YOLO
[6]	No	The implementation of YOLO on the Raspberry Pi 3+ device and the Intel NCS 2 device.
Proposed System	Yes	Comparing YOLOv4 and YOLOv4-tiny to determine the best accuracy. The analysis will be conducted and the chosen model will be implemented on a Raspberry Pi for future research purposes.

Based on the information provided in Table 1, our system comparison focuses on evaluating the performance of YOLOv4 and YOLOv4-tiny models. Through detailed analysis, we determine the model that achieves the highest accuracy. This research offers the advantage of identifying theoptimal algorithm prior to its development or implementationon hardware, providing valuable insights for further advancements.

3 System design

The complete system illustrated in Figure 1 incorporates various components such as a webcam, Raspberry Pi 4, LCDscreen, and buzzer. At intervals of every 3 hours, the webcamcaptures images of the cattle's mouth. These image data are then transmitted to the Raspberry Pi for subsequent data processing. The mouth images undergo detection and classification using a CNN that employs the YOLO algorithm.In the event of a healthy condition being detected, the resultsare displayed on the LCD screen. Conversely, if an infected condition is detected, the buzzer is triggered. This simulation is conducted to compare the performance of the YOLOv4 andYOLOv4-tiny algorithms, ultimately determining the optimalalgorithm for future research.

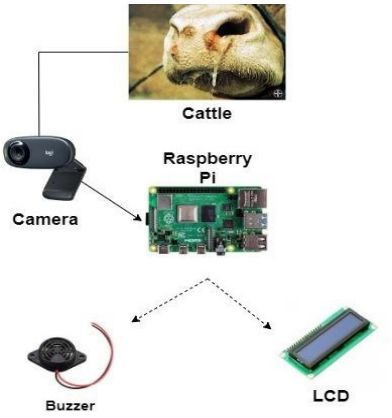

Fig. 1. Design system

This study exclusively focuses on utilizing the YOLOv4 and YOLOv4tiny algorithms for the classification of Foot andMouth Disease (FMD) in cattle. The objective is to determine the most effective algorithm model, necessitating a comparative analysis between the two. To ensure the system's optimal performance, several essential steps will be undertaken. Firstly, relevant datasets pertaining to cattle mouths, specifically a custom dataset for FMD, will be collected and compiled. Following that, data augmentation techniques such as rotation and shearing will be applied. Subsequently, a range of algorithms will be employed to facilitate a thorough comparison. The ultimate goal is to generate accurate and reliable data for subsequent analysis.

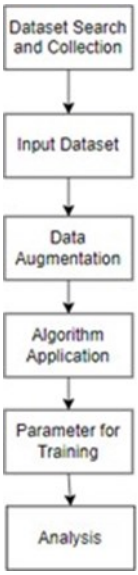

Fig. 2. Methodology system.

The theory must be supported if the primary objectives areto be met and the system is to function as intended.

3.1 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) is a type of supervised machine learning method that relies on labelled data for training. During the training process, CNN utilizes this labelled data to learn and improve its ability to classify objects accurately. By analysing the labelled examples, the CNN adjusts its internal parameters to optimize its classification objective [4].

In comparison to other classification algorithms, CNNs have the advantage of requiring minimal preprocessing of input data. Instead of manually designing filters, CNNs havethe ability to learn filters and distinctive features automaticallyduring the training phase. The design of a CNN is influencedby the intricate organization of neurons in the human brain, particularly the visual cortex. Just as individual neurons in thebrain respond to specific stimuli within a limited area knownas the receptive field, CNNs adopt this concept in theirarchitecture. This resemblance to the human visual system allows CNNs to effectively analyse visual data and extract meaningful features [7]. Similarly, in a CNN, each neuron analyzes a specific portion of the input image. This allows theCNN to capture and process information in a manner that resembles the visual processing mechanism of the human brain. Similarly, in a CNN each neuron analyses a specific partof the input image. By overlapping these receptive fields, the CNN covers the entire visual area, enabling comprehensive analysis of the image.[8].

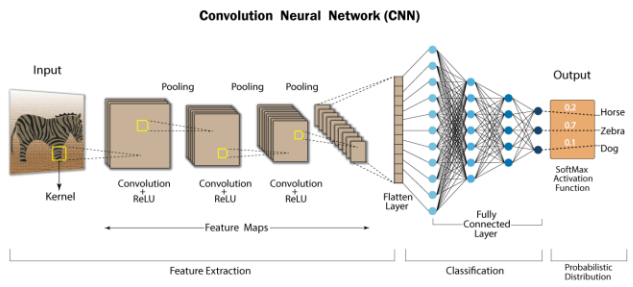

Fig. 3. Convolutional Neural Network Architecture [8]

3.2 YOLO Algorithm

You Only Look Once (YOLO) is an advanced and real- time object detection developed by Joseph Redmon. It is a cutting-edge approach that utilizes a convolutional neural network (CNN) for object recognition. The YOLO framework is built upon Darknet, an open-source CNN framework also developed by Joseph Redmon[9]. The main concept behind YOLO is illustrated in Figure 4, where the input image is divided into a grid of size S×S. The goal of YOLO is to predictthe object centered in each grid cell. By implementing this approach, YOLO achieves efficient and accurate object detection in real-time scenarios [10].

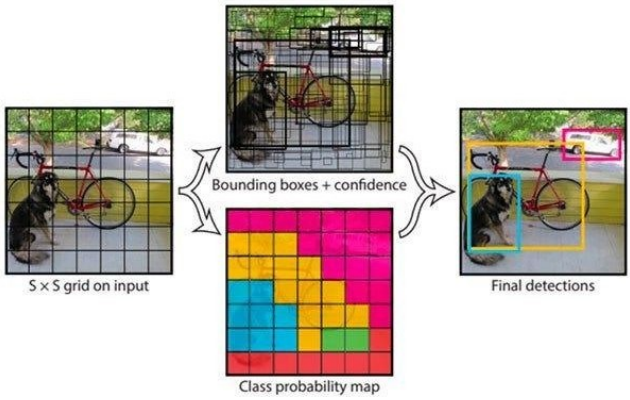

Fig. 4. Object detection sequence of YOLO [11]

While YOLO excels in terms of speed, its detectionaccuracy may be compromised, particularly when it comes toidentifying small objects. This limitation arises from the factthat the input image in YOLO, as illustrated in Figure 4, is divided into a grid of 7x7 cells. Unlike traditional CNN approaches, YOLO treats each grid individually, detecting objects within each grid and generating bounding box information based on the image content.

3.3 YOLOv4

The object detection algorithm known as YOLOv4 is a development on the YOLOv3 model. Alexey Bochkovskiy, et.al.[13] developed the YOLOv4 approach. Compared to EfficientDet, it is twice as quick and has similar performance.Furthermore, YOLOv4 has shown improvements overYOLOv3 in terms of Average Precision (AP) and Frames PerSecond (FPS). Specifically, YOLOv4 has achieved a 10% increase in AP and a 12% increase in FPS compared to YOLOv3 [12]. YOLOv4 utilizes a comprehensive architecture that incorporates various essential components. These components include the CSPDarknet53 backbone, thePANet path-aggregation neck, the YOLOv3 head, and the spatial pyramid pooling extra module. These elements work together to enhance the overall performance and effectivenessof the YOLOv4 model.

3.4 YOLOv4-tiny

YOLOv4-tiny is specifically optimized for mobile and embedded devices. It maintains the overall architecture of YOLOv4 but with simplified network topology and reduced parameters. This enables faster training and detection performance compared to YOLOv4. YOLOv4-tiny utilizes 29 pre-trained convolutional layers, significantly fewer than the 137 layers used in YOLOv4. Moreover, YOLOv4-tiny incorporates two YOLO heads for object detection, while YOLOv4 employs three heads [11]. YOLOv4-tiny achievesa significantly higher frames per second (FPS) compared to YOLOv4, with a performance that is approximately eight times faster. However, when evaluated on the MS COCO dataset, YOLOv4-tiny's accuracy is found to be only around two-thirds of that achieved by YOLOv4.

3.5 Software

We utilized Google Colaboratory to develop machine learning models. Google Colaboratory is a platform that provides us with the ability to utilize and leverage high- performance hardware, such as TPU (Tensor Processing Unit) and GPU (Graphics Processing Unit). Google Colaboratory is a platform that provides us with the ability toutilize and leverage high-performance hardware, such as the TPU (Tensor Processing Unit) and, of course, the GPU (Graphics Processing Unit). By utilizing Google Colaboratory, we can harness the immense computing power of these specialized units for our tasks and computations. Theinfrastructure of the Google Colab we used consists of a CPUXeon Processor@2.3GHz with a single hyper-threaded core, a GPU powered by 1xTesla T4 with 2560 CUDA cores, compute 3.7, and a 15GB (15.079GB usable) GDDR6 VRAM.

3.6 Dataset

The dataset used in this study is obtained from images ofcattle's mouth. We classify the dataset into two classes: FMDand healthy. The FMD class represents cattle that are infectedwith FMD and require intensive monitoring. On the other hand, the healthy class represents cattle that are free from FMD symptoms and in a healthy state. The table below provides information on the dataset that was utilized.

Table 2. Dataset cattle's mouth for FMD disease

Data Labelling	Training data	Testing data
Healthy	405	119
FMD	431	89

3.7 Comparative analysis

The training methodology employed in this experiment has various configuration variables. All the algorithms being compared will adhere to a uniform training approach, with consistent configuration variables. This ensures a fair and equitable comparison among the algorithms, as they will be trained under similar conditions and settings. By employing a standardized training approach with identical configurationvariables, we aim to minimize potential biases and accuratelyevaluate the performance of each algorithm.

Table 3. Hyperparameter for training

Hyperparameter	Value
Batch	2,4,8,16
Subdivision	8
Width	416
Height	416
Max-batches	4000

In our research proposal, we aim to evaluate the impact ofdifferent batch values on the overall accuracy of our model. By comparing various batch sizes, we can identify the optimal batch value that yields the highest accuracy. Furthermore, our study involves comparing the performanceof two different algorithms to assess their effectiveness in achieving accurate results. The two algorithms that will be compared, i.e., YOLOv4 and YOLOv4-tiny.

We compare six different types of matrix evaluations. These evaluations include the True Positive (TP) and False Positive (FP) metrics, which indicate the number of images that can be predicted but are inaccurate. Additionally, we willmeasure the total detection time, which represents the duration of the model's detection process. Accuracy will be used to assess the efficiency of the algorithms, aiming for thehighest possible value. Precision will be calculated by counting the number of accurate positive predictions, strivingfor the highest value. Recall will help determine the proportion of correctly predicted positive outcomes relative to the overall positive data. Lastly, the F1 Score will be usedto combine precision and recall into a comprehensive value.

The dataset is improved using data augmentation methodslike rotation and shearing after being imported into Google Colab for testing. The algorithms and their hyperparameters are then chosen for comparison. By assessing and contrasting the performance of several algorithms, we hope to determinethe most reliable technique for object detection.

- *False Positives and True Positives*. To compare thecorrect predictions with the incorrect predictions ofthe images, numerical outputs can be used for manual calculations using various formulas.
- *Total detection time*, Total time detection is how long a model performs detection
- *Accuracy*. is the proportion of accurate predictions to all data. We will assess this value to determine themethod with the highest accuracy. Using a specific formula, we can quantitatively evaluate the performance of each algorithm.

- *Precision.* It is the proportion of correctly predicted outcomes to all correctly predicted outcomes. And the maximum value for this precision should be used. For the actual formula for precision, which is
- *Recall*. It can be quantified as sensitivity, whichrepresents the ratio of true positive predictions to the
  total number of actual positive data. The algorithm can also be employed to manually calculate recall.
- *F1 Score*. The comparison involves bootstrapped recall and average precision. Therefore, if the provided formula is utilized

4 Experimental Result

The model is designed and simulated using Google Colaboratory. The dataset is classified into two classes: Healthy class and FMD class. To obtain the training and testing data, the dataset is divided, with 20% allocated for testing and 80% for training. The training process involves utilizing various parameters or variables.

We thoroughly compared the YOLOv4 and YOLOv4-tinymodels for precise and ideal item identification and comparison. We achieved findings that included True Positive and False Positive values, as well as measures like accuracy,precision, recall, total detection time, and F1-Score, through simulations that utilized 8 batches. To get a thorough understanding of the algorithm's performance, we also assessed the accuracy of the method across a range of batch sizes.

4.1 False Positive and True Positive

The performance data of YOLOv4 and YOLOv4-tiny were analyzed, and the results revealed the true positive and false positive values using a testing dataset of 20%. It was observed that YOLOv4 achieved more accurate and optimal classification detection, characterized by low false positive values. In contrast, the classification performance of YOLOv4-tiny was less accurate and optimal, as it exhibited a considerably higher number of false positive values.

Table 4. The Performance of YOLOv4

Label Name	True Positive (TP)	False Positive (FP)
Healthy	118	2
FMD	89	2

Table 5. The Performance of YOLOV4-TINY

Label Name	True Positive (TP)	False Positive (FP)
Healthy	85	18
FMD	72	16

4.2 Evaluation of the Models with 8 Batches

Among the compared YOLOv4 and YOLOv4-tiny models using Google Colab, YOLOv4 achieves higher accuracy with precision, recall, and F1 score above 97%. On

the other hand, YOLOv4-tiny is less optimal for object detection but exhibits fast data processing when implemented correctly on hardware.

Table 6. Evaluation Of the Models

Models	Time (Seconds)	Accuracy	Precision	Recall	F1 Score
YoloV4	122	99%	98%	99%	99%
YoloV4-Tiny	3	82%	82%	75%	79%

Based on Table 4, due to the backbone network and the integration of many scales of features, YOLOv4 has a higher accuracy than YOLOv4-tiny. YOLOv4 is better equipped to handle challenging scenarios and objects of various sizes. It performs a fantastic job of picking out both large and little, intricate things. The YOLOv4-tiny, however, is built with a focus on a smaller model size and faster inference times. This can be seen in Table VI. Yolov4-tiny time detection that onlydetects the model in just 3 seconds

4.3 Testing the accuracy of several batches

In the training phase, training samples are processed in groups called batches. The input image is divided into a gridby YOLO, and each grid cell is given the duty of object detection. Each batch contains several training samples, and the model updates its weights using these samples using forward and backward propagation.

In this research, depicted in Figure 5, it is evident that increasing the number of batches results in higher accuracy. This phenomenon can be attributed to the role of batches in mitigating the influence of noisy gradients. By incorporating more representative samples for gradient estimation, the impact of noise is reduced. As a result, weight updates become more stable, leading to improved accuracy across the board.

As we can see, YOLOv4 has higher accuracy than YOLOv4-tiny because of the backbone network and multi- scale feature fusion. YOLOv4 is better able to handle complicated scenes and objects of different scales. It does a good job of recognizing both big and small, highly detailed things. While the YOLOv4-tiny is designed to prioritizefaster inference speed and reduced model size.

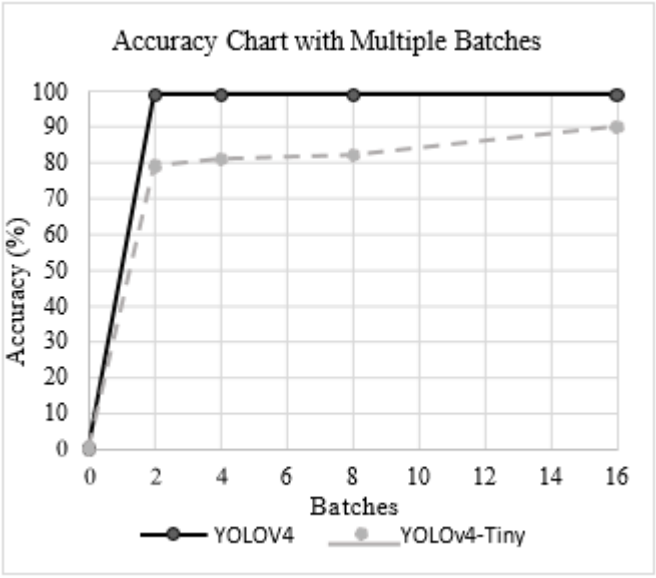

Fig. 5. Accuracy Chart with Multiple Batches

5 Conclusion

In this study, we compare YOLOv4 and YOLOv4-tiny models using Google Colab, YOLOv4 demonstrates higher accuracy with precision, recall, and an F1 score exceeding 97%. This model will be used in the subsequent examination to be implemented on the Raspberry Pi. Due to its deep backbone network and multi-scale feature fusion, which enable it to handle a variety of objects and difficult situations, it performs well. YOLOv4 excels at finding complex details on both big and small objects. The smaller model size and quicker inference times are prioritized by YOLOv4-tiny, incontrast. Even though it might not be as accurate as YOLOv4. With the detection time taking only a short amount of time, the YOLOv4-tiny model demonstrates impressive timeacquisition. Therefore, the decision between YOLOv4 andYOLOv4-tiny is based on the specific needs of theapplication. When high accuracy is crucial and computational resources are available, YOLOv4 is the recommended option. However, if real-time performance or resource limitations are the main considerations, YOLOv4-tiny can still deliver good
results due to its faster inference times

Reference

1. El-Moety and Mohamed Shawky Abd, "Isolation and molecular characterization of foot and mouth disease SAT2 virus during Outbreak 2012 in Egypt", Journal of Veterinary Advances, vol. 3, no. 2, pp. 60- 68, 2013.
2. O. Yazdanbakhsh, Y. Zhou and S. Dick, "An intelligent system for livestock disease surveillance", Inf. Sci. (Ny), vol. 378, pp. 26-47, 2017

3. M. Rony, D. Barai, Riad and Z. Hasan, "Cattle External Disease Classification Using Deep Learning Techniques," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-7.
4. M. Adamu Islam Mashuri, M. Zen Samsono Hadi, R. Widyatra Sudibyo, P. Kristalina and A. Pratiarso, "Smart Victims Detection in Natural Disaster using Deep Learning," 2022 International Electronics Symposium (IES), Surabaya, Indonesia, 2022, pp. 517-521, doi: 10.1109/IES55876.2022.9888369.
5. P. Malhotra and E. Garg, "Object Detection Techniques: A Comparison", International Conference on Smart Structures and Systems (ICSSS), 2020.
6. R. A. Asmara, B. Syahputro, D. Supriyanto and A. N. Handayani, "Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2," 2020 4th International Conference on Vocational Education and Training (ICOVET), Malang, Indonesia, 2020, pp. 391-395, doi: 10.1109/ICOVET50258.2020.9230145.
7. E. Abdellatef *et al.*, "Fusion of deep-learned and hand-crafted features for cancelable recognition systems," *Soft Computing*, vol. 24, no. 20, pp. 15189–15208, 2020. doi:10.1007/s00500-020-04856-1
8. S. Saha, "A comprehensive guide to Convolutional Neural Networks -
9. the eli5 way," Medium, https://towardsdatascience.com/a- comprehensive-guide-to-convolutional-neural-networks-the-eli5-way- 3bd2b1164a53 (accessed May 29, 2023).
10. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
11. P. Sismananda, M. Abdurohman and A. G. Putrada, "Performance Comparison of Yolo-Lite and YoloV3 Using Raspberry Pi and MotionEyeOS," 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020, pp. 1-7.
12. Techzizou, "Yolov4 vs Yolov4-Tiny," Medium, (accessed May 30,
13. 2023).
14. A. M. Awed, A. Maher, M. A. H. Abozied, and Y. Z. Elhalwagy, "Towards realizing a visual UAV flying environment: A novel approach based aerial imagery to construct a dataset for visual servoing," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106098, 2023. doi:10.1016/j.engappai.2023.106098
15. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, 'YOLOv4: Optimal Speed and Accuracy of Object Detection', ArXiv, vol. abs/2004.10934, 2020.