COMPUTOLOGY: Journal of Applied Computer Science and Intelligent Technologies
Volume 4, Issue 2, Jul-Dec 2024, pp. 18-40
DOI: 10.17492/computology.v4i2.2402
https://www.journalpressindia.com/cjacsit

A Literature-based Performance Assessment of the YOLO (You Only Look Once) CNN Approach for Real-time Object Detection

*Sandeep Bhattacharjee**

ABSTRACT

Real-time object identification is considered as one of the major catalysts for computer vision, such as video surveillance, autonomous driving, robotics, and augmented reality. You Only Look Once (YOLO) is a state-of-the-art object detection algorithm based on Convolutional Neural Networks (CNNs) that provides an efficient solution by utilizing both classification and localization in a single forward pass through the network. This review provides a comprehensive overview of YOLO's architecture, key innovations, comparable performance, challenges, and its impact on the field of real-time object detection. It also discusses the improvements that can be made in subsequent versions of YOLO and explores potential future research approaches.

Keywords: *Architecture; Classification; Image; Real time; Object recognition.*

1.0 Introduction

The function of Object identification is a complicated task that includes both classification and as well as location of items in a picture. Traditional object recognition algorithms often involved moving windows and region-based methods, which were computationally expensive and slow, making them unsuitable for real-time apps. The evolution of deep learning, especially Convolutional Neural Networks, altered the concept of object detection. YOLO, first presented by Redmon *et al.* (2016), became a game-changer by combining detection and classification into a single network pass, greatly improving detection speed and efficiency.

**Assistant Professor, Amity Business School, Amity University, Kolkata, West Bengal, India (E-mail: sandeepbitmba@gmail.com)*

1.1 Definitions

- "YOLO can handle object recognition as a single regression issue, directly from picture pixels to bounding box coordinates and class probabilities. This unified model enables the computer to directly anticipate objects in pictures using a single forward pass through the neural network."- (Redmon *et al.*, 2016).
- "YOLO is a real-time object detection system that divides the image into a grid and predicts bounding boxes and probabilities for each region, achieving fast inference speeds while maintaining competitive accuracy." (Bochkovskiy *et al.*, 2020).
- "YOLO is a single-pass convolutional neural network that identifies many items in an image by generating predictions on grid cells, each of which is responsible for recognizing objects in that section of the picture. This model streamlines the object identification process into a single neural network assessment." (Redmon & Farhadi, 2017).
- "YOLO algorithm divides an input picture into a grid where each grid cell performs the duty for identifying objects. Each cell must predict different bounding boxes and the related confidence ratings, decreasing the complexity of many stages in detection frameworks." (Liu *et al.*, 2016).
- Unlike other region proposal-based methods, YOLO orchestrates object detection as a statistical regression problem, involved in directly predicting values within bounding box coordinates and class probabilities from full images in one single step, thus leading to faster and more efficient object identification." (Redmon & Farhadi, 2018).

YOLO's technique erudite for real-time object recognition, engaged in handling both big and tiny objects across a broad range of situations, making it ideal for time-sensitive applications like autonomous driving and video analytics. This evaluation will analyse the YOLO method, its later iterations (YOLOv2, YOLOv3, YOLOv4, and YOLOv5), and its performance compared to other object detection frameworks.

2.0 Literature Review

The major novelty of YOLO is that it considers object detection as a regression issue rather than a classification one. Unlike region proposal-based algorithms like Faster R-CNN that employ several stages (i.e., a separate region proposal network followed by classification), YOLO can predict the bounding boxes and class probabilities directly from the complete picture in a single forward pass. This end-to-end technique makes YOLO more faster and efficient. The architecture splits the input

picture into a S x S grid, where each grid cell predicts bounding boxes and their associated confidence ratings. The confidence score indicates the correctness of the anticipated bounding box and whether it contains an item. YOLO predicts numerous bounding boxes for each grid cell, but only the boxes with the highest confidence ratings are maintained for final predictions (Redmon *et al.*, 2016).

The Real Time Object Identification project employs deep learning and convolutional neural networks to recognize items in video input. In one of the publications, a quicker real-time object identification approach was presented utilizing a convolution neural network model called Single Shot Multi-Box identification (SSD). It also presents a lightweight model called MobileNet, which boosts the accuracy of recognizing household items using depth-wise separable convolution (Kanimozhi *et al.*, 2019). The Single Shot MultiBox Detector (SSD) technique can recognize objects surrounded by boxes, generating a confidence score. The design substitutes VGG Net with residual networks for enhanced computing performance.(KR, 2017).

A unique strategy for recognizing and localizing duplicate items in pick-and-place applications under high occlusion situations has been presented in one such paper. It employs SIFT key point extraction and mean shift clustering to segregate correspondences, confirms object shape hypotheses, and deploys multiple object models for reflected or transparent objects. The study presents metrics of effectiveness and efficiency in real-time processing (Piccinini *et al.*, 2012). Another research examined on how to find out an object's name from examinations of it. A formal foundation is offered, which would be based on a prior over physical events and an identity requirement. The abstract possibility of identity may be described in terms of quantifiable appearance probabilities, that can be trained online to adapt to changing external variables. The research also explores a system for automobile matching utilizing bipartite matching and a leave-one-out approach. The system achieved great precision in matching individual autos, making it the first safe approach for assessing link transit durations (Huang & Russell, 1997).

An overview of the You Only Look Once (YOLO) algorithm and its advanced versions, highlighting differences and similarities betw YOLO and Convolutional Neural Networks was highlighted ongoing improvement and contributes to the literature on targeted picture news and feature extraction in various domains (Jiang *et al.*, 2022). Another application-based study has been recommended for real-time apple identification in trees, allowing for a more accurate estimation of yield utilizing the YOLO-V3 model. This model utilizes photos of immature, growing, and ripe apples, adjusted utilizing rotation, color balancing, brightness, and noise processing. The

DenseNet approach enhances feature transfer and network efficiency. The model trumps the original YOLO-V3 model and the Faster R-CNN with VGG16 net model, yielding an average recognition time of 3000-3000 frames per frame. It can effectively recognize apples under many apples and blocking conditions (Tian *et al.*, 2019).

On the other hand, Object identification seems to be a tough problem in computer vision, involving both object labelling and position. Although Deep neural networks (DNNs) have demonstrated higher performance but using them for real-time object detection on embedded devices remains tricky. This work introduces Fast YOLO, a fast system that speeds YOLOv2 for real-time object identification on embedded devices. The system employs an evolutionary deep intelligence framework to create the YOLOv2 network design, minimizing parameters and power utilization. Experimental findings suggest Fast YOLO decreases deep conclusions and speeds up object identification, operating at an average of 18FPS (Shafiee *et al.*, 2017).

Deep neural networks (DNNs) are beneficial in object recognition, but their use in confined situations like embedded devices is limited. Tinier-YOLO, a version of Tiny-YOLO-V3, tries to decrease model size while boosting identification accuracy and real-time performance. It employs thick connections between fire units to optimize feature transmission and feature flow. The passthrough layer combines feature maps to generate fine-grained features, mitigating the adverse impact of reduced model size. Tinier-YOLO has 25 FPS real-time speed on Jetson TX1 and 65.7% mAP on PASCAL VOC and 34.0% on COCO (Fang *et al.*, 2019).

Object recognition has shown tremendous progression with the introduction of Convolutional Neural Networks (CNN) since 2012 while quicker R-CNN has a slower Frame Per Second (FPS) than real-time effects, necessitating quicker object detection. You Only Look Once (YOLO) has grown as a novel, faster approach for object identification, obtaining FPS 155 and mAP up to 78.6, exceeding Faster R-CNN's performance. YOLOv2 provides an alternative between speed and precision, enabling for a robust visual representation (Du, 2018) A balanced object tracker based on YOLOv3 was framed using PaddlePaddle, with the aim to achieve higher accuracy while preserving speed. The detector, named PP-YOLO, blends known methods without boosting model parameters and FLOPs. The objective of this version is to obtain a superior blend of efficacy (45.2% mAP) and speed (72.9 FPS), outperforming state-of-the-art gadgets like EfficientDet and YOLOv4. (Long *et al.*, 2020).

The preceding literature papers demonstrate the utilization of YOLO CNN in several applications of object identification. But, there appears to be a big gap in comprehending the genuine YOLO CNN and the fundamental powers of YOLO

algorithm. In this study work, we have attempted to identify the genuine YOLO CNN architecture, properties of YOLO architecture, comparison of YOLO CNN technique with other methods, obstacles encountered by YOLO CNN and future directions of improvements in YOLO CNN.

3.0 Discussions

3.1 Impact of real-time object identification

Real-time object recognition is crucial for safe navigation in autonomous vehicles and video surveillance systems. It allows for quick identification of objects, reducing accidents and allowing for efficient responses to environmental changes. This technology also enables security personnel to take necessary steps before potential hazards occur, ensuring a safer and more efficient driving experience. Real-time object identification is vital for safe navigation in autonomous cars, video surveillance systems, robotics, healthcare, augmented reality, and agriculture (Levine *et al.*, 2018; Esteva *et al.*, 2017; Azuma, 2016; Zhang *et al.*, 2019).

It allows a fast identification of objects, minimizing the chance of accidents and boosting proper reactions to environmental changes. In robotics, it promotes robots' interaction with their environment, boosting industrial automation and service applications. In healthcare, it creates speedier cancer diagnosis and surgical navigation systems, enhancing diagnostic accuracy and patient care (Esteva *et al.*, 2017). Augmented Reality and Virtual Reality overlays digital information on the surroundings, enhancing immersive user experiences (Azuma, 2016). In agriculture, it accelerates the identification of crops, weeds, and pests, enabling for precise treatments and enhancing efficiency and sustainability (Zhang *et al.*, 2019). In summary, real-time object identification is crucial for different such applications.

3.2 Features of YOLO (You Only Look Once) CNN:

YOLO simplifies object identification workflow by integrating classification and localization tasks in one step, predicting class probabilities and bounding boxes directly from the complete picture, unlike multi-stage detectors (Redmon *et al.*, 2016). YOLO is a grid-based system that splits an input picture into S x S cells, each predicting objects cantered within that cell, enabling the system to recognize multiple items in a single shot (Redmon & Farhadi, 2017). YOLO, a speed-tuned object detection system, is ideal for applications like autonomous driving and video monitoring due to its ability to analyse

pictures at 45 FPS and 155 FPS, predicting multiple bounding boxes and class probabilities (Redmon *et al.*, 2016).

YOLO, a machine learning algorithm, uses global context to improve object categorization in settings with associations or when context is crucial for accurate predictions. It analyses the entire image simultaneously (Redmon *et al.*, 2016). YOLO's unique feature is its ability to predict bounding boxes and class probabilities in a single neural network run, reducing computational complexity compared to other region-based detectors like Faster R (Redmon *et al.*, 2016). YOLOv2 incorporates anchor boxes for improved localization accuracy, enhancing forecasting of aspect ratio and size, and improving performance on tiny items (Redmon & Farhadi, 2017). YOLOv3 enhances performance by incorporating multi-scale detection, enabling the system to recognize objects of varying sizes, improving its ability to handle tiny objects and dense scenes (Redmon & Farhadi, 2018).

3.3 Algorithm of YOLO CNN

3.3.1 Mathematical formula

Assumptions

- The coordinates of the centre of the box: (x,y).
- The dimensions of the box: (w,h).
- A confidence score for the presence of an object in the box: $c_{confidence}$
- $C_{confidence} = = P(Object) \times IoU_{pred}^{truth}$
- $P(Class_i|Object)$ is the conditional class probability for class i.

a. Image Division into Grid Cells

- Let S = the number of grid cells along each dimension (height and width).
- The image is divided into **S x S grid** cells, and each grid cell is responsible for detecting an object if the object's centre is within the cell.

b. Bounding Box Predictions :

Bounding box = $(x,y, w,h, c_{confidence})$

c. Class Predictions

$P(Class_i|Object)$

d. Final Detection Formula

$S_i = C_{confidence} \times P(Class_i|Object)$

e. **Loss function**

- *Localization Loss* :

$$\lambda_{coord} \sum_{i=0}^{s2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{\omega}_i)^2 + (h_i - \hat{h}_i)^2] \quad \ldots[1]$$

where 1_{ij}^{obj} indicates whether an object exists in grid cell i and box j.

- *Confidence Loss* : Measures the difference between the predicted and true confidence scores for the presence of objects.

$$\sum_{i=0}^{s2}\sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj}\sum_{i=0}^{s2}\sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad \ldots[2]$$

- *Classification Loss* : Measures the error in the predicted class probabilities.

$$\sum_{i=0}^{s2} 1_i^{obj} \sum_{c \epsilon classes} (P(class_i) - \hat{P}(class_i))^2 \quad \ldots[3]$$

- **FINAL LOSS FUNCTION**:

$$L = \lambda_{coord} \sum_{i=0}^{s2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i^{1/2} - \hat{\omega}_i^{1/2})^2 + (h_i - \hat{h}_i^{1/2})^2$$
$$\sum_{i=0}^{s2}\sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj}\sum_{i=0}^{s2}\sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 +$$
$$\sum_{i=0}^{s2} 1_i^{obj} \sum_{c \epsilon classes} (P(class_i) - \downarrow (class_i))^2 \quad \ldots[4]$$

3.3.2 Architecture of YOLO CNN

Figure 1: Single Neural Network for Object Detection

Source: Author own

Single neural network for object detection: A single neural network model in object detection performs object identification, categorization, and localization in one step. The YOLO (You Only Look Once) method is one of the most popular method that performs such function at great speed, with adequate support from fast GPU's (See Figure 1).

Division of image into grid cells: The input picture is grid-arranged for efficient and precise object recognition, especially with methods like YOLO (You Only Look Once). It classifies objects in each grid and compares them to existing database for recognition confirmation and decision making (See Figure 2).

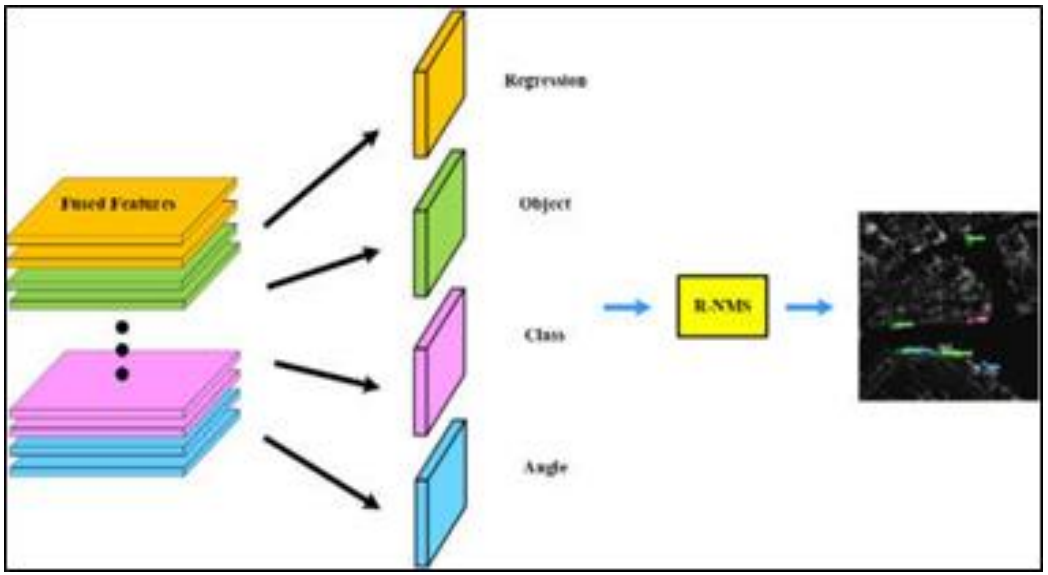
Figure 2: Division of Image into Grid Cells

Source: Author own

Prediction of bounding boxes and class probabilities: In the YOLO (You Only Look Once) algorithm, forecasting bounding boxes and class probability is essential to its object detection approach. This process allows YOLO to locate objects in an image

and categorize them, all in one single step. The feature in the image undergoes regression, object recognition, classification, angle detection all at the same time, with results yielding a complete picture of the environment (See Figure 3).

Figure 3: Prediction of Bounding Boxes and Class Probabilities



Source: Author own

Comparison YOLO with other object detection methods: Here is a comparison of YOLO with other popular object detection methods like Faster R-CNN, SSD, and RetinaNet in table format (see Table 1):

YOLO is by far the quickest, capable of performing in real-time applications on strong GPUs. Other similar methods such as R-CNN and RetinaNet tend to outperform YOLO slightly in accuracy, particularly for small objects and complicated conditions. YOLO is more appropriate for real-time applications such as surveillance and autonomous driving, whereas Faster R-CNN and RetinaNet are preferable for jobs demanding high accuracy (See Table 1).

Some of the top emerging trends in object detection include AI-powered detection, edge computing, 3D object detection, multi-modal detection, small object detection, real-time detection, few-shot learning, explainable AI, adverse condition detection, and transfer learning. Detectum, created by Xailient, is another deep learning neural network algorithm designed to operate with extremely low power consumption

(Xailient, n.d.). TogetherNet model integrates image restoration with object detection to boost performance under challenging weather conditions. By leveraging dynamic enhancement learning, it improves feature extraction and representation capabilities (Wang *et al.*, 2022).

Table 1: Comparison of YOLO with other Object Detection Methods

Method	Architecture	Speed (FPS)	Accuracy (mAP)	Strengths	Weaknesses
YOLO (You Only Look Once)	Single pass CNN predicting bounding boxes and class probabilities	30-60 FPS (YOLOv3)	~57.9% (COCO dataset, YOLOv3)	Fastest among detectors; real-time processing; good general performance	Struggles with small objects; less precise compared to R-CNN variants
Faster R-CNN	Two-stage detector with region proposal network (RPN)	5-7 FPS	~58.5% (COCO dataset)	High accuracy, especially for small objects; robust object localization	Slow; computationally expensive; less suitable for real-time applications
SSD (Single Shot MultiBox Detector)	Single-stage CNN predicting bounding boxes and class probabilities	22-59 FPS	~46.5% (COCO dataset)	Faster than R-CNNs; good balance between speed and accuracy; efficient for large-scale objects	Less accurate with small objects and cluttered environments
RetinaNet	Single-stage detector with focal loss to handle class imbalance	5-8 FPS	~57.5% (COCO dataset)	High accuracy comparable to Faster R-CNN; good at detecting small and hard-to-detect objects	Slower than SSD and YOLO; high computational cost

Source: Author own

Some Real time cases in object detection:

- COCO is one of the most popular benchmarks for object detection, segmentation, and captioning, featuring over 330k images and more than 2.5 million labeled

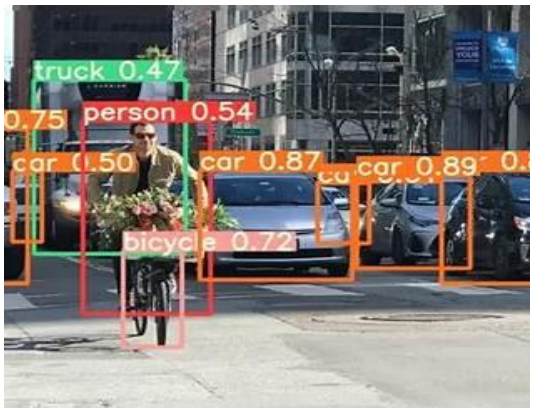
instances across 80 object categories. Recent innovations, such as Transformers and Vision Transformers (ViTs), have been tested on COCO, showcasing improved accuracy and generalization. In real-world scenarios, COCO plays a crucial role in autonomous vehicle research, helping to detect pedestrians, vehicles, and other road objects in diverse environmental conditions (Lin *et al.*, 2014).

- PASCAL VOC remains a well-established benchmark for object detection, offering datasets that span 20 object categories across 11,000 images. While COCO has become more popular, PASCAL VOC continues to be valuable for evaluating traditional CNN-based methods (Everingham *et al.*, 2010). In practical applications, PASCAL VOC contributes to smart surveillance systems by detecting objects in real-time video feeds (Everingham *et al.*, 2010).
- ADE20K is designed for both semantic segmentation and object detection, containing over 20k images with annotations for 150 object categories. It is particularly effective at detecting complex scenes and providing fine-grained object segmentation (Zhou *et al.*, 2017). This dataset plays an important role in urban planning and self-driving car technologies, helping to identify and segment objects such as traffic signs and pedestrians (Zhou *et al.*, 2017).
- The KITTI benchmark provides real-world data tailored for autonomous driving, featuring 7,500 training images and 7,000 test images with annotations for vehicles, pedestrians, and cyclists. It is essential for developing object detection, 3D localization, and tracking models (Geiger *et al.*, 2012). KITTI has been instrumental in autonomous driving research, improving road safety by detecting pedestrians, cyclists, and vehicles in real-world driving environments (Geiger *et al.*, 2012).
- The Waymo Open Dataset, a collection of high-resolution 3D LiDAR and camera data from self-driving cars, is one of the most comprehensive resources for large-scale 3D object detection tasks. This dataset is pivotal for training object detection models for autonomous vehicles, enabling the detection of pedestrians, vehicles, and road features in complex urban settings (Sun *et al.*, 2020).
- VisualWakeup focuses on detecting small objects in low-light or nighttime conditions, utilizing both visible and infrared data. It provides a valuable benchmark for evaluating real-time object detection algorithms used in autonomous systems (Wu *et al.*, 2020).

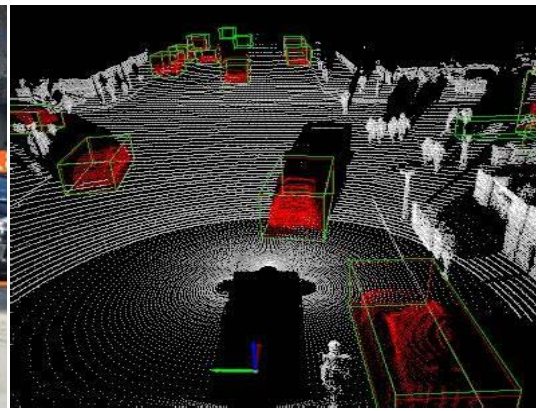
  This dataset is widely applied in security and surveillance, where it helps detect intruders, vehicles, or animals in low-light environments (Wu *et al.*, 2020).

Some Illustrious visual examples of projects using YOLO model (see Figure 4):

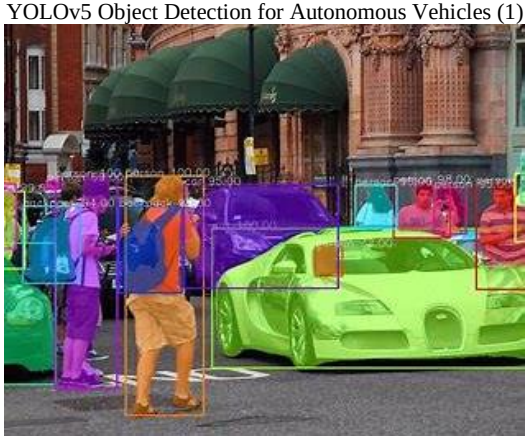
Figure 4: Diagram Showing Examples of Projects using YOLO Object Detection Mode



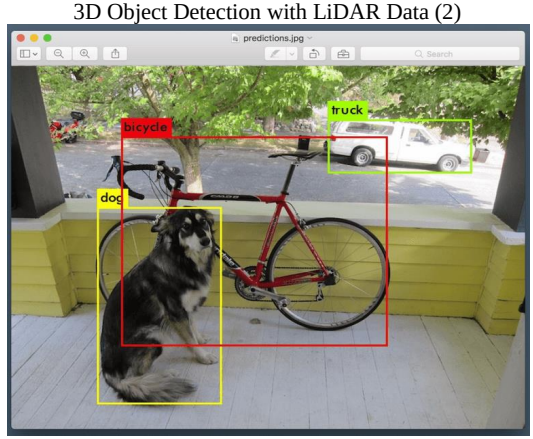
YOLOv5 Object Detection for Autonomous Vehicles (1)



3D Object Detection with LiDAR Data (2)

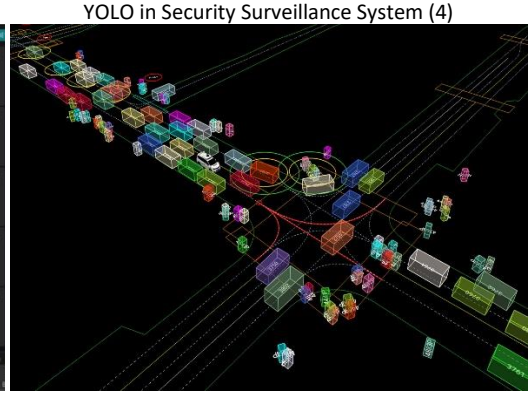


Improved Small Object Detection with CBAM & YOLO (3)



YOLO in Security Surveillance System (4)

Nighttime Detection of Vehicles and Pedestrains with YOLO and infrared Cameras (5)



3D Detection of Road Obstacles Using Waymo Open Dataset (6)

Sources: 1, 2, 3. 4. 5 & 6

4.0 Challenges faced by YOLO

Detecting tiny objects is crucial in object identification tasks, particularly in real-time systems such as YOLO and convolutional neural networks (CNNs).YOLO faces typical known challenges due to its grid-based methodology, local receptive fields, CPU resources, and power efficiency issues. It also struggles with recognizing obscured or busy backgrounds, labelling large data volumes, and dealing with domain transitions. In heterogenous settings with multiple autonomous agents, combining information from diverse viewpoints for greater detection accuracy may be challenging. Additionally, YOLO performance may suffer when objects appear in odd locations, orientations, or scales, especially for tasks with less varied training data.

YOLO is unable to recognize tiny items due to its grid-based methodology, where smaller objects may fall into the same grid cell and be disregarded by others. Traditional convolutional neural networks (CNNs) are strained by their local receptive fields, and although YOLO excels in speed, it may lose global contextual information that can be useful in detection accuracy. Although YOLO is quick, implementing the algorithm on edge devices (e.g., drones, IoT devices are still a challenge in terms of CPU resources and power efficiency. YOLO typically requires recognizing items that are obscured or present in busy backgrounds, resulting in misclassification or missing detections. Labelling vast volumes of data for object recognition can be labour-intensive and time-consuming. YOLO is restricted to 2D photos, which makes recognizing objects in 3D space, notably in autonomous driving, a difficulty. YOLO's post-processing often fails in instances when multiple objects are near to each other. YOLO, like other object detectors, suffers with domain transitions (e.g., from natural to synthetic pictures, or between different situations such as indoor and outdoor. Combining information from diverse viewpoints in settings involving many autonomous agents (e.g., drones or robots), it might be tough to achieve greater detection accuracy. YOLO performance might be prone to errors when objects appear in odd locations, orientations, or scales, notably for tasks with less varied training data.

5.0 Findings and Conclusion

Based on the discussion, certain extracts can be drawn that may include: Real-time object identification is essential for dynamic, time-sensitive applications across several domains. Its fast object interpretation and recognition lets autonomous systems to operate safely and successfully in real-world settings, making it a significant

technological tool. YOLO's primary properties, including its unified architecture, grid-based detection, real-time performance, and global context awareness, make it one of the most efficient and effective object recognition models available. Over numerous iterations (YOLOv2, YOLOv3, YOLOv4), its capabilities have grown, making it useful for many real-time computer vision workloads. YOLO is very rapid because it applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probability for each region simultaneously" (Redmon *et al.*, 2016). YOLOv3 processes images at 30 FPS, making it one of the fastest object detection systems On a modern GPU. (Redmon & Farhadi, 2018).

The YOLO algorithm divides the input image into a S x S grid, and each grid cell predicts B bounding boxes along with their associated confidence scores and class probabilities, resulting in faster processing times" (Bochkovskiy *et al.*, 2020). YOLOv4 has been able to achieve a mix of speed and accuracy, operational at real-time frame rates on modern GPUs. Some of the tiny versions may reach up to 200 FPS, making it suited for real-time applications on edge devices" (Bochkovskiy *et al.*, 2020). YOLO predicts both bounding boxes and class probabilities directly from full images in one evaluation, which makes it fast and accurate for object detection (Redmon *et al.*, 2016). Availability of anchor boxes in YOLOv2 improved detection accuracy, more particularly for small objects and objects of varying sizes (Redmon & Farhadi, 2017). Feature maps from three different layers are used in YOLOv3, to make predictions at multiple scales, which aids in improving the detection of small objects (Redmon & Farhadi, 2018). The use of CSPDarknet53 as the backbone in YOLOv4 improves feature extraction, contributing to higher accuracy while maintaining fast detection speeds" (Bochkovskiy *et al.*, 2020). The introduction of CIoU loss in YOLOv4 enhances the accuracy of bounding box regression by considering multiple factors, including the distance between centers and the aspect ratio of objects" (Bochkovskiy *et al.*, 2020). YOLOv4's mosaic data augmentation helps improve detection accuracy by providing more diverse training data and enabling the model to generalize better across different object sizes and orientations" (Bochkovskiy *et al.*, 2020).

The use of non-maximum suppression helps reduce the number of overlapping boxes and improves the precision of the final object detections" (Redmon & Farhadi, 2018). YOLOv4 can achieve a mean Average Precision (mAP) of 43.5% on the COCO dataset, which is a state-of-the-art result for real-time object detection systems (Bochkovskiy *et al.*, 2020). Small objects can be particularly challenging for detection systems because they occupy fewer pixels and lose significant spatial information during down sampling processes in CNNs (Zhang *et al.*, 2018). The problem of scale variation

poses significant challenges for small object detection, as small objects often appear similar to the background or larger objects and lose distinguishable features during feature extraction" (Liu *et al.*, 2016). YOLO's grid-based prediction method often struggles with small objects because the coarse grid cells may not adequately capture objects that occupy only a small portion of the image" (Redmon & Farhadi, 2016).

Due to the decreasing resolution of feature maps at deeper layers in CNNs, small objects can be overlooked, as they are represented by fewer feature map activations" (Lin *et al.*, 2017). Small objects are more likely to be confused with the background or occluded by other objects, leading to false positives or missed detections" (Hu *et al.*, 2019). The imbalance between large and small objects in training datasets contributes to the challenge of small object detection, as models may become biased toward larger objects" (Shrivastava *et al.*, 2016). Higher resolution images and finer grid sizes improve small object detection but increase computational cost, which can be problematic for real-time applications" (Bochkovskiy *et al.*, 2020). YOLO is the quickest real time object recognition method as compared to other comparable techniques but needs faster GPU's (See Table 1) .

6.0 Future Work

Some of the possible future research could investigate to challenge problems faced by current YOLO architecture :

1. Researchers might study enhancing YOLO's handling of smaller picture regions or adopting adaptive grid cells depending on object size.
2. Integrating transformers, which are extensively employed in vision tasks owing to their ability to represent long-range connections, can also boost YOLO's potential to gather global information and improve accuracy, particularly for complicated scenarios.
3. Lighter versions of YOLO may be developed by optimizing model parameters, utilizing approaches like quantization, pruning, or neural architecture search (NAS) to minimize complexity without reducing detection accuracy.
4. Exercising methods like multi-view detection or combining 3D object identification frameworks to forecast details such as the location and structure of obscured objects might boost performance.
5. Another technique might include context modelling, utilizing extra context-aware neural layers to better differentiate items from chaotic backgrounds.

6. Using semi-supervised or self-supervised learning strategies may reduce the need for labelled data. This may help YOLO enhance its generalization on unknown datasets and adapt to varied contexts (e.g., new weather conditions or illumination.
7. Utilizing alternative post-processing methods such as Soft-NMS or Distance-IoU-NMS may substantially reduce concerns of mis-suppression. Researchers could also examine learning-based strategies for dynamic NMS thresholds built on item density and size in a structured sequence.
8. Discovering new domain adaptation strategies, such as adversarial training, to enhance YOLO's resilience across multiple datasets. This might imply matching feature distributions across domains or applying generative models to bridge gaps in visual appearance.
9. Using collaborative object detection, where many instances of YOLO operating on distinct agents exchange visual input and execute synchronized detections, could boost performance in complex, dynamic situations.
10. Researching more complex data augmentation methods, such as MixUp, CutMix, or generative adversarial networks (GANs) for synthetic data creation, may increase YOLO's resilience. This would facilitate in providing different training circumstances that can better resemble real-world scenarios.

In general, object detection algorithms must be able to integrate some new models into its existing architecture. Self-Supervised Learning can minimize the need for labelled data by utilizing unlabelled data through pretext tasks such as contrastive learning and generative tasks. This method enhances feature extraction and transferability, making object detection more cost-effective and adaptable across diverse applications (Roman, 2024). Using XAI techniques can make it easier to understand and refine how models make decisions. For example, applying XAI to adjust synthetic data can boost model performance, particularly in cases where real-world data is limited (Mital *et al.*, 2024).

Incorporating attention mechanisms like the Convolutional Block Attention Module (CBAM) can improve the accuracy of detecting small objects. This approach is especially beneficial for models like YOLOv5, helping them become more precise when identifying smaller objects (Wang *et al.*, 2024). Expanding the model's receptive field enables it to harness contextual information more effectively. By incorporating modules like the Receptive Field Block (RFB) or Patch Expanding Layer, the model can enhance feature extraction from its initial layers, leading to richer and more precise insights (Wang *et al.*, 2024). Region Proposal Networks (RPN) can enhance the model's ability to perceive larger objects, improving its overall precision in detecting objects of all sizes.

This approach allows the model to better adapt to a wide range of object scales, ensuring more accurate detections (Wang *et al.*, 2024).

For complex environments, data from multiple modalities such as visual, infrared, can be integrated using LiDAR (Light Detection and Ranging) which focuses on higher accuracy and robustness of object detection algorithms Such approach is especially valuable, where relying on just one type of data may not provide the full picture. Creating algorithms capable of detecting objects with just a few training samples per class helps tackle the challenge of rare objects. Approaches like semi-supervised learning and hierarchical ternary classification can significantly enhance the performance of Few-Shot Object Detection (FSOD) (Shangguan & Rostami, 2023). Enhancing the model's ability to adapt across diverse domains—such as transitioning from day to night or from urban to rural environments strengthens its generalization and overall robustness. Therefore, Future research in YOLO can focus on enhancing its detection capabilities, expanding its application in 3D environments, improving small object detection, and increasing its computational efficiency for deployment on edge devices. These efforts could lead to significant advances in real-time object detection, especially in challenging applications like autonomous driving, robotics, and security systems. uture research could focus on improving YOLO's ability to detect objects with higher precision, especially in cluttered and complex environments. One area of focus could be the integration of **Vision Transformers (ViTs)** and attention mechanisms, which have been shown to enhance object detection performance by better capturing global contextual information (Dosovitskiy & Fischer, 2015; Vaswani *et al.*, 2017).

Additionally, advancements in hybrid models that combine CNNs with Transformer-based architectures can further push the detection capabilities of YOLO in challenging conditions YOLO's application could be extended to 3D object detection tasks, which is especially critical in autonomous driving and robotics. Recent approaches like **PointCloud-based YOLO** for LiDAR data have shown promise in 3D object detection (Shi & Zhang, 2020). These methods allow YOLO to detect objects in 3D space, facilitating its deployment in dynamic, real-world environments, where depth information is essential for accurate object localization. Small object detection is a long-standing challenge for YOLO and other object detection models due to the scale imbalance between large and small objects in images.

One potential solution is the integration of **multi-scale feature extraction** techniques, such as the **Feature Pyramid Networks (FPNs)**, which help the model focus on small objects by utilizing features from multiple scales (Lin *et al.*, 2017). Additionally, attention mechanisms like the **Convolutional Block Attention Module**

(CBAM) can enhance the detection of small objects by improving the model's ability to focus on relevant features in the image (Woo *et al.*, 2018). To make YOLO viable for real-time object detection on edge devices, reducing its computational footprint without sacrificing accuracy is crucial. One promising approach is model pruning and **quantization**, which have been shown to reduce the model size and improve inference speed, making it suitable for deployment on devices with limited resources (Cheng *et al.*, 2017). Additionally, research into **lightweight architectures**, such as YOLO-Nano, can help enhance efficiency while maintaining high detection accuracy (Bansal *et al.*, 2020).

References

Azuma, R. T. (2016). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385.

Bansal, A., Soni, P., & Shankar, R. (2020). YOLO-Nano: A lightweight object detection model for mobile devices. Retrieved from https://arxiv.org/abs/200 5.07225

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Cheng, S., Wang, Y., & Li, H. (2017). Model compression: A survey. *ACM Computing Surveys (CSUR), 50*(3), 1-35.

Dosovitskiy, A., & Fischer, P. (2015). Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(9), 1734-1747.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature, 542*(7639), 115-118.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303-338.

Fang, W., Wang, L., & Ren, P. (2019). Tinier-YOLO: A real-time object detection method for constrained environments. *IEEE Access, 8*, 1935-1944.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. *In 2012 IEEE conference on computer vision and pattern recognition* (pp. 3354-3361). Retrieved from https://doi.org/10.1109/CVPR.2012.6248074

Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2019). Relation networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3588-3597.

Huang, T., & Russell, S. (1997, August). Object identification in a bayesian context. *IJCAI*, *97*, 1276-1282.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of Yolo algorithm developments. *Procedia Computer Science*, *199*, 1066-1073.

Kanimozhi, S., Gayathri, G., & Mala, T. (2019, February). Multiple real-time object identification using single shot multi-box detection. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)* (pp. 1-5). IEEE.

KR, S. C. (2017, April). Real time object identification using deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (pp. 1801-1805). IEEE.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., & Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, *37*(4-5), 421-436. Retrieved from DOI: 10.1177/0278364917710318.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936-944.

Lin, T.-Y., Goyal, P., & Girshick, R. (2017). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(2), 318-327. Retrieved from https://doi.org/10.1109/TPAMI.2017.2663821.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Dollár, P. (2014). Microsoft COCO: Common objects in context. *European Conference on Computer Vision* (pp. 740-755). Springer. Retrieved from https://doi.org/10.1007/978-3-319-10602-1_48

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, 21-37. Retrieved from DOI: 10.1007/978-3-319-46448-0_2

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., ... & Wen, S. (2020). PP-YOLO: An effective and efficient implementation of object detector. Retrieved from https://doi.org/10.48550/arXiv.2007.12099

Mital, N., Malzard, S., Walters, R., De Melo, C. M., Rao, R., & Nockles, V. (2024). Improving object detection by modifying synthetic data with explainable AI. Retrieved from https://doi.org/10.48550/arXiv.2412.01477

Piccinini, P., Prati, A., & Cucchiara, R. (2012). Real-time object detection and localization with SIFT-based clustering. *Image and Vision Computing, 30*(8), 573-587.

Redmon, J., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. DOI: 10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263-7271.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. Retrieved from https://doi.org/10.48550/arXiv.1804.02767

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. Retrieved from DOI: 10.1109/CVPR.2016.91

Roman, M. (2024). Enhancing object detection with self-supervised learning: improving object detection algorithms using unlabeled data through self-supervised techniques. Roman Publishing. Retrieved from https://romanpub.com/resources/Vol%205%20%2C%20No%201%20-%2023.pdf

Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. Retrieved from https://doi.org/10.48550/arXiv.1709.05943

Shangguan, Z., & Rostami, M. (2023). Identification of novel classes for improving few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 3356–3365). Retrieved from https://doi.org/10.1109/ICCVW54120.2023.00410

Shi, S., & Zhang, J. (2020). PointRCNN: 3D object proposal generation and detection from point cloud. Retrieved from https://arxiv.org/abs/1812.04256.

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 761-769.

Sun, L., Lee, W., & Wilson, S. (2020). The Waymo Open Dataset: Large-scale autonomous driving data for object detection, tracking, and prediction. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9789-9797). Retrieved from https://doi.org/10.1109/CVPR42 600.2020.00980

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture, 157*, 417-426.

Vaswani, A., Shazeer, N., & Parmar, N. (2017). Attention is all you need. *In Advances in neural information processing systems* (pp. 5998-6008). Retrieved from https://arxiv.org/abs/1706.03762.

Wang, Y., Yan, X., Zhang, K., Gong, L., Xie, H., Wang, F. L., & Wei, M. (2022). *TogetherNet: Bridging image restoration and object detection together via dynamic enhancement learning*. https://doi.org/10.48550/arXiv.2209.01373

Wang, Z., Men, S., Bai, Y., Yuan, Y., Wang, J., Wang, K., & Zhang, L. (2024). Improved small object detection algorithm CRL-YOLOv5. *Sensors*, 24(19), 6437. Retrieved from https://doi.org/10.3390/s24196437

Woo, S., Park, J., & Lee, J. Y. (2018). CBAM: Convolutional Block Attention Module. *In Proceedings of the European Conference on Computer Vision* (pp. 3-19). Retrieved from https://doi.org/10.1007/978-3-030-01234-2_1

Wu, W., Zhang, Z., & Wang, X. (2020). VisualWakeup: Detecting small objects in low-light and night-time images using multi-modal fusion. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4609-4617). Retrieved from https://doi.org/10.1109/CVPR42600.2020.00465

Xailient. (n.d.). *Detectum: Faster than any other cutting-edge object detector model*. Retrieved December 22, 2024, from https://xailient.com/blog/detectum-faster-than-any-other-cutting-edge-object-detector-model/

Zhang, S., Chi, C., Yao, Y., Lei, Z., & Li, S. Z. (2018). Detecting small objects in object detection: A survey. arXiv preprint arXiv:1904.00304.

Zhang, S., Zhang, S., Huang, W., Li, M., & Qiao, B. (2019). Deep learning-based object detection for autonomous driving: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(7), 3212-3232. Retrieved from DOI: 10.1109/TNNLS.2018.2876865.

Zhou, B., Zhao, H., & Xie, J. (2017). Scene parsing through ADE20K dataset. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1-12). Retrieved from https://doi.org/10.1109/CVPR.2017.7399 384

Web links

https://iot-automotive.news/wp-content/uploads/2021/03/WAYMO-OPEN-DATASET_Large.jpg
https://tse1.mm.bing.net/th?id=OIP.r9LWI1NuwgPqDTodcooUKQHaFB&w=321&h=321&c=7

https://tse2.mm.bing.net/th?id=OIP.rl-dNKSsdB2QQN6dZGJqDQAAAA&w=400&h=400&c=7
https://tse4.mm.bing.net/th?id=OIP.7HePJEj3OsOub7wKl9lbMQHaE9&w=317&h=317&c=7
https://videos.cctvcamerapros.com/v/car-detection-ai-security-camera.html
https://www.aiophotoz.com/photos/yolo-real-time-object-detection-explained.html