Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 17 No. 21 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijim.v17i21.45201

PAPER

Improved Detection and Tracking of Objects Based on a Modified Deep Learning Model (YOLOv5)

Nadia Ibrahim Nife[1,2](✉), Mohammed Chtourou[2]

[1]University of Kirkuk, Kirkuk, Iraq

[2]Control & Energy Management Laboratory, National School of Sfax Engineers (ENIS), University of Sfax, Sfax, Tunisia

nadia.ibra@uokirkuk.edu.iq

ABSTRACT

Recent years have seen advances in deep learning, including in the field of traffic management. Detecting distant objects that occupy a small number of pixels in the input image is one of the major challenges in computer vision for several reasons, including limited resolution. The challenges of detecting the rotation of objects may be attributed to the deflection of the camera when taking photographs. We recommend enhancing the features of the YOLOv5 network. The proposed method is to train a model on a traffic dataset, which achieves the best inference results through training, testing, and detection on a 1280 × 1280 image for 300 epochs. Moreover, modifications were made to some structural elements of the YOLOv5. In addition to detecting round objects by increasing degrees from 0 to 270, it also increases the probability of flipping in all directions: up, down, left, and right. In addition, the degree of rotation of the image was increased to 90 degrees. The results showed optimized accuracy in detecting distant and small objects, as 73 objects were detected compared to the original YOLOv5 23 objects. It achieved the best number of objects detected in the video (people, cars, and others), and detecting rotating objects increases the number of detected objects (32 objects). The inference time was (23 Ms.) this dataset can make excellent traffic monitoring applications. This model can be deployed on an Android mobile device to provide accurate data about current traffic at a specific location. This is because a mobile device can be used at any time and place. Therefore, in the future, we are working on designing models for object detection that can be operated on mobile devices.

KEYWORDS

computer vision, artificial intelligence, deep learning, CNN models (YOLOv5), object detection, real-time detection

1 INTRODUCTION

Artificial intelligence and deep learning possess the capability to address challenges across diverse domains and have many applications, including object detection. Deep learning encompasses various architectural models, including the

Nife, N.I., Chtourou, M. (2023). Improved Detection and Tracking of Objects Based on a Modified Deep Learning Model (YOLOv5). *International Journal of Interactive Mobile Technologies (iJIM)*, 17(21), pp. 145–160. https://doi.org/10.3991/ijim.v17i21.45201

Article submitted 2023-07-29. Revision uploaded 2023-09-13. Final acceptance 2023-09-29.

convolutional neural network (Yolo model), which is capable of object detection and localization within an image. Object detection is one of the most significant challenges in computer vision [1]. This technology allows us to locate objects in images and videos [2]. Object detection has significantly contributed to various domains of human existence. The applications of object detection are wide-ranging and include monitoring, autonomous driving, and pedestrian detection, among others [3]. Furthermore, the utilization of object detection models on mobile devices, similar to face detection models, serves the purpose of recognizing the device owner and facilitating phone unlocking. In the present study, we propose a modification to the neural network architecture of YOLOv5 in order to enhance its efficacy in detecting objects with low resolution, such as in cars with few pixels in size, using Colab and YOLOv5, which offer pre-trained models for this purpose. YOLOs are utilized in various applications primarily due to their high-speed detection accuracy [4]. The resolution of the input image plays a crucial role in the network's performance. Higher-resolution images contain more information, which facilitates the network's ability to detect objects. In this study, the higher resolution of the input has been considered to improve performance. The optimized version of YOLOv5 is being compared to the original version. Moreover, it is imperative to enhance the training model to detect objects in all orientations effectively. This study has demonstrated a high level of accuracy in predicting the orientation of rotated objects. In the present study, the conducted experiments demonstrate that the modification of YOLOv5 can enhance the detection rate, average accuracy, and recognition of small objects as well as rotated objects.

Researchers conducted a series of experiments aimed at optimizing the performance of object detection through the utilization of deep learning models. Sharma et al. [5] conducted a notable study in this domain. The authors proposed the use of object occlusion as a method of data augmentation to enhance object detection. This technique involves randomly selecting a bounding box within the image and applying erasing and cutout features to simulate occlusion. Kasper et al. [6] employed YOLOv5 to detect heavy vehicles during the winter by implementing a heightened level of data augmentation. Zhao et al. [7] utilized the YOLOv5 framework to construct a model for detecting wheat height. They further improved the network performance by preprocessing the data.

Additionally, their proposed model successfully identified various sizes of wheat types, with a particular emphasis on smaller types. Lewandowski et al. [8] used mobile devices as a means to enhance object recognition applications due to their incorporation of various sensors, including geolocation and sound sensors. Therefore, further research is required to investigate object detection on mobile devices with the aim of minimizing communication efforts.

Zhou et al. [9] have demonstrated that the effectiveness of YOLOv5 has been evaluated through different benchmarks in both training and testing phases. They utilized pre-trained weights to develop a trainable helmet detector. Song et al. [10] conducted network pruning YOLOv5 model by reducing its depth and width. In addition, the enhanced network should be utilized to catch the robot. This study aimed to investigate the feasibility of detecting multiple objects in images. The objective of this study is to improve the performance of the YOLOv5 model by addressing various challenges related to object detection. Therefore, the main contributions of our search model are presented as follows:

Tiny objects may contain a few pixels inside the bounding box. Therefore, it is necessary to increase the input resolution in order to enhance the number of pixels and features within the bounding box. The high resolution includes a

set of algorithms and techniques used to improve the resolution and scaling of an image proportionally to improve network performance and the accuracy of model input.

The architecture of the YOLOv5 neural network has been modified. Modifying certain components of the model architecture, such as connections and other parameters, has a significant impact on the performance of the affected network and addresses various challenges related to object detection. Additionally, the inclusion of output layers contributes to effective training and optimal object detection.

When conducting training for the proposed model, it is recommended to modify the specified parameters by adjusting the image degree parameter. This can be achieved by setting the parameter within the range of 0 to 270 using a ratio-based approach. Furthermore, by augmenting the parameters for vertical and horizontal image flipping, the issue of detecting rotated images at specific angles can be resolved.

This study is organized as follows: Section 1 introduces the proposed topic of study. In Section 2, there is a brief description and training of YOLOv5 object-detection models. Section 3 presents the related work that describes object detection and problems. Section 4 describes the research methodology, including the data set and the proposed approach. The experimental and evaluation results are discussed in Section 5. Finally, the conclusions are presented in Section 6 to summarize the results and provide insights into future directions.

2 DESCRIPTION OF THE YOLOv5 MODEL

You look only once (YOLO) is an object detection model in CNN networks [11]. This neural network focuses on extracting the most probable features from the input image. These features are then used to predict bounding boxes and class labels for each object.

The YOLOv5 network consists of a backbone, a cross-stage partial (CSPDarknet), a neck Path Aggregation Network (PANet), and a head layer (YOLO) [12]. The backbone serves as the foundation for feature extraction, while the neck is responsible for feature merging. The output is then used for prediction [13]. The backbone extracts information from the input image [14]. The neck connects the head and the backbone [15]. The head represents the Yolo layer and is responsible for outputting detection results, including the category and location of the object [16].

2.1 Training of the model (YOLOv5)

The original YOLOv5 network is trained on a 640 × 640 image and consists of three output layers: P3, P4, and P5. The model used in Figure 1 [17] is the original YOLOv5. C3–C5 represent the features extracted by the backbone network, while P3–P5 represent the features integrated by the neck network. Working summary of this algorithm: The images are processed in the input layer and then sent to the backbone for feature extraction. Where the spine extracts features of various sizes, these features combine through the neck to create the P3, P4, and P5 features. The purpose of each layer is to detect objects based on their scale. Layer P3 detects small objects, P4 detects medium objects, and P5 detects large objects [18]. This proposal aims to enhance object detection in images or videos by modifying the YOLOv5 model.

We also modify parameters from the COCO dataset. The training was conducted for 300 epochs, which is the number of times the training would be repeated [19]. Then, we calculate the mean absolute accuracy (map). This research focuses on a convolutional neural network (YOLOv5) model that can accurately detect objects in images and perform real-time detection.

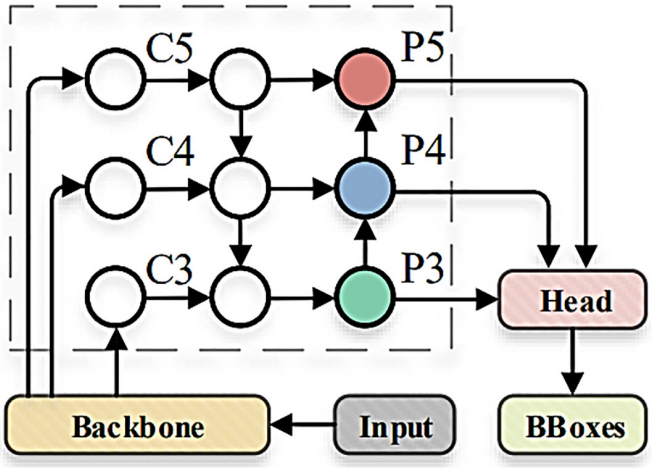

Fig. 1. Original YOLOv5 network

3 RELATED WORKS

The rapid development of deep neural networks has enabled object detection to achieve excellent performance. This research aims to utilize the YOLOv5 model to address the challenges of detecting small, distant, and rotated objects, as well as train it on the original dataset of this model. Many studies have focused on improving the object detection dataset, but more is needed to enhance the quality of detection [20]. In the following studies, Cao et al. [21] demonstrated their experiment aimed at optimizing the detection of small objects with geometric deformation. They achieved this by incorporating deformable convolutional structures to control geometric transformations and integrating multiscale features. Some studies suggested over-sampling images containing small objects during the training process. Despite these improvements, the performance of most current solutions in detecting these objects is low [22]. Therefore, it is necessary to understand the rules on which this algorithm is built in order to improve it and address these problems.

3.1 Object detection

Object detection locates and classifies objects in photos and videos [23]. Deep learning categorizes information through layers of neural networks that contain a set of inputs. Object detection is one of the most challenging problems in computer vision. Therefore, we are working on solving object detection problems using the YOLOv5 model in this paper. Sometimes, the object is not detected, such as by changes in image scale, rotation, or flipping [24]. Through this research, the effect of changes on the detection of these objects in relation to specific tasks has been studied.

3.2 Problems

In recent years, there has been significant progress in object detection, including methods to detect small objects by increasing the amount of data used for training and detection [25]. Despite these improvements, there is still a significant lack of performance. Whereas it is still challenging to detect all objects in a photo or video, these objects may go undetected due to distortion or a lack of image resolution. In addition, objects may be far away, and small objects may interfere with other objects, and objects with few pixels in an image can be challenging to detect accurately. In addition, sometimes, the image to be detected may rotate 90 or 270 degrees, which can cause objects to be detected incorrectly.

4 PROPOSED APPROACH

This study aims to enhance the performance of the YOLOv5 object detector in object detection by modifying certain structural elements of the model and adjusting performance-related parameters.

4.1 Proposed model

In this paper, we study the modification of the architecture of the object detection neural network and propose several significant improvements to enhance the efficiency of object detection using YOLOv5. The primary goal is to detect all objects in the images. Object detection models offer advantages by grouping pixels into convolutional layers, which benefit from training with higher accuracy. Training, testing, and detection in the network with a resolution of 1280×1280 will result in a large model that takes longer to train on your computer.

Therefore, we have several online notebooks available to expedite the training process, such as Google Colab, Kaggle, and Jupyter Notebook. In this study, we suggest using Google Colab because it has a simple user interface and high processing speed. Colab offers free GPUs in addition. The proposed model has been configured. Our proposal structure is outlined in the steps below:

Step 1: We created a data set of traffic images to conduct experiments and use them as inputs for the proposed model.
Step 2: We utilized pre-trained YOLOv5 models on the COCO dataset to detect objects by training them with traffic images.
Step 3: We also modified certain elements of the YOLOv5 network architecture.
Step 4: Change the resolution parameter of the input image size in the training command to 1280×1280.
Step 5: Changing the image rotation coefficients by an angle (90 degrees) and changing the probability of the image flipping up and down or flipping left and right by 50%.
Step 6: The objects in the input images (traffic images) are detected by training the modified YOLOv5 model with a resolution of 1280×1280.

We were able to achieve improved detection using the proposed YOLOv5 model. Figure 2 shows the proposed network diagram and its training.

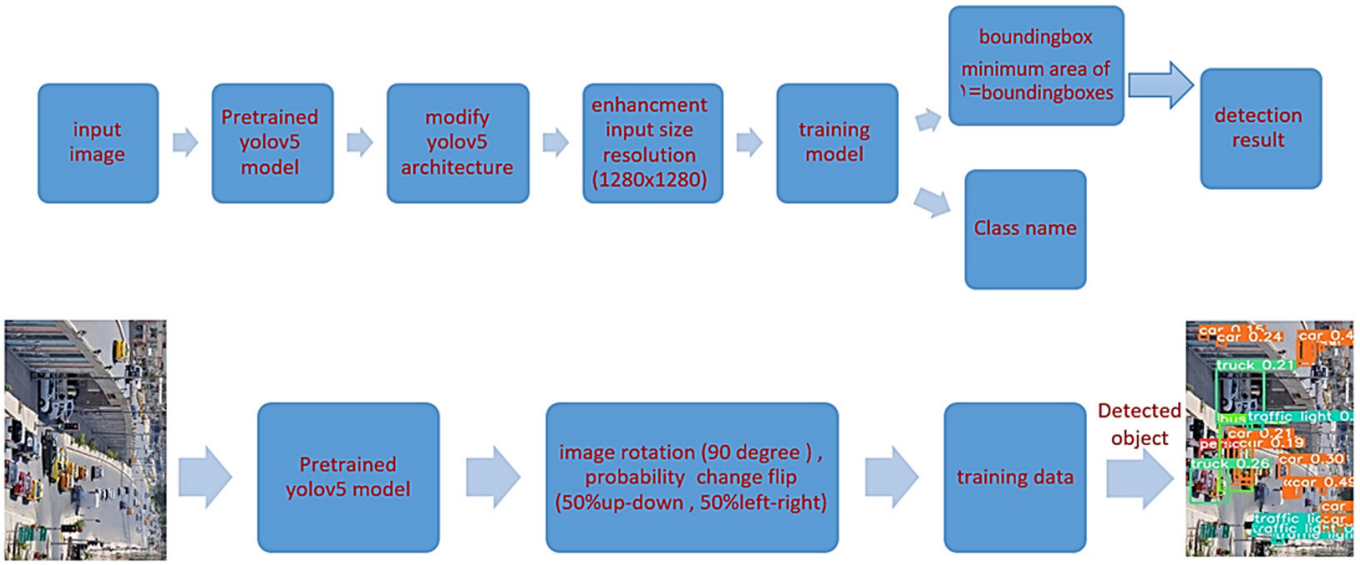

Fig. 2. Proposed algorithm steps

We conducted several experiments and implemented structural changes by introducing a P6 output layer to detect large objects and a P2 layer for very small outputs. Where the backbone extends to P6, the neck goes to P2, then back to P6, and does not stop at P3 and P5. We trained the model using high-resolution inputs (1280 × 1280) of traffic images, which are the inputs for the YOLOv5 network. The input image utilizes a backbone network to extract the feature maps (C2, C3, C4, C5, and C6). The network is sampling output features (C2, C3, C4, C5, and C6) from the neck to generate new feature maps (P2, P3, P4, P5, and P6) in order to identify the targets of various metrics. The head then utilizes the combined features to infer the bounding boxes and categories for the detected objects. Figure 3 indicates the stages of processing the network architecture.

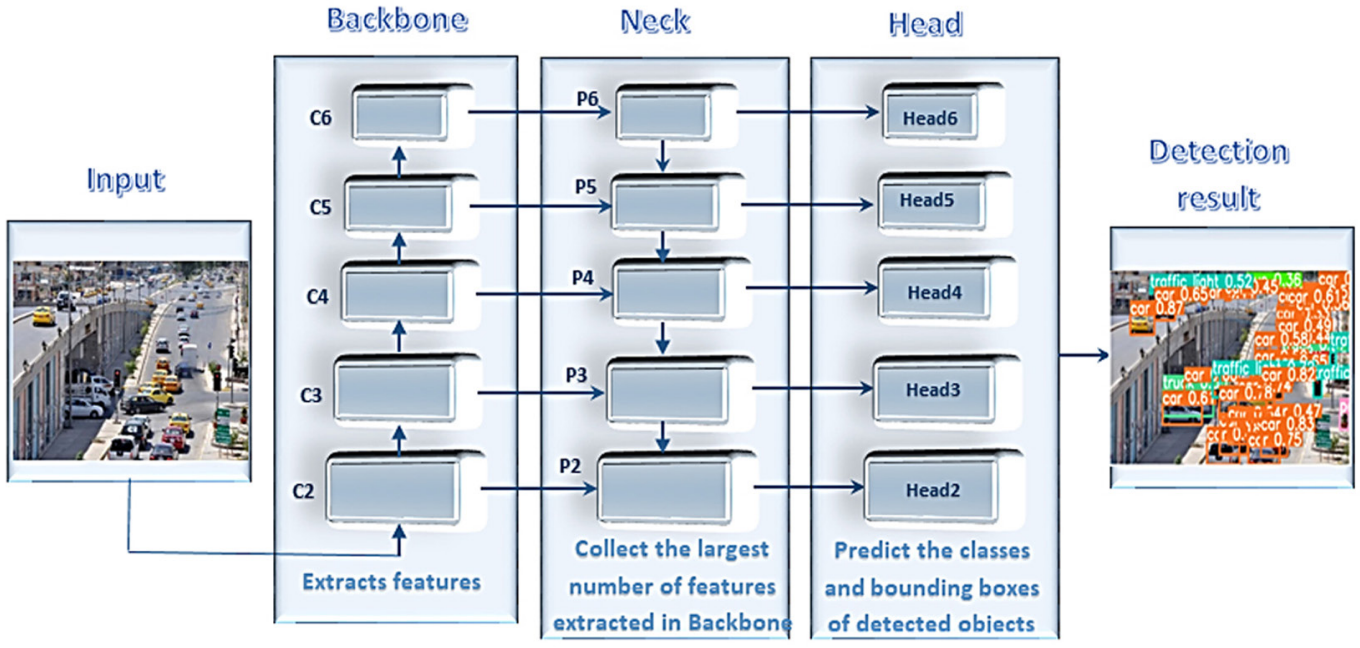

Fig. 3. The architecture of the experimental YOLOv5 model

Additional improvements include increasing the batch size and the number of training images processed in one forward and backward pass [26]. We divided the dataset into 64 batches to ensure faster training speed.

5 DISCUSSIONS

5.1 Dataset

The COCO dataset is a large dataset used to train the YOLOv5 model [27], which is trained on 1200 images. The MS COCO 2017 detection dataset contains 118,287 images used for training, 5,000 images used for verification, and 40,670 images used for testing [28]. This data includes 80 categories [29]. In this study, we utilized the data from the original YOLOv5 model with trained weights. The system has been trained on 1200 images to detect various objects in high-resolution images. This paper proposes a method for detecting objects in images and videos, including small, blurred, or rotated objects. We will train the object detection model using the GPU Free-Google Colab on the previously trained COCO dataset with high resolution.

5.2 Experimental results and evaluation criteria

In this paper, the performance evaluation of the original and improved YOLOv5 model was conducted using Colab with a free GPU to accelerate the training of the large dataset. Figure 4 shows the input image for the trained YOLOv5 network model.

Fig. 4. Shows the original image (model inputs)

Figure 5 shows detection results using the YOLOv5 model pre-trained on COCO data. The input size of the original training model is 640 × 640. It is capable of detecting cars that are large, close, and clear, as well as other large objects.

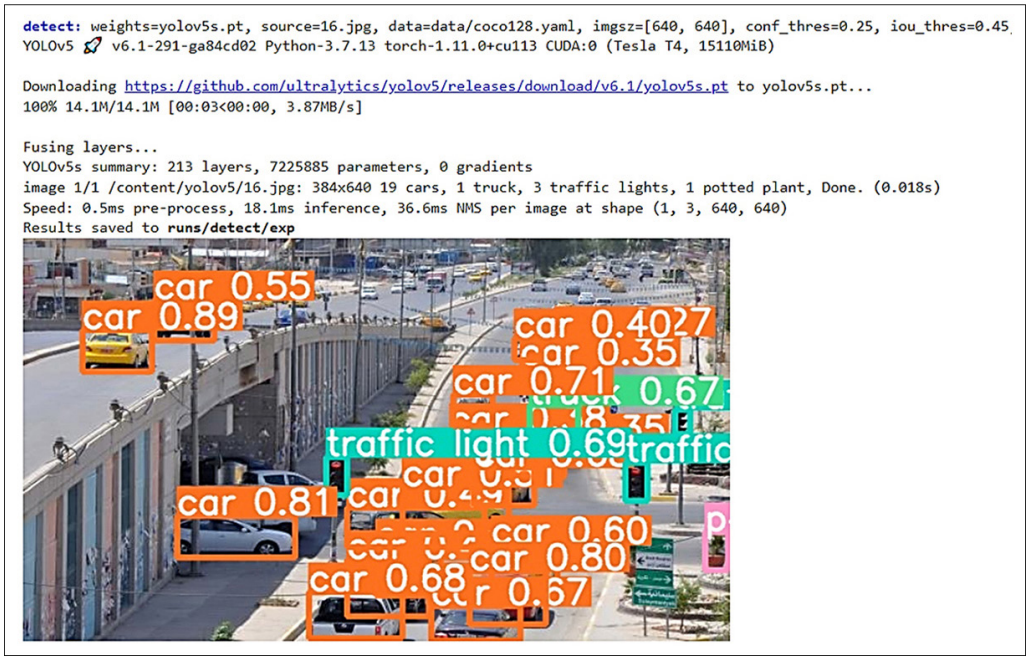

Fig. 5. Detected objects by YOLOv5

In Table 1, the detected object's dimensions, bounding box, and class labels are represented for each object bounding box.

Table 1. Evaluation of the accuracy of the YOLOv5 model

- Code + Text

	xmin	ymin	xmax	ymax	confidence	class	name
0	50.932568	78.618637	111.062660	115.036423	0.892599	2	car
1	521.450745	193.969330	540.221924	227.490921	0.827370	9	traffic light
2	133.199219	239.754852	234.653931	276.281647	0.806198	2	car
3	385.949371	288.170227	451.504608	333.868805	0.793118	2	car
4	371.635193	134.734482	406.135895	162.202988	0.704316	2	car
5	246.633148	306.056671	328.426941	346.498291	0.672582	2	car
6	390.422150	198.168793	442.224762	230.713013	0.672014	2	car
7	261.570038	189.434891	280.425812	222.507568	0.666159	9	traffic light
8	437.493347	143.862961	479.822388	190.147171	0.664993	7	truck
9	563.706421	147.634262	576.847961	168.984329	0.650696	9	traffic light
10	352.847748	317.781067	429.585114	347.720215	0.647752	2	car
11	589.706421	253.786774	610.329651	286.806519	0.609666	58	potted plant
12	405.408081	265.205566	462.156006	306.430847	0.581897	2	car
13	113.438416	53.421070	165.480713	87.966011	0.547789	2	car
14	328.500977	215.665115	419.435730	255.611969	0.477314	2	car
15	279.513763	232.392929	403.917480	266.249115	0.475805	2	car
16	279.811615	278.778503	350.463684	327.091370	0.474676	2	car
17	425.265869	85.542198	448.245239	104.515244	0.406606	2	car
18	304.673065	268.210175	360.989014	309.314911	0.399993	2	car
19	362.844086	189.076630	403.731964	217.286621	0.364943	2	car
20	369.500031	168.331451	408.512909	193.709396	0.354717	2	car
21	430.006958	109.938263	458.211487	131.799576	0.335308	2	car
22	419.650330	176.614120	468.288330	205.952911	0.326405	2	car
23	460.395752	84.063469	481.956421	100.475410	0.276908	2	car

The detection model relies on essential criteria, such as accuracy and detection rate. “Figure 6” shows the number of detected objects for each category using the original YOLOv5 model, which is 24 objects.

As for the proposed model, I modified the parameters (imgsz = 1280, batch_size = 64, epochs = 100) to adjust the network structure of the model and trained it using these modified parameters. Figure 7 shows the successful training of the proposed YOLOv5 model.

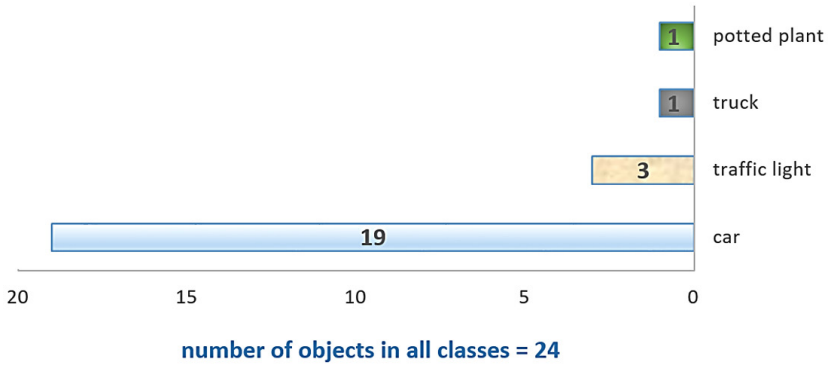

Fig. 6. Objects detected in all classes using YOLOv5 model

```
train: weights=yolov5s.pt, cfg=, data=coco128.yaml, hyp=hyp.scratch-high.yaml, epochs=100, batch_size=64, imgsz=1280, rect=False, resume=False, nosave=False, noval=False, noautoanchor=f
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v6.2-40-g0b8639a Python-3.7.13 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.1, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.3, cls_pw=1.0, obj=0.7, obj_pw=1.0,
Weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5 🚀 runs in Weights & Biases
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 🚀 in ClearML
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/

                 from  n    params  module                                  arguments
  0                -1  1      3520  models.common.Conv                      [3, 32, 6, 2, 2]
  1                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  2                -1  1     18816  models.common.C3                        [64, 64, 1]
  3                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  4                -1  2    115712  models.common.C3                        [128, 128, 2]
  5                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  6                -1  3    625152  models.common.C3                        [256, 256, 3]
  7                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  8                -1  1   1182720  models.common.C3                        [512, 512, 1]
  9                -1  1    656896  models.common.SPPF                      [512, 512, 5]
 10                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12          [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1    361984  models.common.C3                        [512, 256, 1, False]
 14                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16          [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     90880  models.common.C3                        [256, 128, 1, False]
 18                -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  1    296448  models.common.C3                        [256, 256, 1, False]
 21                -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  1   1182720  models.common.C3                        [512, 512, 1, False]
 24      [17, 20, 23]  1    229245  models.yolo.Detect                      [80, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]
Model summary: 270 layers, 7235389 parameters, 7235389 gradients, 16.6 GFLOPs

Transferred 349/349 items from yolov5s.pt
AMP: checks passed ✅
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
train: Scanning '/content/datasets/coco128/labels/train2017.cache' images and labels... 128 found, 0 missing, 2 empty, 0 corrupt: 100% 128/128 [00:00<?, ?it/s]
val: Scanning '/content/datasets/coco128/labels/train2017.cache' images and labels... 128 found, 0 missing, 2 empty, 0 corrupt: 100% 128/128 [00:00<?, ?it/s]

AutoAnchor: 3.99 anchors/target, 0.997 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Plotting labels to runs/train/exp5/labels.jpg...
Image sizes 1280 train, 1280 val
Using 2 dataloader workers
Logging results to runs/train/exp5
Starting training for 100 epochs...
```

Fig. 7. Training the proposed model

In addition to its high input resolutions, it can better identify cars and other objects due to its small size and long-range capabilities. Moreover, increasing the batch size resulted in an inference time of 24.1 ms and 2.1 ms NMS for each image in Figures 1 and 3. Figure 8 shows the detection results of the proposed model.

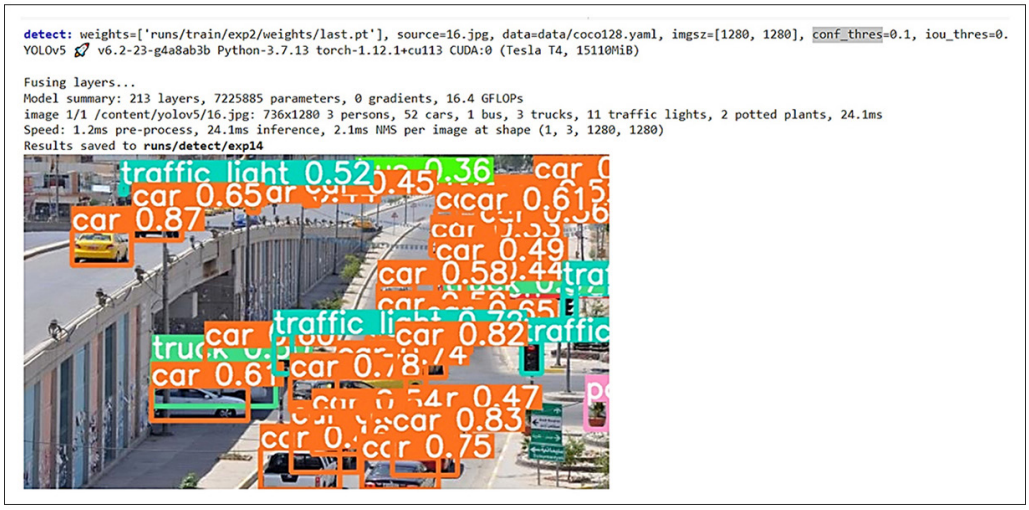

Fig. 8. Detection results using the proposed model

The area within the bounding box is analyzed, the dimensions of the detected object are determined, and a new bounding box is defined with minimal area (see Table 2).

Table 2. Performance evaluation of the accuracy of the experiments proposed

+ Code + Text

	xmin	ymin	xmax	ymax	confidence	class	name
0	48.929317	79.058563	111.826241	115.201927	0.866174	2	car
1	388.234985	198.945648	440.937500	232.369858	0.822945	2	car
2	384.522766	288.074615	452.390472	334.150055	0.817925	2	car
3	520.422607	195.442459	540.913269	227.684235	0.808242	9	traffic light
4	277.920929	232.196182	399.673676	267.525696	0.767627	2	car
...	...	...	...	...	...	...	...
68	24.139078	62.319244	29.774906	71.871925	0.107422	0	person
69	263.000549	63.134624	266.945770	71.596466	0.106460	0	person
70	463.647980	52.717785	478.446106	64.631386	0.103861	2	car
71	602.007141	97.269531	608.381165	107.266800	0.103808	9	traffic light
72	358.992737	189.061600	401.753357	221.004425	0.102144	7	truck

73 rows × 7 columns

Figure 9 shows the number of detected objects for each class using the proposed model (73) objects.

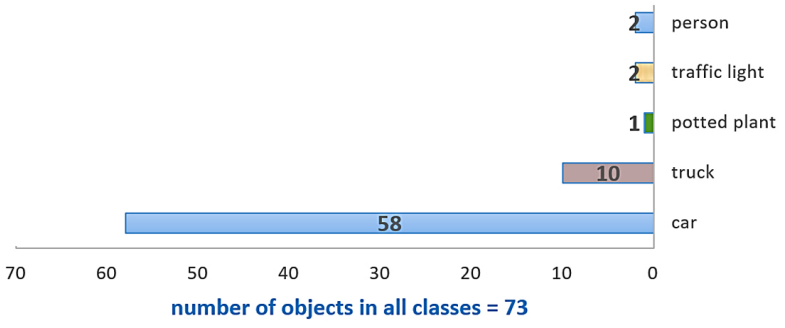

Fig. 9. Objects detected in the proposed model

The original YOLOv5 result is compared with the improved YOLOv5. It can be seen from the presented results that the proposed model can extract image features to detect low-resolution objects. It can achieve shorter training times without ignoring performance.

Figure 10 shows that as we increase the degree parameters from 0 to 270, the number of detected objects also increases.

Figure 10 (A) shows that when we rotate the image at a certain angle and use the original YOLOv5 model to detect objects, it results in a failure of detection (only one object detected), as shown in Figure 10 (B).

In Figure 10 (C), we used the proposed model to detect objects without altering their rotation or flipping. It successfully detected 15 objects.

Figure 10 (D) used the proposed model, adjusted the rotation degrees of the image parameters by 90°, and increased the probability of flipping the image by 50% (either right or left, up or down). As a result, they observed an increase in the number of detected objects, with a total of 32 objects being detected.

Fig. 10. The detections of an object rotated 90° (a) rotated input image (b) by the original model (c) by the proposed model without change image degree model (d) by the proposed model with change image degree

In addition, we also achieved favourable outcomes for high-resolution training of the proposed model using YOLOv5 to assess its real-time video processing capabilities. The traffic monitoring video was tested and trained for 300 epochs (Figures 11 and 12). We found that the new model detected more objects, such as traffic lights,

people, cars, and other objects. Additionally, it was able to detect objects that were far away. This model achieved better and higher performance compared to the original model. This model can be used in a real-time traffic monitoring application.

Fig. 11 Results of object detection in the video using YOLOv5

Fig. 12. The results of detecting objects in the video using the proposed model

To evaluate the performance of the YOLOv5 network. There are several criteria for evaluating performance in object detection, including average precision and loss function:

Mean average precision (MAP) is used to evaluate objects and measure their accuracy. To find the MAP, we must understand that the input image contains objects that are distributed across different classes. We calculate the MAP by finding the AP for each class in the image. It then calculates the mean average precision for all categories. The larger the size of the object, the better the precision of its detection [30].

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

where: TP: the predictive value of the model is truly positive, FP: the predictive value of the model is a false positive.

$$Map = \frac{1}{n}\sum AP_n \tag{2}$$

n: number of classes.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

FN: the predictive value of the model is false Negative.

The loss function estimates the performance of the model. The value is small when the prediction of the detected object is close to reality [31].

$$\text{Loss function} = \frac{1}{m}\sum (y - a)^2 \tag{4}$$

where: m: the number of training inputs in the network, a: the expected value, y: the actual value.

These parameters were calculated to detect the object in a 640 × 640 resolution (Figure 13). Illustrations show the model's losses and precision after the training process of YOLOv5.

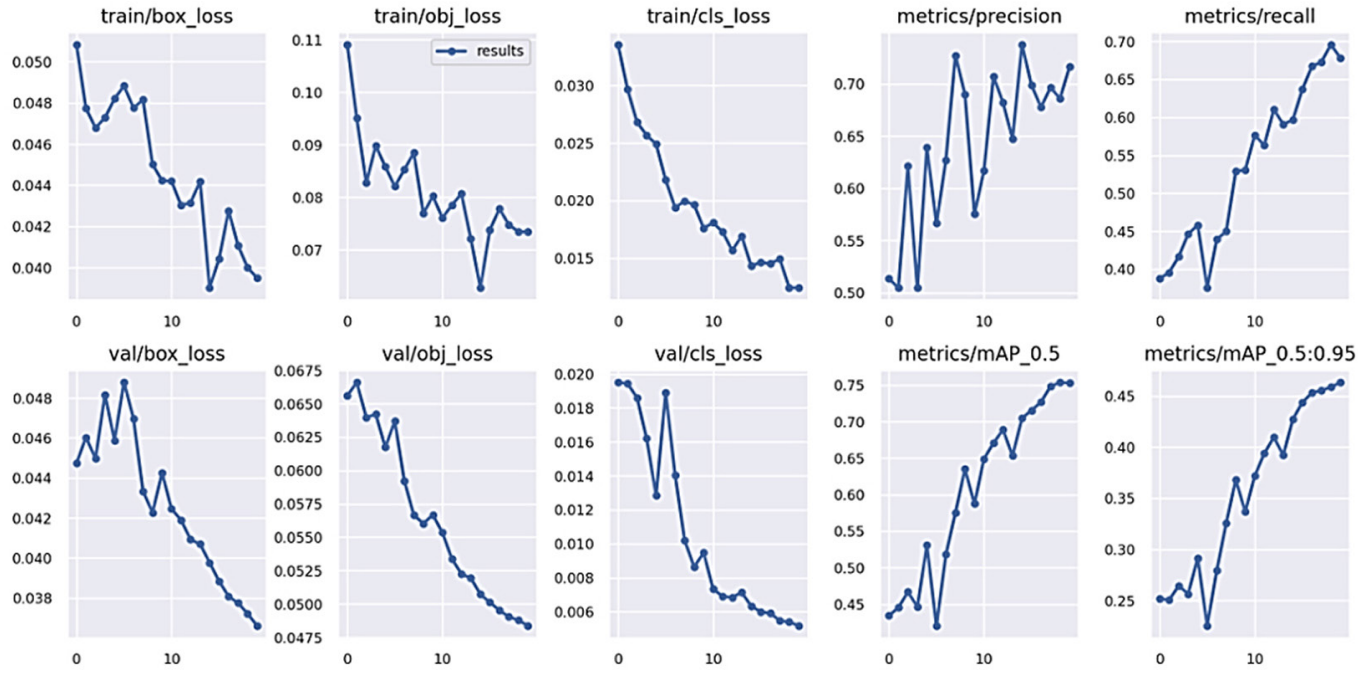

Fig. 13. Results of the map, loss, precision, and recall of the YOLOv5 model

The evaluation criteria for the studied model were calculated with an accuracy of 1280 × 1280, as shown in Figure 14.

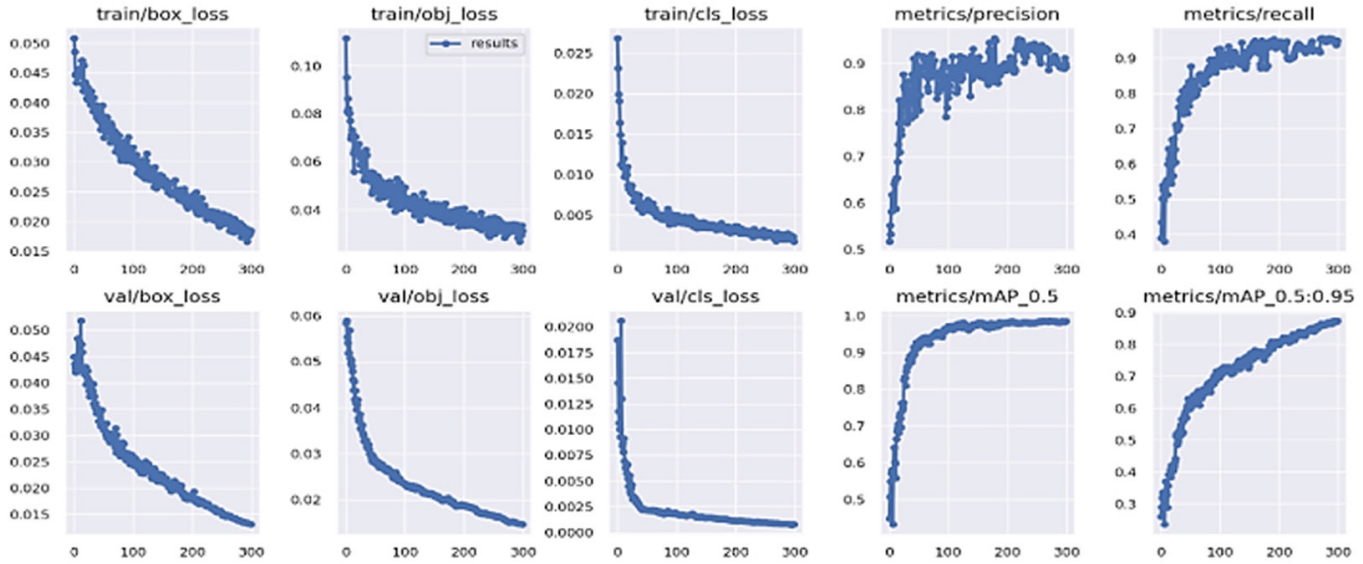

Fig. 14. The map, loss, precision, and recall of the proposed model

After comparing the results, we found that in the proposed method, the training loss is 0.025, and the MAP is 0.90. While the original YOLOv5 model had a training loss of 0.035 and a MAP of 0.75, we have improved the performance criteria and achieved good results in detecting the largest number of objects.

6 CONCLUSION

This paper presents a modified version of YOLOv5 that demonstrates great performance in detecting distant, small, or rotated objects in photos and videos. The proposed model outperformed the original model in object detection. I made modifications to certain structural elements of YOLOv5, which resulted in changes to some important parameters. Specifically, I adjusted the high-resolution parameters of the model inputs to be 1280 × 1280. It also detects objects in multiple directions when the image is rotated from 0 to 270 degrees or when it is flipped. This model has been trained using the Google Colab interface because it offers a free GPU, which results in faster processing of large data. The model proposed in this paper has a training loss of 0.07 and an inference speed of 1.3 ms per image. We also achieved good results with high-resolution training; the number of detected objects increased from 23 in the original model to 73. When the image parameters are rotated by 90°, and there is a 50% probability of the image flipping, the number of detected objects increases to 32. Through the conducted experiments, we have verified that YOLOv5 possesses numerous features, and we have utilized several of them to enhance the accuracy of the detection model. It can utilize additional features for future studies and advancements, such as addressing various image lighting issues, such as brightness and enhancing the neural network's capability in object detection. As well as future work and development of models specifically designed for mobile devices. In future work, we aim to optimize the structure of the detection network to improve its performance.

7 REFERENCES

[1] H. Gong, T. Mu, Q. Li, H. Dai, C. Li, Z. He, and B. Wang, "Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images," *Remote Sensing*, vol. 14, no. 12, p. 2861, 2022. https://doi.org/10.3390/rs14122861

[2] D. Thuan, "Evolution of Yolo algorithm and Yolov5: The state-of-the-art object detection algorithm," 2021.

[3] V. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning-based object detection techniques: A review," *Computer Science Review*, vol. 38, p. 100301, 2020. https://doi.org/10.1016/j.cosrev.2020.100301

[4] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets, and applications," *Multimedia Tools and Applications*, vol. 82, pp. 9243–9275, 2022. https://doi.org/10.1007/s11042-022-13644-y

[5] A. Sharma, "Achieving optimal speed and accuracy in object detection (YOLOv4)," 2022. https://pyimagesearch.com/2022/05/16/achieving-optimal-speed-and-accuracy-in-object-detection-yolov4/

[6] M. Kasper-Eulars, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, and P/E. Kummervold, "Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5," *Algorithms*, vol. 14, no. 4, p. 114, 2021. https://doi.org/10.3390/a14040114

[7] J. Zhao, X. Zhang, J. Yan, X. Qiu, X. Yao, Y. Tian, and W. Cao, "A wheat spike detection method in UAV images based on improved YOLOv5," *Remote Sensing*, vol. 13, no. 16, p. 3095, 2021. https://doi.org/10.3390/rs13163095

[8] M. Lewandowski, B. Płaczek, M. Bernas, and P. Szymała, "Road traffic monitoring system based on mobile devices and bluetooth low energy beacons," *Wireless Communications and Mobile Computing*, vol. 2018, no. 3251598, pp. 1–12, 2018. https://doi.org/10.1155/2018/3251598

[9] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on YOLOv5," in *IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, pp. 6–11, 2021. https://doi.org/10.1109/ICPECA51329.2021.9362711

[10] Q. Song, S. Li, Q. Bai, J. Yang, X. Zhang, Z. Li, and Z. Duan, "Object detection method for grasping robot based on improved YOLOv5," *Micromachines*, vol. 12, no. 11, p. 1273, 2021. https://doi.org/10.3390/mi12111273

[11] M. Chiriboga, C. M. Green, D. A. Hastman, D. Mathur, Q. Wei, S. A. Díaz, and R. Veneziano, "Rapid DNA origami nanostructure detection and classification using the YOLOv5 deep convolutional neural network," *Scientific Reports*, vol. 12, no. 1, pp. 1–13, 2022. https://doi.org/10.1038/s41598-022-07759-3

[12] J. Bhuvana, T. T. Mirnalinee, B. Bharathi, S. Jayasooryan, and N. N. Lokesh, "YOLOv5 for stroke detection and classification in table tennis," 2021.

[13] L. Zhu, X. Geng, Z. Li, and C. Liu, "Improving YOLOV5 with attention mechanism for detecting boulders from planetary images," *Remote Sensing*, vol. 13, no. 18, p. 3776, 2021. https://doi.org/10.3390/rs13183776

[14] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," *ArXiv* preprint, 2021.

[15] C. H. Singh and K. Jain, "An enhanced YOLOv5 based on color harmony algorithm for object detection in unmanned aerial vehicle captured images," *Research Square*, preprint. https://doi.org/10.21203/rs.3.rs-1876969/v1

[16] I. Katsamenis, E. E. Karolou, A. Davradou, E. Protopapadakis, A. Doulamis, N. Doulamis, and D. Kalogeras, "TraCon: A novel dataset for real-time traffic cones detection using deep learning," *ArXiv* preprint, 2022. https://doi.org/10.1007/978-3-031-17601-2_37

[17] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020. https://doi.org/10.1016/j.icte.2020.04.010

[18] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, p. 5817, 2022. https://doi.org/10.3390/s22155817

[19] M. S. Fuad, C. Anam, K. Adi, M. A. Khalif, and G. Dougherty, "Evaluation of the number of epochs in an automated COVID-19 detection system from x-ray images using deep transfer learning," *International Journal of Advances in Engineering & Technology*, vol. 13, no. 6, pp.134–140, 2020.

[20] J. Ma, Y. Ushiku, and M. Sagara, "The effect of improving annotation quality on object detection datasets: A preliminary study," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4850–4859, 2022. https://doi.org/10.1109/CVPRW56347.2022.00532

[21] D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks," *Human-Centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–22, 2020. https://doi.org/10.1186/s13673-020-00219-9

[22] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," *ArXiv* preprint, 2019. https://doi.org/10.5121/csit.2019.91713

[23] M. L. Francies, M. M. Ata, and M. A. Mohamed, "A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 1, 2022. https://doi.org/10.1002/cpe.6517

[24] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on YOLO-v5 with improved maritime dataset," *Journal of Marine Science and Engineering*, vol. 10, no. 3, p. 377, 2022. https://doi.org/10.3390/jmse10030377

[25] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing*, vol. 97, p. 103910, 2020. https://doi.org/10.1016/j.imavis.2020.103910

[26] A. Devarakonda, M. Naumov, and M. Garland, "Adabatch: Adaptive batch sizes for training deep neural networks," *ArXiv*, preprint, 2017.

[27] A. Kuznetsova, T. Maleva, and V. Soloviev, "Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images," in *International Symposium on Neural Networks*, Springer, Cham., pp. 233–243, 2020. https://doi.org/10.1007/978-3-030-64221-1_20

[28] S. Chen and B. Chen, "Research on object detection algorithm based on improved Yolov5," in *Artificial Intelligence in China*, Springer, Singapore, pp. 290–297, 2022. https://doi.org/10.1007/978-981-16-9423-3_37

[29] Z. Qu, L. Y. Gao, S. Y. Wang, H. N. Yin, and T. M. Yi, "An improved YOLOv5 method for large object detection with multi-scale feature cross-layer fusion network," *Image and Vision Computing*, vol. 125, p. 104518, 2022. https://doi.org/10.1016/j.imavis.2022.104518

[30] N. I. Nife and M. Chtourou, "Video objects detection using deep convolutional neural networks," in *6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, 2022. https://doi.org/10.1109/ATSIP55956.2022.9805931

[31] H. Hakim and A. Fadhil, "Survey: Convolution neural networks in object detection," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1804, no. 1, p. 012095, 2021. https://doi.org/10.1088/1742-6596/1804/1/012095

8 AUTHORS

Nadia Ibrahim Nife is a PhD student at the Faculty of Engineering, University of Sfax (E-mail: nadia.ibra@uokirkuk.edu.iq).

Mohammed Chtourou is a Professor in the Department of Electrical Engineering of the National School of Engineers of Sfax, Tunisia (E-mail: Mohamed.chtourou@enis.rnu.tn).