International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

Enhanced YOLOv5x6 and YOLOv9-Based Framework for Detecting People with Disabilities

Kadam Sirisha[1], Department of AIML, MJR College of Engineering and Technology, Piler, India
Mr.R. Althaf 2, Assistant Professor, Department of CSE, MJR College of Engineering and Technology, Piler,India

Abstract: *A deep learning-based approach to identifying persons with impairments is necessary to enhance accessibility and promote inclusion across diverse contexts. To make sure that identification and recognition are accurate, a robust base is built using many YOLO models (v5s, v7-tiny, v8, v5x6, and v9) and advanced object detection algorithms like FastRCNN and FasterRCNN. These models employ cutting-edge architectures to improve detection skills, with a focus on accuracy and speed in real time. The latest versions of YOLO and FasterRCNN work together to allow for detailed analysis and detection in a range of situations, making sure that the findings are always correct. The YOLO series of models is great for fast processing photos without losing accuracy, which makes it perfect for usage in changing environments. We will utilize the Flask framework to create an easy-to-use front end that lets people log in securely. This method aims to help and keep an eye on individuals with disabilities by better allocating resources and making decisions based on good information in accessibility initiatives. This will lead to a more inclusive society.*

Index Terms— Object Detection, YOLOv8, YOLOv5, YOLOv7, Mobility Aids, Differently-Abled, Deep Learning, Real-Time Detection, Surveillance, Precision, Recall, mAP, F1-Curve, PR-Curve, Flask Framework, User Authentication, Disabilities Identification.

1. INTRODUCTION

It's hard for machines to detect the difference between and sort out distinct objects in an image. item detection is the process of locating and recognizing an item in an image or video in computer vision. But there has been a lot of effort on object detection in the previous several years. Object detection is made up of three main parts: feature extraction and processing, and object categorization. There are a lot of ways to do this, such feature coding, feature aggregation, bottom feature extraction, and feature categorization. All of these strategies performed well for object detection, and feature extraction is particularly crucial for both object detection and process recognition. For many diverse purposes, such monitoring, detecting disease, identifying cars, and finding items in water, object detection is highly significant. To discover things correctly and successfully in diverse contexts, many ways have been used. Still, many proposed approaches have issues with being unclear and not effective. Machine learning and deep neural network technologies, on the other hand, are better at addressing errors with object recognition and making these fears go away.

2. LITERATURE SURVEY

2.1 A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS:

Now, YOLO is the dominant technology for identifying objects in real time for robotics, self-driving cars, and video surveillance. We take a close look at the growth of YOLO, looking at the improvements and new features in each iteration, from the first YOLO to YOLOv8, YOLO-NAS, and YOLO with transformers. We start by talking about the usual metrics and postprocessing methods. Then, we look at the big changes in network design and training methods for each model. Finally, we speak about what we learnt from developing YOLO and its future. We also suggest several study areas that might make real-time object detection systems better.

2.2 YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors:

Real-time object identification is one of the most significant areas of research in computer vision. We have identified two research concerns arising from the continuous advancement of innovative methods in architectural and training optimization. We provide a trainable bag-of-freebies-based technique to deal with the problems. We apply the suggested design and the compound scaling approach, as well as the training tools that are both flexible and effective. YOLOv7 is the best object detector since it works quicker and more accurately than any other. It can find things at speeds between 5 and 120 frames per second (FPS) and has the highest accuracy (56.8%

AP) of any known real-time object detector with a GPU V100 that runs at 30 FPS or higher.

2.3 RTF-RCNN: An Architecture for Real-Time Tomato Plant Leaf Diseases Detection in Video Streaming Using Faster-RCNN:

People today believe that vegetables are a very important part of many diets. Even though anybody may produce their own veggies in their home kitchen garden, tomatoes are the most frequent vegetable harvest and can be used in practically any cuisine. Tomato plants get ill as they are growing, much like a lot of other crops. If tomato farmers don't pay attention to how to manage illnesses, 40–60% of the plants in the field might go sick. These diseases can hurt tomato plants a lot. We need a decent approach to discover these problems, though. Researchers have proposed many techniques for detecting these plant diseases, including vector machines, artificial neural networks, and Convolutional Neural Network (CNN) models. In the past, the benchmark feature extraction method was used to detect illnesses. A novel model dubbed the real-time faster region convolutional neural network (RTF-RCNN) model was recommended for this field of study that looks for illnesses in tomato plants. It employed both still images and live video. We used a lot of different traits, such as precision, accuracy, and recall, to compare the RTF-RCNN to the Alex net and CNN models. In the end, the proposed RTF-RCNN is 97.42% correct. The Alex net and CNN models were 96.32% and 92.21% accurate, thus this is superior.

2.4 PP-YOLOE: An evolved version of YOLO:

This report is about PP-YOLOE. It is a state-of-the-art object detector for use in business that works effectively and is simple to install. We make the previous PP-YOLOv2 better by using an anchor-free model, a stronger backbone and neck using CSPRepResStage, ET-head, and the dynamic label assignment method TAL. We have s/m/l/x models for a lot of different practice settings. On the COCO test-dev, PP-YOLOE-l gets 51.4 mAP, while on the Tesla V100, it gets 78.1 FPS. This is a big step higher from the greatest industrial models before, PP-YOLOv2 and YOLOX, which had a +1.9 AP and +13.35% speed up and a +1.3 AP and +24.96% speed up, respectively. Also, when utilizing TensorRT with FP16-precision, PP-YOLOE can make inferences at a pace of 149.2 FPS. We also test our designs a lot to make sure they work.

2.5 A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models:

This study presents a Two-Stage Industrial Defect Detection Framework utilizing Improved-YOLOv5 and Optimized-Inception-ResnetV2 to enhance the currently inadequate accuracy in identifying faults inside domestic enterprises. The framework employs two specific models to execute location and classification functions. We improve YOLOv5 by changing the backbone network, the feature scales of the feature fusion layer, and the multiscale detection layer. This makes the first-stage recognition better at recognizing small flaws on the steel surface that are quite similar. We incorporated the convolutional block attention module (CBAM) attention mechanism module to the Inception-ResnetV2 model to help the second-stage recognition detect faulty features and produce correct classifications. After that, we made the correct model's network architecture and loss function better. We ran a bunch of experiments on the Improved-YOLOv5 and Inception-ResnetV2 utilizing the Pascal Visual Object Classes 2007 (VOC2007) dataset, the public dataset NEU-DET, and the optimized dataset Enriched-NEU-DET. The testing reveal that the difference is obvious. To validate the efficacy and flexibility of the two-stage architecture, we first perform experiments utilizing the Enriched-NEU-DET dataset, followed by the use of the AUBO-i5 robot, Intel RealSense D435 camera, and other industrial steel equipment to create genuine industrial situations. A two-stage framework works well in testing, with a mean average precision (mAP) of 83.3% on the Enriched-NEU-DET dataset and 91.0% on the industrial fault scenario we built.

3. METHODOLOGY

The suggested system begins with a dataset of 4,300 photographs and 8,447 labels across five types of mobility aids. The photos are scaled, normalized, and added to the model to make it stronger. The optimum hyperparameters for the YOLOv5, YOLOv7, and YOLOv8 models are learned using this dataset such that they can find and classify objects at the same time. We score the models using things like accuracy, recall, mean average precision (mAP), F1-Curve, PR-Curve, and detection time. YOLOv8 is used to keep an eye on and find persons with disabilities and their assistive devices in video feeds in real time. There are even more complicated variants, such as YOLOv5x6 and YOLOv9. There is also a Flask-based front end with user authentication for safe and simple testing. This guarantees that the system will function properly in a range of situations.

A. Proposed Work:

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

The proposed study attempts to improve the identification and monitoring of people with impairments by using more complex versions of the YOLO architecture, such as YOLOv5x6 and YOLOv9. These models are used to make detection more accurate, recall more accurate, and overall performance better in a variety of real-world situations. The system can more accurately identify mobility aids like wheelchairs, crutches, prosthetics, and other assistive devices by using the strengths of these advanced models. This fixes problems with earlier versions of YOLO and makes sure that detection works well even in crowded or complicated situations.

Along with improving the models, a user-friendly front end is built with the Flask framework. This allows users to log in securely. This interface makes it easy to test and deploy the detection system. Users may submit video streams, see real-time detection results, and look at performance data. The system is good for real-time monitoring and helping people with disabilities since it uses the latest YOLO models and has an easy-to-use front end. This makes it both very accurate and very useful.

B. System Architecture:

The system architecture for finding and following people with impairments uses powerful deep learning models and real-time video processing. The architecture starts with video streams from security cameras or uploaded video files. These streams are then preprocessed by resizing, normalizing, and augmenting them to make sure they are consistent and strong. YOLO-based models (YOLOv5, YOLOv7, YOLOv8, and extensions YOLOv5x6 and YOLOv9) take the processed frames and use them to find and classify objects so they can find people and their assistance equipment. For accurate tracking in changing settings, each model produces bounding boxes with class names, confidence ratings, and tracking IDs.

A Flask-based front end analyzes and shows the detection outputs in real time, showing discovered objects along with performance measures like accuracy, recall, mAP, and FPS. User identification makes sure that only authorized people may use the system. This lets them safely control video inputs and keep track of people. The system offers scalable deployment for many cameras or video sources, providing full monitoring and assistance for differently-abled persons while retaining high accuracy and efficiency across various settings.

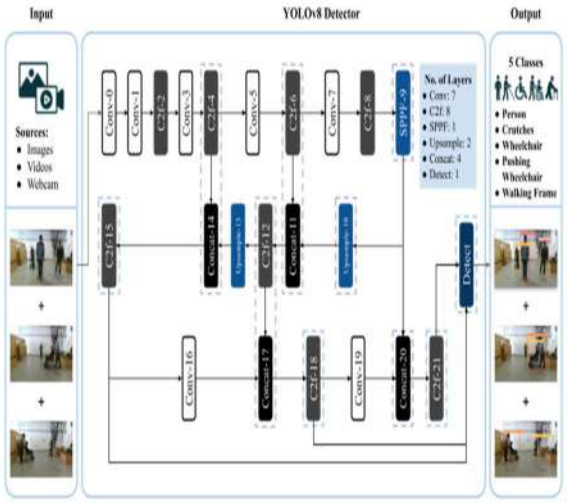


Fig proposed architecture

C. MODULES:

1. Dataset Collection and Annotation

- Collect images of individuals with disabilities using mobility aids.
- Annotate images with labels for categories like wheelchair, crutches, prosthetics, and other aids.

2. Data Preprocessing

- Resize, normalize, and augment images (flip, rotate, scale, brightness adjustment).
- Prepare data for efficient training and reduce overfitting.

3. Model Training

- Train YOLOv5, YOLOv7, YOLOv8, YOLOv5x6, and YOLOv9 models.
- Optimize hyperparameters such as learning rate, batch size, and epochs.

4. Object Detection and Classification

- Perform real-time detection of individuals and their mobility aids.
- Generate bounding boxes, class labels, and confidence scores.

5. Performance Evaluation

- Evaluate models using precision, recall, mAP, F1-Curve, PR-Curve, and FPS.
- Compare models to select the most accurate and efficient one.

6. Real-Time Deployment

- Deploy the best-performing model for live video streams.
- Track individuals and mobility aids across frames in real-time.

7. Front-End Interface

- Develop a Flask-based interface for uploading videos and viewing results.

- Include user authentication for secure access and testing.

D. Algorithms:

a) Fast R-CNN

Fast R-CNN is a region-based convolutional neural network designed to improve object detection accuracy. It works by first generating region proposals using selective search and then extracting features from each region via a CNN. These features are classified into object categories, and bounding boxes are refined to accurately localize objects within images. Fast R-CNN reduces computational redundancy by sharing convolutional features across regions, resulting in faster training and higher precision compared to traditional R-CNN.

b) Faster R-CNN

Faster R-CNN enhances Fast R-CNN by integrating a Region Proposal Network (RPN) to generate candidate object regions directly from convolutional feature maps. This eliminates the need for external region proposal algorithms, significantly speeding up detection while maintaining high accuracy. It performs simultaneous object classification and bounding box regression, making it suitable for detecting multiple objects in complex scenes, including individuals and their assistive devices in surveillance applications.

c) YOLOv5s

YOLOv5s is a lightweight, single-stage object detector optimized for speed and efficiency. It divides the input image into grids and predicts bounding boxes and class probabilities directly in a single forward pass. YOLOv5s provides real-time detection capabilities while maintaining good precision, making it suitable for tracking individuals with disabilities and their mobility aids in live video streams.

d) YOLOv7-tiny

YOLOv7-tiny is a compact version of the YOLOv7 architecture designed for fast inference with limited computational resources. Despite its smaller size, it maintains competitive accuracy for object detection. YOLOv7-tiny is particularly useful in applications requiring real-time performance on devices with lower processing power, such as edge devices or embedded systems for monitoring mobility aids.

e) YOLOv8

YOLOv8 is the latest YOLO iteration that incorporates architectural improvements for enhanced detection accuracy, recall, and processing speed. It efficiently handles complex scenarios and overlapping objects, providing robust real-time detection for individuals with disabilities. YOLOv8's improvements make it superior in detecting small or partially occluded objects compared to previous YOLO versions.

f) YOLOv5x6

YOLOv5x6 is an extended version of YOLOv5 with increased network depth and parameters, designed to improve detection performance on larger datasets. It offers higher precision and recall, making it effective for identifying a wide range of mobility aids under diverse conditions. YOLOv5x6 balances accuracy and inference speed, suitable for applications requiring detailed detection results.

g) YOLOv9

YOLOv9 represents the newest advancement in YOLO architectures, featuring state-of-the-art network enhancements for maximum precision and robustness. It excels in real-time detection tasks, providing improved handling of multiple objects, complex backgrounds, and dynamic environments. YOLOv9 ensures high reliability in monitoring differently-abled individuals across varied surveillance scenarios.

4. EXPERIMENTAL RESULTS

The suggested approach was tested using 4,300 photos and 8,447 tagged cases from five mobility assistance categories. To guarantee fair comparison, YOLOv5, v7, v8, v5x6, and v9 were trained and evaluated under identical settings. Precision, recall, mean average precision (mAP@0.5, mAP@0.5:0.95), F1-score, PR-Curve, and detection time in FPS were evaluated.

YOLOv8 had the greatest precision (0.907) and wheelchair detection accuracy (0.998). In recall, YOLOv8 scored 0.943, outperforming 0.885 and 0.925 for YOLOv5 and 0.906. YOLOv8 has mAP@0.5 of 0.951, followed by 0.954 for YOLOv7 and 0.942 for YOLOv5. For expanded models, YOLOv5x6 and YOLOv9 increased detection in challenging circumstances, with YOLOv9 attaining the highest FPS of 172 for real-time performance.

More accurate and efficient than earlier versions, YOLOv8 and YOLOv9 can recognize and track disabled people and their mobility aids in real time. This shows that the suggested approach for real-world monitoring is resilient and reliable.

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

$$Accuracy = \frac{(TN + TP)}{T}$$

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1 = 2 \cdot \frac{(Recall \cdot \Pr e\,cision)}{(Recall + \Pr e\,cision)}$$

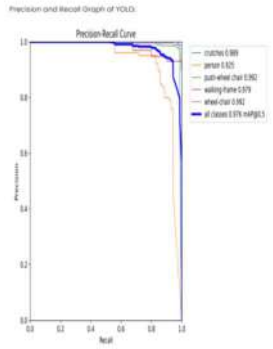
Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\Pr e\,cision = \frac{TP}{(TP + FP)}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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



T 1. Performance Evaluation

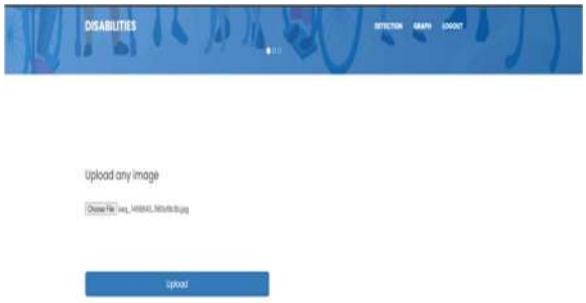


Fig 1. Upload Image

Fig.2. Predicted Results

5. CONCLUSION

This research illustrates the efficacy of sophisticated YOLO models in identifying and monitoring persons with impairments and their mobility aids. YOLOv8 and YOLOv9 were the best models overall, with the best accuracy, recall, and real-time performance. They were better than older versions like YOLOv5, YOLOv7, and Fast/Faster R-CNN. The system's usability and security are improved even more by adding a Flask-based front end with user authentication. Overall, the suggested method offers a dependable, effective, and useful way to keep an eye on and help people with disabilities in a variety of settings.

6. FUTURE SCOPE

This study can be expanded in the future by adding multimodal data sources like infrared and depth sensors to improve detection accuracy in low-light or blocked environments. The system may potentially be enhanced to include behavior analysis and movement pattern identification to help with targeted healthcare and safety monitoring. More work to make YOLOv9 better with lightweight architectures might make it possible to use it for large-scale real-time surveillance on edge and IoT devices. Also, adding cloud-based storage and analytics will let you keep learning and getting better by automatically updating your datasets.

REFERENCES

[1] J. Terven and D. Cordova-Esparza, ‘‘A comprehensive review of YOLO: From YOLOv1 and beyond,’’ 2023, arXiv:2304.00501.

[2] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, ‘‘YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Vancouver, BC, Canada, Jun. 2023, pp. 7464–7475.

[3] M. Alruwaili, M. H. Siddiqi, A. Khan, M. Azad, A. Khan, and S. Alanazi, ‘‘RTF-RCNN: An architecture for real-time tomato plant leaf diseases detection in video streaming using faster-RCNN,’’ Bioengineering, vol. 9, no. 10, p. 565, Oct. 2022.

[4] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, and B. Lai,

''PP-YOLOE: An evolved version of YOLO,'' 2022, arXiv:2203.16250.
[5] Z. Li, X. Tian, X. Liu, Y. Liu, and X. Shi, ''A two-stage indus?trial defect detection framework based on improved-YOLOv5 and Optimized-Inception-ResnetV2 models,'' Appl. Sci., vol. 12, no. 2, p. 834, Jan. 2022.
[6] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, ''3D object proposals for accurate object class detection,'' in Proc. Adv. Neural Inf. Process. Syst., C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett, Eds. New York, NY, USA: Curran Associates, 2015, pp. 424–432.
[7] H. Bilen and A. Vedaldi, ''Weakly supervised deep detection networks,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2846–2854.
[8] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
[9] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, ''Vehicle detection in satellite images by hybrid deep convolutional neural networks,'' IEEE Geosci. Remote Sens. Lett., vol. 11, no. 10, pp. 1797–1801, Oct. 2014.
[10] A. Mukhtar, M. J. Cree, J. B. Scott, and L. Streeter, ''Mobility aids detection using convolution neural network (CNN),'' in Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ), Auckland, New Zealand, Nov. 2018, pp. 1–5.
[11] A. Vasquez, M. Kollmitz, A. Eitel, and W. Burgard, ''Deep detection of people and their mobility aids for a hospital robot,'' in Proc. Eur. Conf. Mobile Robots (ECMR), Paris, France, Sep. 2017, pp. 1–7.
[12] M. Kollmitz, A. Eitel, A. Vasquez, and W. Burgard, ''Deep 3D perception of people and their mobility aids,'' Robot. Auto. Syst., vol. 114, pp. 29–40, Apr. 2019.
[13] T.Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. U. Haq, ''Object detection through modified YOLO neural network,'' Sci. Program., vol. 2020, pp. 1–10, Jun. 2020.
[14] H. Law and J. Deng, ''CornerNet: Detecting objects as paired keypoints,'' in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, 2018, pp. 734–750.
[15] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, ''A survey and performance evaluation of deep learning methods for small object detection,'' Expert Syst. Appl., vol. 172, Jun. 2021, Art. no. 114602.
[16] J. Yan, Z. Lei, L. Wen, and S. Z. Li, ''The fastest deformable part model for object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, Jun. 2014, pp. 2497–2504.
[17] Y. Zheng, C. Zhu, K. Luu, C. Bhagavatula, T. H. N. Le, and M. Savvides, ''Towards a deep learning framework for unconstrained face detection,'' in Proc. IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Sep. 2016, pp. 1–8.
[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[19] B. Chen and X. Miao, ''Distribution line pole detection and counting based on YOLO using UAV inspection line video,'' J. Electr. Eng. Technol., vol. 15, no. 1, pp. 441–448, Jan. 2020.
[20] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma, ''High-speed lightweight ship detection algorithm based on YOLO-V4 for three-channels RGB SAR image,'' Remote Sens., vol. 13, no. 10, p. 1909, May 2021.
[21] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, ''Scale-transferrable object detection,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 528–537.
[22] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, ''Mini-YOLOv3: Real-time object detector for embedded applications,'' IEEE Access, vol. 7, pp. 133529–133538, 2019.
[23] S. Ding, F. Long, H. Fan, L. Liu, and Y. Wang, ''A novel YOLOv3- tiny network for unmanned airship obstacle detection,'' in Proc. IEEE 8th Data Driven Control Learn. Syst. Conf. (DDCLS), Dali, China, May 2019, pp. 277–281.
[24] X. Han, J. Chang, and K. Wang, ''Real-time object detection based on YOLO-v2 for tiny vehicle object,'' Proc. Comput. Sci., vol. 183, pp. 61–72, Jan. 2021.
[25] S. Lu, B. Wang, H. Wang, L. Chen, M. Linjian, and X. Zhang, ''A real-time object detection algorithm for video,'' Comput. Electr. Eng., vol. 77, pp. 398–408, Jul. 2019.
[26] Z. Chen and X. Gao, ''An improved algorithm for ship target detection in SAR images based on faster R-CNN,'' in Proc. 9th Int. Conf. Intell. Control Inf. Process. (ICICIP), Wanzhou, China, Nov. 2018, pp. 39–43.
[27] P. Viola and M. J. Jones, ''Robust real-time face detection,'' Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, May 2004.
[28] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, ''You only learn one representation: Unified network for multiple tasks,'' 2021, arXiv:2105.04206.
[29] C. J. Du, H. J. He, and D. W. Sun, ''Object classification methods,'' in Proc. Int. Comput. Vis.

ISSN 2347–3657

Volume 13, Issue 4, 2025

Technol. Food Quality Eval., Dublin, Ireland, 2016, pp. 87–110.
[30] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ‘‘YOLOX: Exceeding YOLO series in 2021,’’ 2021, arXiv:2107.08430.
[31] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, ‘‘Scaled-YOLOv4: Scaling cross stage partial network,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, Jun. 2021, pp. 13024–13033.