International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

ElegantYOLO: An Agile and Real-time Vehicle Detection System for Live Environments

Sri Jamiya S[1], Esther Rani P[2]
[1]Research Scholar, Electronics and Communication Engineering
Vel Tech Rangarajan Dr. Sagunthala R &D Institute of Science and Technology
Avadi, Chennai. India.
[1]Email:vtd655@veltech.edu.in
[2]Research Scholar, Electronics and Communication Engineering
Vel Tech Rangarajan Dr. Sagunthala R &D Institute of Science and Technology
Avadi, Chennai. India.
[2]Email:drpestherrani@veltech.edu.in

Abstract— Vehicle detection plays a crucial role in traffic monitoring systems, but it often faces challenges like occlusion, vehicle size, and shape variations. Existing systems struggle with misinterpreting parts of other vehicles as actual vehicles, leading to losses and distorted shapes. To address these issues, a new approach called ElegantYOLO is introduced in this study. ElegantYOLO combines elements from LittleYOLO-SPP and ShortYOLO-CSP while modifying baseline network layer depths using the improved Ghost Module Extended connection method to reduce computational costs. The model's learning capabilities are improved by incorporating spatial attributes through the concatenation of Spatial Pooling blocks. The study employs the Alpha-IoU as the bounding box loss function, minimizing the disparity between predicted and ground truth boxes. This enhances vehicle detection accuracy and robustness. Additionally, the study uses the slicing-aided hyper inference (SAHI) technique, which allows the lightweight backbone network to capture more detailed vehicle information by processing higher-resolution images. Through extensive testing on various datasets such as PASCAL VOC 2007, 2012, and MS COCO 2014, the proposed model not only excels in detecting small vehicles but also demonstrates improved detection accuracy across different environmental conditions. The performance of ElegantYOLO surpasses both LittleYOLO and ShortYOLO by achieving an almost 10% higher mean average precision (mAP). Specifically, the model achieves outstanding results on PASCAL VOC and COCO datasets, with mAPs of 96.45% and 79.28%, respectively. Moreover, the proposed technique significantly enhances accuracy while reducing detection time.

Keywords- Vehicle Detection, YOLOv3-tiny, Darknet, LittleYOLO-SPP, ShortYOLO-CSP, Spatial pyramid pooling, Convolutional Neural Networks.

I. INTRODUCTION

Object detection technology is of paramount importance in computer vision, especially in the domain of vehicle detection, due to its versatility across applications such as security systems, vehicle identification, tracking, and intent prediction. With the evolution of intelligent vehicles, the significance of vehicle detection has surged, becoming a pivotal technology for object recognition. Moreover, the need for swift and accurate vehicle recognition systems holds immense importance for ensuring the safety of autonomous vehicles on the road, as well as for safeguarding pedestrians. The detection of vehicles holds a vital role in enhancing both security measures and the efficiency of traffic management systems along roads and highways. The diverse characteristics encompassing vehicle types, shapes, and dimensions significantly influence the methodologies employed for their detection, tracking, and categorization [1-5].

Contemporary techniques for vehicle detection can be divided into two primary categories: conventional machine learning methods and those leveraging deep learning. The progress of artificial neural networks within the realm of artificial intelligence and recognition systems is rapid. Detection systems can be categorized into single-stage and two-stage approaches. The series of RCNN networks represent a classical approach, while newer fast single-stage approaches include SSD [6] and YOLO series. Within the realm of conventional approaches, there's the concept of R-CNN, originally proposed by Girshick et al., which demonstrates adeptness in object detection. Expanding on this foundation, the Fast R-CNN model emerged, drawing inspiration from the SPP-Net approach [7], aimed at enhancing object recognition. Another significant stride is observed in the form of Faster R-CNN, pioneered by Ren et al., which employs a two-step ROI network-based strategy for improved performance [8-10].

Anchors play a pivotal role in modern object detection techniques such as the YOLO series [11-17], RetinaNet, and EfficientNet. These anchor-based models have proven more effective than traditional sliding window approaches, combining machine learning to enhance accuracy. They find extensive application across various industries, including theft prevention, object localization, intelligent transportation, and more, significantly improving accuracy levels. Unlike conventional algorithms, which struggle under adverse weather conditions like rain or bright light, these state-of-the-art networks exhibit remarkable results in object detection [18-19].

Vehicle tracking stands out as a crucial task within computer vision, involving the identification and tracing of objects across successive image frames. Complementary to detection networks, tracking algorithms like deepSORT and Kalman are widely employed. The intelligent assessment of traffic flow incorporates factors such as vehicle size, speed, and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 15 August 2023 **Revised:** 10 September 2023 **Accepted:** 18 October 2023*

type. The primary focus of this system is swift vehicle counting on highways and the analysis of theft incidents [20].

Various strategies for enhancement, including superior feature extraction with minimal loss of efficiency, enhanced data augmentation, rapid detection speed, and decoupled head and anchor-free detection, are discussed in YOLO-X [21]. PASCAL VOC 2007,2012 and MS COCO 2014 are widely utilized datasets for training and evaluating deep neural networks. These datasets encompass a diverse array of classes, which aids in effective object detection[22-23].

The research paper follows this organization: Section 2 offers a comparison of existing methods along with an exploration of their limitations, while Section 3 outlines the methodology proposed in this study. In Section 4, a series of experiments are conducted using the newly proposed model. The paper culminates with Section 5, which provides the study's concluding remarks.

II. EXISTING METHODS AND RECENT WORKS

In the domain of one-stage object detection methods, several studies have concentrated on recognizing objects and vehicles. This section examines recent research efforts and their associated constraints. Object recognition is a crucial aspect of computer vision. In [24], a vehicle detection method is introduced, utilizing YOLOv5 to identify cars. The results reveal that YOLOv5 incorporates the ACmix attention mechanism, enhancing the model's perception of the target area, adaptability to changing target regions, and the capture of additional information features, among other advantages. Testing reveals an approximate 1% increase in accuracy on both a subset of the BDD100K dataset and the PASCAL VOC2007 dataset following the implementation of enhancements. However, one limitation of this system is its reduced effectiveness in detecting vehicles under low-light conditions.

Road hazards resulting from vehicles create substantial issues for safety, ride comfort, and energy efficiency. In the study outlined in reference [25], a cloud-based platform is introduced, incorporating in-vehicle and on-cloud analysis components. This platform efficiently identifies and precisely locates road hazards. Experimental findings underscore the effectiveness of this approach, demonstrating a substantial enhancement in hazard localization accuracy. Specifically, it reduces localization error from 7.4 meters to 1.4 meters, achieving an accuracy rate of 84%. Although the system is versatile in addressing localization challenges, it faces limitations related to feature extraction and processing speed.

Numerous accidents occur daily due to driver negligence and disregard for traffic laws. In reference [26], this research introduces an innovative approach aimed at addressing these issues in a more technical manner. The system utilizes radio frequency identification (RFID) to track vehicles obstructing emergency vehicles. An onboard camera captures images of intruding cars and scans their RFID tags, with an RFID reader gathering vehicle details. However, this system is limited by factors like occlusion and adverse environmental conditions, such as rain and nighttime.

In reference [27], the VV-YOLO model adopts an anchor frame-based implementation technique. To mitigate instability in anchor frame clustering, caused by random cluster center selection, the modified K-means++ technique is employed. Additionally, the model's loss function is restructured using the focus function to address training data imbalances. Testing on the KITTI dataset reveals a precision and average precision of 90.68% for the VV-YOLO model. It's evident that VV-YOLO performs admirably in detecting objects, even when they are distant or partially obscured. However, it falls short of real-time live detection capabilities.

To illustrate the impact of improvements [28], the yolov4 approach serves as the baseline. The DarkNet backbone network topology is modified to introduce a more efficient backbone network called FBR-DarkNet.. This enhanced method surpasses the baseline network's performance by 4.76 percentage points on the BDD100K dataset, with an 8% improvement in mAP metrics. However, it comes with a high computational cost.

This approach integrates an innovative Yolo-based detection network and LSTM-based location prediction networks, leading to substantial improvements in both accuracy and speed [29]. This enhancement is largely attributed to the spatial semantic attention module (SSAM) with its semantic attention mechanism. The method categorizes vehicle trajectories and employs the LSTM network to predict vehicle positions based on these trajectories. The Fast-Yolo-Rec algorithm not only achieves faster car detection compared to high-speed detectors but also maintains a manageable detection network speed.

However, when evaluating this system on a comprehensive highway dataset, it outperforms conventional methods, except when dealing with vehicles at various angles in frames, where it may occasionally fail to detect some vehicles.

The system's accuracy in real-time detection tends to be poor, primarily due to the distant positions and small sizes of some vehicles. For the successful implementation of Artificial Intelligence driving, it is crucial to ensure the accuracy of real-time detectors and detection inferences. Additionally, it's important to keep the processing complexity of vehicle detection low, especially for functioning effectively on low end mobile systems. This consideration is vital when developing such systems.

III. METHODOLOGIES

The one-stage, anchor-based network consists of a backbone network paired with a detection head. This research builds upon earlier studies focused on vehicle detection [30-31]. In particular, LittleYOLO-SPP and ShortYOLO-CSP incorporate robust feature extractors and detection heads to effectively identify vehicles in challenging environments. However, there remains room for enhancement to ensure its adaptability and robustness in the context of real-time live vehicle detection.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

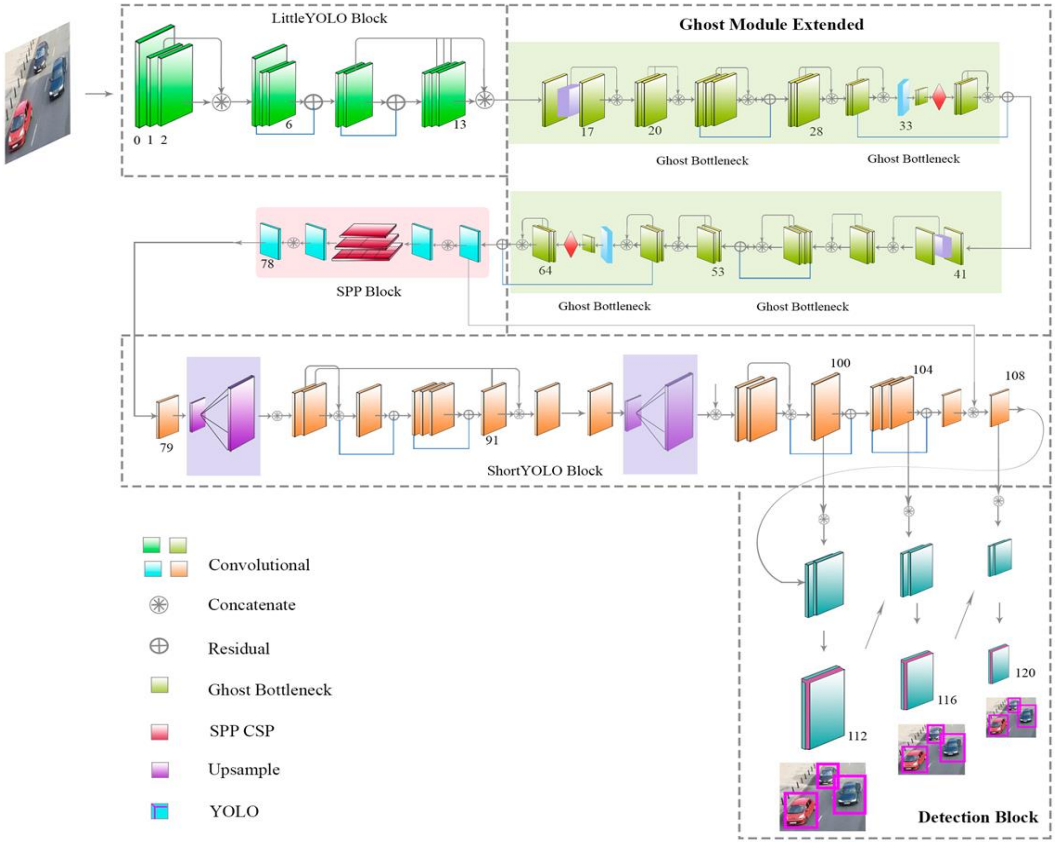

Figure 1. The architectural Layers of ElegantYOLO.

The one-stage, anchor-based network comprises a backbone network, neck, and detection head. The backbone network extracts rich-level features from datasets. The detection head utilizes confidence scores and class probabilities for object classification and localization. This approach builds upon the baseline network of the ShortYOLO system, which is typically implemented using the darknet framework and YOLOv3. The Ghost module is introduced to generate an identical feature map with fewer model parameters.

To enhance efficiency, a deep separable convolution network is employed to complete the output feature map, initially created through traditional convolution. These two feature map components are then merged to yield the desired feature map output. Notably, this approach requires roughly half the convolution parameters compared to conventional methods. Moreover, the Ghost connection block can seamlessly integrate into an existing smaller component. Fig 1 shows the architecture of ElegantYOLO.

ElegantYOLO's notable accomplishments are emphasized below:

1.Enhanced Feature Extraction:
Redesigned ShortYOLO-CSP layers were used to construct an improved feature extraction backbone. This feature extractor effectively captures distinguishing features from various datasets.

2.Innovative Techniques: The proposed work suggested incorporating the Ghost Module Extended, SAHI training technique [32], and [33] Alpha-IoU loss connections with enhanced Spatial pooling layers.

3.Optimized Anchor Boxes: Anchor boxes are generated using the [34] CIoU metric function for precise localization. Swish and leaky activation functions, along with Alpha-IoU loss for regression, optimize the proposed approach compared to state-of-the-art networks.

A. Structure of ElegantYOLO

The YOLO framework has extensive applications, particularly in the field of object recognition networks. The proposed ElegantYOLO model builds upon and leverages certain characteristics from YOLOv3, LittleYOLO, and ShortYOLO. ShortYOLO tends to consume excessive processing time to maintain smooth operation, and ElegantYOLO aims to address and resolve the performance issues encountered by ShortYOLO.

B. Feature extraction blocks

Darknet-53 serves as the feature extraction component in the YOLOv3 framework. Due to its intricate design and numerous layers, it requires robust hardware for efficient processing. LittleYOLO-SPP is a noteworthy adaptation of YOLOv3, offering reduced parameters. In contrast, ShortYOLO takes inspiration from LittleYOLO but introduces

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

larger CSP connection block sets, which provide a moderate reduction in computational demands but may not fully optimize performance on less powerful devices. This is where ElegantYOLO steps in, effectively addressing these issues.

ElegantYOLO comprises 120 layers, consisting of both LittleYOLO and ShortYOLO blocks integrated with Ghost Modules. The module of ghost bottleneck strengthens and lightens the layers of convolution. The primary goal of the ghost implementation is to reduce computational effort along with quick processing. SPP pooling functions are present in the baseline layer, these layers are reinitiated with SPP CSP pooling function to capturing long-range contextual information for pixel-wise prediction [21]. Optimized Vehicle detection is made possible by Slicing Aided Hyper Inference (SAHI), which offers a general pipeline for slicing-aided inference and fine-tuning. Slicing a picture into smaller pieces and executing inference on each piece, then integrating the predictions on the original image, is the underlying idea behind sliced inference. The Ghost modules along with SAHI technique extracts rich level features from dataset which is further used by detection head for processing vehicle class and localization. Table 1 shows the full layerwise details of ElegantYOLO.

TABLE 1. ELEGANTYOLO NETWORK

Blocks	Type	Filters	Size/ Stride	Output
LittleYOLO Block	Convolutional	32	3×3/2	208×208
	Convolutional	64	3×3/2	104×104
	Convolutional	32	1×1	104×104
	Route			104×104
	Convolutional	32	3×3/2	52×52
	Convolutional	64	3×3	52×52
	Convolutional	64	3×3	52×52
	Residual			52×52
	Convolutional	128	3×3	52×52
	Convolutional	64	3×3	52×52
	Residual			52×52
	Convolutional	64	3×3	52×52
	Convolutional	64	3×3	52×52
	Convolutional	128	3×3	52×52
	Route			52×52
2× Ghost Module Extended	Convolutional	64	3×3/2	26×26
	Maxpool		2×2	26×26
	Convolutional	64	3×3	26×26
	Route			26×26
	Convolutional	64	1×1	26×26
	Convolutional		3×3	26×26
	Route	64		26×26
	Convolutional	64	3×3	26×26
	Convolutional	64	1×1	26×26
	Convolutional	64	3×3	26×26
	Route	64		26×26
	Residual			26×26
	Convolutional	32	1×1	26×26
	Convolutional		5×5	26×26
	Route	32		13×13
	Convolutional	32	5×5/2	13×13
	Convolutional		1×1	13×13
	Route	32		
	Avgpool		1×1	1×1
	Convolutional	32	1×1	1×1
	Convolutional	32		
	Scale Layer		1×1	13×13
	Convolutional	64	5×5	13×13
	Convolutional	64		13×13
	Route			
	Residual			
SPP CSP Blocks	Convolutional	256	1×1	13×13
	Route			13×13
	Convolutional	256	1×1	13×13
	Max			13×13
	Route			13×13
	Max			13×13
	Route			13×13
	Max			13×13
	Route			13×13
	Convolutional	256	1×1	13×13
	Route			13×13
	Convolutional	256	1×1	13×13
ShortYOLO Blocks	Convolutional	64	1×1	13×13
	Upsample		2×	26×26
	Route			26×26
	Convolutional	128	3×3	26×26
	Convolutional	64	3×3	26×26
	Route			26×26
	Convolutional	32	3×3	26×26
	Residual			
	Convolutional	64	3×3	26×26
	Convolutional	32	3×3	26×26
	Convolutional	64	3×3	26×26
	Residual			
	Convolutional	128	3×3	26×26
	Route			26×26
	Convolutional	64	3×3	26×26
ShortYOLO Blocks	Convolutional	32	3×3	26×26
	Upsample		2×	52×52
	Route			52×52
	Convolutional	32	3×3	52×52
	Convolutional	32	3×3	52×52
	Route			52×52
	Convolutional	64	3×3	52×52
	Residual			
	Convolutional	64	3×3	26×26
	Convolutional	64	3×3	26×26
	Convolutional	128	3×3/2	26×26
	Residual			
	Convolutional	64	3×3	13×13
	Route			13×13
	Convolutional	128	3×3	13×13
Detection Blocks	Route			52×52
	Convolutional	128	3×3	52×52
	Convolutional	21	1×1	52×52
	YOLO			
	Route			26×26
	Convolutional	256	3×3	26×26
	Convolutional	21	1×1	26×26
	YOLO			
	Route			13×13
	Convolutional	512	3×3	13×13
	Convolutional	21	1×1	13×13
	YOLO			

C. Alpha IoU Loss Functions

The enhancement of detection systems relies heavily on the choice of loss function. In the case of models like LittleYOLO-SPP and ShortYOLO, they have made noteworthy progress by employing loss functions such as GIoU, MSE, and CIoU. Notably, these models have shown the effectiveness of a refined technique known as Alpha-IoU loss, which surpasses the efficiency of CIoU loss. Alpha-IoU loss achieves improved bounding box estimation through the preservation of order and the recalibration of gradients, as outlined in equation (2).

$$\mathrm{IoU}_{\mathrm{loss}} = 1 - IoU \qquad (1)$$

$$\text{Alpha IoU}_{\mathrm{loss}} = \frac{1-IoU^{\alpha}}{\alpha}, \alpha > 0 \qquad (2)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

When utilizing Alpha-IoU during the training of an object detector, it provides the flexibility to attain different levels of bounding box regression accuracy. In the context of the suggested approach, where Alpha-IoU incorporates order preservation and features for reweighting the loss and gradients, it can significantly improve bounding box regression accuracy by amplifying the loss and gradient values for instances with high IoU [22].

D. Ghost Module Extended

The advancement in the field of deep leaning algorithms makes it necessary to build low cost lite weight high performance systems. Hence, A Ghost Module is an image block for convolutional neural network that aims to generate more features by using fewer parameters. It is like a plug and play component easy to incorporate on any detection system. Fig 2(a) illustrates the layers of the ghost module, while Fig 2(b) highlights the key components of the extended ghost module layers. In the extended version, the ghost module undergoes reparameterization through a residual layer just before it is concatenated with a max pooling layer. This reparameterization process is designed to reduce network operations.

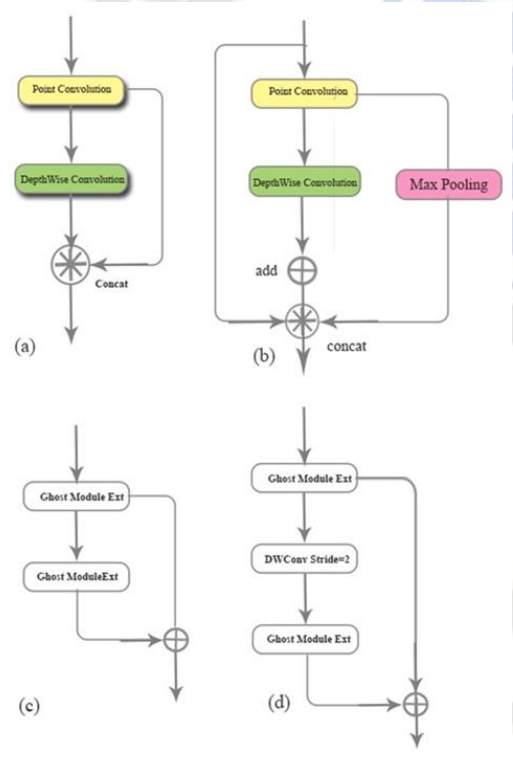

Figure 2. (a) Ghost Module with concatenation layer (b) Ghost Module Extended with both concatenation and residual layers with pooling. (c) Ghost Bottleneck with stride = 1 (d) Ghost Bottleneck with stride = 2

The core component of Ghost module Extended is a stack of Ghost bottlenecks, with Ghost modules serving as the foundation. The first layer is a typical 1×1 convolutional layer, while the subsequent layers consist of a sequence of Ghost bottlenecks with progressively more channels. The sizes of the feature maps used as input for these Ghost bottlenecks are classified into various phases. With the exception of the final bottleneck in each stage, Ghost bottlenecks are imposed with stride=1, and stride=2 as shown in Fig. In some ghost bottlenecks, the residual layer is subjected to the squeeze and excite (SE) module.

E. The detection process of ElegantYOLO

Low-level features are extracted from the dataset through the LittleYOLO block, while high-level features are obtained through the ShortYOLO blocks. These rich-level features are then scaled and assimilated using Ghost Modules. Subsequently, these features are consolidated through a combination of residual and concatenation layers, and the resulting structure is stored as weights. These weights are further utilized within the three-tier YOLO detection layers to identify vehicles.

F. Dataset:

The training datasets include PASCAL VOC 2007 and 2012, as well as MS COCO 2014 [23, 24] from previous experiments. These datasets contain over 100 different types of objects and items. The majority of the classes contained in these datasets, such as humans, animals, and things, are not required for vehicle detection. As a result, all classes other than those associated with vehicles are deleted from the dataset. The ideal network is a vehicle detection system that only covers moving vehicle types such as cars, buses and trucks. Table 2 shows how many images and classes were used in the present study.

TABLE 2. TRAIN AND TEST IMAGE DATASETS.

Dataset	No. of Classes	Classes	Train Images	Test Images
PASCAL VOC 2007, 2012	2	Car and Bus	16,551	4952
MS COCO 2014	3	Car, Bus and Truck	11,432	5545

IV. EXPERIMENTAL RESULTS

The suggested model is evaluated on a variety of datasets, such as MS COCO 2017, and PASCAL VOC 2007. Three classes are chosen for training from these datasets. For testing and determining the effectiveness of the proposed network over current models, the training procedure is carried out using the same way as in the previous networks from LittleYOLO and ShortYOLO. Alpha IoU loss, scaling factor for bounding boxes 1.05, IoU threshold of 0.5, IoU normalisation 0.07, and class normalisation 1.0 are all used by the network in the YOLO detection layers like in ShortYOLO.

The suggested techniques were tested and analyzed on a Tesla P100-PCIE-16 GB GPU with 16 GB of RAM. The outcomes of these experiments were meticulously assessed using various metrics, including mean average precision (mAP), the network's input image size, computational performance (BFLOPS), and inference time (FPS). The system setup for the experiments is detailed in Table 3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

TABLE 3. CONFIGURATION OF THE EXPERIMENTAL PLATFORM.

Computing Machine	Configuration
Operating System	Ubuntu 18.04.3 LTS
GPU	Tesla P100-PCIE-16GB
RAM	16
GPU acceleration library	CUDA10.0, CUDNN7.4

In the first training stage, the network is trained using the PASCAL VOC dataset. The input image's 416 × 416 dimensions have not changed. The hyperparameters of the model are 0.9 momentum, 0.0005 weight decay, and 0.001 learning rate, which optimise the network model and speed up learning. With the help of the multi-scale training strategy provided by this network, the network may be trained using images of diverse shapes and sizes. With training iterations ranging from 100k to 150k, 64 batches are created. After over 120 epochs, training weights are generated. These weights are used to determine the network's mAP and to detect vehicles. The mAP of PASCAL testing is 96.45% which is 5% rise over ShortYOLO network. The network has a recognition time of 0.23 milliseconds and a speed of processing 345 FPS. Once the feature extraction layers are determined and the required feature map sizes are allocated, the network's overall BFLOPS amounts to 5.37, with a weight size of 58.4.

TABLE 4. COMPARATIVE ANALYSIS OF SPEED AND ACCURACY FOR ELEGANTYOLO ACROSS VARIOUS METHODS ON THE PASCAL DATASET.

Networks	AP (car)	AP (bus)	mAP
YOLOv2-tiny	68.15	68.85	68.5
YOLOv3-tiny	77.50	73.0	75.25
LittleYOLO-SPP	79.31	75.57	77.44
ShortYOLO-CSP	89.40	93.25	91.32
ElegantYOLO (Proposed Network)	95.46	97.18	96.45

The MS COCO 2017 dataset is used in the second stage of training with three classes. The network's configurations and settings are identical as in the last training session. The training weights produce a mAP of 79.28%, which is more than a 15% increase over the ShortYOLO. As a result, when compared to other cutting-edge detection networks, it obtains excellent outcomes. Table 5 compares the efficiency of all the networks. Each network in this test is trained with a similar set of hyper parameters and classes. Figure 3 represents the comparison graph of ElegantYOLO trained on COCO dataset.

The proposed network can detect each vehicle with high accuracy by selecting improved bounding boxes. It can detect trucks, vehicles, and cars in congested intersections. From Table 6 the proposed network BFLOPs is 5.37 which is very low processing operational figure when compared with the other networks.

Vehicles can be seen in the test captures driving at night, in the rain, and in congested areas. By selecting appropriate bounding boxes, the recommended system can identify each car with high accuracy. It is capable of spotting automobiles and trucks on slick roadways as well as vehicles at busy intersections. Additionally, in a rainy environment at night, the truck can easily be predicted using the proposed network. The network can also predict the small and blocked tiny vehicles with ease. Videos of different types of road traffic management system frames are used to record the output of image detection. The detection outcomes of ElegantYOLO, which was trained on the PASCAL VOC 2007, 2012 and COCO dataset, are displayed in Fig. 5. The detector performs better in environments that are rainy, mist and dust. The night time white automobile is almost completely not visible on the road. But this detector can correctly classify and find it.

TABLE 5. COMPARATIVE ANALYSIS OF SPEED AND ACCURACY FOR ELEGANTYOLO ACROSS VARIOUS METHODS ON THE MS COCO DATASET

Network	Number of Classes	FPS	mAP
Faster-RCNN	3	198	34.89
EfficientNet Lite	3	85	43.95
Tiny YOLO	3	244	31.44
YOLOv3-tiny	3	220	46.12
YOLOv4-tiny	3	265	57.3
LittleYOLO-SPP	3	49	52.95
ShortYOLO-CSP	3	260	63.27
ElegantYOLO (Proposed Network)	3	355	79.28

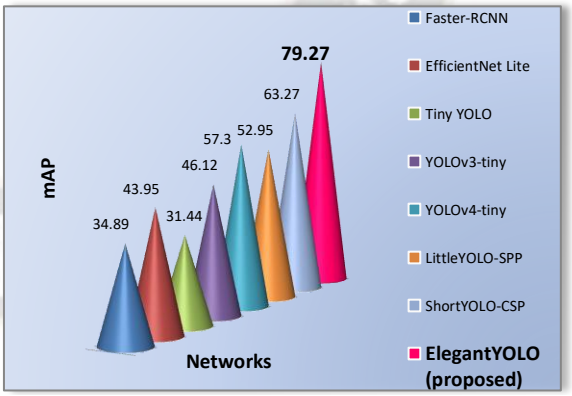

Figure 3. Graph illustrating the performance of ElegantYOLO compared to state-of-the-art detectors on the MS COCO 2017 dataset.

A. Observations and analysis

By mislabeling a bus and failing to forecast some vehicles at late hours, the ShortYOLO network committed blunders. Because of changes in point of view, blockage, and inadequate lighting, these mistakes happen. By increasing the set of data,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *10 September 2023* ***Accepted:*** *18 October 2023*

improving the extraction of features, and using optimisation techniques like Ghost Connections, Spatial pyramid pooling, Alpha-IoU and others, the suggested strategy tackles these problems. In Fig 4, ShortYOLO incorrectly identifies a truck during a rainy nighttime scene as a bus. However, ElegantYOLO addresses and rectifies this error, as demonstrated in Fig 5. The predicted image shows that the ElegantYOLO network, trained with the popularly used datasets, can find vehicles that ShortYOLO-CSP omitted and detected wrongly. For an outcome, compared to the ShortYOLO-CSP network, detection precision and performance have increased.

TABLE 6. COMPARATIVE ANALYSIS OF VARIOUS TECHNIQUES UTILIZED IN THIS RESEARCH

Network	SPP	CSP	Ghost Module	BFLOPS
LittleYOLO-SPP	✓	×	×	16.12
ShortYOLO-CSP	✓	✓	×	23.92
ElegantYOLO (Proposed Network)	✓	✓	✓	5.37

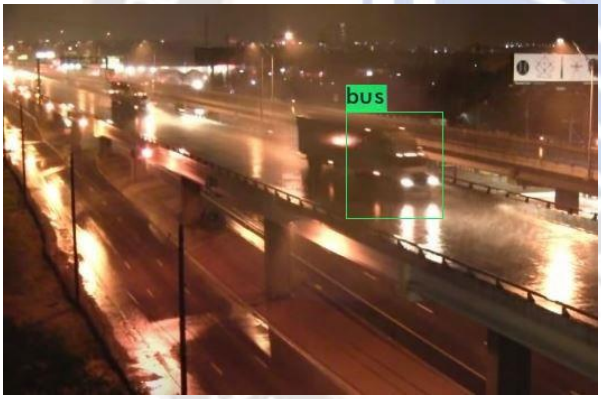

Figure 4. Error Margin in ShortYOLO

Figure 5. ElegantYOLO Rectifies Detection Errors in ShortYOLO

In self-driving vehicles there are many neural networks that can assist with detecting objects, pedestrians and vehicles. To successfully construct the detection network, it is required to assess error data. Comparing LittleYOLO and ShortYOLO the FPS is increased from LittleYOLO which is also not enough during real time rapid detection works . Additionally, ShortYOLO is less accurate. Table 6 provides a further comparison of the essential components of the proposed networks with their predecessors.Thus ElegantYOLO is a better option for real-time vehicle detection when compared to ShortYOLO-CSP because it can operate in a real-time detection environment with a shorter inference time and higher accuracy.

V. CONCLUSION

The paper introduces an optimized Vehicle detection model, building upon an advanced iteration of the LittleYOLO and ShortYOLO series, named ElegantYOLO. The primary goal is to create an accurate system for predicting and precisely locating vehicle objects. This is achieved by integrating a scaled clustering algorithm that allows for the selection of varied anchor box sizes. Modifications to the shortYOLO architecture are implemented to enhance computational efficiency through the integration of Ghost module Extended connections. Enhanced loss functions like Alpha-IOU loss are also adopted to further refine accuracy.

To mitigate the impact of imbalances among background, foreground, and cross-entropy loss, this network employ the focal loss function, which is then normalized using the SAHI technique during training. Subsequently, a multi-layer feature fusion strategy is executed. This involves extracting valuable features and transmitting them to higher layers, consequently enhancing the accuracy of vehicle detection.

Empirical experiments are conducted using the PASCAL and COCO datasets. The outcomes underscore the superiority of the proposed ElegantYOLO model. With a remarkable mean Average Precision (mAP) of 96.45%, the model outperforms other state-of-the-art techniques, firmly establishing its excellence in the realm of vehicle detection.

REFERENCES

[1] Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision(ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. Ademola, O.F., Misra, S., Agrawal, A. (2023). Improving Real-Time Intelligent Transportation Systems in Predicting Road Accident. In: Singh, Y., Verma, C., Zoltán, I., Chhabra, J.K., Singh, P.K. (eds) Proceedings of International Conference on Recent Innovations in Computing. ICRIC 2022. Lecture Notes in Electrical Engineering, vol 1011. Springer, Singapore.

[2] Lu, H., Zhang, Y., Li, Y., Jiang, C., Abbas, H. (2020). User-oriented virtual mobile network resource management for vehicle communications. IEEE Transactions on Intelligent Transportation Systems, 22

[3] Qiu, L., Zhang, D., Tian, Y. et al. Deep learning-based algorithm for vehicle detection in intelligent transportation systems. J Supercomput 77, 11083–11098 (2021). https://doi.org/10.1007/s11227-021-03712-9

[4] Chen, L., Ye, F., Ruan, Y. et al. An algorithm for highway vehicle detection based on convolutional neural network. J Image Video Proc. 2018, 109 (2018). https://doi.org/10.1186/s13640-018-0350-2

[5] Ahmad Arinaldi, Jaka Arya Pradana, Arlan Arventa Gurusinga, Detection and classification of vehicles for traffic video analytics, Procedia Computer Science,Volume 144,2018,Pages 259-268,ISSN1877-0509, https://doi.org/10.1016/j.procs.2018.10.527
[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, ChristianSzegedy, Scott Reed, Cheng-Yang Fu, and Alexander CBerg. SSD: Single shot multibox detector. In European conferenceon computer vision, pages 21–37. Springer, 2016.
[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
[9] Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision(ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
[10] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv 2016, arXiv:1506.01497v3.
[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and AliFarhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computervision and pattern recognition, pages 779–788, 2016.
[12] Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 7263–7271.
[13] Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. arXiv 2018, arXiv:1804.02767 .
[14] Alexey Bochkovskiy, Chien-Yao Wang, and HongYuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020
[15] Ultralytics. YOLOv5: A State-of-the-Art Real-Time Object Detection System. 2021. Available online: https://docs.ultralytics.com (accessed on 10 Feb 2023)
[16] YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. Available online: https://github.com/meituan/YOLOv6. (accessed on 23 March 2023).
[17] Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv 2022, arXiv:2207.02696.
[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, 2017
[19] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of International Conference on Machine Learning (ICML), 2019.
[20] S. Kumar, Vishal, P. Sharma and N. Pal, "Object tracking and counting in a zone using YOLOv4, DeepSORT and TensorFlow," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1017-1022, doi: 10.1109/ICAIS50930.2021.9395971.
[21] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430
[22] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge.International Journal of Computer Vision, 88(2):303–338, 2010.
[23] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll ´ar, and C. L. Zitnick. Microsoft COCO: common objects in context. In ECCV, pages 740–755, 2014.
[24] He, X. (2023). Vehicle target detection algorithm based on yolov5. Frontiers in Computing and Intelligent Systems, 3(1), 56–59. https://doi.org/10.54097/fcis.v3i1.6024
[25] Zhu, X. and Kundu, S., "Road Anomaly Detection and Localization for Connected Vehicle Applications," SAE Technical Paper 2023-01-0719, 2023, https://doi.org/10.4271/2023-01-0719.
[26] Chakravarty, Poonam D., et al. "Emergency Vehicle-Based Vehicle Detection Model." Futuristic Trends for Sustainable Development and Sustainable Ecosystems, edited by Fernando Ortiz-Rodriguez, et al., IGI Global, 2022, pp. 137-146. https://doi.org/10.4018/978-1-6684-4225-8.ch009
[27] Wang, Yinan, Yingzhou Guan, Hanxu Liu, Lisheng Jin, Xinwei Li, Baicang Guo, and Zhe Zhang. 2023. "VV-YOLO: A Vehicle View Object Detection Model Based on Improved YOLOv4" Sensors 23, no. 7: 3385. https://doi.org/10.3390/s23073385
[28] Yu, Q. ., Liu, H. ., & Wu, Q. . (2023). An Improved YOLO for Road and Vehicle Target Detection Model. Journal of ICT Standardization, 11(02), 197–216. https://doi.org/10.13052/jicts2245-800X.1125
[29] N. Zarei, P. Moallem and M. Shams, "Fast-Yolo-Rec: Incorporating Yolo-Base Detection and Recurrent-Base Prediction Networks for Fast Vehicle Detection in Consecutive Images," in IEEE Access, vol. 10, pp. 120592-120605, 2022, doi: 10.1109/ACCESS.2022.3221942.
[30] Sri Jamiya S, Esther Rani P, LittleYOLO-SPP: A delicate real-time vehicle detection algorithm, Optik, Volume 225, 2021, 165818, ISSN 0030-4026, https://doi.org/10.1016/j.ijleo.2020.165818.
[31] Rani, P.E., Jamiya, S.S. ShortYOLO-CSP: a decisive incremental improvement for real-time vehicle detection. J Real-Time Image Proc 20, 3 (2023). https://doi.org/10.1007/s11554-023-01256-0
[32] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. arXiv preprint arXiv:2202.06934, 2022.
[33] J. He et al., “Alpha-IoU: a family of power intersection over union losses for bounding box regression,” (2021).
[34] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU Loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020.