



Forensic Analysis of Drones Attacker Detection Using Deep Learning

Arda Surya Editya^{1*}, Neny Kurniati², Moch. Machlul Alamin³, Anggay Luri Pramana⁴, Angga Lisdyanto⁵

^{1, 2, 3, 4, 5}Department of Informatics Engineering, Faculty of Computer Science, Universitas Nahdlatul Ulama Sidoarjo, Indonesia

Abstract.

Purpose: This research proposes deep learning techniques to assist forensic analysis in drone accident cases. This process is focused on detecting attacking drones. In this research, we also compare several deep learning and make some comparisons of the best methods for detecting drone attackers.

Methods: The methods applied in this research are YOLO, SSD, and Fast R-CNN. Additionally, to validate the effectiveness of the results, extensive experiments were conducted on the dataset. The dataset we use contains videos taken from drones, especially drone collisions. Evaluation metrics such as Precision, Recall, F1-Score, and mAP are used to assess the system's performance in detecting and classifying drone attackers.

Results: This research show performance results in detecting and attributing drone-based threats accurately. In this experiment, it was found that YOLOV5 had superior results compared to YOLOV3 YOLOV4, SSD300, and Fast R-CNN. In this experiment we also detected ten types of objects with an average accuracy value of more than 0.5.

Novelty: The proposed system contributes to improving security measures against drone-related incidents, serving as a valuable tool for law enforcement agencies, critical infrastructure protection and public safety. Furthermore, this underscores the growing importance of deep learning in addressing security challenges arising from the widespread use of drones in both civil and commercial contexts.

Keywords: Drone, forensic, Deep learning, Drones attacker

Received October 2023 / **Revised** November 2023 / **Accepted** February 2024

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



INTRODUCTION

Since the advent of drones in the last few years, it has been possible to make significant advancements in many fields, including agriculture, surveillance, and cinematography, which all have benefited greatly from the use of drones. Despite the numerous benefits these unmanned aerial vehicles (UAVs) are projected to bring, there have also been concerns about their potential misuse for malicious purposes, which has resulted in security concerns [1]. With the advent of drones becoming more accessible to the general public, the need for effective countermeasures to prevent unauthorized drone incursions has become increasingly important, especially given the fact that drones have become more accessible in recent years. As part of this study, we explored the use of deep learning techniques in the analysis of forensic data collected from drones in order to identify and detect potential attackers with the goal of contributing to the safety and security of drones in the future.

The problem of unauthorized drone usage has not been effectively addressed by both law enforcement and security personnel because neither of them can address the problem effectively. It has become increasingly necessary to develop advanced detection and identification mechanisms due to the increase of incidents in which drones have breached restricted airspace or have been used for illegal purposes. In comparison to traditional methods, such as Radio Frequency (RF) signal analysis and visual inspection, more sophisticated methods possess limitations as far as pinpointing the operator behind the drone swiftly and accurately is concerned, especially when compared with traditional methods. Deep learning has enormous potential for enhancing the detection of drone thieves because of its ability to analyze complex data and recognize patterns as well as its ability to process complex data [2].

*Corresponding author.

Email addresses: ardasurya.tif@unusida.ac.id (Editya)

DOI: [10.15294/sji.v11i1.48183](https://doi.org/10.15294/sji.v11i1.48183)

To develop an effective framework for forensic analysis of drones in flight, deep learning algorithms will be used as part of this research in order to analyze drone-related data, including flight patterns, RF signals, and visual data, in order to develop an effective framework for drone-related forensic analysis [3]. Through the exploration of the unique signatures that drones and their operators leave behind in order to identify them, this study seeks to improve the accuracy and efficiency of identifying rogue drones. This research has the potential to contribute to the development of robust security measures aimed at protecting critical infrastructure, public events, and sensitive areas from drone attacks.

The objectives of our study are to explore the methodology, data collection, and experimentation involved in our study, shedding light on how deep learning may be used to address the emerging challenges related to drone accidents, especially in attacked drone cases [4]. As part of this study, we explore the use of deep learning to assist forensic investigators in the process of investigating drone accidents. To validate an object detection model using deep learning methods, real-world data must be used to test its performance. This can be achieved by collecting a representative test dataset that has never been encountered during training or validation. This data should be preprocessed to meet the input requirements of the model. Then, generate predictions based on the test data. It is possible to refine the predictions through post-processing, such as non-maximum suppression. Evaluation metrics include Intersection over Union (IoU), precision, recall, Average Precision (AP), and Mean Average Precision (mAP) to quantify performance. Visual inspection of the results provides qualitative insights. The main contribution of this research is the recommendation of a deep learning method to detect drone attackers. This result may help the forensic investigator determine what caused the drone accident.

METHODS

In Figure 1, we can see how the experiments were carried out according to the method we used. In this experiment, the visual data that has been collected from the drone will be used as the starting point. In this experiment, we will be using a dataset from ColaNet [5], which will be used for the analysis. There are a total of 100 videos in this dataset that show a drone that has been attacked by animals, another drone, and etc. The data we used in this experiment consisted of two types, which included both video data and image data. We also used a pre-trained deep learning model to process the data. This model was trained to detect and classify the type of objects present in the videos. Finally, we used the model to analyze the data collected from the drone.

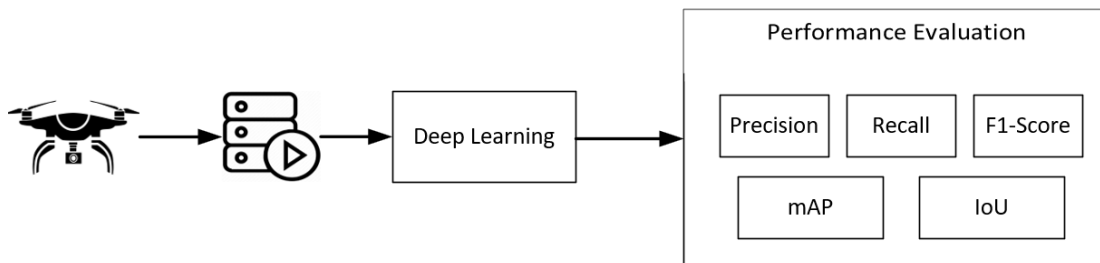


Figure 1. Diagram flow of experiments

Additionally, we processed the dataset using a number of deep learning methods, such as YOLOV3 [6], YOGOV4 [7], YOGOV5 [8], SSD (Single Shot MultiBox Detector) [9] as well as Faster R-CNN [10]. As a result, in order to evaluate the performance of each technique, we examined a number of parameters such as precision, recall, f1-score, mAP, and IoU for the specific object at hand. In this section of the article, we will discuss each of the deep learning methods in more detail in each of the subsections.

YOLO (You Only Look Once)

The YOLO (You Only Look Once) method is a popular object detection algorithm in computer vision that is known for its speed and accuracy. Unlike traditional object detection methods that involve multiple stages and region proposals, YOLO takes a different approach by directly predicting bounding boxes and class probabilities in a single pass through the neural network [6,7,8,11].

In YOLO, the input image is divided into a grid, typically, for example, a 7x7 or 13x13 grid, depending on the YOLO version. Each grid cell is responsible for predicting objects within its boundaries. For each grid

cell, YOLO predicts multiple bounding boxes (usually 2 or 3) and associated class probabilities. These bounding boxes are represented by their center coordinates (x, y), width (w), height (h), and the confidence score (conf) indicating how likely it is that the box contains an object [12]. The class probabilities represent the likelihood of the detected object belonging to a specific class from a predefined set of classes.

The YOLO method can be summarized with an equation for each bounding box prediction within a grid cell:

$$B_i = (x, y, w, h, \text{conf}, P_1, P_2, \dots, P_C) \quad (1)$$

Where:

- B_i represents the i-th bounding box prediction within a grid cell.
- x and y are the normalized center coordinates of the bounding box relative to the grid cell's dimensions.
- w and h represent the width and height of the bounding box, also normalized.
- conf is the confidence score indicating the likelihood of an object being present.
- P_1, P_2, \dots, P_C are class probabilities for each of the C classes.

YOLO uses these predictions to generate a set of bounding boxes and their associated class labels for objects in the image. By applying non-maximum suppression (NMS) to filter out redundant bounding boxes and keeping only the most confident ones, YOLO can provide accurate and efficient object detection results, making it suitable for real-time applications. Furthermore, YOLO can describe using Figure 2 that show the workflow of YOLO [8].

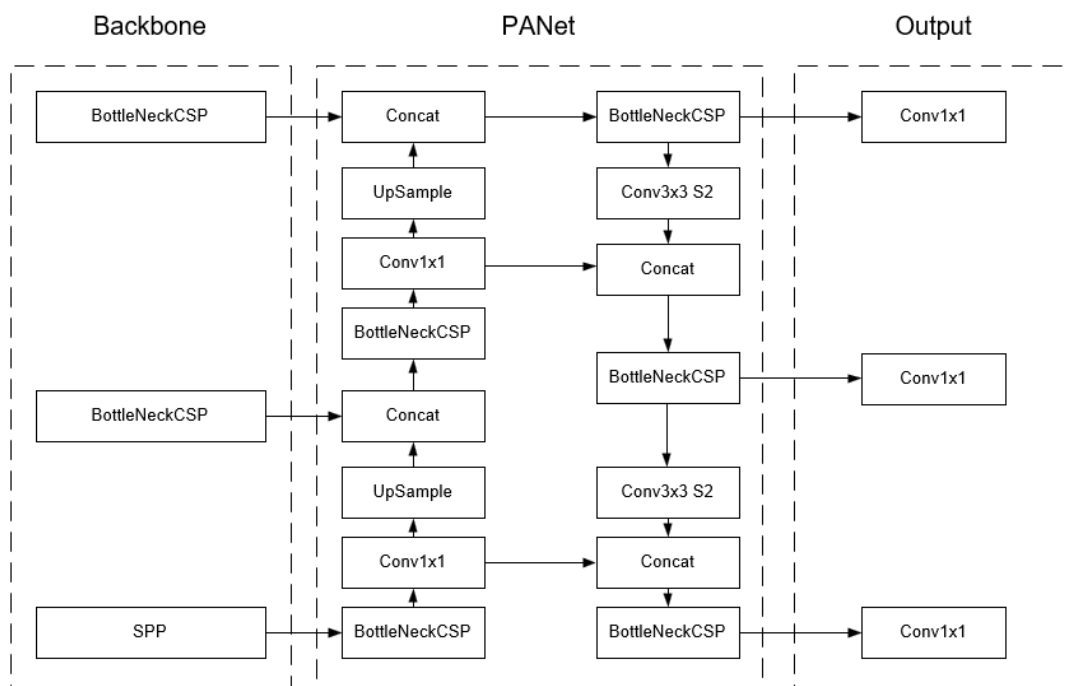


Figure 2. Fast R-CNN workflow diagram

In general YOLO have three main process such as backbone, PANet, and output. In the backbone layer, the YOLO process starts by dividing the image into N grids, having an equal dimensional region of $S \times S$. Each grid is responsible for detecting and localizing objects within, and it can also be called BottleNeckCSP. Therefore, these grids calculate the bounding box as well as the object label. In addition, the algorithm determines the probability of an object being presented in the cell. It can be called Spatial Pyramid Pooling (SPP). Since both detection and recognition of the image are handled by the cells in the image, this process reduces computational time significantly. As a result, it generates a lot of duplicate predictions. This is because multiple cells predict the same object with different bounding boxes and its

process in Path Aggregation Net (PANet) [13]. To resolve this issue, YOLO uses non-maximal suppression. As a result, YOLO suppresses all bounding boxes having lower probability scores. In YOLO, each decision is assessed based on the object's probability, then select the object with the highest probability.

Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD) is a real-time object detection algorithm that combines high detection accuracy with impressive speed [9]. SSD is designed to simultaneously predict multiple bounding boxes and class scores at different scales within a single pass through a neural network. It achieves this by utilizing a set of predefined anchor boxes, which represent various aspect ratios and sizes. SSD achieves its multi-scale detection by incorporating feature maps from different layers of a convolutional neural network (CNN). Each layer of the network is responsible for predicting bounding boxes at a specific scale, with anchor boxes tailored to that scale [14]. This enables SSD to handle objects of varying sizes and aspect ratios effectively. The predictions across all scales are then used to generate a set of candidate bounding boxes, which may overlap. To refine the final set of detections, non-maximum suppression (NMS) is applied. NMS eliminates redundant bounding boxes by keeping the ones with the highest confidence scores and suppressing others that overlap significantly.

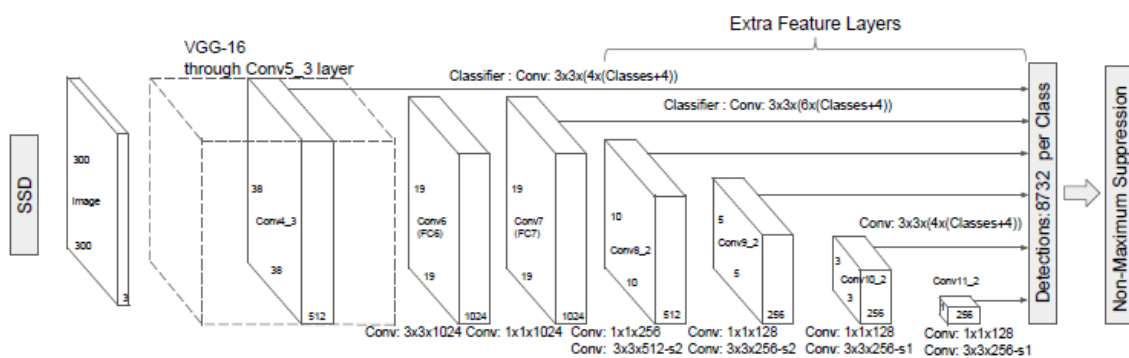


Figure 3. Workflow diagram of SSD

In Figure 3 show us the SSD is build from CNN (Convolutional Neural network) architecture this process have several a layer. The VGG-16 layer contain 16 layers of convolutional and fully connected layers. It is used as the base network to extract features from the input image. The next layer is extra feature layers, in these layer have additional convolutional layers that are added on top of the VGG-16 model to extract more fine-grained features from different scales of the image. Furthermore, in layer object detection heads contain two parallel branches of convolutional layers that are attached to each feature layer [15]. One branch predicts the class labels of the objects in the image, and the other branch predicts the bounding boxes of the objects. Finally in non-maximum suppression is post-processing step that removes overlapping and redundant bounding boxes and selects the most confident ones for each object class.

Faster R-CNN

Faster R-CNN (Region-based Convolutional Neural Network) is a state-of-the-art object detection framework that combines deep learning and region proposal networks for highly accurate and efficient object detection [10]. It introduced the idea of using a region proposal network (RPN) to generate potential object locations in an image and then jointly predicts object bounding boxes and class probabilities. Faster R-CNN's innovation lies in its ability to efficiently propose object regions using the RPN and then refine these proposals with a deep convolutional network. This framework has significantly improved the accuracy and speed of object detection, making it a key milestone in the development of modern object detection algorithms [16].

Faster R-CNN's combination of region proposal and object classification networks has significantly improved object detection performance, making it a cornerstone in the development of advanced object detection methods. This framework is highly versatile and can be adapted for various object detection tasks, including detecting objects of different sizes, shapes, and categories within complex scenes [17]. The RPN in Faster R-CNN learns to propose regions effectively, and the subsequent stages fine-tune these proposals to accurately predict object bounding boxes and class labels. This process not only enhances accuracy but also offers flexibility in handling a wide range of object detection challenges.

One notable advantage of Faster R-CNN is that it can leverage pre-trained CNN models as a backbone network, such as VGG, ResNet, or Inception, to extract meaningful features from images. This transfer learning capability further boosts its performance, particularly when dealing with limited labeled data.

Overall, Faster R-CNN has paved the way for advanced object detection techniques and architectures, contributing to the development of real-world applications in fields like autonomous driving, image analysis, and video surveillance, where precise and efficient object detection is essential [18].

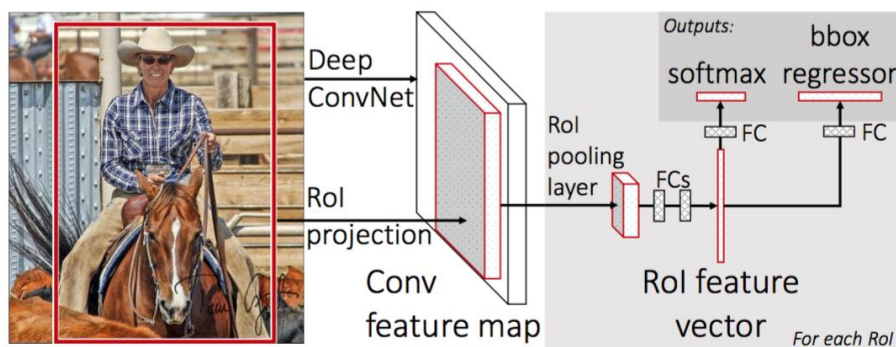


Figure 4. Diagram workflow of Faster R-CNN

In Figure 4 show the Faster R-CNN consists of two modules namely Region Proposal Network (RPN) and Fast R-CNN detector [19]. The first process is RPN takes an image as input and generates a set of region proposals, which are candidate bounding boxes that may contain objects. The Fast R-CNN detector takes the region proposals and the feature map from the RPN as input and performs classification and regression on each proposal to produce the final object detection results [20].

RESULTS AND DISCUSSIONS

This section will present the test results from the existing methods for drone forensic investigation, as well as the results from applying deep learning to the drone forensic investigation process, especially in the situation of an attacked drone. In this experiment, five deep learning methods have been implemented such as YOLOV3, YOLOV4, YOLOV5, SSD300, and fast R-CNN. Several methods were tested to find the most effective one to detect objects and to see how good each method would perform. In this experiment, mean Average Precision (mAP) was used as the evaluation metric in order to measure the performance. The results of this test are depicted in Figure 5.



Figure 5. The result of experiments

The image in Figure 5 is the result of the deep learning algorithm that was used to process the image. In order for the object to be marked, a box detection is used. Also, there is a value that has been placed in the box that represents how accurate the object is that is inside the box. This value can range from 0 to 1, with 1 being the most accurate. The accuracy is determined by the algorithm's ability to recognize and identify the objects in the image. The higher the accuracy, the more reliable the algorithm is. A comparison of the performance of each method is also shown in Table 1.

Table 1. Result of experiment

| Measure | YOLOV3 | YOLOV4 | YOLOV5 | SSD300 | Fast R-CNN |
|-----------|--------|--------|-------------|--------|------------|
| Precision | 0.75 | 0.70 | 0.82 | 0.73 | 0.76 |
| Recall | 0.43 | 0.60 | 0.72 | 0.55 | 0.57 |
| F1-Score | 0.57 | 0.66 | 0.70 | 0.64 | 0.62 |
| mAP | 0.48 | 0.63 | 0.71 | 0.52 | 0.63 |

In Table 1 we can see that YOLOV5 has outperformed values in each parameter such as precision, recall, F1-score, and mAP. This could be because YOLOv5 may have an architecture that is better suited to object detection tasks, allowing it to capture more complex patterns and relationships in the data. Also YOLOv5 might achieve better results while being more computationally efficient. It's essential to note that the specific reasons for YOLOv5's superior performance would depend on the details of the experiment, including the dataset, model configuration, training process, and evaluation criteria. Nonetheless, when YOLOv5 consistently outperforms other models in these metrics, it indicates its strength in object detection tasks. Furthermore YOLOv5 may have a more extensive and deeper network architecture, allowing it to learn more complex features and patterns within the data. This increased capacity can lead to better object detection performance, especially for small or closely packed objects. As is a powerful backbone network, such as CSPDarknet53, which enhances feature extraction capabilities. This can be crucial for detecting objects in cluttered scenes and in YOLOV5 we have optimized object detection head which is optimized to generate more accurate bounding box predictions and confidence scores, contributing to better precision and recall.

In this experiment we also test each model to detect the object on drone accident dataset. In Table 2 shown the result of accuracy of object detection in each deep learning method.

Table 2. Accuracy value on object

| Object | YOLOV3 | YOLOV4 | YOLOV5 | SSD300 | Fast R-CNN |
|--------|-------------|-------------|-------------|-------------|-------------|
| Person | 0.42 | 0.49 | 0.86 | 0.67 | 0.72 |
| Bird | 0.56 | 0.52 | 0.72 | 0.58 | 0.65 |
| Dog | 0.78 | 0.75 | 0.73 | 0.68 | 0.70 |
| Car | 0.72 | 0.62 | 0.82 | 0.74 | 0.80 |
| Ball | 0.41 | 0.62 | 0.42 | 0.55 | 0.60 |
| Tree | 0.14 | 0.18 | 0.12 | 0.21 | 0.11 |
| Drone | 0.12 | 0.17 | 0.23 | 0.18 | 0.32 |
| Baloon | 0.48 | 0.53 | 0.86 | 0.25 | 0.32 |
| Lamp | 0.23 | 0.32 | 0.27 | 0.21 | 0.42 |

As shown in Table 2 in this scenario we use ten objects namely person, bird, dog, car, ball, tree, drone, baloon and lamp. The selected object chosen because the object have show many in frame on dataset colaNet. As the result, the accuracy detection of each object using YOLOV3, YOLOV4, YOLOV5, SSD300, and Fast R-CNN have variated value in each object. It can be because model that have installed on each method. As we know the object detection depend on model that build before. The dataset for training the model may have various a number on each object. In this result we know the YOLOV5 can detect many object well, it can be seen on value of accuracy in ten object that test.

CONCLUSION

In conclusion, the experimental results demonstrate that YOLOv5 outperforms YOLOv3, YOLOv4, SSD300, and Fast R-CNN in multiple critical metrics for object detection tasks. These superior performance metrics include F1 Score, Recall, Precision, and mAP (mean Average Precision). The advantages of YOLOv5 likely stem from a combination of architectural improvements, better training techniques, efficient model design, and potentially optimized hyperparameters. YOLOv5's success in achieving higher F1 Score, Recall, Precision, and mAP signifies its ability to accurately detect objects while minimizing false positives, identify a higher percentage of actual objects, strike a balance between Precision and Recall, and perform well in terms of ranking and categorizing objects across various classes. These results make YOLOv5 a compelling choice for object detection tasks, as it exhibits enhanced performance and efficiency compared to its predecessors (YOLOv3, YOLOv4), SSD300, and Fast R-CNN. However, the specific advantages may vary based on the experimental setup, data, and application requirements, but overall, YOLOv5's superiority in these metrics underscores its effectiveness in the field of object detection.

In the future, the implemented of deep learning model in real drone especially the best model of this study namely YOLOV5. In addition, deep learning techniques can used in many data that took from drone and make the drone have mini blackbox inside the drone.

REFERENCES

- [1] AL-Dosari K, Hunaiti Z, Balachandran W. Systematic Review on Civilian Drones in Safety and Security Applications. *Drones*. 2023; 7(3):210.
- [2] A. S. Editya, T. Ahmad and H. Studiawan, "Direction Estimation of Drone Collision Using Optical Flow for Forensic Investigation," 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 2022, pp. 1-6, doi: 10.1109/ISDFS55398.2022.9800790.
- [3] A. S. Editya, T. Ahmad and H. Studiawan, "Forensic Investigation of Drone Malfunctions with Transformer," 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSSES), Tumakuru, India, 2023, pp. 1-5, doi: 10.1109/ICSSSES58299.2023.10199237.
- [4] U. Nepal, H. Eslamiat, Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs, *Sensors* 22 (2) (2022).
- [5] Pedro, D., Mora, A., Carvalho, J., Azevedo, F., & Fonseca, J. (2020). ColANet: A UAV Collision Avoidance Dataset. In *IFIP Advances in Information and Communication Technology* (pp. 53–62). Springer International Publishing. https://doi.org/10.1007/978-3-030-45124-0_5
- [6] C. Kumar B., R. Punitha, Mohana, YOLOv3 and YOLOv4: Multiple object detection for surveil lance applications, in: 2020 Third International Conference on Smart Systems and Inventive Tech nology (ICSSIT), 2020, pp. 1316–1321.
- [7] A. Mishra and S. Panda, "Drone Detection using YOLOV4 on Images and Videos," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9825244.
- [8] Y. Yang, "Drone-View Object Detection Based on the Improved YOLOv5," 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2022, pp. 612-617, doi: 10.1109/EEBDA53927.2022.9744741.
- [9] Arshad, U. (2021). Object Detection in Last Decade - A Survey. In *Scientific Journal of Informatics* (Vol. 8, Issue 1, pp. 60–70). Universitas Negeri Semarang. <https://doi.org/10.15294/sji.v8i1.28956>
- [10] B. Liu, W. Zhao and Q. Sun, "Study of object detection based on Faster R-CNN," 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 6233-6236, doi: 10.1109/CAC.2017.8243900.
- [11] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection (Version 5). arXiv. <https://doi.org/10.48550/ARXIV.1506.02640>
- [12] M. Mahendru and S. K. Dubey, "Real Time Object Detection with Audio Feedback using Yolo vs. Yolo_v3," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 734-740, doi: 10.1109/Confluence51648.2021.9377064.
- [13] P. Samothai, P. Sanguansat, A. Kheaksong, K. Srisomboon and W. Lee, "The Evaluation of Bone Fracture Detection of YOLO Series," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, 2022, pp. 1054-1057, doi: 10.1109/ITC-CSCC55581.2022.9895016.
- [14] J. Yang and L. Wang, "Feature Fusion and Enhancement for Single Shot Multibox Detector," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 2766-2770, doi: 10.1109/CAC48633.2019.8996582.
- [15] Chengcheng Ning, Huajun Zhou, Yan Song and Jinhui Tang, "Inception Single Shot MultiBox Detector for object detection," 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 2017, pp. 549-554, doi: 10.1109/ICMEW.2017.8026312.
- [16] O. Hmidani and E. M. Ismaili Alaoui, "A comprehensive survey of the R-CNN family for object detection," 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, 2022, pp. 1-6, doi: 10.1109/CommNet56067.2022.9993862.
- [17] K. Zhang, "Faster R-CNN Transmission Line Multi-target Detection Based on BAM," 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Hangzhou, China, 2022, pp. 364-369, doi: 10.1109/ICMSP55950.2022.9858980.
- [18] Y. Liu, "An Improved Faster R-CNN for Object Detection," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2018, pp. 119-123, doi: 10.1109/ISCID.2018.10128.

- [19] P. Liu, K. Li, X. Zhou and X. Qi, "RFP-based Faster R-CNN in Aerial Image Detection," 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 2020, pp. 325-329, doi: 10.1109/ICAICE51518.2020.00069.
- [20] G. Yu, P. Yu, H. Li and H. Li, "An Improved Faster R-CNN Method for Car Front Detection," 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Beijing, China, 2022, pp. 7-12, doi: 10.1109/IAEAC54830.2022.9929605.