Yolo Model for Durian Theft Detection in Night Vision

I Nyoman Eddy Indrayana[1],
Gde Brahupadhya Subiksa[2],
Putu Manik Prihatini[3], I Wayan Suasnawa[4],
and Putu Indah Ciptayani[5]

[1,2,3,4,5] Information Technology Department, Politeknik Negeri Bali, Bali, Indonesia
eddyindrayana@pnb.ac.id

Abstract. Theft of durians is a prevalent issue in Indonesia, particularly in regions where durians are cultivated. This results in substantial economic losses for durian farmers. Nocturnal robbers who often operate during the night can be detected using night vision technology. The objective of this study is to create a model for detecting durian thieves using Yolo (You Only Look Once) version 8 under night vision settings. The Yolo version 8 model is trained using image and video datasets captured by a night vision camera, specifically targeting durian thieves. The dataset is partitioned into two distinct subsets: the training dataset and the testing dataset. The Yolo model is trained using the training dataset and subsequently assessed using the testing dataset. The results demonstrate that the Yolo model may get a remarkable degree of precision in detecting individuals who steal durians in low-light circumstances using night vision technology. This model can identify individuals who steal durians, regardless of their different body positions and backgrounds. The Yolo model demonstrated its efficacy in detecting individuals who steal durians in low-light conditions using night vision technology. This approach may enhance the security of durian crops and offer timely detection of durian theft.

Keywords: Deep Learning, Durian Theft Detection, Yolo Night Vision

1 Introduction

The issue of nocturnal durian fruit theft has emerged as a significant worry for durian producers. The durian fruit is a very valuable product with significant economic (Barakat et al., 2023) importance, serving as a staple crop for several farmers in different locations. Regrettably, instances of durian theft frequently take place during nighttime, when the absence of light facilitates the covert operations of thieves. This leads to substantial economic losses for farmers and interrupts the smooth operation of agricultural enterprises. The remote position of durian gardens, distant from residential areas, poses a challenge in monitoring durian theft. Durian gardens are often situated in secluded or interior regions, apart from human towns or hubs of activity. The geographical characteristics in this area provide challenges for accessing durian gardens, particularly at night when community activities are often limited. During the

A. A. N. G. Sapteka et al. (eds.), *Proceedings of the International Conference on Sustainable Green Tourism Applied Science - Engineering Applied Science 2024 (ICoSTAS-EAS 2024)*, Advances in Engineering Research 249,
https://doi.org/10.2991/978-94-6463-587-4_39

nighttime, the absence of light surrounding the durian crop creates an optimal opportunity for burglars to target it. The scarcity of light and the considerable distance from human populations provide challenges for farmers in closely overseeing their durian farms. Indeed, durian plantations located at a distance from residential areas are also susceptible to inadequate security and supervision, since growers may face challenges in routinely and promptly monitoring their farms, particularly during nighttime hours.

Currently, the security measures used in durian plantations generally depend on old techniques that are not very efficient, such as minimal manual monitoring and human guard systems that are susceptible to mistakes and carelessness. As a result of this constraint, farmers cannot frequently promptly and effectively predict or address incidents of theft. Hence, there is a want for more advanced and efficient measures to assist farmers in safeguarding their durian plantations against theft, particularly during nighttime. An effective approach involves employing artificial intelligence technology in the form of object detection models (Ahmad et al., 2020; Wu et al., 2020), which can swiftly and automatically identify the presence of burglars.

2 Methodology

The research involved the following procedures to develop a model for detecting durian theft at night using the YOLO (Redmon et al., 2016; Redmon & Farhadi, 2017) approach: Begin by detecting any odd behavior (Mishra et al., 2022) occurring around the durian tree during nighttime. Indicators of suspicious activity may encompass typical movements (Sary et al., 2023) or the existence of unfamiliar individuals (Ivašić-Kos et al., 2019) in the vicinity of the durian crop during nighttime hours. Activities may involve approaching, climbing, or carrying specialized equipment near a durian tree. Secondly, Data Collection and Dataset Creation (Li & Yang, 2018): After identifying suspicious behavior, the subsequent action involves gathering data through the capture or acquisition of photos or videos from the durian plantation containing suspicious human activities (Gao et al., 2020) during nighttime. The provided data will be utilized to generate a dataset, which will then be employed for training a model. Ensure that the collected data encompasses a diverse range of suspicious actions and varying lighting conditions. Following data collection, proceed to annotate the data by accurately designating the position of any suspicious items or actions inside the picture or video. The inclusion of these annotations is crucial for effectively training the model, enabling it to precisely identify and classify suspicious human activities (Zadobrischi & Negru, 2020). Subsequently, partition the dataset into two distinct subsets: the training dataset and the validation dataset. The training dataset will be utilized to train the model, whilst the validation dataset will be employed to assess the model's performance (Ahmad et al., 2020). Fourth, data preparation involves performing various operations on data to prepare it for further analysis or modeling (Wu et al., 2020). These operations may include tasks such as resizing images, normalizing pixel values, and using data augmentation techniques (Ryu & Chung, 2021). Preprocessing enhances the quality of data and boosts the performance of the model. The subsequent step involves conducting model training by utilizing the training dataset to train the YOLO model.

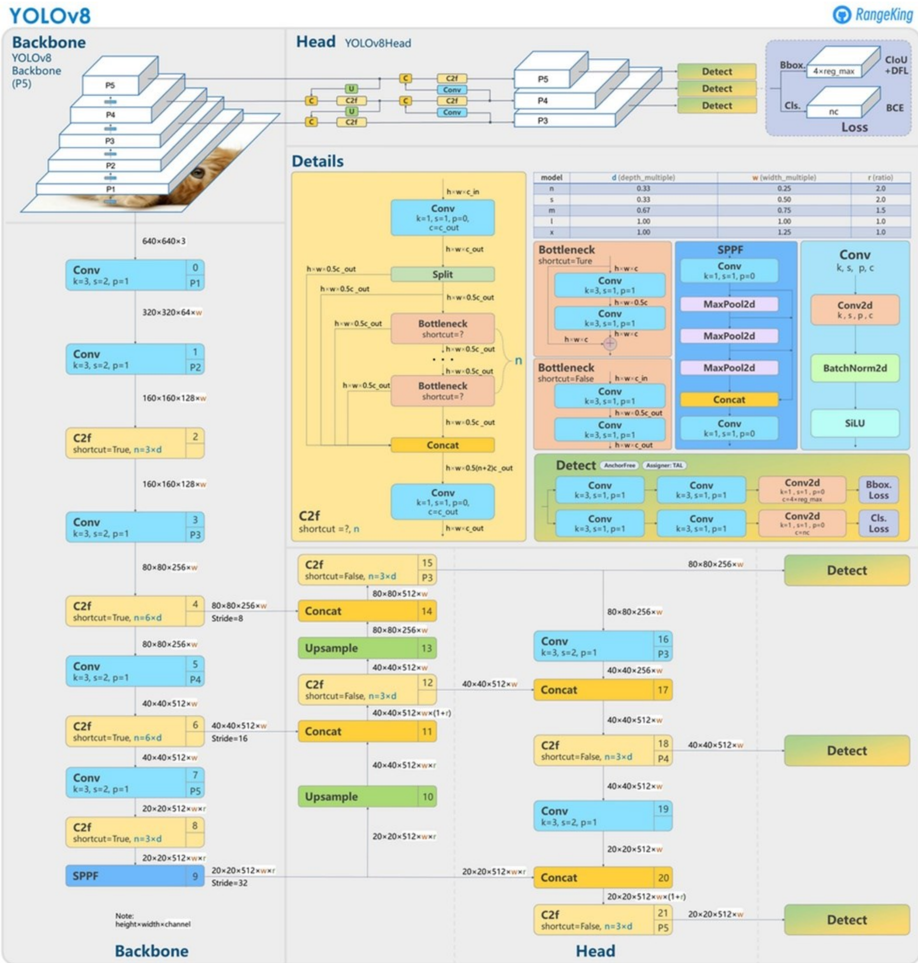


Figure 1. Yolov8 Architecture (Bai et al., 2023)

This training method entails a machine-learning algorithm where the model will acquire the ability to discern dubious activities by analyzing the given data. The utilized tools include Google Colabs (Carneiro et al., 2018), Tensorflow, Ultralytics, OpenCV (Muthanna et al., 2020), and the Yolov8 algorithm. After completing the training process, utilize the validation dataset to assess the performance of the model through model validation, fine-tuning, and evaluation. Assess the model's performance using a confusion matrix, accuracy, mean average precision (mAP), and recall to verify its ability to accurately identify suspicious activities. Conduct fine-tuning of the model to enhance its performance. Reassess and refine prior stages as needed to attain the most effective model performance.

2.1 Yolov8 Algorithm

The Yolov8 architecture consists of many block parts. The blocks include the convolutional block, C2f block (Zhang, 2023), SPPF block (Fang & Pang, 2024), and detect block. The Convolusional component, consisting of the SiLU activation function, BatchNorm2d layer, and Conv2d layer (Dai et al., 2022), is the fundamental building component in the design. Figure 1 displays the Yolov8 Architecture, which contains Convolutional Blocks Convolutional Block. The Fast type C2 module is denoted by the C2F block (Gong, 2023). The Cross Stage Partial Bottleneck (CSP) (Jooshin et al., 2024) with two convolutions is denoted as the C2 module in YOLOv8. The YOLOv8 architecture utilizes it as a fundamental component. The C2f block includes a convolutional block, which is then followed by a split of the feature map. The Bottleneck block is accessed through a single feature map, whereas the Concat block is accessed directly through the other feature map. The number of Bottleneck blocks used in the C2f block is defined by the depth_multiple model parameter. Subsequently, a concluding convolutional block obtains the combined feature map from the bottleneck block and the separated feature map.

2.2 Matrix Evaluation

In our evaluation, we employed the Confusion Matrix (Krstinić et al., 2020) to assess the model's classification performance by comparing the predicted and actual values. The confusion matrix provides information on the count of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). By utilizing the confusion matrix, we may calculate several evaluation metrics, including accuracy, precision, recall, and F1 score. A True Positive (TP) refers to the instances where the model correctly identifies an example as positive (indicating successful identification of an item in the context of object detection) when the example is indeed positive according to the present state of the data. False Positive (FP) refers to the number of instances in which the model incorrectly identifies an example as positive (indicating the presence of something) when it is negative, according to the true state of the data. The True Negative (TN) is the number of instances where the model correctly identifies an example as negative (indicating that the item is not present) when, according to the current state of the data, the example is negative. A False Negative (FN) occurs when the model incorrectly classifies an example as negative (object not detected) even when the example is really positive, according to the true state of the data. The concepts of TP (true positive), FP (false positive), TN (true negative), and FN (false negative) can be employed to calculate various evaluation metrics such as precision and accuracy, Precision is the quotient obtained by dividing the number of true positives (TP) by the total number of positive predictions made by the model, and it is represented as

$$precision = \frac{TP}{TP + FP} \tag{1}$$

Precision is a metric used to evaluate the effectiveness of a classification model. It is one of the measurements used to examine the model's performance. Accuracy is a measure of how frequently the model properly categorizes all observed examples. The confusion matrix employs a mathematical technique to compute the accuracy (Legaspi et al., 2021) as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Equation (2) represents the relationship between the accuracy of a system and the sum of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

3 Result

3.1 Hardware Preparation

The initial stage of setting up the environment for detecting durian thieves is ensuring that the required hardware is prepared. We utilized a CCTV camera or surveillance camera that had exceptional nocturnal recording capabilities. Select a camera equipped with night vision or infrared capabilities to guarantee optimal visibility in low-light circumstances. Proper camera placement is crucial. Ensure that the camera is positioned strategically to provide clear monitoring of the whole durian planting area. The camera utilized for capturing the theft is the EZVIZ C8PF - Dual Lens Pan & Tilt Wi-Fi Camera. It has two lenses: one with a focal length of 2.8 mm and an aperture of F1.6, and the other with a focal length of 12 mm and an aperture of F1.6. The camera has a GK Shift Angle of 340 degrees and a Tilt of 80 degrees. It uses video compression formats G.265/H.264 and supports Wi-Fi standards IEEE802.11b, 802.11g, and 802.11n.

3.2 Configuring Software Settings using Google Colab

Google Colab is a cloud-based platform that is highly beneficial for the development and training of machine learning models, including YOLO version 8 (YOLOv8). By subscribing to the commercial version of Colab, users have access to a cloud server that is equipped with a GPU (Graphics Processing Unit). This GPU is highly advantageous as it accelerates the process of training models, eliminating the need to invest in costly GPU hardware. To begin, the initial action is generating a fresh notebook inside Google Colab and configuring the Python environment by installing essential libraries. The libraries encompass PyTorch, OpenCV, NumPy, Matplotlib, Pillow, and Roboflow.

3.3 Utilizing Roboflow for Dataset Preprocessing

In this study, we employ Roboflow (Sharma et al., 2023) as a tool for effectively handling and organizing datasets. Roboflow is a very beneficial software solution for effectively organizing and preparing datasets in computer vision tasks. Our study picture collection may be uploaded to Roboflow for labeling. Roboflow offers a range of preprocessing and augmentation techniques to enhance the quality of research datasets. These techniques include scaling, rotation, and flipping. After preparing the dataset, we can obtain it in a format that is compatible with YOLOv8 and import it straight into Google Colab via the Roboflow API. The collection has 1497 photos illustrating different instances of durian theft occurring at midnight. The annotation of each photograph indicates the precise position of pertinent things, such as durian thieves. To optimize the learning process and accurately assess the performance of our deep learning model, the dataset is partitioned into three primary segments: training, validation, and testing. For the training set, we utilize 70% of the data that is utilized to train the model. The training set has 1048 photos. During training, 20% of the validation set is allocated for setting hyperparameters and preventing overfitting. The validation set has 299 images. Subsequently, a fraction of 10% of the testing set is allocated to evaluate the model's performance once the training process is over. The testing dataset comprises 150 photos. Figure. 4 displays an illustration of a dataset documenting instances of durian theft that occurred during nighttime.

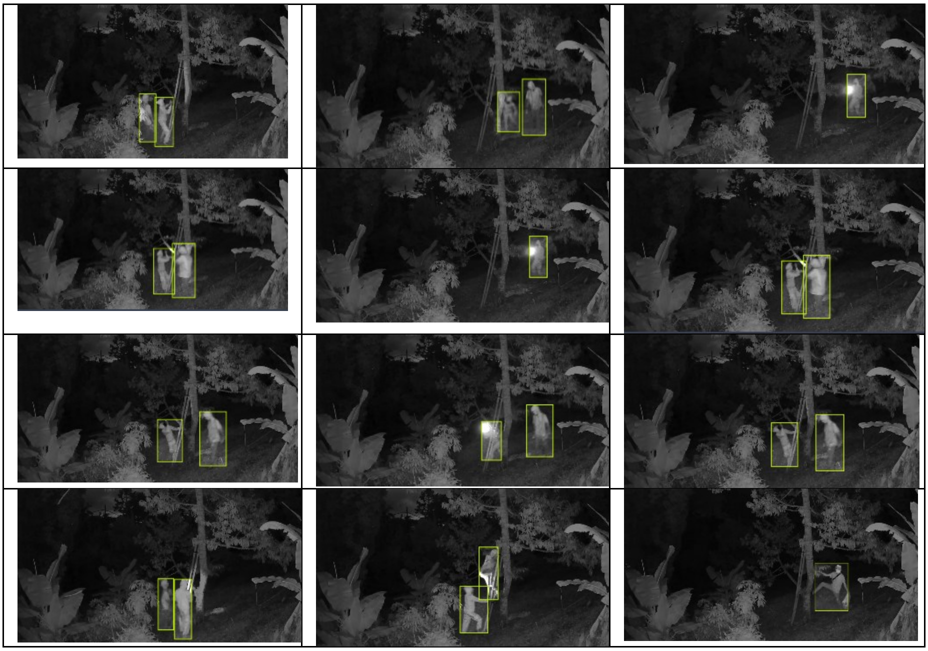
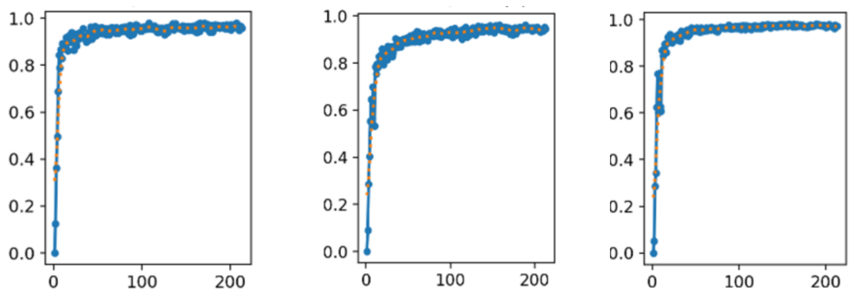


Figure 2. Example of a dataset of durian theft at night

Data setup and preparation in Colab involves importing the dataset and configuring the YOLOv8 model. Subsequently, the dataset undergoes preprocessing in Colab. This entails extracting photographs from the dataset folder, adjusting their size to match the input requirements of YOLOv8, and applying normalization. This phase is crucial to guarantee that the data is prepared for utilization in model training.

3.4 Model Evaluation and Testing

We utilize Google Colab to examine and test the durian thief model at nighttime. We use Roboflow's API to access the dataset that we created with it. We utilize 250 epochs for this dataset. Figures 3a, 3b, and 3c show the precision, recall, and mAP findings in the form of graphs.



(a) (b) (c)

Figure 3. Graph of the computing results of the durian theft model at night Precision, (b) Recall, (c) mAP

(a)

(b)

(c)

(d)

Figure 4. Durian theft model testing results at night

The results of testing the durian thief model at night can be seen in Figure 4. In Figure 4, there are several examples of success in detecting the durian thief model, whether detecting 1 person or more.

4 Conclusion

The research findings indicate that the development of a durian thief detection model using YOLO version 8 entails multiple crucial steps, such as data collection and model performance evaluation. Initially, a dataset including 1497 pictures illustrating different instances of durian theft during the evening was gathered from the field. Each image in this collection is tagged to indicate the precise position of important things, such as criminals and durian trees. Subsequently, the gathered dataset is sent to Roboflow for subsequent analysis and manipulation. The dataset in Roboflow is partitioned into three primary segments: the training set (70%), the validation set (20%), and the testing set (10%). In addition, data augmentation is implemented to enhance the quantity and variety of training data, facilitating the model's ability to learn from variations in pictures. After the dataset has been processed and shared, the data is obtained by downloading it using the API offered by Roboflow and then incorporated into Google Colab. The conclusive outcomes of this procedure demonstrate that the constructed durian thief detection model has exceptional performance. The model attained a precision score of 0.976, a recall score of 0.937, and a mean Average Precision (mAP50) score of 0.982. The data presented demonstrates that the YOLOv8 model can accurately detect instances of durian theft, hence offering timely alerts on theft incidents in durian orchards during nighttime.

Acknowledgment

This research was financially supported by DIPA Politeknik Negeri Bali under the grant number SP DIPA-023.18.2.677608/2024, with the third revision on 04 March 2024

References

Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., Haq, A. U., & Ali, R. (2020). Object detection through modified YOLO Neural Network. *Scientific Programming*, *2020*, 1–10. https://doi.org/10.1155/2020/8403262.

Muthanna, M. S.A., Lyachek, Y. T., Musaeed, A. M.O., Esmail, Y. A. H, & Adam, A. B. M. (2020). Smart system of a real-time pedestrian detection for smart city. *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 45–50. https://doi.org/10.1109/EIConRus49466.2020.9039333.

Bai, R., Shen, F., Wang, M., Lu, J., & Zhang, Z. (2023). *Improving detection capabilities of YOLOv8-n for small objects in remote sensing imagery: towards better precision with simplified model complexity*. https://doi.org/https://doi.org/10.21203/rs.3.rs-3085871/v1.

Barakat, M., Chung, G. C., Lee, I. E., Pang, W. L., & Chan, K. Y. (2023). Detection and sizing of durian using Zero-Shot Deep Learning models. *International Journal of Technology*, *14*(6), 1206–1215. https://doi.org/10.14716/ijtech.v14i6.6640.

Carneiro, T., Medeiros Da Nobrega, R. V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, *6*, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767.

Dai, G., Hu, L., & Fan, J. (2022). DA-ActNN-YOLOV5: Hybrid YOLO v5 model with data augmentation and activation of compression mechanism for potato disease identification. *Computational Intelligence and Neuroscience*, *2022*, 1–16. https://doi.org/10.1155/2022/6114061.

Fang, Y., & Pang, H. (2024). An improved pedestrian detection model based on YOLOv8 for dense scenes. *Symmetry*, *16*(6), 716. https://doi.org/10.3390/sym16060716.

Gao, F., Wang, C., & Li, C. (2020). A combined object detection method with application to pedestrian detection. *IEEE Access*, *8*, 194457–194465. https://doi.org/10.1109/access.2020.3031005.

Gong, S. (2023). Research advanced in pedestrian detection based on deep learning. *Highlights in Science, Engineering and Technology TPCEE, 2023*.

Ivašić-Kos, M., Krišto, M., & Pobar, M. (2019). Human detection in thermal imaging using YOLO. *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, *Part F148262*, 20–24. https://doi.org/10.1145/3323933.3324076.

Jooshin, H. K., Nangir, M., & Seyedarabi, H. (2024). Inception-YOLO: computational cost and accuracy improvement of the YOLOv5 model based on employing modified CSP, SPPF, and inception modules. *IET Image Processing*, *18*(8), 1985–1999. https://doi.org/10.1049/ipr2.13077

Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). Multi-label classifier performance evaluation with Confusion Matrix. *Computer Science & Information Technology*, 01–14. https://doi.org/10.5121/csit.2020.100801.

Legaspi, K. R. B., Sison, N. W. S., & Villaverde, J. F. (2021). Detection and classification of whiteflies and fruit flies using YOLO. *2021 13th International Conference on Computer and Automation Engineering, ICCAE 2021*, 1–4. https://doi.org/ 10.1109/ICCAE51876.2021.9426129.

Li, R., & Yang, J. (2018). Improved YOLOv2 object detection model. *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, 1–6. https://doi.org/10.1109/ICMCS.2018.8525895.

Sary, I. P., Armin, E. U., Andromeda, S., Engineering, E., & Karawang, U. S. (2023). Performance comparison of YOLOv5 and YOLOv8 architectures in human detection using aerial images. *Ultima Computing : Jurnal Sistem Komputer*, *15*(1).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690.

Ryu, S. E., & Chung, K. Y. (2021). Detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization. *Applied Sciences (Switzerland)*, *11*(15). https://doi.org/https://doi.org/10.3390/app11157093.

Sharma, K., Rawat, S. S., Parashar, D., & Sharma, S. (2023). State of-the-art analysis of multiple object detection techniques using Deep Learning. *International Journal of Advanced Computer Science and Applications*, *14*(6), 527–534. https://doi.org/10.14569/IJACSA.2023.0140657.

Mishra, S., Verma, M., Akhtar, N., Chaturvedi, S., & Perwej, Y. (2022). An intelligent motion detection using OpenCV. *International Journal of Scientific Research in Science, Engineering and Technology*, 51–63. https://doi.org/10.32628/IJSRSET22925.

Wu, D., Lv, S., Jiang, M., & Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Computers and Electronics in Agriculture*, *178*, 105742. https://doi.org/10.1016/j.compag.2020.105742.

Zadobrischi, E., & Negru, M. (2020). Pedestrian detection based on TensorFlow YOLOv3 embedded in a portable system adaptable to vehicles. *2020 15th International Conference on Development and Application Systems, DAS 2020 - Proceedings*, 21–26. https://doi.org/10.1109/DAS49615.2020.9108940

Zhang, Z. (2023). Drone-YOLO: An efficient neural network method for target detection in drone images. *Drones*, *7*(8), 526. https://doi.org/10.3390/drones7080526