Journal of Robotics and Control (JRC)
Volume 6, Issue 2, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i2.25856

Enhancing Collision Avoidance in Mobile Robots Using YOLOv5: A Lightweight Approach for Unstructured Environments

Saleel H. Abood [1*], Hussein. M. H. Al-Khafaji [2], Mohanned M. H. Al-Khafaji [3]
[1, 2] Mechanical Engineering Department, University of Technology-Iraq, Ministry of Higher Education & Scientific Research, Baghdad, Iraq
[3] Production Engineering and Metallurgy Department, University of Technology-Iraq, Ministry of Higher Education & Scientific Research, Baghdad, Iraq
Email: [1] Saleelhussein91@gmail.com, [2] hussein.m.hussein@uotechnology.edu.iq, [3] mohanned.m.hussein@uotechnology.edu.iq
*Corresponding Author

Abstract—**Mobile robots play a crucial role in Industry 4.0, particularly in dynamic and unstructured environments where moving obstacles present significant challenges. This study applies the YOLOv5 object detection algorithm to enhance robotic perception and obstacle avoidance. The primary objective is to improve the accuracy and speed of object detection in real-time scenarios, ensuring safer and more efficient navigation for robots. The research contribution lies in developing a lightweight YOLOv5 model optimised for robotic applications, capable of detecting objects with high accuracy. The model was trained on a diverse dataset of 10,700 images, including static and dynamic objects such as chairs, fans, fire extinguishers, and humans, captured under various conditions and orientations. The dataset was divided into training (70%), validation (15%), and testing (15%) subsets. The proposed model achieved a mean average precision (mAP) of 0.73 at a confidence threshold of 0.374, demonstrating superior performance compared to the YOLOv4 model in terms of accuracy and processing speed. Notably, the model excelled in detecting static objects such as chairs, achieving a perfect recognition rate of 1.00, while encountering challenges with moving objects such as humans due to motion blur and rapid changes in body posture. These findings highlight the model's potential for real-time applications in industrial and unstructured environments. In conclusion, this study demonstrates that the enhanced YOLOv5 model significantly improves object detection and collision avoidance capabilities in robotic systems.**

Keywords—Convolution Neural Network; YOLOv5 Detector; Object Detection; Mobile Robot; Collision Avoidance; Unstructured Environments; Real-Time Performance.

I. Introduction

Robotics technology has made significant advancements in recent years, enabling robots to dynamically perceive and interact with their environments [1][2]. This progress has led to increased efficiency in industrial processes and improved safety. Robots have become essential in various industrial applications [3], boosting productivity [4], reducing errors, enhancing process accuracy, and protecting workers from hazards [5][6]. However, one of the primary challenge's robots face is avoiding collisions in complex work environments, which can negatively affect both equipment safety and productivity [7][8][9]. To address this, object detection techniques via computer vision have emerged as a crucial solution, allowing robots to recognize and accurately locate objects. Among these techniques, the YOLO (You Only Look Once) algorithm has gained prominence due to its speed and efficiency in real-time image processing. While earlier versions of YOLO, such as YOLOv2 and YOLOv3, have improved robot performance, YOLOv5 has emerged as the most effective algorithm in industrial settings due to its enhanced accuracy and efficiency, particularly in complex environments. YOLOv5's faster performance and high image processing speeds make it ideal for live, real-time applications [10][11][12]. However, despite its advantages, YOLOv5 faces certain limitations when applied in real-world industrial environments, these include challenges in detecting small objects or objects in highly cluttered environments, especially under varying lighting conditions [13]. Additionally, YOLOv5's accuracy can decrease when used with resource-constrained hardware, leading to potential delays in processing time.

The focus of this research is to improve robot collision avoidance by applying YOLOv5 to computer vision systems. The study aims to explore how YOLOv5 can enhance robots' ability to detect objects and avoid collisions in industrial environments, while also analyzing potential limitations under various conditions.

Previous studies have demonstrated the effectiveness of YOLO in different robotic applications. In [14] YOLO for object detection was used in four-wheeled robots in Gazebo, which were connected to LIDAR and Kinect cameras. These robots cooperated to avoid revisiting paths and used YOLO to detect surrounding objects. Similarly, in [15] YOLO with the Kinect sensor was employed to identify obstacles for mobile robots, showing reliable obstacle detection. In [16] developed a system for detecting olive fruits in real time using YOLOv5, which showed high accuracy and speed for

CC BY SA

agricultural robots. Additionally, in [17] proposed an optimized YOLOv5 model to address the challenges of long computational times and low detection rates in resource-constrained robots. The optimization improved speed and accuracy without compromising performance. In [18] proposed an improved YOLOv5 for industrial grasping robots, achieving remarkable performance improvements in accuracy, recall, mAP, and F1 score while also reducing model size. In [19] introduced YOLOv5_Tel, a modified version of YOLOv5 for teleoperated robots, which improved object localization accuracy and efficiency. The model utilized advanced techniques like BiFPN, CA modules, and SIOU loss functions, leading to better performance, including faster convergence and reduced model size. In [20] The YOLOv5 series models were applied to a warehouse environment, and the models were trained and optimized on a warehouse object dataset. This achieved accurate warehouse object detection. The mAP@50 of the YOLOv5 series models reached over 99%. Among them, the mAP@50-90 of the YOLOv5l and YOLOv5x models reached approximately 92%. Experimental results show that YOLOv5m achieved satisfactory warehouse object detection and has certain practical value in the field of warehouse object detection.

This research aims to optimize robotic movement by enabling advanced obstacle avoidance capabilities, ensuring more efficient and seamless navigation in dynamic environments.

II. METHODOLOGY OF YOLOV5

The detection of objects involves the generation of features from input images. Subsequently, boxes are drawn around objects and their categories are predicted by entering these features through a prediction system. YOLO v5 is designed based on the procedures. As a computer vision model for object detection that is built upon the YOLO series [21], YOLO v5 includes several architectural improvements for simultaneously enhancing speed and accuracy [22][23].

A. *Architecture*

Input images used in YOLO v5 exhibit varying sizes, with typical resolutions of 640×640 pixels or 416×416 pixels. Before they are fed into the network, input images are resized and normalised [24][25].

The backbone of YOLO v5 is designed for efficiently extracting features [26]. It is composed of two parts. The focus layer, which divides an input image into several parts and then gathers them back together, facilitates the capture of fine details by focusing on various areas of the image. The other part, CSPDarknet53, includes partial connections across stages. It divides and then merges feature maps, improving gradient flow and reducing computational cost [27][28]. The architecture of YOLO v5 has multiple stages; each stage comprises several convolutional layers and residual connections [29].

The neck refers to a series of layers that is responsible for pooling the backbone layers. The spatial pyramid pooling (SPP) layer helps in improving the object detection ability of the model at different scales [30]. Pooling operations are applied to multiple scales and then the aggregated features are concatenated to capture the context at different levels. The purpose of the path aggregation network (PANet) is to enhance information flow [31]. Augmentation of the path from the bottom up is performed to enhance the hierarchy of features with accurate localisation signals in the lower layers, shortening the information path between the lower layers and the higher features. Adaptive feature pooling is also implemented, linking the feature network and all the feature levels, such that useful information spreads directly at each feature level [32].

Lastly, the head of YOLO v5 is tasked to predict boundary boxes, class probabilities and object scores. Fine-tuned anchor boxes are used for the dataset, optimising the model to make accurate predictions [33]. The model's architecture is illustrated in Fig. 1.

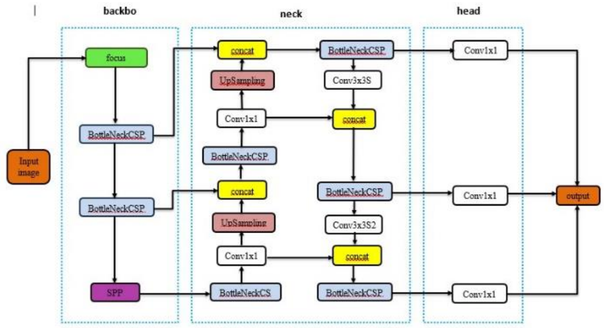

Fig. 1. Architecture diagram of YOLOv5 object detection model

1) *CNNs*

CNN refers to a special type of feedforward neural network inspired by the biological processes occurring in the brain of living organisms, particularly the optic lobe [34]. It is used as a solution to many computer vision problems in artificial intelligence, including image and video processing. CNN comprises a group of connected neurons [35]. These neurons are organised within a group of layers, as shown in Fig. 2 [36].

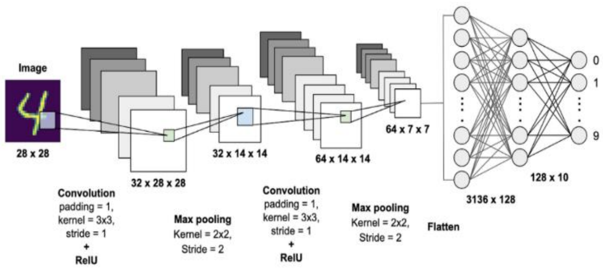

Fig. 2. Convolutional Neural Network (CNN) architecture for digit recognition

The input layer is responsible for receiving input (typically images) and then prepares it for processing. The convolutional layer is composed of filters that pass over an image to extract a set of features from the image. The filter must pass across the entire image (i.e. part after part) because different features will be extracted from each part of the image by different filters. The output of the convolutional layer is a feature map [37]. The features of such map are obtained by multiplying the filter by the image. Another important building block of CNNs is the pooling layer, which is a 2D filter. This layer performs an important function in

CNNs, because it decreases the spatial size of feature maps [38]. The pooling layer has several types, including maximum, minimum and average pooling. One of the popular techniques used in CNNs is maximum pooling, which typically uses a 2×2 filter with a stride of two. Hence, an output for a maximum pooling layer provides important features for a previous feature map [39][40]. The layer that is normally placed at the end of the network is the fully connected layer, which consists of neurons and exhibits an activation function for making a classification decision [41]. Lastly, the last layer in the network is the output layer, which is responsible for providing the consequence.

2) Anchor boxes

Anchor Boxes are a set of predefined bounding boxes with specific height and weight. These boxes are used in deep learning to define potential object regions within an image [42][43]. They rely on a set of pre-defined boxes to predict the locations and dimensions of objects within the image. The image is divided into a grid with cells of varying resolutions to accommodate different object sizes, and several initial boxes are predicted for each cell to match the shape and size of the object that may appear in that cell [44][45]. In YOLOv5, three Anchor Boxes are used for each grid cell, meaning that for each cell, three pre-defined boxes are predicted to match the shape and size of the object that might appear in that cell. The image is divided into a grid with varying resolutions to match different object sizes. Anchor Boxes are assigned according to the changing grid sizes: small grid cells (52×52) are dedicated to small objects, medium grid cells (26x26) are for medium objects, and large grid cells (13×13) are used for large objects. Each grid cell contains 3 different Anchor Boxes. After training, each Anchor Box has a probability value representing the presence of an object in that cell. These cells contribute to accurately predicting the locations and sizes of objects [46][47][48].

Each Anchor Box consists of 85 channels which include the following information:

- x and y: The coordinates of the object's center relative to the bounding box.
- w and h: The width and height of the object.
- c: The probability of the object being inside the Anchor Box, indicating the likelihood that this box contains an object.

During training, the neural network learns to reduce the gap between the actual dimensions of objects and the predicted dimensions from the Anchor Boxes. After training, the model is capable of more accurately assigning Anchor Boxes for detecting different types and shapes of objects in images [49][50][51].

3) Evaluation metrics

Evaluation metrics are used to calculate the accuracy and efficiency of the object detection models [52], for the simple reason that they define how such a model can accurately pinpoint objects within an image [53]. Moreover, those evaluations offer information on how the corresponding model regulates furthermore false positive and false negative results [54][55]. Some of the most significant evaluation criteria are existing. Intersection over union shown in Fig. 3.

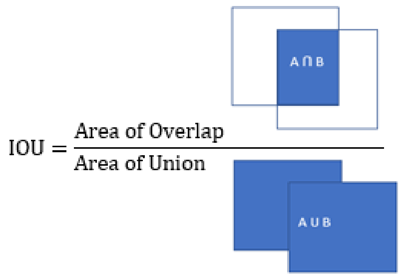

Fig. 3. Illustration of Intersection over Union (IOU) in object detection

From the above, we see that IoU calculates the degree of match that is between the predicted bounding box and the actual bounding box. It has the onus of measuring the extent of accuracy in relation to object localisation [56][57]. Underneath the precision–recalled curve is metreage by AP, which provides a single summary of the accuracy and recall of any model. AP is further defined by mean AP (mAP), which is mean of AP across all the classes as it provides an overall scenario of how a model performs across all classes. Precision measures how many out of all predicted positive instances are actually true and thus assesses the potential of a model to classify negative instances as being negative. The true positive fraction is the complement of fall-out which shows the recall rate, which is the measure of the ability of a model to find all instances of a certain class [58][59]. Finally, the F1 score is obtained by averaging precision and recall into a harmonic mean that also takes into account both false positives and false negatives [60]. Combined, these metrics allow improving the overall understanding of object detection while enhancing the performance of the corresponding models

B. Data preparation and training the YOLOv5 model

The study was conducted using a safety laboratory setup as the defined manufacturing environment, with static and dynamic obstacles placed for the robot. These objects included a fire extinguisher, reflection of warning signs, a fan, a chair, and a human. The dataset for object detection in this study consists of images for various objects: 2000 images of chairs, 1900 images of fans, fire extinguishers, and cones, and 1500 images each of warning signs and humans. Depth images were captured using an HD camera mounted on a mobile robot, equipped with two DC motors (205RPM), SLAM Lidar, a professional robot expansion board, Raspberry Pi 4B, and a 7-inch screen. Images were taken at different locations and orientations within the manufacturing workspace (Fig. 4a).

To prepare the dataset, a labeling technique was employed to mark the images for training (Fig. 4b). The dataset consists of RGB images formatted as a 3D matrix, where each value corresponds to the pixel intensity. The images were pre-processed and normalized to ensure efficient model training. This normalization process adjusts pixel values to a range between 0 and 1 by dividing the original values by 255, which helps stabilize the model and reduce noise. The images were resized to a standard size of 640×640 to maintain uniformity.

The dataset was split into three subsets: training (70%), validation (15%), and testing (15%). The distribution of images in the subsets is shown in Table I.

TABLE I. THE PERCENTAGE OF TRAINING, VALIDATION, AND TESTING IMAGES

Object	Total image	Training 70%	Validation 15%	Testing 15%
Chair	2000	1400	300	300
fan	1900	1330	285	285
fire extinguisher	1900	1330	285	285
cone	1900	1330	285	285
warning	1500	1050	225	225
human	1500	1050	225	225

Step 1: Loading and Splitting the Dataset

The first step is to load and divide the dataset into training, validation, and testing categories. The training set is used for model training, the validation set for hyperparameter tuning, and the test set to assess the model's performance on unseen data.

Step 2: Validating the Input Data

Before training, the dataset is validated to ensure the quality of images, bounding boxes, and labels. This step checks for invalid image formats, corrupted files, and verifies that bounding box values are positive and within the image boundaries.

Step 3: Determining Anchor Boxes

YOLOv5 uses anchor boxes—predefined bounding boxes with specific dimensions—to predict potential object locations within an image. The number and size of anchor boxes are calculated during training using the K-means clustering algorithm, though customization is possible. These anchor boxes correspond to different object sizes, and they play a crucial role in detecting the presence and positioning of objects in images.

Step 4: Applying Augmentations to the Training Data

To enhance model performance, augmentations are applied to the training data only. These augmentations include random adjustments in color, random horizontal flipping, and random scaling (up to 10%). These transformations help increase the model's ability to generalize across different object variations. The validation and test sets remain unchanged. Additionally, various training options, as outlined in Table II, are used to define the training process and optimize model performance.

TABLE II. TRAINING OPTIONS

Parameter	Value
Algorithm	Adam
GradientDecayFactor	0.9
SquaredGradientDecayFactor	0.999
InitialLearnRate	0.001
MiniBatchSize	4
MaxEpochs	70
Shuffle	“every epoch”
VerboseFrequency	20
ValidationFrequency	1000
CheckpointPath	“checkpoint.path”

Step 5: Normalizing the Input Images

To further prepare the dataset, all input images undergo normalization. This process scales pixel values to a range of 0 to 1, ensuring that inputs are within an appropriate range for model training. For an image of size W×H×3, the normalization formula is:

$$X_{norm} = \frac{x}{255} \tag{1}$$

where X is the original pixel value matrix and X_norm is the normalized matrix. This normalization helps improve training by reducing potential noise and ensuring stable learning.

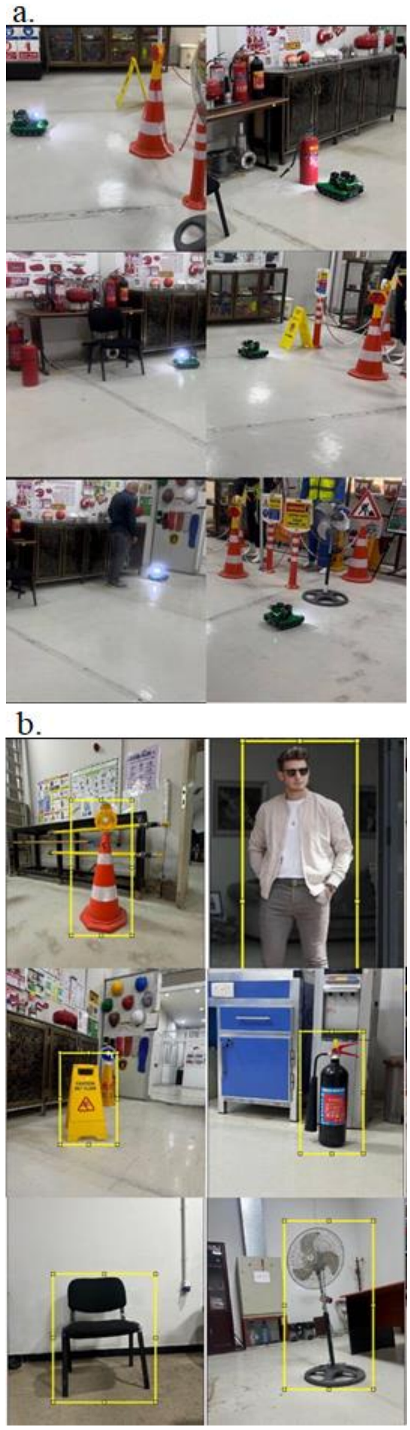

Fig. 4. Data Preparation, a) Data Collection, b) Data Labelling

Step 6: Training the Model

The model training in YOLOv5 utilizes full, un-mined, or cropped images to improve performance through multi-scale training. This technique allows the model to identify objects of various sizes by scaling images during the process. Several augmentation methods, such as size changes, angle adjustments, horizontal flipping, and color modifications, are employed to reduce overfitting. Additionally, batch normalization is used to normalize and enhance the speed of the training process.

YOLOv5 is trained on images sized 416×416×3 (or 640×640 in some cases), which are divided into a grid with three main output paths representing different object sizes:

1. 52×52×255: Small grid cells for small objects.
2. 26×26×255: Medium grid cells for medium objects.
3. 13×13×255: Large grid cells for large objects.

Each grid cell is responsible for predicting three Anchor Boxes, corresponding to different object sizes. Each cell has 85 channels, including 4 values for the object's **x, y** (center coordinates), **w, h** (width and height), 1 value for **C** (the object's presence probability), and 80 classification values for the 80 detectable objects. The model predicts Anchor Boxes based on the presence of an object in the cell. If **C = 1**, it indicates that the box contains an object, and the values for **x, y, w, h**, and classification are provided. If **C = 0**, the box does not contain an object, and its values are ignored during training.

The training process uses backpropagation to minimize the loss function, which measures the discrepancy between the predicted and actual locations and classifications of objects. Anchor Boxes with an Intersection over Union (IoU) greater than 0.3 are selected for positive training signals. If no Anchor Box meets this IoU threshold, the one with the highest IoU is chosen. Positive Anchor Boxes are assigned a confidence value of **1**, and the relevant values for **x, y, w, h**, and classification are encoded using **1-hot encoding**. This procedure enables YOLOv5 to learn the relationships between object classes and their features (such as texture, color, and shape), and to optimize its weights accordingly. By the end of training, YOLOv5 efficiently identifies and classifies objects in new images.

All the training and testing tasks described in this study were performed using Python 3.12 (64 bit). Firstly, the model receives images that are labelled with object classes and bounded boxes, as shown in Fig. 5. Secondly, YOLO v5 learns the relationships between object classes and features (e.g. texture, colour and shape) by using these images. Thirdly, the model optimises its weights by using backpropagation to minimise the loss function. This procedure measures the difference between the locations and classes of predicted and actual objects. Fourthly, features that are relevant to object recognition are extracted from the images. Finally, the extracted features are mapped onto the labelled features in the dataset, allowing objects to be identified quickly and accurately in the new images.

Fig. 5. Object detection results using YOLOv5 algorithm

III. The Potential Drawbacks or Challenges of YOLOv5

Although YOLOv5 has demonstrated good performance in the tested safety lab environment, its application in dynamic environments may present some challenges. The most prominent of these challenges include the algorithm's ability to adapt to rapidly moving objects and changes in lighting. Moving objects can interfere with detection processes, leading to decreased accuracy, and the algorithm may struggle to identify objects that move quickly or in environments with fluctuating lighting. Additionally, continuous real-time tracking of objects is essential for collision avoidance in dynamic environments, which presents a further challenge for YOLOv5. The algorithm may require integration with other systems to improve tracking accuracy. Moreover, interference between objects or unclear data can negatively impact performance, highlighting the need for further improvements and integration with other algorithms, such as moving object tracking, to enhance collision

IV. Results and Discussion

To evaluate the overall effectiveness of the model, monitoring the metrics of the training dynamics of the YOLO v5 model and its performance on validation data is essential. A series of graphs organised in a grid format is displayed in Fig. 6. Each graph depicts various aspects of a model's training and evaluation.

The loss associated with predicting the bounding box coordinates during training is tracked by the train/box loss curve. When loss is decreasing, the model is improving in placing boxes around objects. Object loss, which measures how well the model differentiates between objects and background, is represented by the train/obj loss curve. Low loss values are preferred. Classification loss, which gauges how accurately the model categorises detected objects, is displayed by the train/cls loss graph. Here, decreasing values of this metric also indicate improvement. The proportion of true positive predictions amongst all positive predictions made is evaluated by the metric/precision graph. A high precision indicates that the model is accurate in object prediction. The model's effectiveness in identifying all relevant cases (true positives) is measured by metrics/recall, which is calculated by dividing the number of true positive predictions by the total number of actual positives (true positives plus false negatives). A high recall indicates that the model is proficient in capturing as many relevant instances as possible.

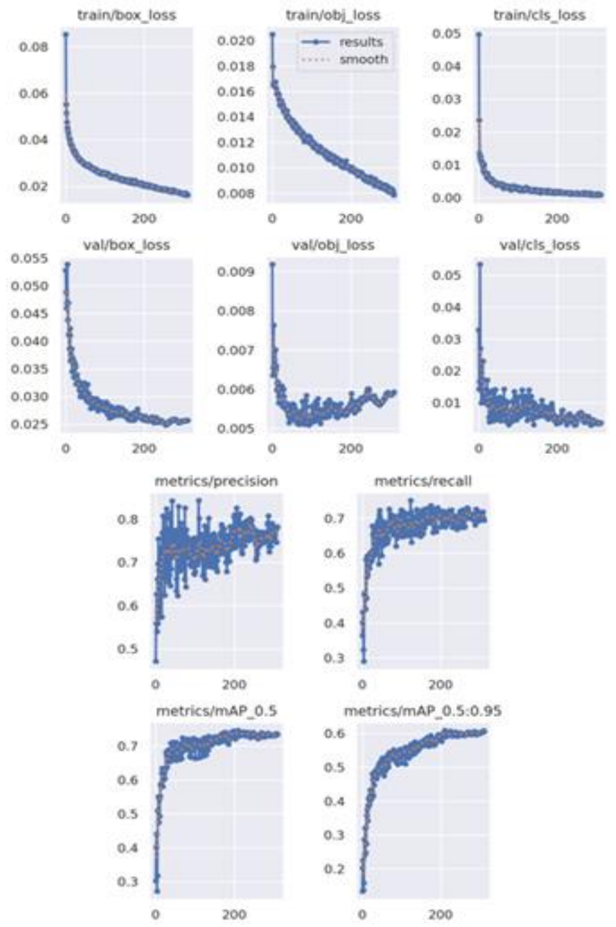

Fig. 6. Training and validation metrics for object detection model

Box loss on validation data is reflected by the val/box_loss graph, which helps assess the generalisation of the model to new, unseen data. The model's performance on the validation dataset is measured by val/obj_loss, which evaluates loss related to model accuracy in predicting the presence of objects in images. Low values are preferred. The val/cls_loss graph is like train/cls_loss. However, it is applied to the validation dataset and indicates model performance in correctly classifying objects on validation data. Metrics/mAP_0.5 represents mAP at a confidence threshold of 0.5. It provides an overall summary of precision across different confidence levels. Meanwhile, metrics/mAP_0.5:0.95 denotes mAP calculated over confidence thresholds ranging from 0.5 to 0.95. It provides a detailed view of model performance across different detection confidence levels.

In summary,

- Low Loss Values: Gradual decreases in all validation loss metrics (box, objectness, classification) signify that the model is improving in terms of localization, detection, and classification on unseen data.
- High mAP Values: High scores in metrics/mAP_0.5 and metrics/mAP_0.5:0.95 suggest strong detection capabilities and generalization, with the latter providing a more nuanced view of the model's robustness.
- Discrepancies Between Training and Validation Metrics: Significant differences between training and validation losses or mAP values may indicate overfitting, requiring strategies such as data augmentation, regularization, or architecture adjustments.

The F1-confidence curve, which is a common tool for assessing the performance of object detection models, such as YOLO v5, is depicted in Fig. 7. On the graph, each coloured line represents the F1 score for a specific class (e.g. chair, warning cone, fire extinguisher, fan, people and warning sign), providing insights into the performance of the model across various object categories and confidence levels.

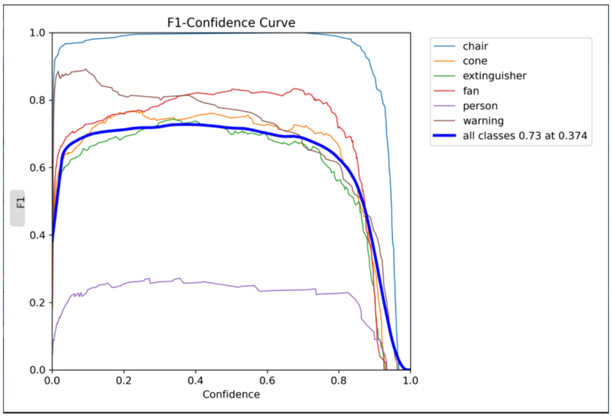

Fig. 7. F1-Confidence curve per class for object detection

In the F1 score metric, precision and recall are combined into a single score, reflecting the accuracy of the model. A high F1 score indicates better performance. Confidence measures the certainty of a model regarding its predictions. with scores ranging from (0 to 1). High values indicate high certainty. The overall curve, shown by the blue line, denotes the average F1 score across all classes. Specific scores are displayed at various confidence thresholds.

The results indicate that the YOLOv5 model experiences a significant decline in person recognition performance, with a low F1-score of 0.19 compared to other categories. This weakness can be attributed to the fact that a person is a dynamic object, as body positions and viewing angles change rapidly when in motion, leading to considerable variation in image characteristics. Additionally, continuous movement results in motion blur in the captured images, making it difficult to extract clear and specific features of individuals. Moreover, the training data primarily consist of people in static positions, which hampers the model's generalisation when applied to dynamic scenes. The performance of the YOLOv5 model also varied across other categories (fire extinguisher, cone, warning, and fan). The recognition process depends on the characteristics of the training data. These categories include objects of different sizes and colours, increasing the complexity of the recognition process. This suggests that the model requires additional training data encompassing various shapes and sizes to enhance its recognition accuracy. The model exhibited very high performance in recognising the category "chair," as both the training and testing data contained the same type of chairs. Consequently, the model did not encounter issues related to visual variation between samples, resulting in accurate and stable recognition.

A confusion matrix, which summarises the prediction results for a classification problem, is depicted in Fig. 8. This matrix illustrates how many instances of each class are correctly identified (true positives) and where the model made errors (false positives and negatives). Values along the diagonal (from top left to bottom right) signify the accuracy of the model for each class. High values reflect good performance. Meanwhile, misclassifications are represented by off-diagonal values.

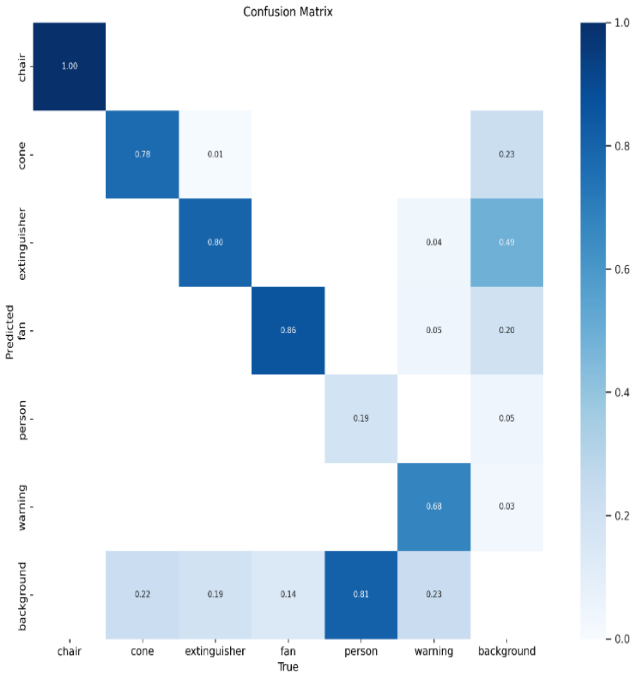

Fig. 8. Confusion matrix for object detection model performance

A perfect recognition rate of 1.00 is achieved by the chair class. That is, all instances of ‘chair’ are classified correctly. The fan, fire extinguisher and cone classes also exhibit good performance, with several misclassifications, particularly those being confused with ‘background’. The person class presents a low performance rate of 0.19, indicating that the model has incorrectly classified ‘person’ as ‘background’ 81% of the time. Lastly, the warning sign class demonstrates reasonable performance, with some confusion occurring primarily with ‘background’. In general, the confusion matrix offers a visual summary of the detection performance of the model. Areas where the model performs well are highlighted, and aspects that may require improvement are identified. The analysis of the results can help guide refinements to the model’s architecture, adjustments to the training data or tuning of the hyperparameters in the future.

Key strategies for improving model performance in classification tasks are shown in the mind map in Fig. 9. It emphasizes enhancing training data by increasing samples for underperforming classes and applying data augmentation techniques. Feature extraction can be refined using additional data sources like depth or infrared, while loss functions should incorporate class-specific weighting to address errors in minority classes. Optimizing model architecture through advanced techniques such as attention mechanisms and multi-scale feature extraction is also suggested. Regular evaluation, including monitoring confusion matrix trends and iterative refinement, along with addressing background confusion by balancing datasets, are critical steps for achieving robust and accurate results.

The model is learning effectively as indicated by the decrease in training loss (box, objects and classification) over time. Being higher than validation loss is beneficial for training loss, because a significantly higher validation loss compared with training loss can indicate overfitting. When precision and mAP values are consistently high, the model is suggested to be reliably detecting and classifying objects. As confidence increases for most classes, the F1 score tends to decline. The chair class maintains high F1 scores across a wide range of confidence thresholds. This result indicates better balance between precision and recall for this category. By contrast, the person class exhibits a relatively low F1 score across most thresholds. This finding suggests that the model may have trouble in accurately detecting people. The F1 score for the person class gradually decreases as the confidence threshold rises, implying that the model may be overly strict in its detection criteria for people, resulting in more false negatives. Overall, the F1 scores for all the classes stabilise at around 0.73 at a confidence threshold of approximately 0.374. This result signifies that the threshold is a reliable choice when making general predictions.

This study focuses on enhancing robot mobility by incorporating advanced obstacle avoidance mechanisms. Since effective obstacle avoidance depends primarily on accurately detecting objects in the surrounding environment, the performance of YOLOv5 directly influences a robot’s ability to navigate safely in industrial and unstructured environments. Obstacle avoidance is a critical function in robotic systems, particularly for autonomous robots operating in dynamic or unpredictable settings. This capability requires robots to detect, analyze, and maneuver around obstacles in real time, which is essential for applications such as autonomous vehicles, drones, industrial robots, and service robots. Modern approaches to enhancing obstacle avoidance integrate multiple sensors, such as LiDAR, ultrasonic sensors, infrared sensors, and cameras, to collect environmental data. This data is processed using advanced algorithms, including simultaneous localisation and mapping (SLAM), path planning, and machine learning-based decision-making frameworks. These techniques enhance a robot’s situational awareness, enable it to anticipate potential collisions, and optimise alternative path planning. Accordingly, performance analysis suggests that improving YOLOv5's accuracy in detecting objects—especially moving objects—can significantly enhance the efficiency of obstacle avoidance systems in intelligent robots.

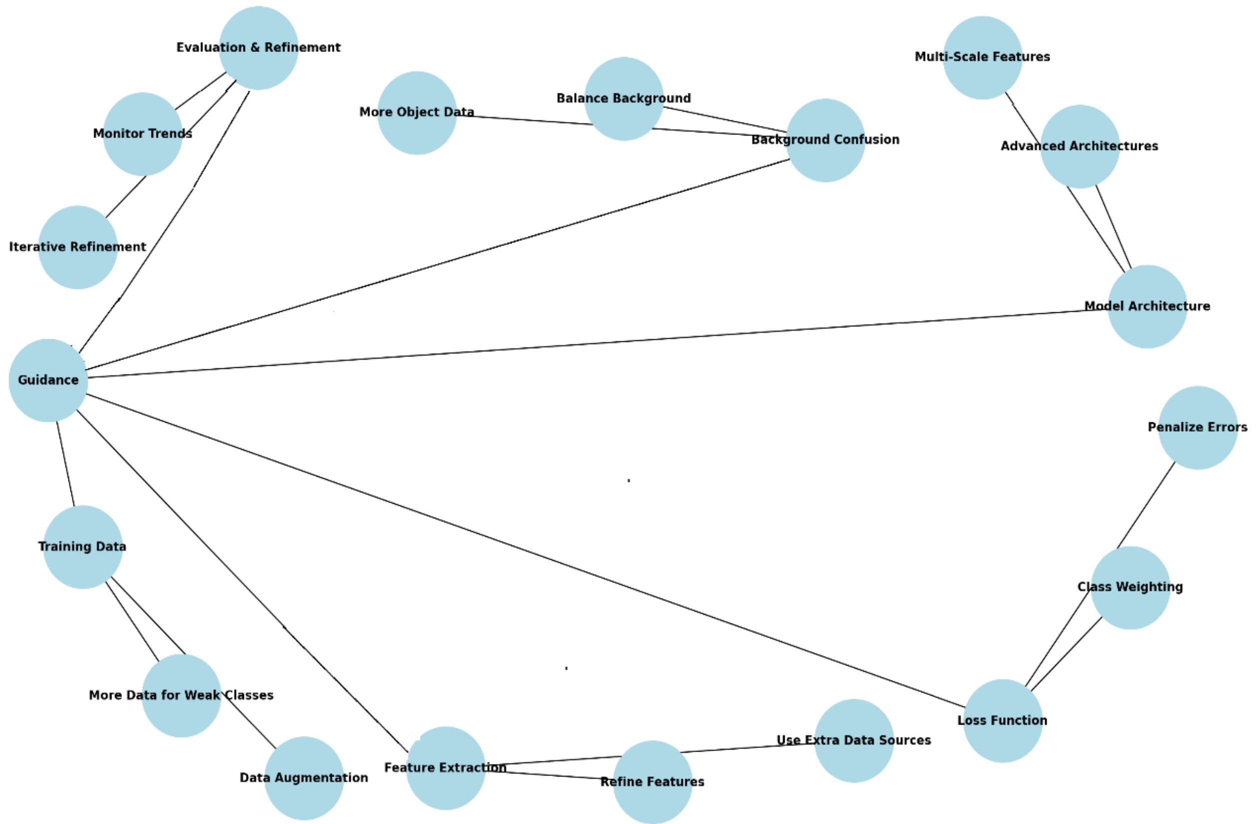

Fig. 9. Mind map illustrating the guidance for improving model performance in object detection

V. Conclusion

This study explores the application of YOLOv5 to enhance the efficiency and safety of robots through object detection and collision avoidance. The results demonstrated high accuracy and speed in recognizing static objects; however, the model faced challenges with dynamic objects, such as people, due to rapid changes in posture and the effects of motion blur. To improve performance, it is recommended to increase the diversity of training data to include various poses and movements, as well as enhance image processing through techniques like motion blur simulation and lighting adjustments. Additionally, integrating YOLOv5 with sensors such as LiDAR can further enhance detection accuracy. This research represents an important step toward developing more efficient systems in dynamic environments, paving the way for future improvements in industrial robots, autonomous vehicles, and unmanned aerial systems.

VI. Recommendations

- Increase diversity in training data:

Include samples with variations in size, shape, and color of static objects collected from different environments (indoor, outdoor, natural light, artificial light).

If the model is designed to detect people, incorporate data containing individuals of different sizes and postures, as well as various movement patterns, to enhance generalization.

- Improve image processing during training:

Applying data augmentation techniques such as motion blur to simulate real-world motion effects, with introducing lighting and contrast adjustments to improve the model's robustness under diverse imaging conditions.

Acknowledgment

I would like to express my sincere gratitude to the Training and Research Department of the Ministry of Electricity in Iraq, specifically the Safety Division, for allowing me to work in their lab during this research. The opportunity to access their facilities was invaluable in completing this project. I appreciate the support of the dedicated professionals in the department.

References

[1] Z. Bai, "Advancements in robotics engineering: Transforming industries and society," in *Int. Conf. Machine Learning and Automation*, 2023, doi: 10.54254/2755-2721-/32/20230861.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016.

[3] Ş. Çiğdem, I. Meidute-Kavaliauskiene, and B. Yıldız, "Industry 4.0 and industrial robots: A study from the perspective of manufacturing company employees," *Logistics*, vol. 7, no. 1, p. 17, 2023.

[4] D. Duan *et al*., "Industrial robots and firm productivity," *Structural Change and Economic Dynamics*, vol. 67, pp. 388–406, 2023.

[5] D. A. Zebari, D. Q. Zeebaree, A. M. Abdulazeez, H. Haron, and H. N. A. Hamed, "Improved threshold-based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images," *IEEE Access*, vol. 8, pp. 1–12, Nov. 2020, doi: 10.1109/ACCESS.2020.3036072.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[8] P. H. Rosen, E. Heinold, E. Fries-Tersch, and S. Wischniewski, “Advanced robotics and automation: implications for occupational safety and health,” *Brussel: European Agency for Safety and Health at Work.* vol. 10, p. 789276, 2022.

[9] S. K. Sahoo and B. B. Choudhury, "Challenges and opportunities for enhanced patient care with mobile robots in healthcare," *J. Mechatronics Artif. Intell. Eng*., vol. 4, no. 2, pp. 83–103, 2023.

[10] M. L. Ali and Z. Zhang, "The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection," *Computers*, vol. 13, no. 12, p. 336, 2024.

[11] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, “Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy,” *Vehicles*, vol. 6, no. 3, pp. 1364-1382, 2024.

[12] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[13] Z. Fatima, S. Zardari, and M. H. Tanveer, "Advancing industrial object detection through domain adaptation: A solution for industry 5.0," *Actuators*, vol. 13, no. 12, 2024.

[14] W. Hanafi and M. Tamali, "Implementing distributed collaboration and applying the YOLO algorithm to robots," *Studies in Engineering and Exact Sciences*, vol. 5, no. 1, pp. 277–296, 2024.

[15] D. H. Dos Reis, D. Welfer, M. A. De Souza Leite Cuadros, and D. F. T. Gamarra, "Mobile robot navigation using an object recognition software with RGBD images and the YOLO algorithm," *Appl. Artif. Intell*., vol. 33, no. 14, pp. 1290–1305, 2019.

[16] A. Aljaafreh *et al*., "A real-time olive fruit detection for harvesting robot based on YOLO algorithms," *Acta Technologica Agriculturae*, vol. 26, no. 3, pp. 121–132, 2023.

[17] G. Liu, Y. Hu, Z. Chen, J. Guo, and P. Ni, "Lightweight object detection algorithm for robots with improved YOLOv5," *Eng. Appl. Artif. Intell*., vol. 123, p. 106217, 2023.

[18] 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.

[19] Z. Chen, X. Li, L. Wang, Y. Shi, Z. Sun, and W. Sun, "An object detection and localization method based on improved YOLOv5 for the teleoperated robot," *Appl. Sci*., vol. 12, no. 22, p. 11441, 2022.

[20] F. Wang and X. Li, "Research on Warehouse Object Detection for Mobile Robot Based on YOLOv5," *2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pp. 1196-1200, 2023.

[21] Z. Li, C. Pang, C. Dong, and X. Zeng, "R-YOLOv5: A lightweight rotational object detection algorithm for real-time detection of vehicles in dense scenes," *IEEE Access*, vol. 11, pp. 61546–61559, 2023.

[22] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," *arXiv preprint arXiv:2407.20892*, 2024.

[23] G. Lavanya and S. D. Pande, "Enhancing real-time object detection with YOLO algorithm," *EAI Endorsed Trans. Internet Things*, vol. 10, 2024.

[24] A. S. Geetha, "Comparing YOLOv5 variants for vehicle detection: A performance analysis," *arXiv preprint arXiv:2408.12550*, 2024.

[25] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Comput. Sci*., vol. 199, pp. 1066–1073, 2021.

[26] K. Saranya, J. J. R. Jegaraj, K. R. Kumar, and G. V. Rao, "Artificial intelligence based selection of optimal cutting tool and process parameters for effective turning and milling operations," *J. Inst. Eng. (India): Ser. C*, vol. 99, pp. 381–392, Aug. 2018.

[27] S. Han, X. Dong, X. Hao, and S. Miao, "Extracting objects’ spatial–temporal information based on surveillance videos and the digital surface model," *ISPRS Int. J. Geo-Inf*., vol. 11, no. 2, p. 103, 2022.

[28] J. Lim, J. Lee, C. An, and E. Park, "Enhancing real-time traffic volume prediction: A two-step approach of object detection and time series modelling," *IET Intell. Transp. Syst*., vol. 18, no. 12, pp. 2744–2758, 2024.

[29] G. Jocher *et al*., "ultralytics/yolov5: v6.2-yolov5 classification models, apple m1, reproducibility, clearml and deci.ai integrations," *Zenodo*, 2022.

[30] C. Limberg, A. Melnik, A. Harter, and H. J. Ritter, "YOLO - You only look 10647 times," *arXiv preprint arXiv:2201.06159*, 2022.

[31] Z. Wang *et al*., "Improved small object detection algorithm CRL-YOLOv5," *Sensors*, vol. 24, no. 19, p. 6437, 2024.

[32] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 8759-8768, 2018.

[33] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," *arXiv preprint arXiv:2112.11798*, 2023.

[34] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, vol. 22, no. 1, 2011.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[36] G. I. Molina *Learning to detect deepfakes: Benchmarks and algorithms*. M.S. thesis, Universitat Pompeu Fabra, 2020.

[37] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. CVPR 2001*, vol. 1, 2001.

[38] Y. Xiao, X. Wang, P. Zhang, F. Meng, and F. Shao, "Object detection based on Faster R-CNN algorithm with skip pooling and fusion of contextual information," *Sensors*, vol. 20, no. 19, p. 5490, Sep. 2020.

[39] C. C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Cham: Springer, 2018.

[40] K. Gaurav, A. K. Sahoo, and S. K. Mishra, "Nonlinear system identification using functional link multilayer perceptron artificial neural networks," *Int. J. Eng. Res*., vol. 10, no. 44, p. 6, 2015.

[41] D. A. Zebari, D. Q. Zeebaree, A. M. Abdulazeez, H. Haron, and H. N. A. Hamed, "Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images," *IEEE Access*, vol. 8, pp. 1–12, Nov. 2020.

[42] S. Liu *et al*., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," *arXiv preprint arXiv:2201.12329*, 2022.

[43] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li and J. Shi, "FoveaBox: Beyound Anchor-Based Object Detection," in *IEEE Transactions on Image Processing*, vol. 29, pp. 7389-7398, 2020.

[44] V. Varadarajan, D. Garg, and K. Kotecha, "An efficient deep convolutional neural network approach for object detection and recognition using a multi-scale anchor box in real-time," *Future Internet*, vol. 13, no. 12, p. 307, 2021.

[45] S.-S. Park, V.-T. Tran, and D.-E. Lee, "Application of various YOLO models for computer vision-based real-time pothole detection," *Appl. Sci*., vol. 11, no. 23, p. 11229, 2021.

[46] B. R. Paudel, D. Senarathna, H. Wang, S. Tragoudas, Y. Hu and S. Jiang, "Predicting YOLO Misdetection by Learning Grid Cell Consensus," *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 643-648, 2021.

[47] C. Limberg, A. Melnik, A. Harter, and H. Ritter, "YOLO--You only look 10647 times," *arXiv preprint arXiv:2201.06159*, 2022.

[48] S. N. Tesema and E. -B. Bourennane, "Multi-Grid Redundant Bounding Box Annotation for Accurate Object Detection," *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 145-152, 2021.

[49] C. Limberg, A. Melnik, H. Ritter, and H. Prendinger, "YOLO: You Only Look 10647 Times," *INSTICC*, pp. 153–160, 2023.

[50] D. Novaković, N. Vasić, S. Novaković, D. Kostić, R. Bianchini, "{DeepDive}: Transparently identifying and managing performance interference in virtualized environments," in *2013 USENIX Annual Technical Conference (USENIX ATC 13)*, pp. 219-230, 2013.

[51] D. Jayakumar and S. Peddakrishna, "Performance evaluation of YOLOv5-based custom object detection model for campus-specific scenario," *Proc.-Int. J. Exp. Res. Rev*., 2024.

[52] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digit. Signal Process*., vol. 132, p. 103812, 2023.

[53] 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," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464-7475, 2023.

[54] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimed. Tools Appl*., vol. 82, no. 6, pp. 9243–9275, Mar. 2023.

[55] S. Zhang, C. Qu, C. Ru, X. Wang, and Z. Li, "Multi-objects recognition and self-explosion defect detection method for insulators based on lightweight GhostNet-YOLOv4 model deployed on board UAV," *IEEE Access*, vol. 11, pp. 61546–61559, 2023.

[56] T. Gandor and J. Nalepa, "First gradually, then suddenly: Understanding the impact of image compression on object detection using deep learning," *Sensors*, vol. 22, no. 3, p. 1104, 2022.

[57] D. Zhao, F. Shao, L. Yang, X. Luo, Q. Liu, H. Zhang, and Z. Zhang, “Object detection based on an improved YOLOv7 model for unmanned aerial-vehicle patrol tasks in controlled areas,” *Electronics*, vol. 12, no. 23, p. 4887, 2023.

[58] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on YOLO-V5 with improved maritime dataset," *J. Mar. Sci. Eng*., vol. 10, no. 3, p. 377, 2022.

[59] A. R. Pathak, M. Pandey, and S. Rautaray, “Application of deep learning for object detection,” *Procedia computer science*, vol. 132, pp. 1706-1717, 2018.

[60] S. Liu, H. Zhou, C. Li, and S. Wang, "Analysis of Anchor-Based and Anchor-Free Object Detection Methods Based on Deep Learning," *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1058-1065, 2020.