Implementing YOLOv9 and Faster R-CNN for Multi-Class Vehicle Recognition

Rosalia Arum Kumalasanti
Department of Informatics
Sanata Dharma University
Yogyakarta, Indonesia
rosalia.santi@usd.ac.id

Bernardus Hersa Galih Prakoso
Department of Informatics
Sanata Dharma University
Yogyakarta, Indonesia
bernardus.hersa37@gmail.com

Hari Suparwito
Department of Informatics
Sanata Dharma University
Yogyakarta, Indonesia
shirsj@jesuits.net

Agnes Maria Polina
Department of Informatics
Sanata Dharma University
Yogyakarta, Indonesia
a.m.polina@usd.ac.id

***Abstract*—Computer vision has become one of the rapidly developing fields of knowledge. One of the implementations of computer vision is object detection and recognition. Two prominent object detection algorithms that have gained significant attention are YOLOv9 and Faster R-CNN. The purpose of this study is to apply the YOLOV9 and Faster R-CNN algorithms to detect objects in the form of vehicles on the road such as motorcycles, cars, buses and trucks. Data are taken from CCTV recordings belonging to the department of transportation. 5963 image data have been collected from May to June 2024. Data were divided to 70% as training data, 20% as validation data and 10% as testing data. Further, all data was preprocessed by converting to YOLO and faster R-CNN format then resizing the image to 640x640. The analysis results show that YOLOv9 algorithm is superior to faster R-CNN with F1-score value of 0.938 and mAP@50:95 of 0.937. The results of testing datasets also show that YOLOv9 is good to detect vehicle on the road both during the day and night.**

Keywords— classification, computer vision, faster R-CNN, object detection, yolov9

I. INTRODUCTION

Computer vision has become one of the fastest growing and reliable fields of knowledge that is utilized in various industries [1]. The advantage of computer vision lies in extracting information from images, videos, and other visual inputs, which can then be further processed [2]. In other words, computer vision has the potential to integrate human interaction with systems in a modern way so that future technology will always be up to date [3]. Of the various fields that can be solved with computer vision applications, one that is quite important is object detection. Object detection in computer vision can be applied in various fields such as autonomous vehicles [4], video surveillance [5], and image understanding [6]. To perform object detection, certain modelling algorithms are needed and currently there are two leading object detection algorithms that have received significant attention, namely YOLOv9 and Faster R-CNN [7], [8].

YOLOv9 is known for its impressive speed and real-time performance [9]. The performance of YOLOv9 is done by integrating candidate box extraction, feature extraction, target classification, and target localization into a single deep neural network, which enables end-to-end training and turns the detection problem into a regression task. With such methods, the YOLOv9 algorithm has been shown to outperform other object detection techniques such as Faster R-CNN and SSD in terms of speed, while maintaining a good balance between speed and accuracy [10]. The superiority of YOLOv9 is demonstrated by achieving good results on several benchmark datasets, such as COCO and Pascal VOC [11], [12].

Another algorithm that also excels in object detection is Faster R-CNN. This algorithm is an improved version of the original R-CNN framework. Faster R-CNN excels in object detection because it uses Region Proposal Network (RPN) to generate region proposals, which are then classified and refined to produce the final detection result [13]. With this method, the faster R-CNN has shown superior performance on more complex and dense object detection tasks [14].

In some previous studies, the YOLOv9 algorithm is generally faster and more efficient than Faster R-CNN [15]. This capability makes YOLOv9 more suitable for real-time applications. The methods in the YOLOv9 algorithm can process images at much higher frame rates, up to 45 FPS on the GPU [16], compared to Faster R-CNN which is typically slower at around 7 FPS. YOLOv9 also has a simpler and leaner architecture, with a single neural network performing all detection tasks, which contributes to its efficiency [17]. As such, the YOLOv9 algorithm would fare better if deployed on embedded devices and mobile devices more easily, making it a better choice for applications that require fast and low-latency object detection, such as driverless cars or robotics. However, other studies have shown that the Faster R-CNN algorithm is generally more accurate than YOLOv9, especially in detecting small objects and handling occlusions. This is due to its region proposal network and multi-stage detection pipeline, which enables more precise feature extraction and object localization [18]. A comparison of these two algorithms shows that the YOLOv9 algorithm is a better choice for applications that require fast and low-latency object detection.

Considering the strengths and weaknesses of YOLOv9 and Faster R-CNN algorithms, this study aims to provide a comparison of the performance of YOLOv9 and Faster R-CNN algorithms in multi-class vehicle detection. The reason for taking traffic streaming video data is because the objects obtained vary in size and shape, for example motorcycles, cars, buses and trucks. If there is a traffic jam, the object will become occluded and the object will also become denser. This supposition is expected to test the performance of the YOLOv9 algorithm and faster R-CNN in detecting objects.

II. Material and Methods

A. Data

A robust object detection system lies in the availability of a comprehensive and diverse dataset. Various methods can be used to collect data, such as carefully setting up camera equipment, recording videos, and utilizing web scraping techniques to obtain images depicting objects of interest under diverse lighting conditions, sizes, and orientations. In the data collection stage of this study, we prioritized capturing a diverse dataset of streaming video images that included objects of interest. This involved collecting images under various scenarios, ensuring that the data covers different lighting conditions (day and night), and includes various object sizes and orientations. The purpose of diversifying the dataset is to improve the robustness and generalizability of the model. This comprehensive approach is essential to accurately train the object detection algorithms to perform well under real-world conditions. Each method was chosen for its effectiveness in capturing the desired diversity and detail. By combining these data capture techniques, we ensure that our data set is comprehensive and representative, thus laying a good foundation for the next stage of model analysis and training.

In this study, data was collected from CCTVs owned by the ministry of transportation that record local traffic at specific places. The data was collected from May to June 2024 between 10am and midnight. Not all video streaming data was used in this study. The dataset used in this study consists of 3,000 images at night and 2,963 images during the day. The CCTV video recordings were selected as needed based on differences in lighting (day or night), location, and diversity of passing vehicles. This data variation serves to provide a comparison of results in image detection. Below is an example of CCTV data taken in the form of images. For each CCTV location, the video image taken is about 2 minutes. Some examples of raw data are shown in Fig.1, Fig. 2 and Fig.3 while the raw data in the form of streaming video could be accessed at the following link (gdrive: https://l1nk.dev/76B1b)

B. Methods

In general, there are several steps taken to analyze video image data obtained from CCTV recordings. Fig.4 shows the steps of the research. The first step is a data collection. In this process, data were collected from the CCTV recording of the Department of transportation. The next process after data collection is data preprocessing. The first thing is to convert the video image data into image data (photos) so that it can be labeled. for each object in the photo. In this study, vehicle objects are labeled into four: motorcycle, car, bus and truck. Next, the image must be converted according to the YOLOv9 format and faster R-CNN. This format change must be done so that the image can be read by both algorithms. for each image that has been labeled, converted into YOLO format and record format for faster R-CNN.

Fig.1 Image data taken at noon

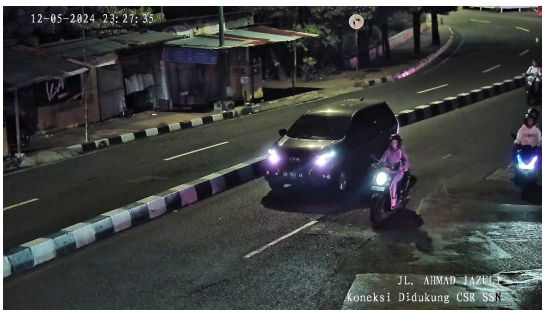


Fig. 2. Image data taken at night

Fig.3 Image data taken at the rush hour

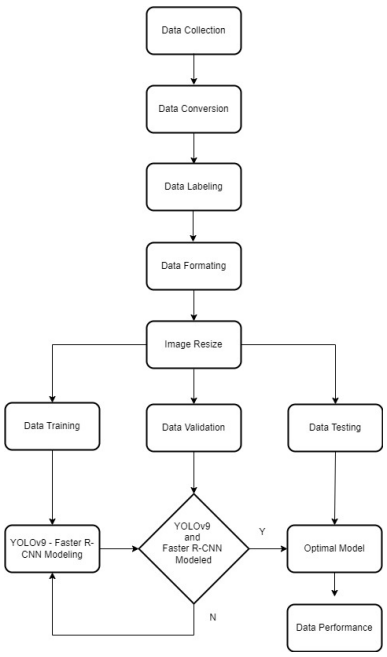


Fig. 4 The steps of the research

After the image is available in the appropriate format, the next step is to resize the image with padding to 640x640 using the available framework. The last step is to divide the image data randomly to be grouped as training data, validation data and testing data. The dataset will be divided into training, test, and validation, where the number of data in each dataset is 70% for training dataset, 20% data for validation dataset and 10% As a testing dataset.

Faster R-CNN effectively produces region proposals within the same deep learning framework and this architecture is made up of a number of interested components intended to improve recognizing objects by speed and accuracy. Fig.5 is an overview of the Faster R-CNN architecture in the conducted research.

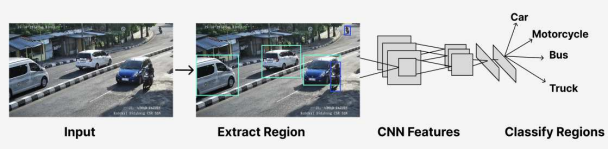


Fig. 5 Faster R-CNN architecture

The step after data preprocessing is modeling. YOLOv9 and faster R-CNN algorithms were implemented on the data in the training dataset. If the optimal model has been found then a parameter test is carried out on the validation data to determine whether overfitting occurs or not. Next, the optimal model would be tested on the testing data. In the modelling step, two experiments were carried out by changing the activation function parameters and altering the number of layers, while others parameters have a default value such as: Table 1 shows the experiment and the parameter that has been implemented.

In general, these parameters were chosen based on best practices for object detection models and tailored for the dataset and computational resources. They aim to optimize training efficiency while maintaining high performance and generalization capability. In some cases, we chose parameters that are quite significant in providing optimal computational results. For example, we used a small batch size because A small batch size allows for more granular updates to the model weights and is useful when working with limited memory resources. While larger batch sizes can stabilize training, smaller batches provide better generalization. Relu and Softmax were chosen because **ReLU** ensures sparse activation, reducing computational cost and mitigating the vanishing gradient problem. **Softmax** is used in the final layer for multi-class probability predictions, as it transforms raw scores into probabilities. A small learning rate ensures stability during training, avoiding drastic weight updates and facilitates faster convergence while retaining stability in optimization. SGD and Adam optimizer were chosen for their ability to generalize well, especially for tasks like object detection. It ensures stability and avoids overfitting when paired with momentum. We decided the momentum value was quite high because High momentum helps the optimizer maintain direction in parameter updates, accelerating convergence and overcoming small local minima during training.

The last step is model evaluation. Here, F1-score and mAP values would be measured for the optimal object detection model.

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

III. Results and Discussion

The results of F1-score and mAP values of the training and validation datasets are shown in table 2 and table 3. The model's performance on the dataset is shown by metrics called training and validation loss. The loss function table (table.4) shows that the model is optimal in the training and validation dataset. It can be seen that the comparison results of F1-score and mAP of YOLOv9 and faster R-CNN algorithm show YOLOv9 is more optimal than faster R-CNN in the case of object detection of video streaming traffic data.

For the testing dataset, we only implemented the optimal model namely YOLOv9. Testing dataset contains data that is completely different from training and validation data.

TABLE 1. Experiments and its parameters

Experiments	Parameters
YOLOv9m	Epoch = 50 Batch = 5 Activation function = relu-softmax Number of layers = 603 Optimizer = AdamW Learning rate = 0.00125 Momentum = 0.9
Faster R-CNN	Backbone = Resnet50 Training steps = 10000 Batch = 2 Optimizer = SGD Learning rate = 0.02 Momentum 0.8 Activation function = Relu - Softmax Scale = [0.25, 0.5, 1, 1.5, 2] Aspect ratio = [0.5, 1, 1.5, 2]

TABLE 2. F1-score and mAP results of training dataset

Model	F1-score	mAP@50:95
YOLOv9	0.938	0.937
Faster R-CNN	0.806	0.899

TABLE 3. F1-score and mAP results of validation dataset

Model	F1-score	mAP@50:95
YOLOv9	0.970	0.924
Faster R-CNN	0.858	0.896

Table 2 and table 3 show the results of a very significant difference in F1-score values between YOLOv9 and faster R-CNN. this shows that the YOLO algorithm is more accurate because YOLO can detect objects faster than faster R-CNN. as is known that the research object data is a running vehicle. in other words, in order to detect a running object, the process in the object detection algorithm needs to be faster. one of the things that makes YOLO able to quickly carry out the detection process because of YOLO's simple architecture with only one process. according to reference [10] YOLO can process up to 45 FPS while faster R-CNN is only 7 FPS, this can also explain why the YOLO algorithm can detect moving objects better.

TABLE 4. The training and validation loss function values

		Classification loss	Box loss
YOLOv9	Training	0.115	0.036
	Validation	0.264	0.092
Faster R-CNN	Training	0.182	0.323
	Validation	0.256	0.336

TABLE 5. The comparison of F1-score and mAP on testing dataset at day and night

	Testing dataset	
	F1-score	mAP@50:95
Day	0.937	0.868
night	0.943	0.788

the comparison of F1-score values between data taken during the day and night shows a value difference that is not too significant. this means that YOLO v9 can detect objects well in different light conditions.

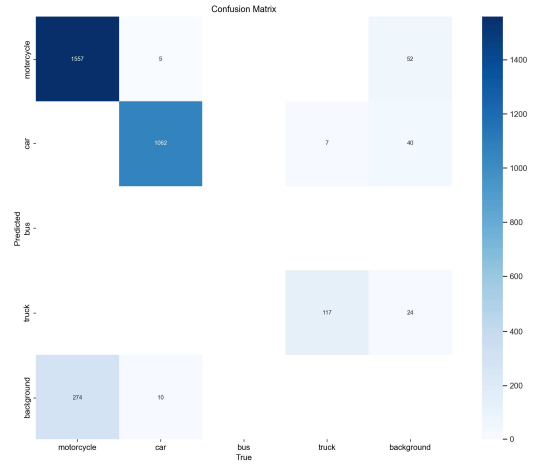


Fig. 6 the Confusion matrix result on testing dataset at the day

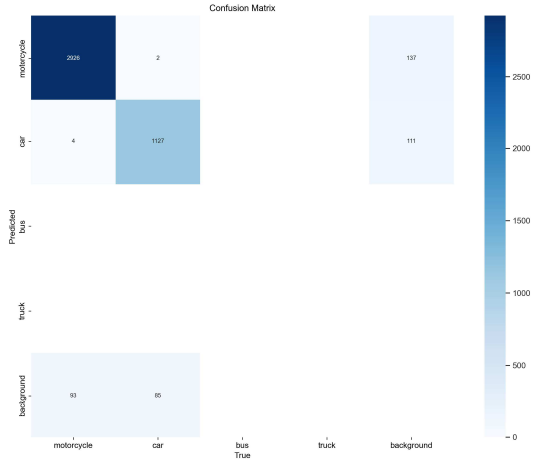


Fig. 7 the Confusion matrix on the testing dataset at night

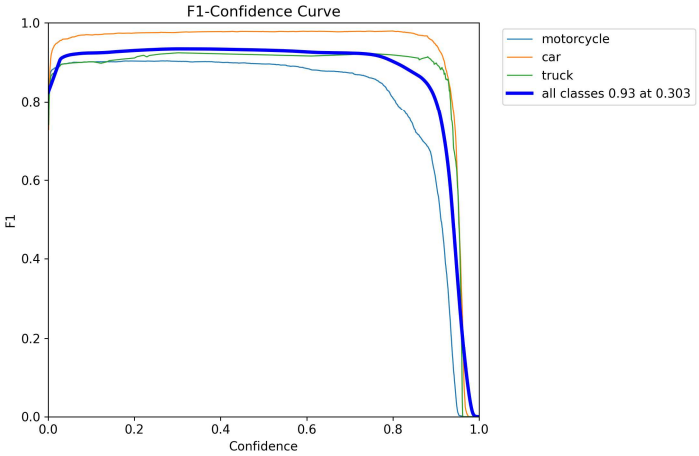


Fig. 8. F1- score confidence on testing dataset at the day

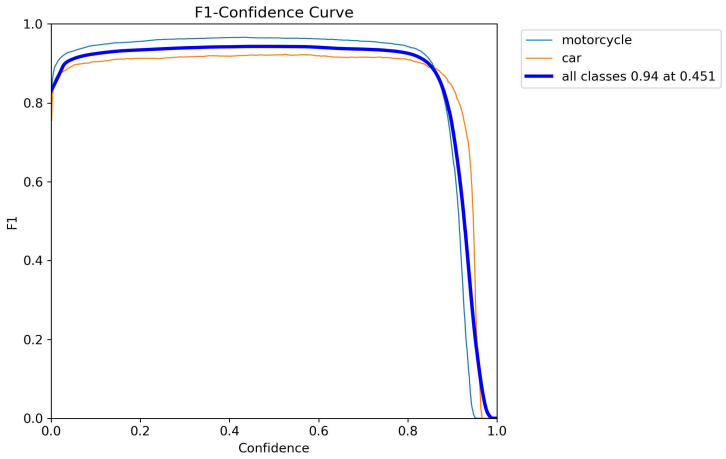


Fig. 9. F1-score confidence on testing dataset at night

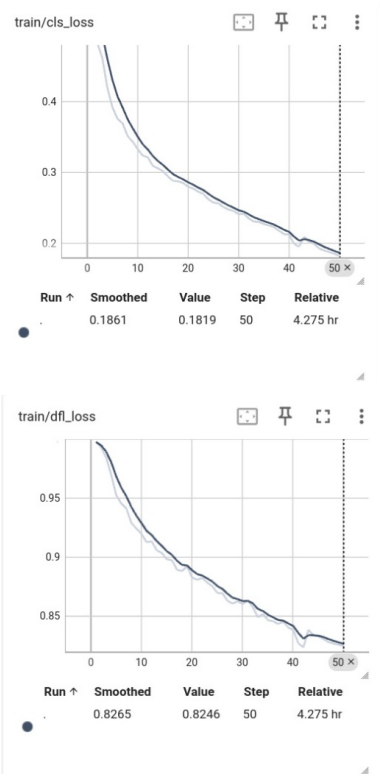
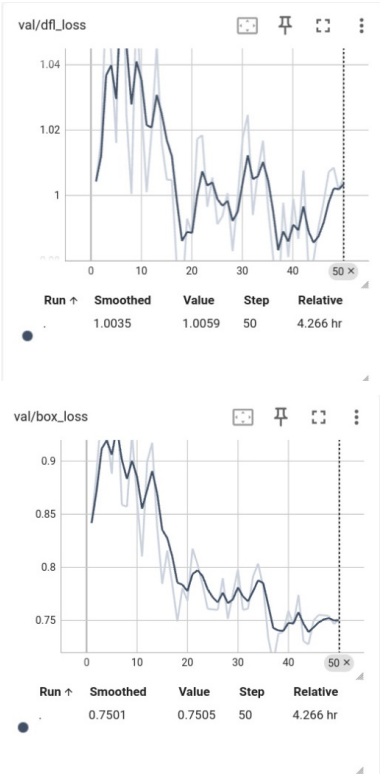


Fig. 10. YOLOv9 Training



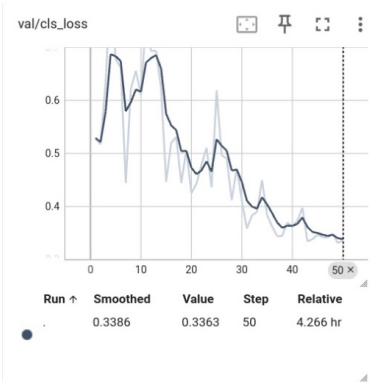


Fig. 11. YOLOv9 Validation loss function

Training and validation loss are metrics that indicate how well the model is performing on both the training dataset and the unseen validation dataset, respectively [12]. The training and validation loss graphs in fig. 8 and 9 show that the YOLOv9 model is good enough to be used as a model in testing data. This is indicated by the similarity of the patterns of the two graphs

Fig. 12. the annotated image on testing dataset at noon. Three classes were detected i.e., motorcycles, cars, and trucks

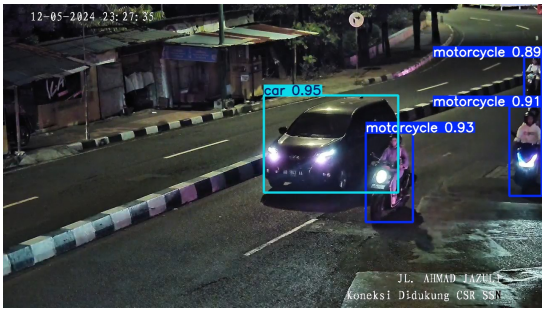


Fig. 13. the annotated image on testing dataset at night. Two classes are detected namely motorcycles and cars

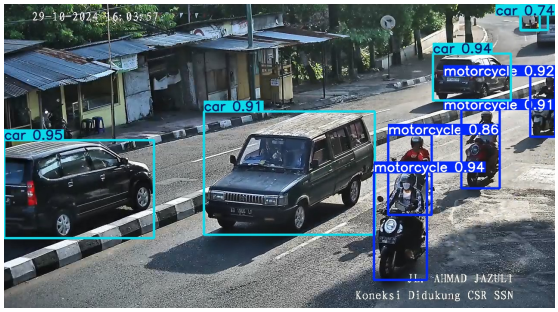


Fig. 14. YOLOv9 can capture and distinguish the 2 classes well even when the road is crowded

It can be seen that the comparison results of F1-score and mAP of YOLOv9 and faster R-CNN algorithm show that YOLOv9 is more optimal than faster R-CNN about 13% for f1-score and 4% for mAP in the case of object detection of video streaming traffic data. The factors affecting model accuracy and F1-score are mostly related to the proper tuning of hyperparameters such as learning rate, batch size and momentum, to ensure efficient learning and stable convergence. In this study, the superior performance of YOLO over the faster R-CNN is most likely due to the effective selection of hyperparameters, especially the optimizer and learning rate.

Next, YOLOv9 algorithm would be implemented on testing dataset. Testing dataset contains completely different data from training and validation data. The results of the implementation of YOLOv9 algorithm are shown in Figures 12, 13, and 14. In Figure 14, YOLOv9 is capable of detect three classes object. In figure 12, the algorithm could detect two classes object, motorcycles and cars, though at night condition where the area is lack of light.

Figure 10 shows that the faster R-CNN can detect multi-class objects well where the algorithm can distinguish between motorcycles, cars and trucks well. Buses objects cannot be shown because there were no buses passing by during the data collection period.

From all the detection results on the testing dataset, it can be seen that YOLO can detect above 90% confidence level in various conditions whether it is day or night or in crowded road conditions. In all figure, it can be seen that the motorcycles under the shade with a rather dark situation could not be detected.

IV. Conclusion

The performance of the YOLOv9m and Faster R-CNN algorithms for multi-class vehicle recognition on CCTV traffic video data obtained in different illumination circumstances is assessed in this study. The findings demonstrate that, in terms of accuracy and computational efficiency, YOLOv9m performs noticeably better than Faster R-CNN. In particular, on the training dataset, YOLOv9m obtained an F1 score of 0.938 and mAP@50:95 of 0.937, whereas Faster R-CNN obtained an F1 score of 0.806 and mAP@50:95 of 0.899. In the validation dataset, YOLOv9m surpassed Faster R-CNN, achieving an F1-score of 0.970 compared to 0.858. Furthermore, YOLOv9m demonstrated robust performance in a variety of settings, efficiently identifying cars in both day-time and night-time conditions with no F1-score decline (0.937 during the day and 0.943 at night). This illustrates the model's flexibility to fluctuations in lighting and environmental circumstances. The faster R-CNN has slightly better precision in some cases, but its processing speed limits its real-time performance. In conclusion, YOLOv9m excelled at real-time multi-class vehicle recognition, balancing accuracy and speed. Future research may examine the approach under occlusion and congestion conditions to improve reliability and robustness.

Acknowledgment

Thank you to the Intelligent System laboratory of the Informatics Department of Sanata Dharma University for allowing us to use the laboratory to do this research. We would also like to thank the Institute for Research and Community Service of Sanata Dharma University for funding this research.

References

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” Comp. Intelli. Neurosc., vol.1, 2018, 7068349.

[2] Wiley, Victor, and T. Lucas, "Computer vision and image processing: a paper review," International Journal of Artificial Intelligence Research, vol. 2, no. 1, 2018, pp. 29-36.

[3] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, and L. A. Gabralla, "A review on machine learning styles in computer vision—techniques and future directions," *IEEE Access*, vol. 10, pp. 107293–107329, 2022.

[4] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-object detection and tracking, based on DNN, for autonomous vehicles: A review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5668–5677, 2020.

[5] S. Jha, C. Seo, E. Yang, and G. P. Joshi, "Real-time object detection and tracking system for video surveillance system," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3981–3996, 2021.

[6] Cazzato, Dario, C. Cimarelli, J.L. Sanchez-Lopez, H. Voos, and M. Leo, "A survey of computer vision methods for 2d object detection from unmanned aerial vehicles," Journal of Imaging, vol. 6, no. 8, 2020, pp. 78.

[7] M. Maity, S. Banerjee, and S. S. Chaudhuri, "Faster R-CNN and YOLO based vehicle detection: A survey," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, pp. 1442–1447.

[8] Sapkota, Ranjan, Z. Meng, D. Ahmed, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive performance evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments," arXiv preprint arXiv:2407.12040, 2024.

[9] M. A. R. Alif and M. Hussain, "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain," *arXiv preprint arXiv:2406.10139*, 2024.

[10] Ali, M. Liaqat, and Z.Zhang, "The YOLO framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection, 2024.

[11] Y. Wang, Q. Rong, and C. Hu, "Ripe tomato detection algorithm based on improved YOLOv9," *Plants*, vol. 13, no. 22, p. 3253, 2024.

[12] G. C. Sunil, A. Upadhyay, Y. Zhang, K. Howatt, T. Peters, M. Ostlie, W. Aderholdt, and X. Sun, "Field-based multispecies weed and crop detection using ground robots and advanced YOLO models: A data and model-centric approach," *Smart Agricultural Technology*, vol. 9, p. 100538, 2024.

[13] Hussain, Muhammad, and R. Khanam, "In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection," Solar, vol. 4, no. 3, pp. 351-386. MDPI, 2024.

[14] Xia, Baizhan, H. Luo, and S. Shi, "Improved faster R-CNN based surface defect detection algorithm for plates," Computational Intelligence and Neuroscience, no. 1, 2022, 3248722.

[15] Liu, Yu. "An improved faster R-CNN for object detection," 11th international symposium on computational intelligence and design (ISCID), vol. 2, 2018, pp. 119-123. IEEE.

[16] E. Gallagher, James, and E. J. Oughton, "Surveying You Only Look Once (YOLO) Multispectral Object Detection Advancements, Applications And Challenges," arXiv preprint arXiv:2409.12977, 2024.

[17] Lin, Deyu, J. Zhao, F. Yu, W. Min, Y. Zhao, and L.G. Yong, "A novel high-precision and low-latency abandoned object detection method under the Hybrid Cloud-Fog Computing Architecture," IEEE Internet of Things Journal, 2024.

[18] Joiya, “Object Detection: Yolo Vs Faster R-CNN,” *International Research Journal of Modernization in Engineering Technology and Science*, Sep. 2022, doi: 10.56726/irjmets30226.