A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms

[1*]Oluwaseyi Ezekiel Olorunshola, [2]Martins Ekata Irhebhude and [3]Abraham Eseoghene Evwiekpaefe

[1]Electrical and Electronics Engineering Department, Air Force Institute of Technology, Kaduna,
[2,3] Computer Science Department, Nigerian Defence Academy, Kaduna, Nigeria

email: [1*]seyisola25@yahoo.com, [2]mirhebhude@nda.edu.ng, [3]aeevwiekpaefe@nda.edu.ng

**Corresponding author*

Received: 12 October 2022 | Accepted: 22 December 2022 | Early access: 08 February 2023

Abstract - *This paper presents a comparative analysis of the widely accepted YOLOv5 and the latest version of YOLO which is YOLOv7. Experiments were carried out by training a custom model with both YOLOv5 and YOLOv7 independently in order to consider which one of the two performs better in terms of precision, recall, mAP@0.5 and mAP@0.5:0.95. The dataset used in the experiment is a custom dataset for Remote Weapon Station which consists of 9,779 images containing 21,561 annotations of four classes gotten from Google Open Images Dataset, Roboflow Public Dataset and locally sourced dataset. The four classes are Persons, Handguns, Rifles and Knives. The experimental results of YOLOv7 were precision score of 52.8%, recall value of 56.4%, mAP@0.5 of 51.5% and mAP@0.5:0.95 of 31.5% while that of YOLOv5 were precision score of 62.6%, recall value of 53.4%, mAP@0.5 of 55.3% and mAP@0.5:0.95 of 34.2%. It was observed from the experiment conducted that YOLOv5 gave a better result than YOLOv7 in terms of precision, mAP@0.5 and mAP@0.5:0.95 overall while YOLOv7 has a higher recall value during testing than YOLOv5. YOLOv5 records 4.0% increase in accuracy compared to YOLOv7.*

Keywords: YOLOv5, YOLOv7, Object detection, Computer Vision, Detection Algorithm.

1 Introduction

There are several object detection algorithms such as Single Shot Detector (SSD), Region-based Convolutional Neural Network (R-CNN), and Fast Region-based Convolutional Neural Network (Fast R-CNN) (Padilla et al., 2021). In 2015, a researcher, Joseph Redmon, and colleagues introduced an object detection system that performed all the essential stages to detect an object using a single neural network. You Only Look Once (YOLO) is an object detection algorithm that detects various objects in a picture. It was founded in 2016. It reframes the object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. This unified model predicts multiple bounding boxes and class probabilities simultaneously for those objects covered by boxes. At the time of its release, YOLO algorithm has produced impressive specifications that outstood the premier algorithms in terms of both speed and accuracy for detecting and determining object coordinates (Redmon, et al., 2016).

The base YOLO model processes images in real-time at 45 frames per second (FPS). A smaller version of the network: Fast YOLO, processes an astounding 155 FPS while still achieving double the mean Average Precision (mAP) of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background (Redmon et al., 2016).

YOLO algorithm can be used in wildlife, drones, military, autonomous driving, hospital, other Computer Vision (CV) tasks etc. (Górriz et al., 2020). Over the years, YOLO has developed many other variants such as YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6 and YOLOv7. However, there is need to evaluate which of the YOLO algorithms performs best of all the YOLO versions. From previous works, it was found that YOLOv5 performs better than previous YOLO versions (YOLOv3 and YOLOv4) in terms of accuracy and speed (Sahal, 2021; Ramya, et al., 2021) and the newly released version of YOLO which is YOLOv7 is also very performant. Hence, this study evaluated YOLOv5 and YOLOv7.

The rest of this paper is presented as follows; Section 2 briefly highlights the background of the YOLO, Section 3 reviews related works involving YOLOv5 and YOLOv7. Section 4 contains the methodology. Section 5 analyses and discusses the results, and finally, Section 6 details the conclusion of this study.

2 Background of YOLO

Redmon et al. (2016) presented YOLO, a new approach to object detection. The YOLO design enables end-to-end training and real-time speeds while maintaining high average precision. The system divides the input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Confidence is defined as the measure of the predicted object and the ground truth object. If no object exists in that cell, the confidence scores should be zero. Otherwise the confidence score should equal to the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The w and h are the width and height that are predicted relative to the whole image. Finally, the confidence prediction represents the IOU between the predicted box and the ground truth box.

The network architecture is inspired by the GoogLeNet model for image classification (Redmon et al., 2016). The network has 24 convolutional layers followed by 2 fully connected layers. Instead of the inception modules used by GoogLeNet, 1 × 1 reduction layers is utilized with a 3 × 3 convolutional layers. The YOLO architecture is shown in Figure 1.

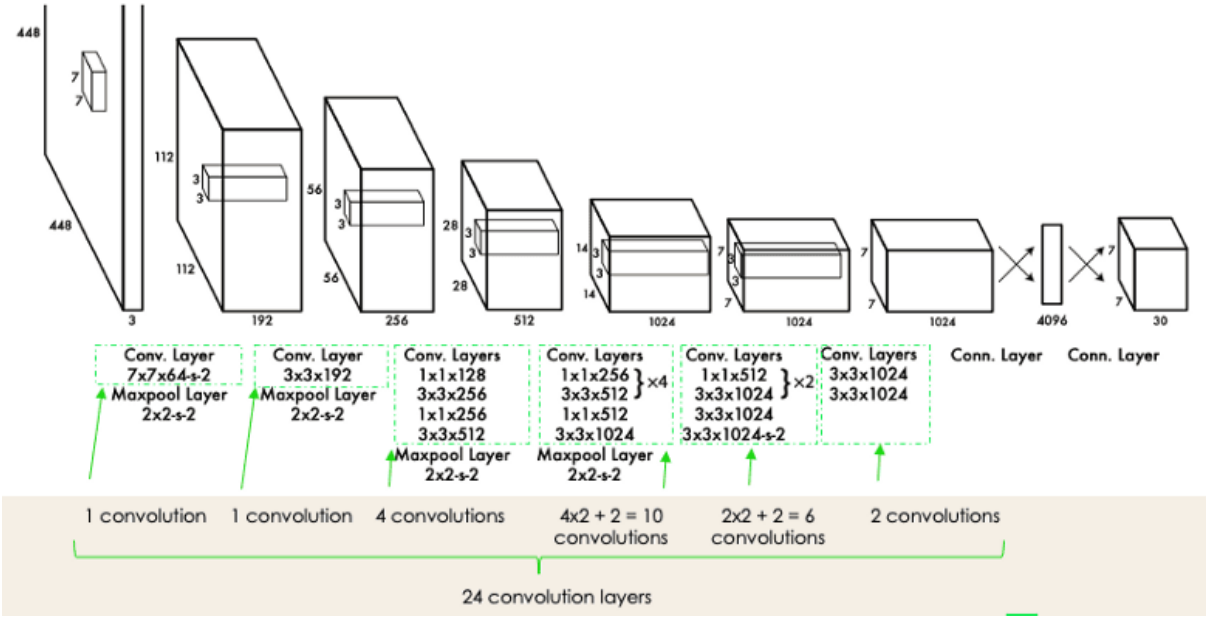


Figure 1: YOLO Architecture (Redmon et al., 2016)

Further improvements were made to the YOLO architecture as more research was done to improve detections by implementations of techniques and methods to improve accuracy, reduce the size of the network and offer faster detections. Such improvements are summarized in Table 1, while YOLOv5 and YOLOv7 are further discussed in the following subsections.

2.1 Improvements on YOLO Versions

From the introduction of the YOLO, there have been various changes and improvements which resulted in several versions of YOLO from YOLOv1 to YOLOv7. The findings on the YOLO versions are summarized in Table 1.

Table 1: Summary of Improvements on YOLO

S/N	YOLO Variant	Improvement	Results
1.	YOLOv1 (Redmon et al., 2016)	Single shot detector combines and solves the problem of drawing boundary boxes and class identification	Higher accuracy and speed compared to two-stage object detector such as Faster R-CNN
2.	YOLOv2 (Redmon & Farhadi, 2018)	Iterative improvements on Batch Normalization, higher resolution detection and use of anchor boxes	Reduction in architecture, faster detection and higher accuracy and better detection of high resolution images
3.	YOLOv3 (Redmon & Farhadi, 2018)	Addition of objectness score to bounding box prediction, added connections to the backbone network layers and predictions at three separate granularities.	Improves detection of smaller objects
4.	YOLOv4 (Alexey et al., 2020)	Improved feature aggregations, bag of freebies with mosaic augmentations and use of mish activation	Achieved improved accuracy and ease of training, high quality performance and accessibility
5.	YOLOv5 (Nepal & Eslamiat, 2022)	Reduced network parameters, use of Cross Stage Partial Network (CSPNet) for the head, PANet for the neck of the architecture, residual structure and auto-anchor. It also utilizes mosaic augmentations.	Extremely easy to train, inference on individual, batch images, video feed and webcam ports. Ease of transfer and use of weights. Faster and more lightweight than previous YOLO.
6.	YOLOv6 (Chuyi et al., 2022)	Redesigned network backbone and neck to EfficientRep Backbone and Rep-PAN Neck. The Network head is decoupled separating different features from the final head	Improvement in detecting small objects, anchor free training of model. Less stable and flexible as compared to YOLOv5.
7.	YOLOv7 (Wang et al., 2022)	Layer aggregation using E-ELAN, trainable bag of freebies, 35% fewer network parameters. Model scaling for concatenation-based model	Increase in speed and accuracy, ease of training and inference.

According to Nepal and Eslamiat (2022), the main differences between YOLOv1, YOLOv2, YOLOv3, YOLOv4, and YOLOv5 architecture are that YOLOv1 uses the softmax function, and YOLOv2 has higher resolution classifier, higher accuracy, and higher efficiency than YOLOv1. This is because batch normalization layer is added to the CNN of YOLOv2. YOLOv3 uses Darknet53 as its main backbone to extract features from the input image which has a better efficiency and detection performance. In YOLOv3, there is multi-object classification i.e. objects may belong to multiple categories at the same time. YOLOv3 replaces softmax function with an independent logistics function to calculate the probability that the input image belongs to a specific label and also YOLOv3 uses the 2-class entropy loss for each category thereby reducing the computational complexity brought about by softmax functions. YOLOv4 architecture uses CSPDarknet53 as a backbone which is a combination of Darknet53 and CSP network. YOLOv4 has higher accuracy, higher efficiency for object detection and also reduced hardware requirements. YOLOv5 uses Focus structure with CSPDarknet53 as a backbone. The Focus layer is first introduced in YOLOv5. The Focus layer replaces the first three layers in the YOLOv3 algorithm. The advantage of using a Focus layer is reduced required Compute Unified Device Architecture (CUDA) memory, reduced layer, increased forward propagation, and back propagation. YOLOv5 is extremely fast and is nearly 90% smaller (lighter) than YOLOv4. YOLOv6 has many improvements in backbone, neck, head and training strategies; YOLOv6 uses RepVGG Style structure, EfficientRep Backbone, Rep-PAN Anchor-free paradigm, SimOTA algorithm and SIoU bounding box regression loss function, while YOLOv7 exceeds all known object detectors in both speed and accuracy in the range from 5 FPS to 160 FPS, and has the highest accuracy 56.8% AP among all known real-time object detectors with 30 FPS or higher on GPU V100. YOLOv7 greatly improved real time object detection accuracy without increasing the inference cost, it reduced about 40% parameters and 50%

computation of state-of-the-art real-time object detector, and has faster inference speed and higher detection accuracy.

2.2 YOLOv5

A month after YOLOv4 was released, a researcher named Glenn, and his team, published a new version of the YOLO family, called YOLOv5. According to Nepal and Eslamiat (2022), YOLOv5 is different from the previous releases in that YOLOv5 utilizes PyTorch instead of Darknet. It utilizes CSPDarknet53 as backbone. It uses Path aggregation network (PANet) as neck to boost the information flow. PANet adopts a new feature pyramid network (FPN) that includes several bottom ups and top down layers. This improves the propagation of low level features in the model. PANet improves the localization in lower layers, which enhances the localization accuracy of the object. In addition, the head in YOLOv5 is the same as YOLOv4 and YOLOv3 which generates three different output of feature maps to achieve multi scale prediction. YOLOv5 model can be summarized as follows: Backbone: Focus structure, CSP network, Neck: SPP block, PANet, Head: YOLOv3 head using GIoU-loss. The improvement of YOLOv5 over YOLOv4 was utilization of CSPDarknet53 which solved the problem of repetitive gradient information found in YOLOv4 and YOLOv3 thereby reducing the network parameters and reducing inference speed while accuracy is increased. The architecture of YOLOv5 is shown in Figure 2.

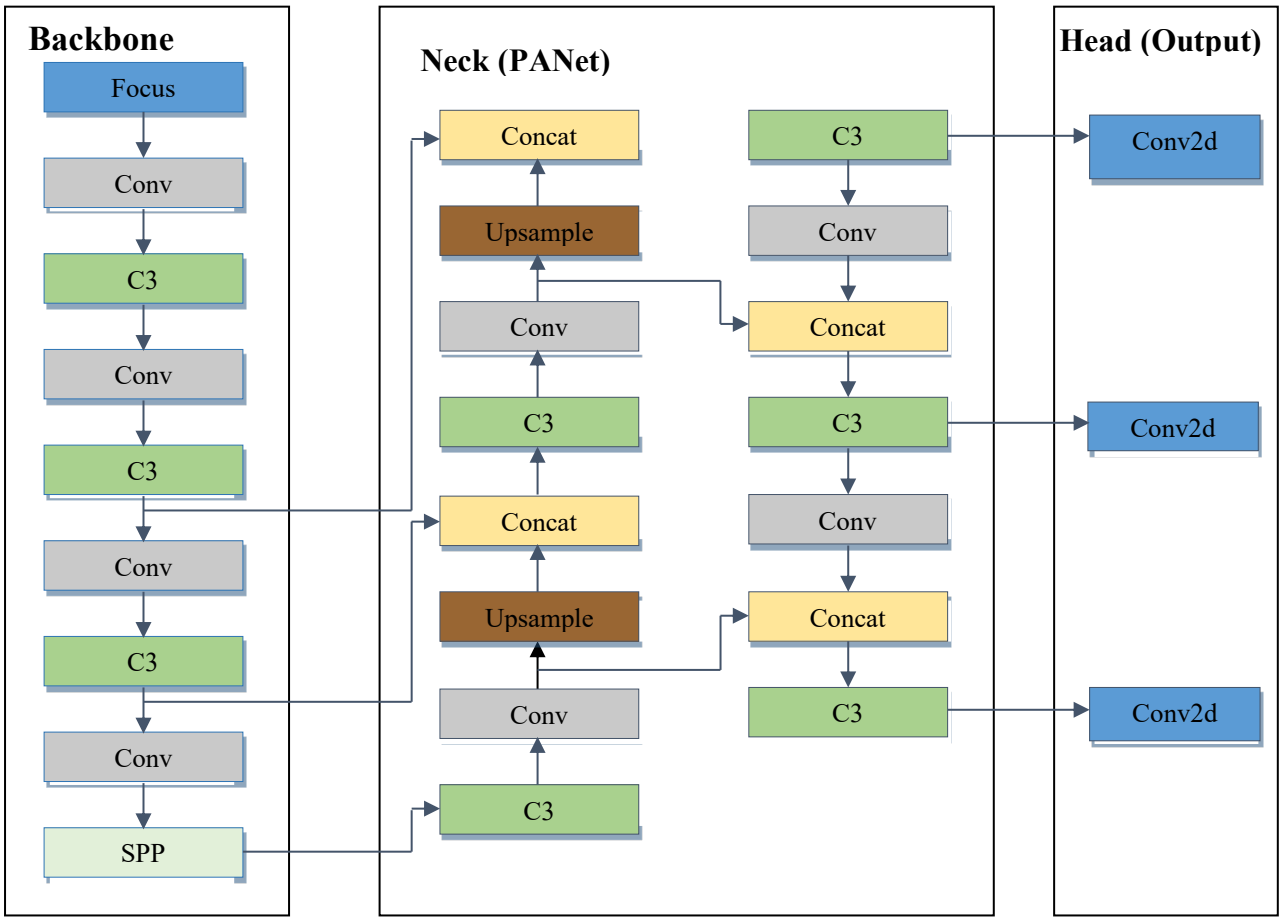


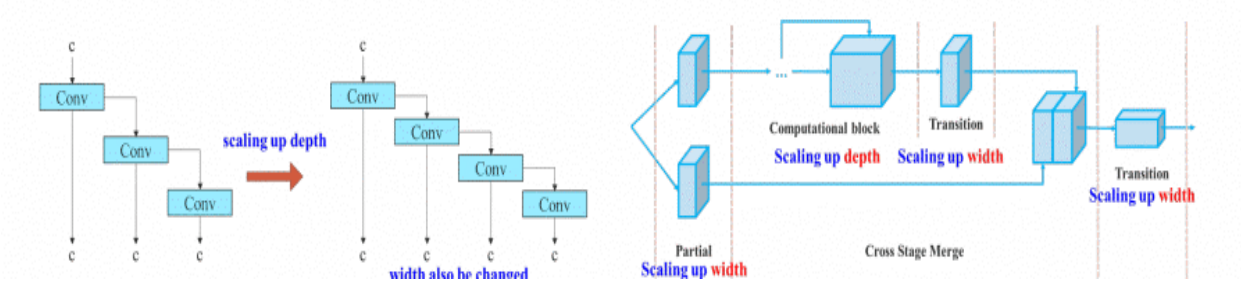
Figure 2: YOLOv5 Architecture (Nepal & Eslamiat, 2022)

2.3 YOLOv7

YOLOv7 is the latest version of the YOLO at the time of this research. YOLOv7 is a real-time object detector currently revolutionizing the CV industry with its incredible features. YOLOv7 was trained only on MS COCO dataset from scratch without using any other datasets or pre-trained weights (Wang et al., 2022). Wang et al. (2022) stated that YOLOv7 surpasses all known object detectors in both speed and accuracy in the range from 5 FPS to 160 FPS, and has the highest accuracy at 56.8% AP among all known real-time object detectors with 30 FPS or higher on Graphics Processing Units (GPU) V100. YOLOv7 greatly improved real time object detection

accuracy without increasing the inference cost; it reduced about 40% parameters and 50% computation of state-of-the-art real-time object detector, and has faster inference speed and higher detection accuracy.

YOLOv7 has extended efficient layer aggregation networks (E-ELAN). E-ELAN uses expand, shuffle, and merge cardinality to achieve the ability to continuously enhance the learning ability of the network without destroying the original gradient path (Wang et al. 2022). E-ELAN only changes the architecture in computational block, while the architecture of transition layer is completely unchanged. In addition to maintaining the original E-LAN design architecture, E-ELAN also guides different groups of computational blocks to learn more diverse features. YOLOv7 also has model scaling for concatenation-based models. The main purpose of model scaling is to adjust some attributes of the model and generate models of different scales to meet the needs of different inference speeds. Proposed compound scaling method can maintain the properties that the model had at the initial design and maintains the optimal structure. Figure 3 is the Model scaling for concatenation-based models of YOLOv7.



(a) concatenation-based model (b) scaled-up concatenation-based model (c) compound scaled up depth and width for concatenation-based model

Figure 3: Model Scaling of YOLOv7 (Wang et al., 2022)

From (a) to (b), it is observed that when depth scaling is performed on concatenation-based models, the output width of a computational block also increases. This phenomenon will cause the input width of the subsequent transmission layer to increase. Therefore, (c) is proposed, that is, when performing model scaling on concatenation-based models, only the depth in a computational block needs to be scaled, and the remaining of transmission layer is performed with corresponding width scaling.

YOLOv6 also offers great improvements in terms of detection but lacks scalability and ease of training when compared with YOLOv5 and YOLOv7. Also, YOLOv6 performs more accurately when used for single image inference compared to multiple image inference accuracy offered by YOLOv5 and YOLOv7 (Banerjee, 2022). As a result of this, experiment was conducted with YOLOv5 and YOLOv7 as they fit in for the multiple object detection, providing ease in customizing the training and running of inference.

3 Literature Review

Kasper-Eulaers et al. (2021) studied how YOLOv5 can be implemented to detect heavy goods vehicles at rest areas during winter to allow for the real-time prediction of parking spot occupancy. The model was trained using Google Colaboratory (Colab), which provides free access to powerful GPUs and requires no configuration. A notebook was developed by Roboflow.ai which is based on YOLOv5 and uses pre-trained COCO weight. The model improved swiftly in terms of precision, recall and mean average precision before overfitting after about 150 epochs. The box, objectness and classification losses of the validation data also showed a rapid decline until around epoch 15. Results show that the trained algorithm can detect the front cabin of heavy goods vehicles with high confidence, while detecting the rear seems more difficult, especially when located far away from the camera.

Malta et al. (2021) proposed a model of a task assistant based on a deep learning neural network. A YOLOv5 network was used for recognizing some of the constituent parts of an automobile. The dataset created consisted of 582 images taken from three videos with similar lighting conditions, where it was possible to identify a total of eight different types of parts: oil dipstick; battery; engine oil reservoir; wiper water tank; air filter; brakes fluid reservoir; coolant reservoir; and power steering reservoir. The images taken from each frame were converted to a 416 × 416 format, which is the format that the chosen architecture needs to use as input. The hardware used during development included computers, for running software, and cell phones, for capturing videos and pictures. The object detection model was trained using laptop computer with access to a Google Colab virtual machine. The precision obtained for the two models (YOLOv5s and YOLOv5m) was in line with that obtained by other authors

for similar problems. YOLOv5s demonstrated to be capable of identifying eight different mechanical parts in a car engine with high precision and recall always above 96.8% in the test sets, which, compared to the larger model, has almost the same results. Results proved that the network is good and fast enough to be applied to the task of assisting in recognizing constituent parts of an automobile.

Wan et al. (2021) proposed a YOLOv5 model based on a self-attention mechanism for polyp target detection. Mosaic method was used in the data preprocessing stage to enhance the amount of training data in the data set, Cross Stage Partial Networks (CSPNet) was used as the backbone network to extract the information features in the image, which solved the problem of gradient disappearance, and the feature pyramid architecture with attention mechanisms was used to enhance the detection performance of varying-size polyps. The proposed method was trained by stochastic gradient descent (SGD) and backpropagation in an end-to-end way on a cloud-computing platform configured with eight 16 GB GPUs, a 16-core CPU, and a 64 GB memory. YOLOv5 used spatial pyramid pooling (SPP) to enhance the model's detection of objects with different scales, Path aggregation network (PANET) as the neck for feature aggregation and new Feature Pyramid Networks (FPN) structure that enhanced the bottom-up path, which improved the propagation of low-level features. The author's method achieved excellent performance. In the Kvasir-SEG data set, the precision was 0.915, the recall rate was 0.899 and the F-score was 0.907. In the WCY data set, the precision was 0.913, the recall was 0.921 and the F-score was 0.917. Specifically, this method used full-image information when predicting the target window using each network, which greatly reduced the false positive rate.

Yao et al. (2021) developed a defect detection model based on YOLOv5, which is able to detect defects accurately, and at a fast speed. A small object detection layer was added to improve the model's ability to detect small defects. Squeeze-and-Excitation (SE) Layer and the loss function complete intersection over union (CIoU) were introduced to make the regression more accurate. The model was trained based on transfer learning and used the Cosine Annealing algorithm to improve the effect. The mAP@0.5 of YOLOv5 reached 94.7%, which was an improvement of nearly 9%, compared to the original algorithm.

Jia et al. (2021) introduced a real-time end-to-end helmet detection of motorcyclists method based on YOLOv5 algorithm. The original anchor box size of YOLOv5 is calculated by the K-means algorithm in the COCO dataset, and the HFUT-MH dataset proposed is quite different from the COCO dataset. This method achieved mAP of 97.7%, F1-score of 92.7% and frames per second of 63, which outperformed other state-of-the-art detection methods.

In Liu et al. (2021), a real time railway signal lights detection based on YOLOv5 was introduced. Experiments were conducted to prove the effectiveness of the proposed method. A dataset consisting of subway scenes with signal lights was constructed, and trained with YOLOv5 model. The signal lights detection model trained by YOLOv5s has an average recall rate and accuracy of 0.972, while running speed reached 100 FPS.

Patel et al. (2022) proposed an ensemble model for translated infrared images. The model uses advanced deep learning models which are pix2pix Generative Adversarial Networks (GAN) and YOLOv7 on the LLVIP dataset which contains visible-infrared image pairs for low light vision. The dataset amounting to 33672 images mostly captured in dark scenes and tightly synchronized with time and location. The model was able to outperform models trained with just images against the translated images in all aspects especially conditions of low light. The model has higher precision, recall, mAP@0.5 and mAP@0.5:0.95 for translated images than for visible images which was graphically represented.

Dima et al. (2021) proposed a YOLOv5 based solution because of its lightweight, good speed and accuracy. MU HandImages ASL, which is a benchmark dataset, was used to train and evaluate the model. The data set contains 2515 close-up, colored images. Authors achieved 95% precision, 97% recall, 98% map@0.5, and 98% map@0.5:0.95 score which is adequate to recognize the gesture in real-time. The achieved results, even with a relatively small data set, are on average 0.98 F1 scores in the identification of 36 distinct classes of ASL. This result indicates a good potential for using YOLOv5 to recognize the ASL dataset.

Hao et al. (2021) proposed a lightweight algorithm which improves YOLOv5 in both speed and accuracy. The model was experimented with dataset containing fire scenarios and shows that the Light-YOLOv5 improves mAP by 3.3% and achieves FPS of 91.1. Compared with YOLOv7-tiny, the mAP of the improved model was 6.8% higher, which shows how effective the algorithm is.

Cengil and Cinar's (2021) study aimed to identify poisonous mushrooms. YOLOv5 was used in real-time applications due to its speed and high accuracy rate. It is also good at finding small objects. Mean Average

Precision of all classes was 0.77 and AP values of each class were 0.818 for Autumn Skullcap, 0.825 for Destroying Angle, 0.610 for Cococybe Filaris, 0.737 for Deadly Dapperling, 0.826 for Death Cap, 0.854 for Podostroma Cornu-Damae, 0.993 for Fly Agaric, 0.556 for Webcaps. The experimental results showed that with the image data used, a high success rate was achieved.

Yang et al. (2022) added YOLOv7 as object detection network to DeepSORT tracking algorithm to get YOLOv7-DeepSORT detection by tracking model. The evaluation metrics used in the experiment were: Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Identity F1 Score (IDF1), Number of Identity Switches (IDs), Mostly Lost Targets (ML), Mostly Tracked Targets (MT), False Positive (FP), and False Negative (FN). Experiments showed that YOLOv7 gave higher scores than YOLOv5 in MOTA, MOTP and IDF1. For IDs, ML, MT, FP and FN, it was reported that YOLOv5 is better than YOLOv7.

Hussain et al. (2022) presented a CV-based autonomous rack inspection framework centered around YOLOv7 architecture to solve the problem of manual process results in operational down-time as well as inspection and certification costs and undiscovered damage due to human error. Additionally, the authors proposed a domain variance modeling mechanism for addressing the issue of data scarcity through the generation of representative data samples. The proposed framework achieved a mAP of 91.1%.

YOLOv7 is a recent model of the YOLO variant. From the reviews, authors that have conducted comparison on the use of YOLOv7 reported significant improvement in performance when compared with other variants. To investigate and clearly confirm the improvement and performance of the YOLOv7, an experiment was conducted to compare and clearly identify the performance of both models: YOLOv5 and YOLOv7, for effective training and subsequent application as the detection model. Table 2 contains the summary of the literature reviewed in the course of this research.

Table 2: Summary of Literature Review

Authors/Date	Methodology	Analysis & Results	Conclusions
Kasper-Eulaers et al. (2021)	Real Time detection of heavy-duty vehicles for occupancy of parking spot suing YOLOv5	Improvement on precision, recall and mAP in detection. The model was able to detect front cabin with high confidence but fails when located at a far distance.	The model shows improvement on close range detection, but at a far distance, shows poor detection.
Malta et al. (2021)	Recognition of different constituents of a car using YOLOv5 series.	The model was able to achieve precision and recall of 96.8% in detection of the eight parts of the car engine used in the experiment compared with larger models used for the same purpose.	The experiment compares that the smaller models could perform in terms of precision and recall a high value compared with larger models while still offering speed due to size of the model.
Wan et al. (2021)	Experiment using YOLOv5 model for self-attention mechanism for polyp target detection.	The model was able to achieve excellent recall, precision and accuracy of above 90% due to the use of full image information during prediction in each network.	The use of full image information in each network of a model can enhance the model in detection and lower the rate of false positives.
Yao et al. (2021)	Experiment of defect detection using YOLOv5 with addition	The model achieved a mAP@0.5 of 94.7% which improved the model	Addition of a small detection layer like the SE layer can improve the model detection while also offering faster detection.

	of a Squeeze-and-Excitation (SE) layer.	by 9% compared to the original model	
Jia et al. (2021)	A real-time end to end helmet detection of motorcyclists using YOLOv5 using K-means algorithm to calculate the anchors	The model achieved 97.7% mAP and 92.7% F1 scores which outperforms other state-of-the-art models	Use of well calculated anchors can greatly improve model detection.
Patel et al. (2022)	An ensemble model proposed for translated infrared images using pix2pix, GAN and YOLOv7 on visible infrared image pairs for low light visiowhns	The model performed better when compared with just images used for training as against the model use of translated images especially on low light conditions.	The use of translated images can improve the accuracy of a model especially on low light condition and also ensemble model of good algorithms can also perform better than an original detection model algorithm
Hao et al. (2021)	Experiment with YOLOv5 combined with a light weight algorithm was compared with YOLOv7-tiny in fire scenarios	The model improved by 6.8% compared with YOLOv7-tiny with improvements on mAP and FPS.	Light weight algorithms can prove to be effective in speed and accuracy when combined with state-of-the-art algorithms
Yang et al. (2022)	Experiments to compare YOLOv5 and YOLOv7 in terms of tracking using different evaluation metrics	The results showed that YOLOv7 performed better in some of the evaluation metrics such as MOTA, MOTP and IDF1 while YOLOv5 outperformed YOLOv7 in IDs, ML, MT, FP and FN.	Both algorithms showed good results in tracking experiment. The evaluation metrics of choice in an experiment will decide which of the models to consider for implementation.
Hussain et al. (2022)	A CV-based autonomous rack inspection framework using YOLOv7 and a domain variance modeling mechanism for addressing data scarcity.	The model achieved mAP of 91.1%	The model automated the process. Human errors due to undiscovered damage were reduced.

4 Methodology

Experiments were carried out by training custom datasets model with both YOLOv5 and YOLOv7 independently in order to consider which one of the two performs better in terms of precision, recall, mAP@0.5 and mAP@0.5:0.95 as these metrics determine which one performs better in terms of overall detection. For the quantitative analysis of the models, the metrics used are explained as follows:

i. Precision measures the proportion of accurately categorized positive samples (True Positive) to the total number of positively classified sample (either correctly classified or not, True Positive + False Positive).
ii. The recall value is calculated by taking ration of True Positive to all Positive samples (True Positive + False Negative). It measures how well the model can identify positive samples.
iii. The mAP@0.5 calculates a score by comparing the detected box to the ground-truth box bounding box at IoU threshold of 0.5. The model's detections are the more precise, the higher the score.
iv. mAP@0.5:0.95 refers to the average mAP over various thresholds, from 0.5 to 0.95, in steps of 0.05.

4.1 Experiment Setup

Google Colab which is a platform that offers free coding notebook, a cloud virtual machine with storage, a GPU and Tensor Processing Unit (TPU) for running long and complex computing was used for the experiment on the detection models. All experiments were conducted on HP Probook 6570b using Google Chrome browser to access Google Colab for running the training, validation and the testing of the custom model. The results of the training, validation and testing were saved on Google Drive which can be loaded for further use. The platform is Linux-based (Linux OS) with access to all resources a physical computer possesses. The platform also allows access to the Google drive which is important for loading in the dataset and saving the files.

4.2 Dataset Description

The dataset used in this experiment were Google Open Images Dataset (Google Open Images, n.d.), Roboflow Public Dataset (Roboflow, n.d.) and locally sourced images. Primarily, the images gotten from Google Open Images, which is a large-scale dataset with different trainable classes, were a total of 5808 and they comprise Person, Handgun, Rifle and Knife classes. A total of 2971 images of Pistols were also gotten from Roboflow Public Dataset and modified to class "Handgun" and added to the dataset to make a total of 8779 images. Added to the dataset were locally sourced images from different military theatres of operation. The locally sourced images consist of about 1000 images which captured various persons, handguns, pistols, rifles and knives of different types using a high-definition D5100 DSLR Nikon camera. The camera was set to capture images in resolution of 1280 x 720 pixels which is the resolution used by YOLOv7 and YOLOv5. The images were gathered, cleaned and annotated as Person, Handgun, Rifle and Knife classes using Roboflow Annotation tool. Data preprocessing done on the dataset was Auto-Orient and the images were resized to 416 x 416 (weight x height) size which is the size used by both YOLOv5 and YOLOv7. The dataset used for the training were 9779 images containing 21,561 annotations of the four classes. The dataset was split into training, testing and validation on ratio 60:20:20 of the number of images annotated. 5867 images which makes up 60% of the dataset was used for training while 1955 images which makes up 20% of the total images was used for testing and 1955 images which amounts to 20% of the total images remaining was used for validation. Figure 4 shows the sample images of the dataset gotten from Google Open Images Dataset (left), Roboflow Public Dataset (centre) and locally sourced images (right).

Figure 4: Sample images of the dataset used in the research.

5 Results and Discussion

The output values of the performance results gotten from testing of YOLOv7 model and YOLOv5 model are shown in Table 2.

Table 2: Performance Result of YOLOv7 and YOLOv5

Class	**Images**	**Precision**		**Recall**		**mAP@0.5**		**mAP@0.5:0.95**	
		YOLOv 7	**YOLOv 5**	**YOLOv 7**	**YOLOv 5**	**YOLOv 7**	**YOLOv 5**	**YOLOv 7**	**YOLOv 5**
All	1767	0.528	**0.626**	**0.564**	0.534	0.512	**0.553**	0.315	**0.342**

Handgun	1767	0.778	**0.819**	0.778	**0.785**	0.814	**0.829**	0.584	**0.599**
Knife	1767	0.588	**0.716**	**0.746**	0.695	0.669	**0.740**	0.431	**0.488**
Person	1767	0.382	**0.511**	**0.524**	0.410	0.380	**0.398**	0.173	**0.181**
Rifle	1767	0.363	**0.458**	0.209	**0.247**	0.183	**0.242**	0.0735	**0.101**

5.1 Precision

For precision, comparing the results of YOLOv7 and YOLOv5 from Table 2, it can be seen that YOLOv5 outperforms YOLOv7 in all cases. YOLOv5's all classes had 62.6% and 81.9%, 71.6%, 51.1% and 45.8% for Handgun, Knife, Person and Rifle classes respectively compared with YOLOv7 having 52.8% for all classes, 77.8%, 58.8%, 38.2% and 36.3% respectively for Handgun, Knife, Person and Rifle classes respectively. From the comparison, YOLOv5 has more true positives to total number of detected objects compared YOLOv7 by 9.8% difference in overall class detection. Both models have more detection for the class of Handgun compared to other classes with a difference of 4% when compared with YOLOv7. The model in this case will efficiently identify Handgun more than the other classes.

5.2 Recall

For the results of recall in Table 2, it can be seen that YOLOv5 outperforms YOLOv7 in only Handgun and Rifle detection with results of 78.5% and 24.7% compared with YOLOv7 results of 77.8%, 20.9%. For the overall class recall, Knife and Person, YOLOv7 outperforms YOLOv5 with YOLOv7 having 56.4%, 74.6%, 52.4% compared with YOLOv5 having 53.4%, 69.5% and 41%. For the recall value, Handgun mostly recalled during detection with a percentage of 78.5% for YOLOv5 compared with 77.8% of YOLOv7 with a slight difference of 0.7%. YOLOv7 in this case was able to surpass YOLOv5 in identifying the Knife and the Person classes making an overall class recall better than YOLOv5 with a slight difference of 3%. Meanwhile, YOLOv5 also has better recall in Handgun and Rifle class detection compared to YOLOv7.

5.3 Accuracy in Terms of mAP@0.5 and mAP@0.5:0.95

For mAP@0.5 and mAP@0.5:0.95, comparing the results in Table 2, it is seen that YOLOv5 gave a better result in terms of accuracy than YOLOv7 in all cases, with the overall class results in mAP@0.5 and mAP@0.5:0.95 of 55.3% and 34.2% compared with 51.2% and 31.5% of YOLOv7. The mAP values comparing the detected box to the ground truth bounding box at IOU of 0.5 shows that the model precise detection of an object in a frame. With YOLOv5 having mAP@0.5 of 4% difference compared with that of YOLOv7 shows how well the model is able to rightly and accurately detect objects when compared with the ground truth objects. With average mAP at different thresholds the mAP@0.5:0.95 also records better performance for YOLOv5 compared with YOLOv7 with a slight difference of 2.7%.

It is observed that the YOLOv5 model performs better than YOLOv7 for all the performance metrics except for the case of recall score during testing. It is deduced from the experiments that YOLOv5 has better detection accuracy, precision and less recall than YOLOv7 especially when used during production as deduced from the testing results.

6 Conclusion

This paper conducted a comparative analysis of the widely used YOLOv5 and the relatively new YOLOv7. The experiment carried out shows significant contribution compared to other works earlier mentioned in the literature review. It shows the ease of setting up and use of detection models, and the use of the different evaluation metrics for the experimentation for comparison. The experiment also demonstrates the effectiveness of the YOLOv5 model compared with YOLOv7. These two versions of YOLO were compared in terms of precision, recall, and mAP. It is observed from the experiment conducted that YOLOv5 gave a better result than YOLOv7. YOLOv5

gave a precision value of 62.6% compared to 52.9% of YOLOv7, accuracy score of 55.3% to 51.2%, while YOLOv7 has slightly higher recall than YOLOv5 during testing. Also, YOLOv5 outperformed YOLOv7 in mAP@0.5:0.95. The experiment performed showed better performance in favour of YOLOv5. The results from the experiment will benefit researchers seeking to use either one of the models as a reference for choice of experiment by considering the evaluation metrics. However, with more research on both models, clear performance difference will be pointed out clearly, as one model may perform better than other in different applications and use cases.

References

Alexey B., Chien-Yao W., Hong-Yuan M. L. (2020) Yolov4: Optimal speed and accuracy of object detectionarXiv:2004.10934.

Banerjee A. (2022). *YOLOv5 vs YOLOv6 vs YOLOv7*. Retrieved October 12, 2022, from https://www.learnwitharobot.com/p/yolov5-vs-yolov6-vs-yolov7/.

Cengil, E., & Cinar, A. (2021). Poisonous mushroom detection using YOLOV5. *Turkish Journal of Science and Technology*, *16*(1), 119-127.

Chuyi L., Lulu L., Hongliang J., Kaiheng W., Yifei G., Liang L., Zaidan K., Qingyuan L., Meng C., Weiqiang N., Yiduo L., Bo Z., Yufei L., Linyuan Z., Xiaoming X., Xiangxiang C., Xiaoming W., Xiaolin W. (2022). YOLOv6: A single-stage object detection framework for industrial applications. _arXiv_:2209.02976

Dima, T. F., & Ahmed, M. E. (2021, July). Using YOLOv5 Algorithm to Detect and Recognize American Sign Language. In *2021 International Conference on Information Technology (ICIT)* (pp. 603-607). IEEE.

Google Open Images. (n.d.). Google Open Images Dataset of Person, Handgun, Rifle and Knife. Retrieved from https://storage.googleapis.com/openimages/web/visualizer/index.html.

Górriz, J. M., Ramírez, J., Ortíz, A., Martínez-Murcia, F. J., Segovia, F., Suckling, J. & Ferrández, J. M. (2020). Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. *Neurocomputing*, *410*, 237-270.

Hao, X., Bo, L., & Fei, Z. (2021). Light-YOLOv5: A Lightweight Algorithm for Improved YOLOv5 in Complex Fire Scenarios.

Hussain, M., Al-Aqrabi, H., Munawar, M., Hill, R., & Alsboui, T., (2022). Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections. *Sensors, 22, 6927.*

Jia, W., Xu, S., Liang, Z., Zhao, Y., Min, H., Li, S., & Yu, Y. (2021). Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Processing*, *15*(14), 3623-3637.

Kasper-Eulaers, M., Hahn, N., Berger, S., Sebulonsen, T., Myrland, Ø. & Kummervold, P. E. (2021). Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5. *Algorithms*, *14*(4), 114.

Liu, W., Wang, Z., Zhou, B., Yang, S., & Gong, Z. (2021, May). Real-time signal light detection based on yolov5 for railway. In *IOP Conference Series: Earth and Environmental Science* (Vol. 769, No. 4, p. 042069). IOP Publishing.

Malta, A., Mendes, M., & Farinha, T. (2021). Augmented reality maintenance assistant using yolov5. *Applied Sciences*, *11*(11), 4758.

Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors*, *22*(2), 464

Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics, 10(3), 279.

Patel, D., Patel, S., & Patel, M. (2022). Application to image-to-image translation in improving pedestrian detection.

Ramya, A., Venkateswara, G. P., Amrutham, B.V., Sai, S. K. (2021). Comparison of YOLOv3, YOLOv4 and YOLOv5 Performance for Detection of Blood Cells. *International Research Journal of Engineering and Technology (IRJET) 8(*4*),* (pp. 4225 – 4229).

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

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Roboflow (n.d). Roboflow Public Dataset (n.d). Public Dataset of Pistols. Retrieved from https://public.roboflow.com/object-detection/pistols

Sahal, M. A. (2021). Comparative Analysis of Yolov3, Yolov4 and Yolov5 for Sign Language Detection. *IJARIIE, 7*(4), (pp. 2395 – 4396).

Wan, J., Chen, B., & Yu, Y. (2021). Polyp Detection from Colorectum Images by Using Attentive YOLOv5. *Diagnostics*, *11*(12), 2264.

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Yang, F., Zhang, X., & Liu, B. (2022). Video object tracking based on YOLOv7 and DeepSORT. *arXiv preprint arXiv:2207.12202*.

Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J., & Li, X. (2021). A real-time detection algorithm for Kiwifruit defects based on YOLOv5. *Electronics*, *10*(14), 1711.