UJEAS

UNIZIK JOURNAL OF ENGINEERING AND APPLIED SCIENCES

UNIZIK Journal of Engineering and Applied Sciences 2(3), December (2023), 335-344
Journal homepage: https://journals.unizik.edu.ng/index.php/ujeas
PRINT ISSN: 2992-4383 || ONLINE ISSN: 2992-4391

Object detection and identification in multiple image scenes using deep learning

Okoro, C.K.; Odo, K.O.*; Nwaorgu, O.A.; Kanu, N.D. and Chikelu, P.C.
Department of Electrical and Electronic Engineering, Michael Okpara University of Agriculture, Umudike, Abia State.

*Corresponding Author's E-mail: kayceebby@yahoo.co.uk

Abstract
Object detection is the process of locating objects of interest within an image or video frame. Object detection and classification is a fast growing and an important aspect of research in computer vision. It has yielded variety of applications in shopping systems, health systems, security systems and many surveillance systems. This paper presents a general trainable framework for object detection in images for multiple scenarios. The detection technique will be based on YOLO v4, and Matlab/Simulink software has been used to design and simulate the object detection system. The research work carried out has been able to apply the machine learning technique and also the YOLO image detection technique for the detection of multiple images in different scenes. The study was able to explain the score and recall of the detector depending on the seven features annotated (chair, fire extinguisher, exit sign, clock, printer, screen and trashbin) during the machine learning using the YOLOv4 deep learning object detector. The result of this study revealed that the detector in score plot performed poorly on three classes (printer, screen, and trashbin) but performed well in four classes (chair, clock, exit sign and fire extinguisher). Also, the result of recall plot showed that the printer, screen and trashbin have lower values of 0.3, 0.5 and 0.6 respectively whereas chair, fire extinguisher, exit sign and clock recorded the highest recall value of 1 each. It also suggests that this architecture can be further developed and used in face detection, face recognition, anomaly detection, crowd counting, security surveillance, etc.

Keywords: YOLO v4, Matlab/Simulink, deep learning, precision, detector score

1. Introduction

Object detection and tracking in wide research areas is computer vision and other applications in traffic detection, vehicle navigation, and interpersonal connections. Object detection is a computer process which is related to computer visualization and image processing that deals with detecting examples of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. The wide area of applications in object detection is face detection, face recognition and video object detection, tracking motion of the ball, tracking ball during the match, tracking person in a video. Also, object detection finds its application in many areas of computer vision, including image fetching and video surveillance, (Sunil and Gagandeep, 2019).

In recent years, the development of computer hardware and software and the introduction of efficient recognition and detection algorithms have led to the development of object recognition technology towards the four criteria for evaluating object recognition methods, namely robustness, correctness, efficiency and scope. The process of object recognition is divided into the processes of obtaining a single frame image of an object, image preprocessing, feature extraction, feature selection, feature matching, and object identification information feedback from the matching result. The key to whether an object recognition technology is efficient lies in the efficiency of the object feature extraction, feature processing matching, and classification and recognition methods. With the continuous development of deep learning, the application of deep learning in the field of object recognition has become a research hotspot in various enterprises and scientific research institutions, (Shiji et al, 2020).

There are a lot of contributions in the field of object detection, due to the high demand for faster and more robust techniques. There are three major approaches to object recognition:

1. Geometry-based approaches: This was the earliest approach, and used geometric models to account for the variation in an object's appearance due to viewpoint, illumination and occlusion changes.
2. Appearance-based approaches: At the end of the geometric era, scientists started discovering appearance-based techniques and a lot of interest was generated around them. Most notably, the Eigen face methods attracted a lot of attention.

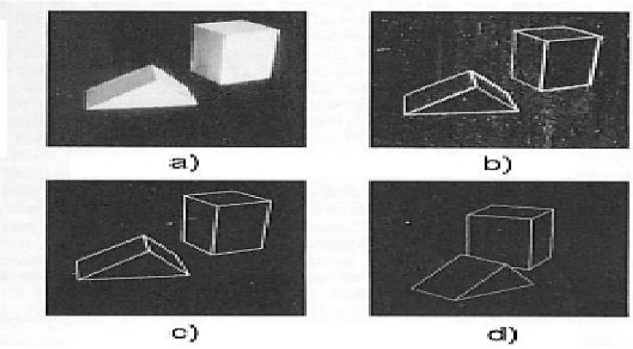


Figure 1: A representation of the Block world (Salvador et al., 2016) a) A block world scene b) Detected edges c) A 3D polyhedral description of the scene d) The 3D scene displayed from a different view point.

It represents one of the very first face recognition systems that was relatively accurate and not so computationally complex making it more efficient, like the work of (Salvador et al., 2016).

3. Feature-based approaches: In feature-based object detection, standardization of image features and registration (alignment) of reference points are important. The images may need to be transformed to another space for handling changes in illumination, size and orientation. One or more features are extracted and the objects of interest are model led in terms of these features. Object detection and recognition then can be transformed in to a graph matching problem (Girshick et al., 2015). The process of recognizing a moving or non-stationary object in a video sequence is known as object detection. This is the initial and most crucial step in tracking moving objects. To gain a thorough understanding of images, we would not only classify them but also attempt to precisely guess the concepts and locations of objects contained in each image. Object detection (Murugan et al., 2019) is the name given to this task, which is divided into subtasks such as skeleton detection, face detection, and pedestrian detection.

Among them, there are various methods that can be applied to the autonomous recognition of tennis balls by robots, such as: direct recognition of the color and contour of tennis balls by the camera; and training of images of tennis balls using a cascade classifier on Open CV. For example, in the literature (Zhao et al., 2019) were lied on the color and contour classifier to recognize the target which is simple and efficient with less preparation, but only guarantees good recognition under ideal conditions such as stable camera operation, open field of view, no interference and clutter, good ambient light, etc., i.e., the anti-interference capability is not strong; then, in the literature (Chen et al., 2016).were lied on the cascade classifier to recognize the target which improved the anti-interference capability and detection efficiency, but recognition distance is shorter and requires more preparation work. Combining the advantages and disadvantages of the above methods, the deep learning method is chosen to train the image samples of tennis balls, which can ensure the anti-interference ability and recognize tennis balls at longer distances, and the recognition algorithm has strong robustness (Chen et al., 2016).

1.1 You Only Look Once (YOLO)

YOLO is the strongest, fastest, and simplest object detection algorithm used in real-time object detection. YOLO runs at 155 fps achieved mAP = 52.7%, and its improved version runs at 45 fps achieved mAP = 63.4% on the VOC-2007 dataset. YOLO designers completely replaced the previous object detection model's proposed detection plus verification. All previous object detection algorithms use regions to localize objects within the image, but the YOLO approach is entirely different; the entire image is applied to a single CNN. YOLO network splits the entire image into regions, and for each region, it predicts bounding boxes and class probabilities, (Chinthakindi et al, 2020)

The main drawbacks of the YOLO object detector are: detection of small objects in an image, and localization accuracy dropping off when compared to two-stage detectors. YOLOv2, YOLOv3, and SSD, (Liu et al, 2016) detectors paid much attention to YOLO drawbacks. To the basic YOLO detector, Redmond et al, (2016) later made improvements and implemented YOLOv3 and YOLOv4 which have achieved better detection accuracy without scarifying detection speed.

The YOLO Detection System

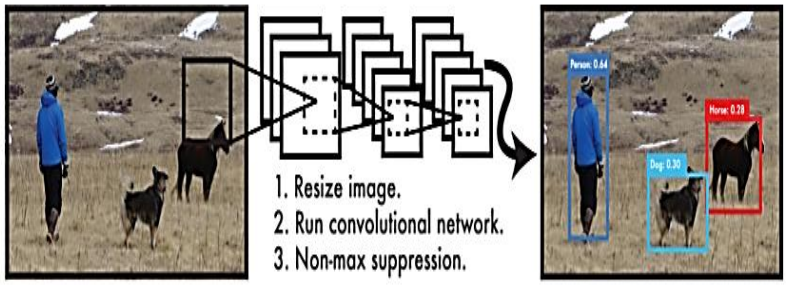


Figure 2: The YOLO Detection System. Source: (Rahman, 2022).

In the YOLO system, the inputs (images) are resized, then a single convolutional neural network is run on the image and the detection procedure is done by non-max suppression. With just a single convolutional network, multiple bounding boxes and class probabilities are predicted concurrently. Detection of the object in images is very fast in this system but it can't do well precisely in localizing some objects, especially the small ones. In the Unified Detection in YOLO, the input images are divided into an S × S grid. If the center of an object falls into a grid cell, that grid cell is liable for detecting that object. Each grid cell predicts the bounding boxes (B) and confidence scores for those boxes by using the features from the whole image. These confidence scores reflect the probability that there's an object in the box and the accuracy of the prediction of an object class. The confidence score is defined as (Kuntal and Paliwal, 2022).

$$confidence = Pr(classi\ /\ object) \times Pr(Object) \times IoUtruth\ pred\ ;\ Pr(object) \in [0,1] \quad (1)$$

where Pr(object) is the probability of an object in the grid cell, Pr(classi /object) is the probability of a specific object present in the cell given that the cell contains an object. IoU is the intersection over union, IoU truth pred is intersection over Union (IoU) metric for true and predicted bounding boxes. If the confidence score is zero, it means that there's no object in that cell. The confidence score is used for the calculation of mAP at a threshold value. For example, bounding boxes with confidence below the threshold are ignored. Each bounding box has bx, by, w, h, and confidence attributes for each object. The (bx, by) coordinates represent the center of the box relative to the bounds of the grid cell. The width (w) and height (h) are predicted relative to the whole image. Each grid cell predicts the class probabilities (Pr(Class)) of the objects it contains. For example: If a grid cell predicts that it contains objects with pr(Car)=60%, then there's 60% chance that the cell contains a car and 40% chance that it does not contain a car.

1.2 Literature Review

1.2.1 Review of related works

Liu et al (2016) presented a simple and straightforward network called as Single Shot multi-box Detector (SSD) which is capable of delivering real-time performance at high accuracy. This network does not utilize regional proposal method. Luo et al (2017) presented an OpenCL based implementation of the Deep Convolutional Neural Network, which is one of the most advanced deep learning frameworks. Their framework aimed at three major contributions- a real-time object recognition system, framework with low power consumption that can be applied even in portable devices. Alpaydin (2018) proposed an adaptive fuzzy based network topology which is run alongside Deep Convolutional Neural Networks, to achieve highly efficient object recognition for long range images that are low in contrast and having variable, noisy backgrounds.

Redmon et al (2018), in their paper, presented YOLOv3 which is an updated version of their revolutionary network YOLO. This model surpassed all the other state-of the-art networks such as Faster R-CNN, VGG-16, ResNet, etc., in terms of computational speed and accuracy, thus making it an ideal network for performing real-time detections and tracking while maintaining high accuracy which the other networks have failed to do. The YOLOv3 is also capable of detecting objects of small size as it can detect objects of three different scales effectively.

The researchers have identified the knowledge gaps existing among various researchers in their literatures. Based on this, a research work on object detection and identification in multiple image scenes using deep learning and YOLOv4 has been carried out.

2.0 Material and methods

The materials required in this study are Matlab/Simulink, Laptop/PC, YOLO v4 algorithm and Images (Jpeg)

2.1 Methods

2.1.1 The YOLO Model

YOLO treats object detection as an exclusive regression problem, straight from image pixels to bounding box coordinates and object probabilities. An individual convolutional network at one time predicts multiple bounding boxes and probabilities for those boxes. YOLO runs the detection on full images and undeviating optimizes detection performance. This joined model has various advantages over classical methods of object detection. YOLO's system models detection as a regression problem. It divides the image into a $X \times X$ grid, for every grid cell predicts B bounding boxes, confidence for those boxes, and C object probabilities. Each grid cell predicts B bounding boxes and confidence rates for these boxes. These confidence rates indicate how confident the model is that the box comprises an object and also how precise it thinks the box and the predicted objects are. Each grid cell also predicts C conditional class probabilities, Pr(Classi |Object). These probabilities are transformed on the grid cell holding an object. One set of sophistication probabilities per grid cell can only be predicted regardless of the quantity of boxes B.

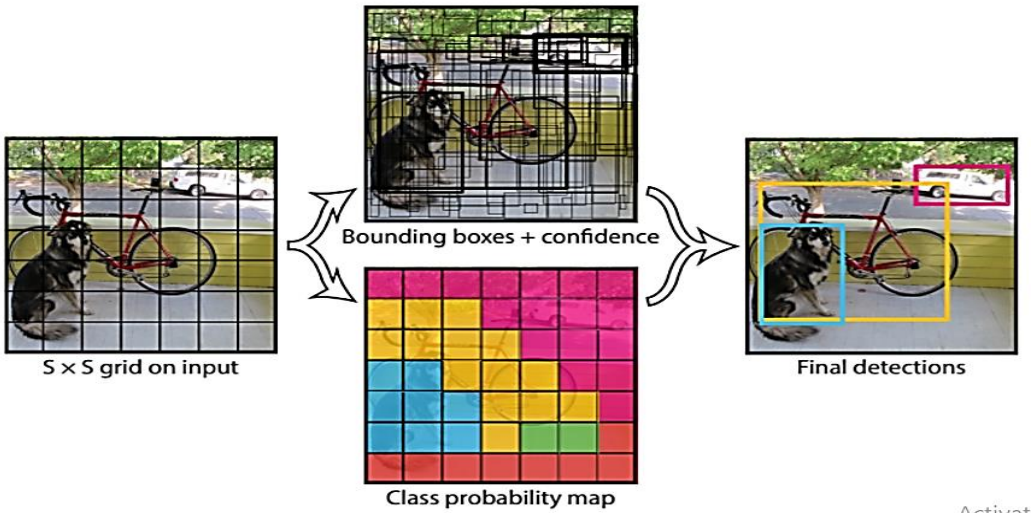


Figure 3: YOLO detection steps (Kuntal and Paliwal, 2022)

2.1.2 Network Architecture and Training

YOLO's interface has 24 convolutional layers followed by 2 entirely connected layers. It simply uses 1 × 1 reduction layers followed by 3 × 3 convolutional layers Fast YOLO practices a neural network with 9 convolutional layers instead of 24 and fewer filters in those layers (figure 4). Leaving apart the volume of the network, all training and testing parameters are the same between YOLO and Fast YOLO. YOLO is optimized for sum-squared error within the output of our model. It implements sum-squared error because it is easy to optimize, even though it doesn't align to maximize average precision. It weights localization error uniformly with classification error which is not prototypical. Also, in every image, many grid cells don't contain any object. This drives the “confidence” of many of those cells towards zero, often overwhelming the gradient from cells that do contain objects. This will cause model instability, causing training to diverge early.

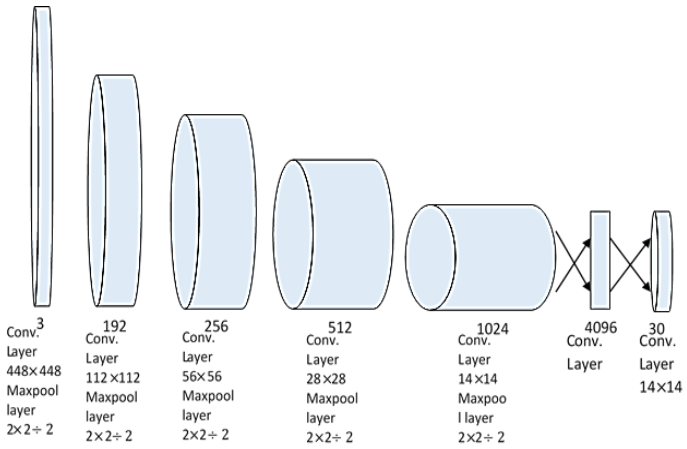


Figure 4: The Architecture of YOLO system

To change this, YOLO intensifies the loss from bounding box coordinate predictions and decreases the loss from confidence predictions for boxes that don't contain objects. YOLO uses two parameters, λcoord and λnoobj to achieve this. YOLO sets λcoord = 5 and λnoobj = .5. The sum-squared error also equally weights errors in large boxes and small boxes. Its error metric should reflect that tiny deviation in large boxes matters but small boxes. To partially address this, the basis of the bounding box width and height was predicted instead of the width and height directly. YOLO predicts multiple bounding boxes per grid cell. At the time of training, only individual bounding box predictors was liable for each object. One predictor is assigned to be "responsible" for predicting an object supported which prediction has the very best current IOU with the bottom truth. This results in specialization between the bounding box predictors. Each predictor gets more qualified at predicting specific sizes, aspect ratios, or classes of objects, improving overall recall.

2.1.3 Deep Learning Technique

Deep learning is a powerful machine learning technique that can be used to train robust multiclass object detectors such as YOLO v2, YOLO v4, YOLOX, SSD, and Faster R-CNN. This technique will test for images detection for various objects.

Deep learning process

- Upload image of object to the system (several if possible)
- Identify its special features for training of the machine
- Loading of YOLO v4 pre-training object detector
- Deep learning process
- Create an image data store for object
- Analyze the distribution of object class label and size

3.0 Results and Discussions

3.1 Evaluate Object Detector

Evaluate the trained object detector on test images to measure the performance. Computer Vision Toolbox provides an object detector evaluation function (evaluateObjectDetection) to measure common metrics such as average precision and log-average miss rate of object detection. For this evaluation, the average precision (AP) metric will be used to evaluate performance. The average precision provides a single number that incorporates the ability of the detector to make correct classifications (precision) and the ability of the detector to find all relevant objects (recall).

Figure 5 presents multiple images of the identified objects during the training process; it is the duty of the trained machine to identify the objects based on the training undergone. The test conducted was carried out in 3 different scenes in which the clock, printer, screen and chair were available in a multiple image scenario (figure 5), the second scene consist of the fire extinguisher and the exit sign in a multiple image scenario (figure 6), the third scene consist of the dust bin in which it is considered alone (figure 7).

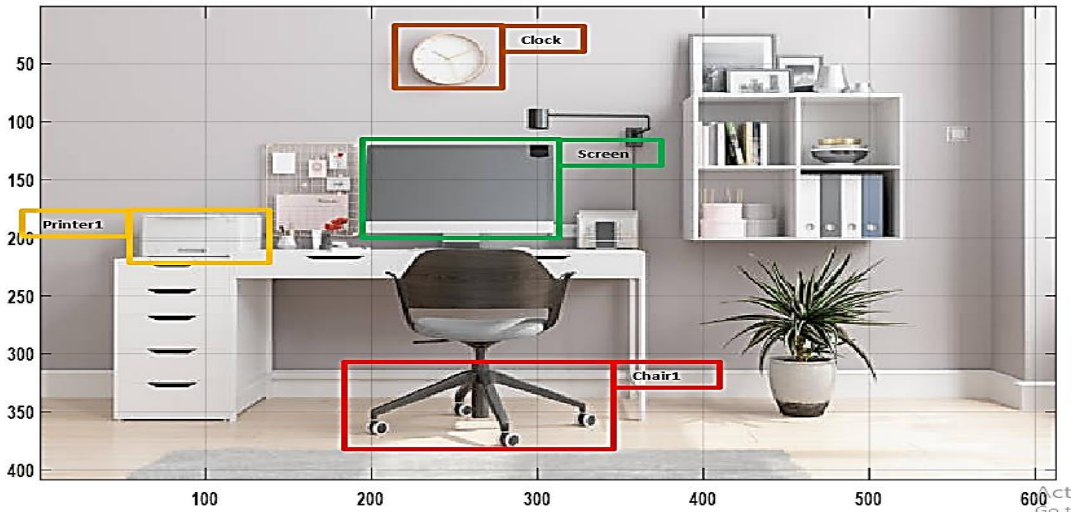


Figure 5: Multiple object detection test for the trained object detector (scene 1)

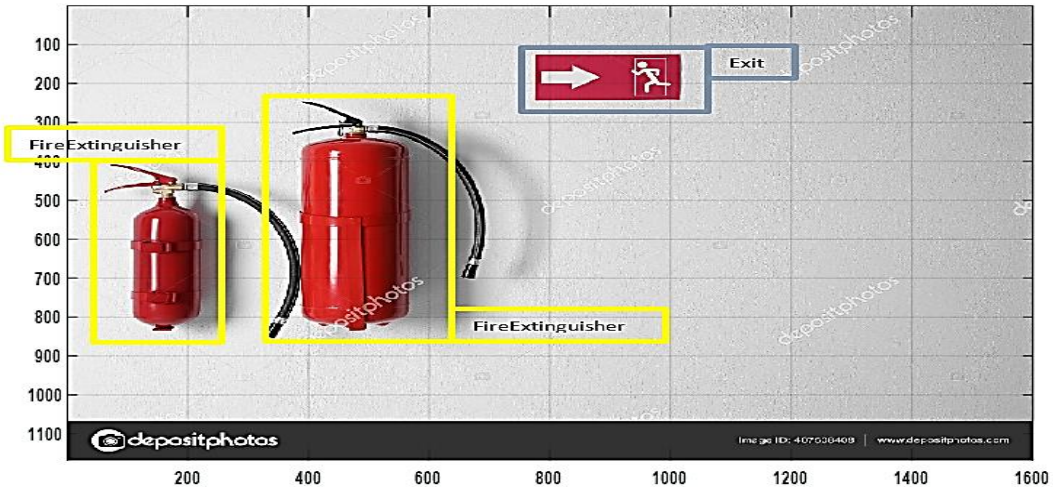


Figure 6: Multiple object detection test for the trained object detector (scene 2)



Figure 7: object detection test for the trained object detector (scene 3)

3.2 Detector precision and Detection score

Running the detector on the test data set will set the detection threshold to a low value to detect as many objects as possible. This helps to evaluate the detector precision across the full range of recall values.

Calculating object detection metrics on the test set results with evaluateObjectDetection, which evaluates the detector at one or more intersection-over-union (IoU) thresholds plot indicates the score of detector for the object detection period of 1 second. The score defines the amount of overlap required between a predicted bounding box and a ground truth bounding box for the predicted bounding box to count as a true positive. The evaluation of object detection on intersection over union (IoU) class metrics is shown in Table 1.

Table 1: Evaluation of object detection on intersection over union (IoU) class metrics

metrics.ClassMetrics
ans=*7×5 table*

	NumObjects	IoU	AP	Score	Recall
chair	**168**	**0.60842**	**{3×1 double}**	**{3×13754 double}**	**{3×13754 double}**
clock	**23**	**0.551**	**{3×1 double}**	**{3×2744 double}**	**{3×2744 double}**
exit	**52**	**0.55121**	**{3×1 double}**	**{3×3149 double}**	**{3×3149 double}**
fireextinguisher	**165**	**0.5417**	**{3×1 double}**	**{3×4787 double}**	**{3×4787 double}**
printer	**7**	**0.14627**	**{3×1 double}**	**{3×4588 double}**	**{3×4588 double}**
screen	**4**	**0.08631**	**{3×1 double}**	**{3×10175 double}**	**{3×10175 double}**
trashbin	**17**	**0.26921**	**{3×1 double}**	**{3×7881 double}**	**{3×7881 double}**

3.2.1 Score Plot

The score plot in figure 8 reveals that the detector did poorly on 3 classes (printer, screen, and trash bin) which had fewer samples compared to the other classes. Detector performance also degraded at higher IoU thresholds. Based on these results, the next step to improve performance is to address the class imbalance problem identified in the Analyze Class Distribution section.

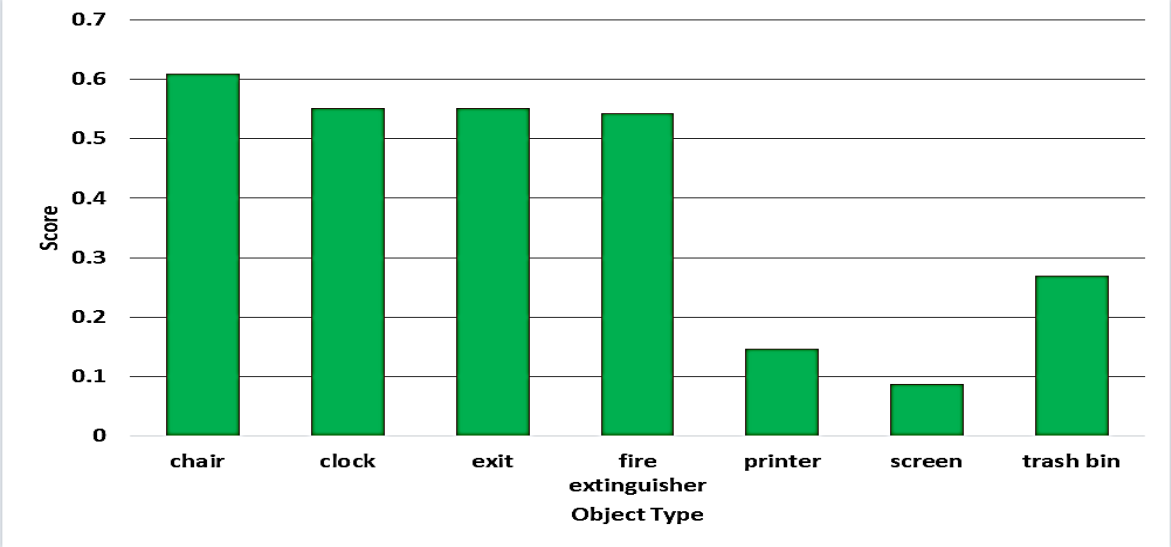


Figure 8: Score plot for detector on the identified object

3.2.2 Recall plot

The recall values for the score test for only selected 1 × 176 data values, Figure 9 plots the recall data for the detector under 1 second, for the recall plot, the printer, screen and trash bin are seen to have lower recalls due to their lesser features during the feature annotation process.

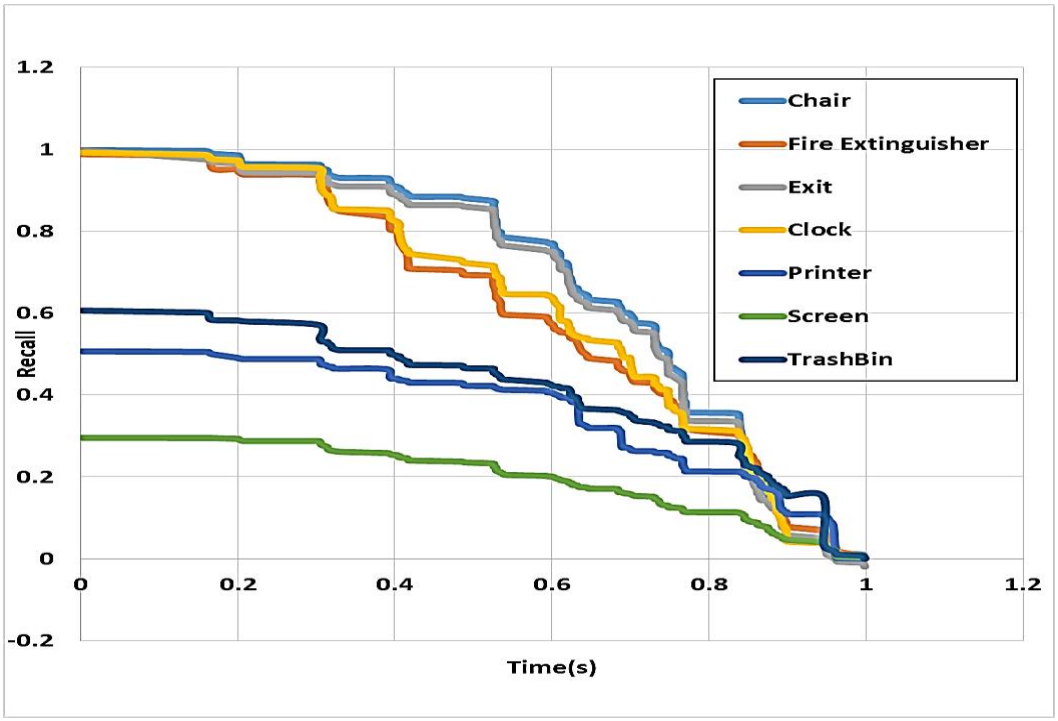


Figure 9: plot of detection recall for identified objects in 1 seconds

3.3 Object Size Impact on Detector Performance (Precision)

Investigate the impact of object size on detector performance using the metricsByArea function, which computes detector metrics for specific object size ranges. The object size range can be defined based on a predefined set of size ranges for the application, or use the estimated anchor. The NumObjects column shows how many objects in the test data set fall within the area range. Although the detector performed well on the "chair" class overall, there is a size range where the detector has a lower average precision compared to the other size ranges. The range where the detector does not perform well has only 11 samples. To improve the performance in this size range, more samples of this size or use data augmentation were added to create more samples across the set of size ranges. The procedure was repeated for the other object classes to gain deeper insight into how to further improve detector performance. The impact of object size on detector using the metricsByArea function is shown in Table 2.

Table 2: Impact of object size on detector using the metricsByArea function

areaMetrics = metricsByArea(metrics,areaRanges,ClassName= classes(3))
areaMetrics=*6×6 table*

AreaRange	NumObjects	mAP	AP	Precision	Recall
0	2774	0	0	{3×1 double}	{3×152 double}	{3×152 double}
2774	9177	19	0.51195	{3×1 double}	{3×578 double}	{3×578 double}
9177	15916	11	0.21218	{3×1 double}	{3×2404 double}	{3×2404 double}
15916	47799	43	0.72803	{3×1 double}	{3×6028 double}	{3×6028 double}
47799	1.2472e+05	74	0.62831	{3×1 double}	{3×4174 double}	{3×4174 double}
1.2472e+05	3.7415e+05	21	0.60897	{3×1 double}	{3×423 double}	{3×423 double}

3.4 Compute Precision and Recall Metrics

Finally, figure 10 plots the precision curve and the detection confidence scores (recall) side-by-side. The precision/recall curve highlights how precise a detector is at varying levels of recall for each class. By plotting the detector scores next to the precision curve, one can observe the performance of the detector in terms of overall confidence level. It is seen that the precision values of screen, printer and trashbin are 0.3, 0.5 and 0.6 respectively while chair, exit sign, clock and fire extinguisher has a precision value of 1 each.

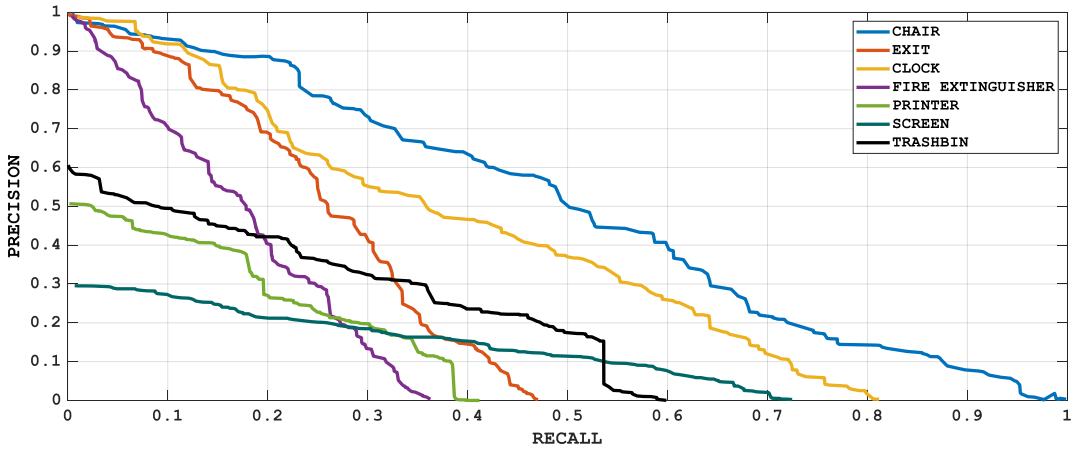


Figure 10: Plot of detector precision against detector against object recall

4.0. Conclusion

The study carried out has been able to apply the machine learning technique and also the YOLO image detection technique for the detection of multiple images in different scenes which follows from the fact that the YOLO deep learning object detector has been learnt. Matlab and m-file codes have been used in this study to execute the deep learning commands and also the application of the YOLO detection technique using version 4. The study was able to explain the score and recall and precision of the detector depending on the seven features annotated (chair, fire extinguisher, exit sign, clock, printer, screen and trashbin) during the machine learning using the YOLOv4 deep learning object detector. The result of this study revealed that the detector in score plot performed poorly on three classes (printer, screen, and trashbin) but performed well in four classes (chair, clock, exit sign and fire extinguisher). Also, the result of recall plot showed that the printer, screen and trashbin have lower values of 0.3, 0.5 and 0.6 respectively whereas chair, fire extinguisher, exit sign and clock recorded the highest recall value of 1 each. Furthermore, the result of this study showed that the precision values of screen, printer and trashbin are 0.3, 0.5 and 0.6 respectively while chair, exit sign, clock and fire extinguisher has a precision value of 1 each.

5.0 Recommendation

It is recommended that other versions of You Only Look Once (YOLO) algorithms for image detection and tracking should be used in order to compare the results obtained from it with that of the YOLOv4.

Acknowledgements

The authors wish to acknowledge the assistance and contributions of the laboratory staff of Department of Electrical and Electronic Engineering, Michael Okpara University of Agriculture, Umudike toward the success of this work.

References

Alpaydin, G., 2018. An adaptive deep neural network for detection, recognition of objects with long range auto surveillance. in ICSC, pp. 316–317.

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild, in Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake, UT, 3339–3348. doi:10.1109/CVPR.2018.00352

Chinthakindi B. M., Mohammad F. H., Neeraj D. B. and Zong W. G. 2020. Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review. Applied Science, 10(3280); pp. 1 – 46, doi:10.3390/app10093280

Girshick, R., Donahue, J., Darrell, T., and Malik, J. 2015. Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Machine Intell. 38, 142–158. doi: 10.1109/TPAMI.2015. 2437384

Kuntal, T. S., and Paliwal, M. 2022. Object detection using YOLOv4. 06, 847–852.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., S. Reed, C. Y. Fu, and A. C. Berg, 2016. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Springer: Berlin, Germany, 2016; pp. 21–37

Luo, Y., Li, S., K. Sun, R. Renteria, and K. Choi, 2017. Implementation of deep learning neural network for real-time object recognition in OpenCL framework. DOI:10.1109/ISOCC.2017.8368905

Rahman, M. T. 2022. Driving-scene image classification using deep learning networks : YOLOv4 algorithm

Redmon, J and Farhadi, A., 2018. "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767.

Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. 2016. You only look once: Unified, real-time object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 779–788

Salvador,A.,Giró-i-Nieto,X.,Marqués,F.,and Satoh,S.I. 2016."Fasterr-cnn features for instance search," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Las Vegas, NV, 9–16. doi: 10.1109/ CVPRW.2016.56

Shiji L., Hong C., Qing W., Jiahui A. and Jiayue L. 2020. Summary of object recognition. Journal of Physics: Conference Series, 1651 012159 doi:10.1088/1742-6596/1651/1/012159

Sunil and Gagandeep 2019. Study of object detection methods and applications on digital images. International Journal of Scientific Development and Research (IJSDR), Volume 4, Issue 5, pp. 491 – 497