International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-12 Issue-5, May 2025

Improved Fast YOLO One-Stage Object Detection Algorithm for Detecting Objects in Images

Fidelis Nfwan Gonten, Ezekwe, Chinwe Genevra, Otene Patience Unekwuojo

Abstract: ***The use of a Convolutional neural network (CNN) has gained wide recommendation in the research community, especially in the area of vision systems (Object detection). The recent CNN recorded Various advancements in object detection in images with tremendous accuracy but still faced challenges of high time complexity. A one-stage object detection algorithm called YOLO (You Only Look Once), used for object classification and localization, performs great, especially in detecting objects in real time. In this study, we proposed an improved, fast YOLO CNN-based algorithm for detecting objects in images. We introduced hard negative mining for resampling and voting to eliminate some negative samples for balancing between negative and positive samples. A small convolution operation was used in exchange for the original convolution, which adjusted the parameters and effectively decreased detection time in images. The proposed model outperformed Fast YOLO with a precision of 88.32% and recall of 89.92%, conducted on smart city datasets.***

Keywords: Convolutional Neural Network (CNN), Object Detection, You Only Look Once (YOLO).

Abbreviations:
CNN: Convolutional Neural Network
YOLO: You Only Look Once
DL: Deep Learning
LSTM: Long-Term Short Memory
SSAM: Spatial Semantic Attention Module
NCC: Normalized Cross-correlation
FPS: Frames Recognized Per Second

I. INTRODUCTION

The Deep Learning (DL) framework has been extensively employed in a wide range of fields in recent years, and convolutional neural networks (CNNs) are one of the most commonly used architectures in tackling real-time problems in computer vision since they allow for the most accurate acquisitions (Rashid, Khan, et al. 2019).

In contemporary practice, CNN models are widely engaged in diverse fields of vision systems, including image recognition and classification and image tracking. Image detection has been among the hot areas of study in vision systems because of the complex nature of the problems. (Yin, Li, & Fu, 2020) [15]. Image detection tasks are more intricate and challenging than object recognition tasks. Image detection is used for identifying various objects in a section and then assigning labels to all the bounding boxes of those objects. (Yin et al., 2020). Detecting objects includes two different parts of tasks: locating objects and classifying the objects. (Ansari, 2020) [1]. Object localization was performed before the CNN algorithm became famous by making every pixel in the image that contains the object look like the Object detection. The previous techniques used edge detection, drawing contours, and HOGs (Ansari, 2020). This method is computationally demanding, sluggish, and inaccurate, (Ansari, 2020).

In recent years, Schilling et al. have made a dramatic improvement by presenting a CNN-based approach for the detection of personal vehicles constructed for the execution of multi-sensor data attached to the pseudo-Siamese body.

(Lu et al., 2019) Presented a fast YOLO algorithm for object detection in real-time images. The proposed model used image pre-processing to remove the influence of background on the images and further trained the Fast-YOLO algorithm to detect images and obtained object details, through the use of Google Net as the base network and the application of 1x1 and 3x3 Conv-layers, which further reduces the number of parameters and reduces the time it takes to detect an object in the images. The proposed model experienced a setback due to balancing between positive samples and negative samples because most of the samples were negative. (Lu et al., 2019) [7].

This paper aims to advance the detection speed of objects in images and optimize the network structure through the introduction of the Hard-Negative Mining (Resampling and Voting) technique to stabilize the negative-sample and positive-sample in our proposed model.

Is an internationally reputed journal that publishes research articles globally. All accepted papers should be formatted as per the Journal Template. Be sure that each author profile (min 100 words), along with a photo, is included in the final paper/camera-ready submission. It is to be sure that the contents of the paper are fine and satisfactory. The author (s) can make rectification in the final paper, but after the final submission to the journal, rectification is not possible. In the formatted paper, volume no/ issue no will be in the right top corner of the paper. In the case of failure, the papers will be declined from the database of the journal and publishing house. It is noted that: 1. Each author profile, along with a photo (min 100 words) has been included in the final paper.
2. The final paper is prepared as per the journal template. 3. The contents of

Manuscript received on 20 December 2024 | First Revised Manuscript received on 10 April 2025 | Second Revised Manuscript received on 19 April 2025 | Manuscript Accepted on 15 May 2025 | Manuscript published on 30 May 2025.
***Correspondence Author(s)
Fidelis Nfwan Gonten*, Department of Computing Science, Admiralty University of Nigeria/ Abubakar Tafawa Balewa University, (Delta/Bauch), Nigeria. Email ID: gontenlur@gmail.com, ORCID ID: 0009-0001-1037-6031
Dr. Ezekwe, Chinwe Genevra, Senior Lecturer, Department of Computing Science, Admiralty University of Nigeria, Ibusa (Delta), Nigeria. Email ID: ezekwe-software@adun.edu.ng, ORCID ID: 0009-0006-2498-3074
Otene Patience Unekwuojo, Lecturer, Department of Computing Science, Admiralty University of Nigeria, Ibusa (Delta), Nigeria. Email ID: puotene@gmail.com

the paper are fine and satisfactory. The author (s) can make rectification in the final paper, but after the final submission to the journal, rectification is not possible.

II. RELATED WORK

One of the most advanced DNN-based object detection methods with good performance is YOLO. Both in terms of speed and precision, its most recent version, Tiny-YOLO-V3, features a compact form that can run on embedded technology. (Fang, Wang, & Ren, 2019) [2]. (Redmon, Divvala, Girshick, & Farhadi, 2016) [5]. In 2016, YOLO was proposed to identify the object detection issue as a regression issue. It was a turning point for one-stage detectors in comparison to the two-stage detector, which was very quick [8]. However, it did poorly at locating and detecting tiny items. YOLO-V3 and the series of Single, YOLO-V2, and Multi-Box Shot Detector (SSD) tried to resolve the issue (Z. Li & Zhou, 2017; Liu et al., 2016; Redmon & Farhadi [9], 2017, 2018) [10]. (Wang, Niu, Wang, & Chen, 2019) [13] Proposed a novel network model called Res-YOLO-r based on mixed ResNet and YOLO detection networks to enhance localization capacity and increase network convergence by using the number and size of prediction boxes for the YOLO, which are changed by the clustering technique [4]. The proposed convolutional neural network technology with the YOLOv3 detection algorithm for targeted objects is hoped to achieve image identification and detection, analysis, and processing via computer vision algorithms. (Xie, Ahmad, Jin, Liu, & Zhang, 2018) [14] Presented the MD-YOLO model, used for multi-directional detection of the license plate of a vehicle. As a metric for evaluation, a probationary angle forecast and a rapid intersection over the union (IoU) were used [17]. (Yin, Li & Fu 2020) proposed a novel Faster-YOLO used for real-time object detection [18]. The proposed model applied the random deep kernel convolution with two hidden layers for extracting image features to detect objects [19]. The Faster-YOLO performance conducted with the PASCAL VOC dataset shows an enhanced accuracy in detection as compared to YOLOv2 and YOLOv3. (Vajgl, Hurtik & Nejezchleba) Proposed an improved Dist-YOLO for the prediction of images via monocular camera information [20]. The proposed architecture extends the combination of vector prediction, weight backbone sharing, bounding box regression, and loss function for distance prediction [21]. The model provides two methods of solving the distance: agnostic class and aware class. The agnostic class achieved good prediction and better results. The proposed model used KITTI datasets and achieved good results with 45 frames per second (FPS). (Shafiee et al.., 2017) proposed a novel CNN-based architecture called Fast YOLO, which improved on the YOLO-v2, used for real-time object detection in images on embedded systems. The proposed model controls the deep CNN model to modify the YOLO-v2 algorithm, which optimizes the model with fewer parameters. A motion adaptive inference technique is employed to decrease the expended power to decrease deep inference by 38.13%. The proposed model performed with an average speed of 3.3% for detecting objects in the video above the YOLO-v2. (Zarei, Moallem & Shams 2022) Proposed a YOLO-based detection algorithm with Long-Term Short Memory (LSTM) called Fast-YOLO-Rec, used for detection and prediction of vehicles via vehicle dataset. The proposed system employed a spatial semantic attention module (SSAM), which adds to the speed and accuracy of detection. The proposed model detects the position of the vehicle faster than other fast detectors with recommended accuracy. The performance and evaluation output of the Fast-YOLO-Rec performs better than the baseline techniques.

(Lu et al., 2019) Proposed a YOLO-based algorithm called Fast YOLO, used for real-time image detection in videos developed. The proposed model introduced an image pre-processing technique by removing background images to detect objects and acquire object details. They deployed the Google Net model with 1x1 convolution, which reduced network parameters and further enhanced the Fast YOLO. The proposed model is greatly used in detecting real-time images in video.

Baseline Techniques: The techniques compared to our technique are shown in Table I.

Table-I: Classical Algorithm Approaches

References	Classical Algorithm	Techniques
(Sudowe & Leibe, 2011) [11]	Sliding Windows	This technique splits the images into different grids, and sliding windows were applied for feature extraction to each grid to enhance the prediction and probability of object categories.
(Zha, Luisier, Andrews, Srivastava, & Salakhutdinov, 2015) [16]	CNN	The feature extraction was performed by a convolutional approach on images through the edge, texture, and colour.
(Lou & Cui, 2007) [6]	R-CNN	The images are split into different small regions, which are applied to CNN to get the feature regions. The classifier is applied to detect the image as an object or background.
(Sun, Wu, & Hoi, 2018) [12]	Faster R-CNN	The feature extraction areas were plotted on the feature map via the extreme of the conv layer to enhance the extracted feature of the image.
(J. Li, Su, Geng, & Yin, 2018) [3]	YOLO	The images are split into various grids to calculate the possibility of the middle of the image dropping within the grid. To calculate the category possibilities of the objects, bounding boxes are built.
(Shu, Duan & Jiao, 2019)	SSD	The extracted outcome of several layers and split the issues into various levels which predict the image classes.
(Lu et al., 2019)	Fast YOLO	The frame difference approach was used to eliminate the likeness between the two frames, which further implements the background difference technique to predict images.

Besides, it is imperative to handle the issue of sample ratio imbalance between positive and negative samples to improve detection speed and optimization. (Lu et al., 2019). Our proposed model in this study fits into these trends to

attain high performance in the detection of objects in images through the introduction of hard negative mining on the fast YOLO algorithm.

Table I shows the major techniques applied by the baseline paper and the classical algorithm. (Lu et al., 2019).

A. Fast Yolo

In 2019, Lu et al. improved on the YOLO model, which led to the development of the Fast YOLO algorithm. The Fast YOLO applied two main techniques to efficiently detect images in video by first introducing the Google Net module, which enhanced the parameter setting. (Lu et al., 2019). The fully linked layer is swapped out for a pooling layer in the Google Inception Network, and the Inception module is made to increase the effectiveness of parameter use. The input is convoluted by 1x1 in the first branch. To enhance the network, the 1x1 convolution arranged the details across the channels and minimized the size of the output channels. Secondly, to distinguish the background image, the Fast-YOLO algorithm combines the frame difference and background difference approaches. The pixel's brightness follows a Gaussian distribution because the pixels in the background and moving objects change tremendously (Morinaga, Hara, Inoue, & Urahama 2018). As represented in equations 1 and 2, denote the mean difference of the Gaussian distribution, and $\sigma 2$ denotes the variance of the Gaussian distribution. (x,y) denotes the coordinates of a pixel as $B(x,y) \sim N(\mu, \sigma 2)$ fulfilled (Lu et al., 2019).

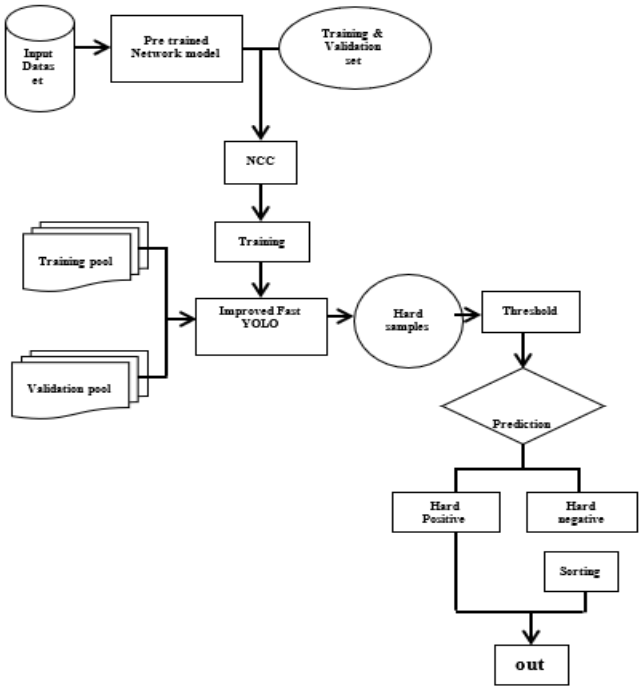
$$\mu(x,y) = \frac{1}{N}\sum_{i=1}^{N-1}[Fi(x,y)] \quad \ldots \quad (1)$$

$$\sigma(x,y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1}[Fi(x,y) - \mu(x,y)]} \quad \ldots \quad (2)$$

III. IMPROVED FAST YOLO

A. Improved Fast YOLO

In our experiment, object detections through the images were thresholded with a high confidence. A score of 0.8. We created a prediction through the execution of template matching in adjacent Frames, confined in a window of ± 5 frames for every image detection in the frame. the normalized Cross-correlation (NCC) was used on the bounding box generated for the current image detection. Which was further expanded to 100 pixels, and the region of interest is searched within the Frames for accurate matching. We applied a threshold set as 0.5 on the NCC to discard scenarios. There are frequent appearance changes to handle occlusion between frames. If the highest (I) within the prediction and detection in each adjacent frame was 0.2, we termed it as a hard Negative sample resulting from detection flicker while the detections were found to be consistent. Adjacent frames were termed to have a high probability of having true predictions (i.e., hard positive samples), as shown in Figure 1.



[Fig.1: Improved Fast YOLO Flowchart]

B. Dataset

Our dataset consists of vehicle and pedestrian images, a smart city dataset compiled by the Xiamen Urban Transport Bureau. The datasets were obtained from the GitHub repository. The videos consist of several 416∗x 416 color frames. In our model, we joined the frame and background differences approach to pre-process the input image, as designed. The training set contains 5000 images, while the test set contains 2000, and finally, the validation set contains 1000 images. We labelled 12,000 images in the training set for various classes and compiled the statistics on five classes: truck, car, bus, and pedestrian. Our model was fine-tuned through parameter tuning with label images and further extracts image features from images and learns about their location using the fast YOLO network.

C. Experimental Setup

In training our model, we used smart cities (from the Xiamen Municipal Transportation Bureau) datasets (Lu et al., 2019). We will apply several performance evaluation metrics, such as Precision and Recall, to show the efficiency and acceptability of our proposed model. The major tools and platform to run the model include the Nvidia GPU hardware accelerator, 32GB RAM, and 130GB disk of memory from Google Collaboratory (colab.research.google.com) as our execution environment. Python programming language and TensorFlow with Keras will be used in coding our model.

D. Loss Function

In our proposed model, the loss function performs both localization loss for offset prediction of the bounding box and the classification loss for possible conditional class.

The two-loss functions are both computed as the sum of squared errors. The base model with the inception module is replaced by 1x1 and 3x1 Conv layers. Our method will divide images by S*S and make the final prediction of shape S*S*(5B*K), which adds the location details, level of confidence, and probability of the class.

Two parameters are applied to regulate the way we can enhance the prediction through the bounding box coordinate and also how to reduce the confidence loss value prediction of boxes with no objects. We down-weighted the loss on the background boxes to a bounding box with no object. In this work, our model will be set at 5 and λ_{noobj} 0.5.

$$\mathcal{L}_{\cdot L_{0c}} = \lambda_{coord} \sum_{i=0}^{S^2} \cdot \sum_{j=0}^{B} \cdot \mathbf{1}_{ij}^{0bj} \left[(x_i - \hat{x}_i)^{\cdot 2} + (y_i - \hat{y}_i)^{\cdot 2} \right] + \left[\left(\sqrt{w_i} - \sqrt{w_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{h_i} \right)^2 \right] \quad \ldots \quad (3)$$

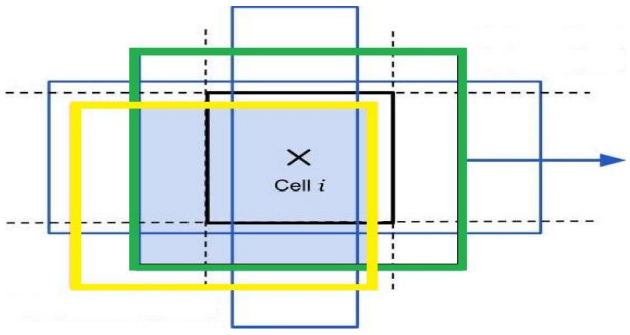
Where 1^{obj}_i is the indicator function and indicates if cell I has an image, 1^{obj}_{ij} indicates if the J-th bounding box of cell i is capable of object prediction, C_{ij} is the confidence value of cell I, $\hat{C}_{ij}$ is the predicted confidence value, C is the set of every class, $p_i(c)$ is conditional of if cell I has an image, $\acute{P}_i(c)$ is the predicted conditional class, as shown in Fig. 2.

E. Bounding Box

In our proposed model, we demarcate bounding boxes around the objects for labelling, we will construct training sets. Ground truth is another name for these bounding boxes in the practice set. (Lu et al., 2019). The object detection method predicts bounding boxes throughout the model learning process and contrasts them with the ground truth. The formula described below will be used to determine the IoU between a predicted bounding box A and a ground truth B (Lu et al., 2019).

$\boldsymbol{I_0U} = \frac{(A \cap B)}{(A \cup B)}$ = Area of intersection/ Area of union (Lu et al., 2019).

The model estimates the bounding boxes around the object as it learns to make great predictions.



[Fig.2: The Green Box Represents the Bounding Box, While the Yellow Box Represents the Truth Bounding Box]

F. Performance Evaluation Metric

We will use the precision and recall rates and the frames recognized per second (FPS) to evaluate the experimental results. (Lu et al., 2019). We will compute the IoU of the detection boundary boxes and reference boundary boxes to judge whether the results are true or false. (Lu et al., 2019).

IoU = {≥ 00.55 true and **IoU** = {< 00.55 True We define the following metric as follows:

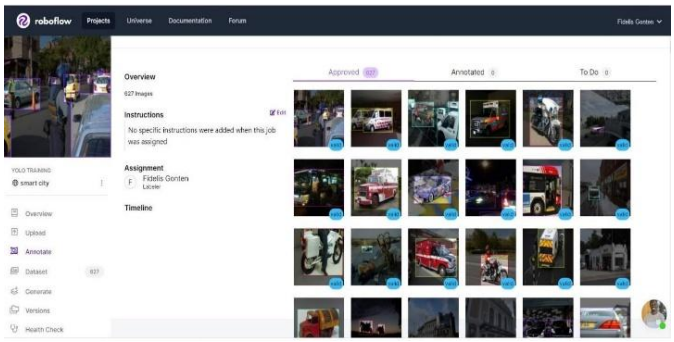
Precision = TP / TP + FP and Recall = TP / TP + FN, where TP, FP, and FN signify the number of True positives, false positives, and false negatives correspondingly (Lu et al., 2019).

IV. RESULTS AND DISCUSSION

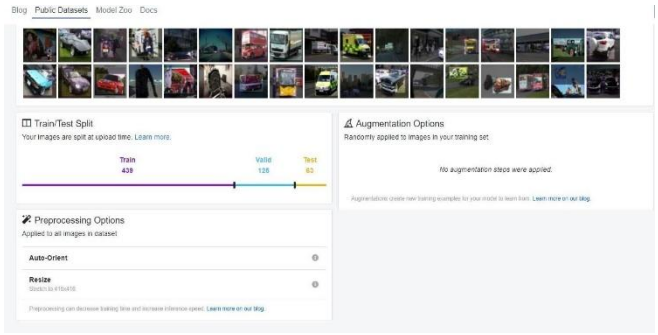
We analyze and illustrate our procedure for annotating training images, training procedure, and visualization, and experimental results of dataset training with hyperparameter tuning of the improved fast-YOLO algorithms, results obtained were compared concerning the benchmark functions.

A. Annotating our Training Image

In training our improved fast YOLO model, we used bounding box labels to monitor the learning process. Every image for which we create a detection method is surrounded by a box, and each box is labelled with the object class that the detector should be able to predict accurately. We divide the datasets into two phases: Training, Validation, and testing having 70%, 20%, and 10% respectively, as shown in Figure 4. For the training set, we balanced the training data by almost equal representation for all classes, and the test data set images are run via the proposed model, and the model's performance is evaluated by comparing the expected results to the labels of the model outcome. We used the CVAT computer vision tool for bounding box annotations and labelling, as shown in Figure 3. To ensure best practices, we labelled all the paths around the target object, labelled all obstructed images, and evaded large spaces around the target image.



[Fig.3: The Annotating Training Interface]



[Fig.4: Training, Validation, and Test Interface]

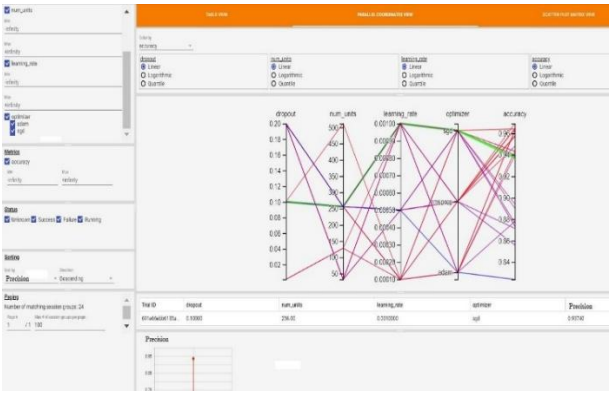
B. Training Procedure and Visualization

Our model selects the

default boxes that match the ground truth during the training. We applied a threshold of 0.5 to regulate the overlapping amid the ground truth and the default boxes. At every network layer, our network learned due to overlapping execution, and numerous default boxes were formed, which did not overlap with the ground truth at (IoU≥0). The negative samples are the default boxes that do not overlap (IoU < 0.5), while the positive samples are the default boxes that match the ground truth. The majority of the samples were negative, which resulted in class imbalance and skewed the predictions. Our model sorted the negative boxes, picked the topmost samples, and discarded the rest, making both negative and positive ratios 2:1. This procedure increases the optimization speed.

C. Hyperparameter Optimization

We chose and fine-tuned three hyperparameters, such as the number of units in the first dense layer (hp. Discrete (16, 32), dropout or learning rate in the dropout layer hp.Realinterval (0.1, 02), optimizer (adam, SGD). These influence our model metrics, precision, and recall. After the experiment, we visualized the result in the TensorBoard on Google Collaboratory, and we used an HParams dashboard to visualize the result, as shown in Figure 5.



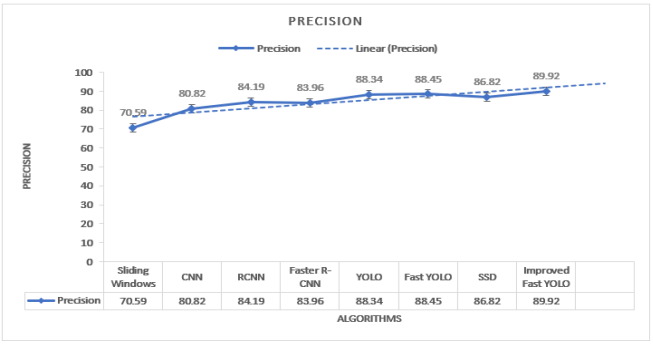
[Fig.5: Tensor Board Hyperparameter Interface]

D. Experimental Result

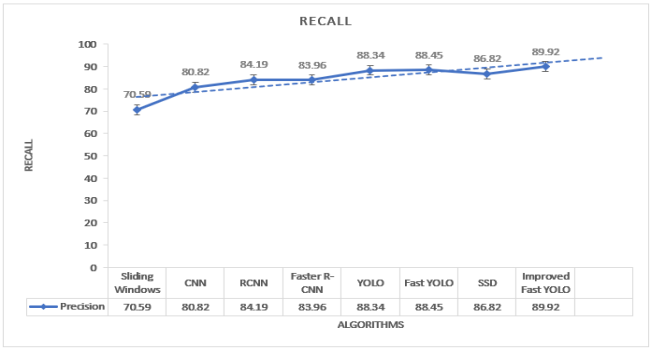
In evaluating our proposed model, we employed five datasets (Car, trunks, bus, motor, and human) (Lu et al., 2019). To assess the experimental findings, we employed the benchmark evaluation metrics (precision and recall) as well as frames identified per second (FPS) (Lu et al., 2019). On the Improved Fast YOLO algorithm and the baseline approaches, we calculate the IoU of the detection bounding boxes for reference and bounding boxes for detection to establish if the results are correct or not and also compute the average for every result of the datasets. As demonstrated in Table II, Figure 6, and Figure 7. Our model performs better than the other algorithms, notably in terms of recall rate. As a result, the risk of missed detection may have reduced the recall rate.

Table-II: The Proposed Model Results

Method	Precision	Recall
Sliding Windows	70.59%	72.91%
CNN	80.82 %	78.38 %
R-CNN	84.19 %	83.69 %
Faster R-CNN	83.96 %	82.65 %
YOLO	88.34 %	86.22 %
Fast YOLO	88.45 %	86.64 %
SSD	86.82 %	84.10 %
Improved Fast-YOLO	89.92 %	88.32 %



[Fig.6: The Precision of the Dataset]



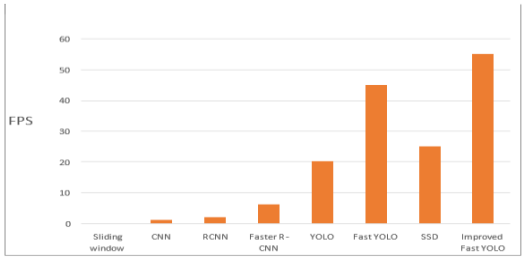
[Fig.7: The Recall of the Dataset]

The precision rate for different image classes for all the algorithms is shown in Table III. The Improved Fast YOLO, Fast YOLO, and YOLO algorithms provide much-improved precision for all categories when compared to the previous algorithms. The precision of buses and cars is much greater compared to trunks in the categories of image detection. Our datasets play a big role in this result. In our image detection, buses and cars are more than trunks.

Table-III: The Precision Rate of the Image Classes

Algorithm	Car	Trunk	Bus	Motor	Person
Sliding Window	65.35	70.26	68.65	68.34	72.16
CNN	81.88	75.38	79.98	79.98	72.31
R-CNN	75.38	80.21	85.36	82.52	85.33
Faster R-CNN	76.24	84.32	85.32	85.32	80.25
YOLO	87.64	84.58	89.99	81.21	88.42
Fast YOLO	87.88	84.76	90.12	80.87	88.64
SSD	80.25	84.29	83.58	76.82	80.16
Improved Fast YOLO	88.45	84.98	92.24	82.45	90.36

The Improved Fast YOLO method, as shown in Fig. 8, can detect frames up to 45 in one second, significantly beyond some CNN models' capacity. The detection speed substantially improved our model, which is based on the Fast YOLO network. The traffic monitoring image is typically 25 FPS.



[Fig.8: Histogram of FPS]

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table-IV: Quality Performance Metrics for Test A.

Experiments	Evaluation Metrics	Car	Trunk	Bus	Motor	Person
Test A	Precision	88.45	84.98	92.24	82.45	90.36
	Recall	89.46	83.78	92.94	84.47	91.36
Test B	Precision	90.87	81.11	91.23	92.67	87.80
	Recall	89.34	85.98	91.29	82.45	92.87
Test C	Precision	87.63	87.45	90.73	94.93	91.89
	Recall	88.45	87.09	90.24	87.90	92.76
Test D	Precision	90.76	91.88	94.89	89.98	97.36
	Recall	81.37	90.27	93.65	91.56	89.24
Test E	Precision	90.89	93.72	90.13	91.78	92.54
	Recall	86.67	87.52	89.46	67.45	87.65

Table IV depicts the quality performance metrics of the proposed model as performed on car, trunk, bus motor, and person datasets. The experiment was tested repeatedly from A, and the results were recorded. The testing value metrics (precision and recall) show some small differences in the outcome for all the datasets tested.

Table V depicts the run time for test A and for each epoch or iteration. The running time for test A shows good results concerning time complexity and the number of frames processed in contrast to tests B, C, D, and E. From the time complexity analysis, the best-case time is 2.67s for detecting a targeted object in images in all the iterations and 4.95s for the worst-case time complexity.

Table-V: Running Time Complexity in Detecting Objects in Images

Tests	A		B		C		D		E	
Epoch	FPS	Time(s)	FPS	Time(s)	FPS	Time(s)	FPS	Time(s)	FPS	Time(s)
1	45.10	2.91	45.11	3.26	45.16	3.81s	45.21	3.75	45.34	4.30
2	45.01	2.67	45.09	3.87	45.49	3.89	45.03	3.92	45.09	4.93
3	45.12	2.95	45.22	3.53	45.42	3.80	45.12	3.62	45.55	4.95
4	45.34	2.84	45.32	3.23	45.29	3.58	45.44	3.91	45.24	4.32
5	44.13	2.78	44.13	3.92	44.51	3.96	44.41	3.76	44.34	4.25

V. CONCLUSION

In this article, we outline a fast method of detecting objects in images. We specifically point out the extraordinary performance of our algorithm in terms of detection speed, with more than [[[[NOTE TO AUTHOR: Please insert missing number here.]]]] images per second of detection speed and excellent detection performance. We optimized fast YOLO (You Only Look Once) by using hard negative mining to reconcile the ratio between positive samples and negative samples to significantly reduce the amount of calculation and speed up detection.

The rate for precision and recall, with frames per second (FPS), was used to assess the tests executed on images from our datasets obtained from GitHub. To train our model and learn about the objects, we take image patches from the vehicle image that have labels tagged on them. The results show that our approach can outperform the Fast YOLO algorithm itself, as well as the other standard techniques. Our approach not only allows for rapid detection but also guarantees excellent accuracy. As long as the equipment is in good shape, our model can detect objects in images.

In the future, we will take into account multi-parallel processing on our proposed model to increase the speed for detection and better improve the model due to issues associated with heavy dense objects and computational load. Additionally, we can simplify the study of the background by not having to construct boxes of any size to serve as bounding boxes.

ACKNOWLEDGMENT

I would also like to express my gratitude to the members of the research committee for their constructive criticism and valuable feedback, which contributed to enhancing the quality of this research.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Authors Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Ansari, S. (2020). Deep Learning in Object Detection. In *Building Computer Vision Applications Using Artificial Neural Networks* (pp. 219-307): Springer. DOI: https://doi.org/10.1007/978-1-4842-9866-4_6
2. Fang, W., Wang, L., & Ren, P. (2019). Tinier-YOLO: A real-time object detection method for constrained environments. *IEEE Access, 8*, 1935-1944. DOI: http://doi.org/10.1109/ACCESS.2019.2961959
3. Li, J., Su, Z., Geng, J., & Yin, Y. (2018). Real-time detection of steel strip surface defects based on improved Yolo detection network. *IFAC-PapersOnLine, 51*(21), 76-81. DOI: https://doi.org/10.1016/j.ifacol.2018.09.412
4. Li, Z., & Zhou, F. (2017). FSSD: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*. DOI: https://doi.org/10.48550/arXiv.1712.00960
5. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *SSD: Single shot multibox detector.* Paper presented at the European conference on computer vision. DOI: https://doi.org/10.1007/978-3-319-46448-0_2
6. Lou, X., & Cui, B. (2007). Boundedness and exponential stability for nonautonomous RCNNs with distributed delays. *Computers & Mathematics with*

Applications, *54*(4), 589-598. DOI: https://doi.org/10.1016/j.camwa.2007.02.007
7. Lu, S., Wang, B., Wang, H., Chen, L., Linjian, M., & Zhang, X. (2019). A real-time object detection algorithm for video. *Computers & Electrical Engineering*, *77*, 398-408. DOI: https://doi.org/10.1016/j.compeleceng.2019.05.009
8. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection—paper* presented at the Proceedings of the IEEE conference on computer vision and pattern recognition. DOI: https://doi.org/10.1109/CVPR.2016.91
9. Redmon, J., & Farhadi, A. (2017). *YOLO9000: better, faster, stronger*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition. DOI: https://doi.org/10.1109/CVPR.2017.690
10. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. DOI: https://doi.org/10.48550/arXiv.1804.02767
11. Sudowe, P., & Leibe, B. (2011). *Efficient use of geometric constraints for sliding-window object detection in video.* Paper presented at the International Conference on Computer Vision Systems. DOI: https://dl.acm.org/doi/10.5555/2045399.2045402
12. Sun, X., Wu, P., & Hoi, S. C. (2018). Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, *299*, 42-50. DOI: https://doi.org/10.1016/j.neucom.2018.03.030
13. Wang, Y., Niu, H., Wang, X., & Chen, L. (2019). *Multi-class Object Detection Algorithm Based on Convolutional Neural Network.* Paper presented at the 2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE). DOI: https://doi.org/10.1109/ICIASE45644.2019.9074015
14. Xie, L., Ahmad, T., Jin, L., Liu, Y., & Zhang, S. (2018). A new CNN-based method for multi-directional car license plate detection. *IEEE Transactions on Intelligent Transportation Systems*, *19*(2), 507-517. DOI: https://doi.org/10.1109/TITS.2017.2784093
15. Yin, Y., Li, H., & Fu, W. (2020). Faster-YOLO: An accurate and faster object detection method. *Digital Signal Processing*, *102*, 102756. DOI: https://doi.org/10.1016/j.dsp.2020.102756
16. Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. (2015). Exploiting image-trained CNN architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*. DOI: https://doi.org/10.48550/arXiv.1503.04144
17. Lambat, S., & Sonawane, Dr. S. S. (2020). Analysis of Different Neural Network Techniques Used for Image Caption Generation. In International Journal of Innovative Technology and Exploring Engineering (Vol. 9, Issue 9, pp. 299–304). DOI: https://doi.org/10.35940/ijitee.i7165.079920
18. Kholwal, R., & Maurya, S. (2021). A Comparative Approach on Classification of Images with Convolutional Neural Networks. In International Journal of Engineering and Advanced Technology (Vol. 10, Issue 4, pp. 201–205). DOI: https://doi.org/10.35940/ijeat.d2483.0410421
19. Panda, C. (2019). Object Detection and Tracking using Faster R-CNN. In International Journal of Recent Technology and Engineering (IJRTE) (Vol. 8, Issue 3, pp. 4894–4900). DOI: https://doi.org/10.35940/ijrte.c5580.098319
20. Dr. Nithyanandh S. (2025). Object Detection & Analysis with Deep CNN and Yolov8 in Soft Computing Frameworks. In International Journal of Soft Computing and Engineering (Vol. 14, Issue 6, pp. 19–27). DOI: https://doi.org/10.35940/ijsce.e3653.14060125
21. Jadia, A., & Chawla, M. P. S. (2020). Image Classification and Detection of Insulators using Bag of Visual Words and Speeded up Robust Features. In International Journal of Innovative Science and Modern Engineering (Vol. 6, Issue 10, pp. 7–13). DOI: https://doi.org/10.35940/ijisme.j1260.0961020

AUTHOR'S PROFILE

Fidelis Nfwan Gonten is a Computer Scientist With a strong focus on prolific research (in AI) and global learning development. Deeply committed to contributing to the research community and advancing the overall body of knowledge through a systematic approach. His expertise lies in building vision-based systems using advanced deep-learning algorithms. Over the past 13 years, he has been actively engaged in the teaching and learning process. Through extensive and rigorous studies, he is well-prepared and equipped to establish and uphold computing standards at a higher level. I am a lecturer with the Admiralty University of Nigeria (ADUN), Computing Science department. He is currently pursuing his PhD. In Software Engineering.

Dr. Ezekwe, Chinwe Genevra is a Senior Lecturer in the department of Computing Sciences of Admiralty University of Nigeria, Ibusa, Delta State, Nigeria. Currently, she is the Head of Department of Computing Sciences of Admiralty University of Nigeria, Ibusa, Delta State, Nigeria. She obtained a Bachelor of Science (B.Sc.) degree in Computer/Physics from University of Nigeria, Nsukka, Nigeria in 1998, Master of Science (M.Sc) degree in Computer Science from Nnamdi Azikiwe University, Awka, Nigeria in 2012 and doctorate Computer Science at Ebonyi state University, Abakiliki, Nigeria in 2019. Her research interests are in Artificial Intelligence systems and embedded systems programming. She has worked in several high institutions as a computer Science lecturer. She is a member of the Institute of Electrical Electronics Engineering (IEEE), NIWIIT (Nigeria Women in Information Technology) and the Computer Society of Institute of Electrical Electronics Engineering (IEEE).

Otene Patience Unekwuojo, BSc. Computer Science (Benue State University), MSc Computer Science (University of Ibadan), and a PhD in View. I am a lecturer at the Admiralty University of Nigeria interested in Artificial Intelligence and Data Science. My research focuses on AI, machine learning, and data-driven solutions for real-world applications.