Received: January 22, 2021. Revised: April 22, 2021. 25

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Intelligent Traffic Management Support System Unfolding the Machine Vision Technology Deployed using YOLO D-NET

Divya Jegatheesan[1]* **Chandrasekar Arumugam[2]**

[1]*Department of Information Technology, St.Joseph's College of Engineering, Chennai, India*
[2]*Department of Computer Science and Engineering, St.Joseph's College of Engineering, Chennai, India*
* Corresponding author's Email: divraj.prof@gmail.com

Abstract: The intelligent transport system aims to control congestion and enhance the driving experience through a variety of technologies and communication systems. They provide us a lot of data which can be fed to machine learning technologies to provide further enhanced services to the common public as well as the transport officials. In this busy modern world, the usage of vehicles is highly increased and thus monitoring and detecting these vehicles onboard with better accuracy and less time cost is quite a challenging task. We propose an intelligent traffic management solution to detect vehicles (static and onboard) which finds its set-about in the field of machine vision technology called YOLO D-NET (You Only Look Once Dilated Net). YOLO D-NET puts forward a novel architecture which incorporates the implementation of traditional YOLO model alongside of dilated Convolution Neural Network (CNN). Our paper also focuses on making a constructive comparison between the four different models namely SSD (Single Shot Multibox Detector) Inception, Faster R-CNN (Region-based CNN), YOLO ResNet (Residual Neural Network) and the proposed model YOLO D-NET. We have used the predefined COCO (Common Objects in Context) dataset and the custom dataset to train the traditional models. Also YOLO D-NET makes use of the predefined YOLO weights followed by the dilated CNN layers. Our proposed model was found to be 97.5 percent accurate with enhanced precision and accuracy than the other models particularly with less training time than the traditional YOLO model.

Keywords: Deep learning, YOLO, Dilated CNN, SSD Inception, RCNN, RESNET.

1. Introduction

A fundamental shift towards Intelligent Transport System (ITS) promotes safer, faster way of transport. Furthermore automation can be brought in by automating the actual decisions to improve driver safety and efficiency. Booming up intelligent integrated transport systems in the current era has paved way to active research in both assisted driving system and unmanned driving. Apart from this general motion detection of the front vehicle with the help of on-board vehicle camera, detecting the presence or absence of nearby vehicles, human beings and animals over certain areas and the direction in which the vehicle is moving can be taken into consideration which is known as pre-crashing concept. Such a prediction can widely prevent accidents in the highways and roadways in urban areas. The approach to attain this is to apply vision-based detection methodologies which targets better prediction and high efficiency.

The proposed system could be combined with Intelligent Transport Systems that could be equipped with the underlying traffic or CCTV cameras to assess road conditions where there is no need for human monitoring. The ultimate aim to provide aid for the transport officials and thereby improve the traffic monitoring capabilities that can independently monitor huge lots of visual data of moving vehicles in highways or lanes, extract and analyse the vehicle data.

We suggest a cost-effective access to data that can be used by existing traffic cameras. We remove the need for continuous manual monitoring of hundreds or thousands of camera feeds by collecting and reporting the actionable data to the transport

department. It enables the department of Transport/Officials to concentrate on retaliating against accidents that need urgent attention. It finds its wide range of application in robot vision or machine vision which is an emerging technology in the field of intelligent traffic surveillance. Our method can also be integrated with dashboard cameras and on-board detection of front vehicles, traffic signals, human beings or animals can be achieved which aids to have better and safe driving experience. Fast and accurate object detection algorithms can be employed to overcome all of the faults in the general processing of the vehicle detection.

The proposed method can be summarized as follows: YOLO D-NET focuses on providing an efficient traffic management support system by combining the traditional YOLO model and dilated convolutions to achieve real time vehicle tracking and detection mechanism. D-NET supports dense prediction and our model will be able to detect multiple objects that exists in a frame efficiently with high accuracy in less period of time. A constructive comparison between three other existing models has been made and finally the results are tabulated.

Section 2 discusses on the related work and the workflow of the proposed YOLO D-NET model has been described in Section 3. In Section 4, experimental requirements and results are detailed. Section 5 provides the conclusion and future direction.

2. Related work

Various works related to front-vehicle detection were carried out in the literature based on traditional methods like motion based, model based, multi-sensor based and contemporary techniques based on multiple features, machine learning or deep learning methods. Through traditional methods discussed in [1-3] based on framing difference, 2D or 3D models, radar and vision sensors front-vehicle detection is better achieved with few shortcomings like blurred images, intensive computing, less accuracy and model bound. Recent techniques which are feature based has been discussed in the literature. Multiple features based on directional characteristics were considered in [4]. Haar-like features were used in [5] and vehicle's horizontal edge feature [6] was taken into account to detect vehicles with better accuracy. Spatial characteristics [7] were used to obtain motion characteristics of front vehicle. Feature based techniques prove relatively good results but the accuracy decreases when more vehicles come into picture.

Moving to machine learning techniques, the authors had created classifiers using the Support Vector Machine (SVM) [8] to categorize extracted vehicle features through histogram of oriented gradient method. Deformable part model [9] and a trained AdaBoost classifier by applying Haar features [10] were suggested to detect vehicles. Machine learning techniques have shown better accurate results but involves large datasets to train incurring more cost, time and complexity.

Recently, deep learning techniques are mightily studied and used in computer vision based object detection. A method by combining visual attention mechanism and CNN using reinforcement learning [11] was discussed for front vehicle detection and by using information entropy, classification confidence was evaluated to assist reinforcement learning.

An adaptive hybrid network [12] was suggested which included two levels for feature extraction and multilayer sensing respectively. The first R-CNN model for object detection using deep learning was proposed in [13], which resulted in considerable outputs in terms of accuracy. Feature extraction was done using SIFT operator and Euclidean distance was calculated to categorize the features using KNN (K-Nearest Neighbours) algorithm [14]. Faster R-CNN [15] was analysed to detect and classify the vehicles based on its types with high accuracy in low time consumption. The authors had proposed a novel architecture called SqueezeNet [16] based on deep learning which recognized the make and model of vehicles in an efficient way. An intuitive framework which could perform compressive measurements without the requirement for image reconstruction was proposed [17] and YOLO was used as a deep learning tool for object detection and classification.

The competence of intelligent transport system has been growing day by day. Several works and researches have been carried out for efficient onboard and static vehicle detection.

Traditional methods based on framing difference, 2D and 3D models, sensor based imaging have been implemented which have their own pros and cons. Feature based model selection methods have been instrumented based on temporal, spatial, and horizontal edge features to detect moving vehicles which poses comparatively less accuracy.

Several machine learning algorithms such as SVM, traditional CNN and reinforcement learning based models were proposed which failed to outperform the ultimate aim of accurate prediction.

Followed by which, YOLO model came into picture to accelerate the detection in terms of speed and accuracy.

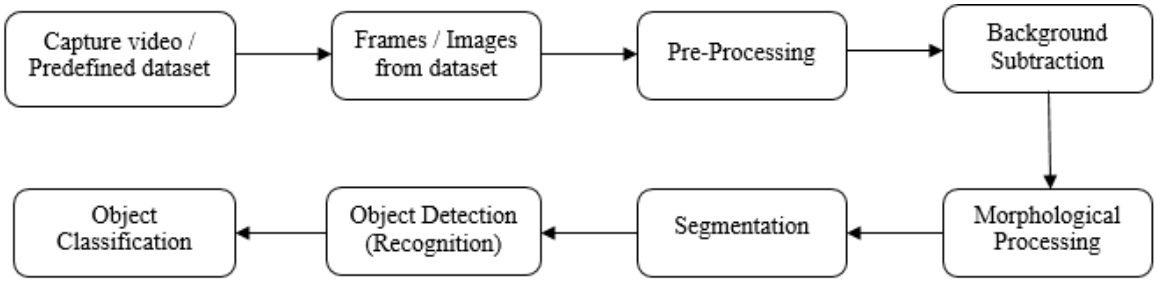


Figure. 1 Consummate workflow of machine vision based vehicle detection

One such of a variant of YOLO is proposed in this paper by modifying the CNN layers with dilation rate and using pre-trained YOLO weights which improves accuracy and training time.

3. Proposed method

Our proposed model incorporates image processing and deep learning with a clear focus to detect the vehicles which are either moving or static. Vehicle detection using machine vision technologies finds its real time applications, with manifestations ranging from personal protection to increased efficacy in the intelligent transport system. The potential results are fathomless with respect to the potential application of cases for intelligent traffic management system. The rationale behind the implementation of YOLO and dilated CNN is that the model brings forth novelty by homogenizing a traditional model with diluted convolutions thereby proposing a surpassed approach for intelligent traffic management and vehicle detection system.

3.1 Pre-processing phase

Images can be obtained either through video capturing the drive way or by creating custom dataset. Fig. 1 shows that how the end to end processing takes place right from video capture or predefined dataset to object detection and intimation to the user.

The custom dataset can be created by segmenting the target video into frames. A minimum of 500 to 1000 images per class will be generated by performing data augmentation strategies and fine-tuning the YOLO network by framing the videos. A frame is considered for a second. Thus, for a 1 minute video, 60 frames will be generated. LABELIMG Tool is used to specify that area using rectangle box and also label for each box. After specifying the labels, every image is saved and converted into the xml file for annotations which is used by YOLO model to detect that object. Then the dataset is separated as training and testing dataset by some ratio like 60:40 or 80:20 or 70:30. Based on training ratio, object prediction accuracy increases. These images can be processed in different stages namely multiple frame conversion, preprocessing of the image frame, segregation of the background images and obstacles, morphological processing, object detection, recognition and classification. The initial phase includes – pre-processing which enhances the image by eliminating unwilling distortions or improving some image features before further processing. Images captured by a camera will vary in dimensions that cannot be directly given as input to the deep learning models. Hence by applying the resize () method helps to modify the image according to the parameters specified as height and width. Also image structures can be enhanced by applying Gaussian smoothing function.

a) Framing

The spatial and temporal characteristics of the input video can reproduce frames with high resolution. A subsampled matrix is constructed by processing high resolution video frames. Video frame extraction algorithm with motion estimation [18]. Either Motion Vectors or camera pan helps in block matching thereby estimating the motion. Motion vector field with $\lambda(l,k)= \frac{10}{|l-k|}$ and Camera pan (average of motion vector field), $\lambda(l,k)=\frac{1000}{|l-k|}$ where λ is the confidence parameter associated with each frame, l and k represents low and high resolution respectively.

b) Preprocessing

Preprocessing includes a set of operations with images at low level of abstraction in order to improve the image quality by suppressing unwanted image distortions. Even though there exists many preprocessing techniques, we have utilized Geometric Transformations so as to eliminate geometric distortions that occurs during image capture. To derive the unknown transformation, several pixels in both the images with known correspondence are used. We find it efficient to use affine transformation with three pairs of

corresponding points to find the coefficients as shown in Eq. (1).

$$a' = i_0 + i_1 a + i_2 b \text{ and } b' = j_0 + j_1 a + j_2 b \quad (1)$$

Also affine transformation includes rotation, translation, scaling and skewing and applying such geometric transformation results in co-ordinate system changes which can be determined by the Jacobean function as mentioned in Eq. (2).

$$J = i_1 j_2 - i_2 j_1 \quad (2)$$

c) Image resizing

Certain outputs from framing step may vary in size which cannot be directly fed as input to our YOLO model. So a default size is defined and all images are resized to exactly match the default size. The input image is resized to a dimension of 448*448.

d) Noise removal

Gaussian blur which is also known as Gaussian smoothing is obtained based on Eq. (3). Blurring an image can be resulted by applying Gaussian function,

$$G(a,b,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{a^2+b^2}{2\sigma^2}} \quad (3)$$

where σ is the standard deviation which determines the blurring effect and a and b are local indices. Gaussian blur reduces image noise and distortions in an image efficiently. Segmentation can be applied for morphological operations. Morphological operation removes imperfections and segments the shape and structure of the object.

3.2 Traditional YOLO model

You Only Look Once framework is studied and applied as it is incredibly fast and comparatively accurate. It can detect objects by taking the whole image as input in a time and computes the coordinates of bounding box and predicts the class accordingly. Generalized object representation can be recognised by YOLO and 45 frames can be processed per second. Also it shows relatively similar performance to R-CNN algorithms on detecting objects.

The training and testing data is obtained by fragmenting the source dataset. Each bounding box has 4 factors namely center, width, height of a bounding box and class of an object. Image will be split into cells, typically a 19×19 grid as shown in Fig. 2. Each cell will be predicting 5 bounding boxes (for more than one object in the cell).

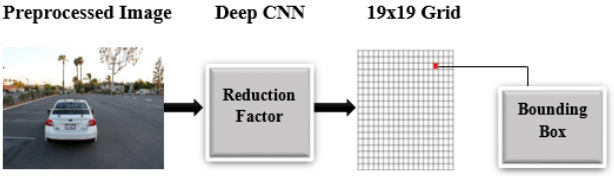


Figure. 2 Bounding box calculation for traditional CNN

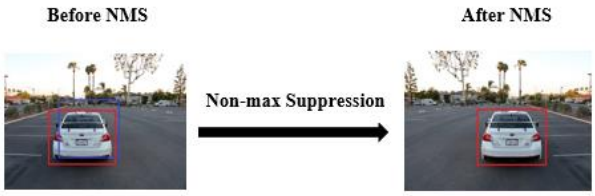


Figure. 3 Applying Non-max Suppression

3.3 Non-max suppression and intersection over union

The ultimate aim of Non Maximum Suppression (NMS) in many computer vision algorithms is to select one entity such as, bounding box, out of many overlapping entities to normalize the bounded boxes. It considers probability number overlap measure as criteria. The bounding boxes which do not have an object within its boundary can be removed and bounding boxes with the highest shared area that includes an object inside can be achieved through non-max suppression.

Intersection over Union (IoU) computes the overlap measure of actual bounding box and predicted bounding box. In Fig. 3, red box is the actual bounding box and blue one is the predicted bounding box. Intersection over Union can be calculated using Eq. (4).

$$IoU = \frac{AoI}{AoU} \quad (4)$$

where AoI denotes Area of Intersect and AoU denotes Area of Union. If IoU value is greater than 0.7, then it can be considered as object and if it is less than 0.3 it cannot be taken as an object.

3.4 Anchor boxes

When multiple objects are present in a single grid, anchor boxes can be applied to improve the performance of YOLO model. Say there are two objects, a human and a car which lies in the same grid and in order to find both the objects, two different anchor boxes can be defined. So instead of obtaining one output, two different outputs can be derived.

The default view of frame size to detect any object or vehicle or human being could be enhanced using YOLO model and IOU. In order to measure the

accuracy of object detection. Intersection over Union is considered to be an important evaluation metric which is used to measure the accuracy of an object detector on a particular dataset.

3.5 YOLO D-NET

Our proposed model YOLO D-NET has been framed as YOLO incorporated with Dilated CNN. As discussed above, several pre-processing techniques were applied to the images. Then the segmented pre-processed images were fed into the detection network. After applying the pre-processing techniques, a deep convolutional dilated CNN network with a dilation rate of (2,2) was applied at the last three layers and was trained with 448×448 input. The results and accuracy were found to remain consistent with the other models.

Dilated convolution is a way to increase the receptive view (global view) of the network by exponential and linear accretion parameters. One general use of dilated CNN is the segmentation of the image where each pixel is labelled by its corresponding class. The state-of-art way of implementing dilated CNN is to apply convolution and then add de-convolution layers to the upsample. It does, however, introduce many more learning parameters. Instead, dilated convolution is used to keep output resolutions high and avoid the need for upsampling. Dilated CNN is applied in our proposed work as detection of fine details by processing inputs in higher resolutions can be obtained. It could be used to yield a fast run time with less parameters.

3.5.1. Why Dilated CNN?

Our main intent to make use of dilated convolutions is dense prediction. In any vision application, the need to integrate information from different spatial scales such as semantic segmentation with each label per pixel, achieve super resolution, denoising, key point detection, and maintain properties such as pixel-level accuracy. Instead of using a multi-scale convolutional neural network, dilated convolutions help to manage the multi-scale efficacy wherein the count of the parameters remains the same.

3.5.2. Dilated CNN model

One of the major drawbacks of pooling in convolutional neural nets is, at times they lead to huge loss of information. This is where dilated CNN can find its place as it can widen the receptive field even without the help of pooling wherein each of the

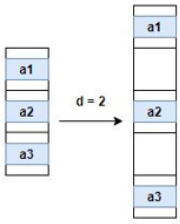


Figure. 4 Dilated convolution layer with dilation rate (2,2)

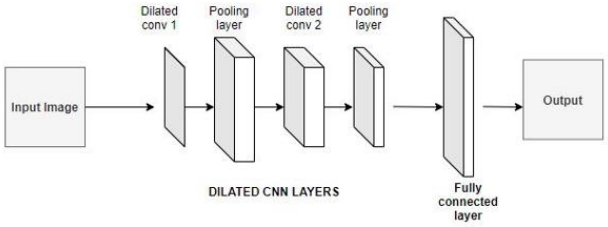


Figure. 5 Workflow of YOLO D-NET

convoluted layer can hold more information. Dilation is applied in the layers to reduce the loss of information thus enabling improved visualization of accurate data. Thus, dilated CNN is applied here for vehicle detection. The dilation methodology in CNN layers is an aberration of the stereotypical convolution [19]. The dilation factor can be denoted as K_d and can be expressed as shown in Eq. (5). Thus for a $K * l$ convolution kernel, the size of K_d will be

$$K_d = K + (K - l)(d - l) \quad (5)$$

where K is the kernel and l is the layer index. The dilation layer can be viewed as shown in Fig. 4.

3.5.3. Workflow of YOLO D-NET

Our proposed model makes use of the predefined yolo weights later to which the inputs are fed into the dilated CNN model. In addition, YOLO D-NET makes use of the traditional CNN network wherein the last three layers are said to be dilated layers with a dilation rate of 2. We intend to build our model as shown in Fig. 5 using an exponentially increasing dilation rate within each block and stack multiple blocks to create a full network in the future works. Each of the input image is of size 448*448.

3.5.4. YOLO D-NET Algorithm

The YOLO D-NET algorithm is defined as wherein the images are input from the dataset, and the pre-processing phase commences where the images are represented as tensors, and passed into the filter kernel. Tensors are n-dimensional array representation of images that constitutes generalisation of matrices. The filters are used to

obtain abstract detailing regarding the objects from images. We could obtain fine-tuned details from a CNN by piling layers of convolutions on top of each other. The convolution layer is the first layer that extracts the different features from the input images. The convolution mathematical operation is performed between the input image and a filter of a specific size MxM in this layer. The dot product between the filter and the sections of the input image with respect to the size of the filter is taken by sliding the filter over the input image (MxM). The feature map is used to feed other layers about the other features of input image. The pooling layer's primary goal is to reduce the size of the convolved feature map in order to reduce computational costs. This is accomplished by reducing the relations between layers and operating independently on each function map. Followed by, the pre-trained YOLO weights which help in obtaining the pre-defined weights from the YOLO network. The pre-defined weights are set to certain values based on the YOLO model. It helps us recognize the traditional images from the default dataset (MS COCO) and also helps in detecting the objects from the new custom inputs. Moreover it facilitates in reducing the training time since the model is trained already and the weights are set. The final layer is the Fully Connected (FC) layer. The weights and biases, as well as the neurons, make up the FC layer, which is used to link the neurons between two layers. The last few layers of dilated CNN Architecture are mounted before the output layer to achieve dilated convolutions. The image I and the kernel k are passed in as input to the convolutional layer followed by which a pooling layer. The last three layers are said to be dilated convolutional layers with a dilation rate of 2. The model is executed with the predefined yolo weights and thus the output tensor is obtained. Based on the above discussion, step by step procedure of YOLO D-NET algorithm can be defined as follows:

Step 1: Input the images from the dataset

Step 2: Pre-process the image to represent the images as tensors
Dim(image) = (n_H , n_W , n_C)
where n_H is the size of height, n_W is the size of width, n_C is the number of channels

Step 3: Define the filter
Dim(filter) = (f,f,n_C)

Step 4: Thus Conv layer of Image I and Kernel K is
Dim(Conv(I,K)$_{x,y}$)= (n_H + 2p – f, n_W + 2p – f)

Step 5: Dimensions of the pooling layer is defined as
Conv(I,K)$_{x,y}$ = (n_H + 2p – f, n_W + 2p – f, n_C)

Step 6: Define the network which hold a convolution layer, activation layers, pooling layer, fully connected layers

Step 7: Define the layers of CNN
Input: $a^{[i-1]}$, padding: $p^{[l]}$, stride $s^{[l]}$Filters: $n_C^{[l]}$, bias $b_C^{[l]}$, activation function (phi $^{[l]}$)

Step 8: The last three layers are said to be dilated with a rate of 2
Input: $a^{[i-1]}$, padding: $p^{[l]}$, stride $s^{[l]}$
Filters: $n_C^{[l]}$, bias $b_C^{[l]}$, activation function (phi $^{[l]}$)
Dilation rate: d (2*2)

Step 9: Load the predefined yolo weights (yolo timy.weights)

Step 10: Output: $a^{[l]}$ with size($n_H^{[l]}$, $n_W^{[l]}$, $n_C^{[l]} = n_C^{[l-1]}$)

Step 11: Thus the tensor obtained at the fully connected layer is
$n_{i-1} = n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}$

4. Results and discussion

The default dataset considered is the MS COCO which contains 91 common object categories with 82 of them having more than 5,000 labelled instances. In total the dataset has 2,500,000 labelled instances in 328,000 images. We have considered the baseline model which holds 121,408 images with 80 classes for predefined object detection.

For the creation of custom dataset, a minimum of 500 to 1000 images per class will be generated by performing data augmentation strategies and fine-tuning the YOLO network by framing the video inputs. A frame is considered for a second. Thus, for a 1 minute video, 60 frames will be generated.

The execution setup required to train the model includes a laptop/PC with hard disk capacity of 40GB and above, internal RAM storage of 512 MB with a Pentium IV processor and webcam/mini camera for the computer vision applications. The model is runnable in any windows operating system with versions such as XP, 7, 8, 8.1, 10. The external libraries required are Numpy and scipy libraries, Open CV, Python imaging library (PIL), YOLO pre-trained weights.

The results of the four experiments conducted were given in detail in this section. 80% of the dataset images were used for training and the remaining 20% were used for testing and evaluation in experiments. The results were found to be accurate and precise in the proposed model. The table shows the constructive comparison between each of the models based on their parameters concerned. We have incorporated a constructive comparison between different object detection techniques based on deep learning such as a) SSD Inception model, b) Faster RCNN, c) YOLO RESNET with our proposed model, and d) YOLO D-NET (YOLO weights with dilated CNN).

a) Model 1: SSD Inception model

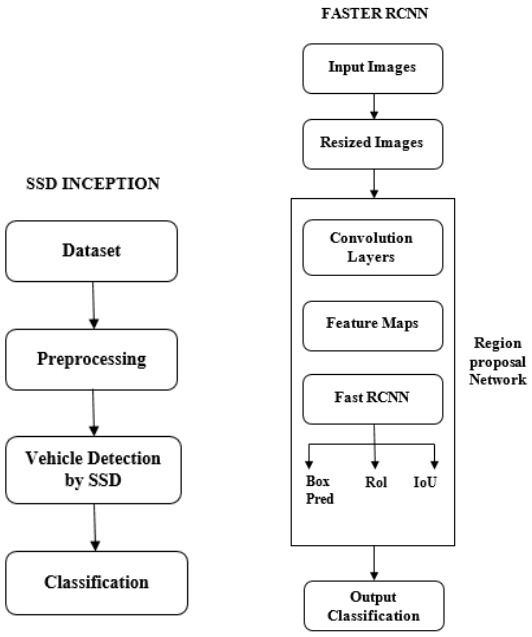
The initial model used to identify the vehicles is the Single Shot MultiBox Detector, SSD Inception model, which is an object detection model to find their location as shown in Fig. 6(a). This model had been trained with the baseline MS COCO and PASCAL VOC dataset [20]. The architecture of SSD builds on the venerable VGG-16 architecture, but discards the fully connected layers. The reason VGG-16 was used as a basic network is due to its strong performance in high-quality image classification tasks and its popularity in problems where transfer learning helps to improve results. Instead of the original fully-connected VGG layers, a set of auxiliary convolutionary layers (from conv6 onwards) has been added, allowing multiple scale features to be extracted and the input size to each subsequent layer to be progressively reduced.

b) Model 2: Faster R-CNN

Faster R-CNN is a network that is widely used in object detection. It is faster than its descendants RCNN and Fast RCNN, as explained by its name. As shown in Fig. 6(b), the state-of-art model holds the first layer that extracts the features from the input image. It preserves the relationship between pixels through the use of small input data squares for learning image features. The feature map operation is performed by retaining the size of the image with stride 1 subsuming ReLU activation and Max-pooling layer. After pooling layer, the feature map matrix will be flattened as vectors. These features are combined together along with the fully connected layer to form a model. The input image is then fed to the CNN to generate convoluted feature map. The RPN is trained to produce high-quality region proposals end-to-end, which are used for detection by Fast R-CNN [21]. Thus the regions of the proposal are identified. It consists of an area proposal algorithm to create "bounding boxes" or positions of possible objects in the image; then a stage of feature generation to obtain features of these objects using CNN and a layer of classification to predict the class to which this object belongs and then a layer of regression to make the bounding box object coordinates more accurate.

c) Model 3: YOLO ResNet

The YOLO ResNet model makes use of the pre-trained ResNet50 network, wherein the classifier layers are replaced with the YOLO classifier layer which serves as a feature extractor. [22] suggested a hybrid network relied on YOLO and RESNET for



(a) (b)

Figure. 6: (a) SSD Inception and (b) Faster R-CNN

multiple object detection. It is based on the radical concepts of replacing the layers of traditional YOLO model with the ResNet layers enhance the efficiency of prediction.

The proposed method makes use of default pre-defined dataset (ie) MS COCO and custom dataset is been created from reference videos. This paper presents a new way for YOLOs to detect object classes, which is significantly faster than recent state-of-the-art solutions. The novelty is greatly induced by the concealed architecture of YOLO and dilated CNN. The model reveals improved performance of detection than the nearest Fast R-CNN competitor, especially on small objects.

There have been several predecessors, no computer vision algorithm has been as fast or as successful at detecting objects in real time as YOLO. Because of its pace, YOLO makes computer vision far more applicable and practical in real-life scenarios. YOLO D-NET is trained on whole images during training, so it doesn't just look at individual items. It not only stores information about a class's presence, but also information about its context. As a result, possible background noise being mistook for an entity has little effect on YOLO networks.

Because of its design and training, YOLO ResNet is a highly generalised network. It not only learns how to recognise objects in isolation, but also how to represent them in general. Thus, it could be further improved by increasing the dilation rate in the neural network layers in the YOLO D-NET architecture.

Table 1. Multi-model tableau based on the evaluation parameters

Parameters	**SSD Inception [20]**	**Faster R-CNN [21]**	**YOLO ResNet [22]**	**YOLO D-NET (Proposed Model)**
Output image size	(8,8) (in inches)	(8,8) (in inches)	(8,8) (in inches)	(448,448) (in Pixels)
Score	0.977	0.973	0.989	0.982
Accuracy	97.5%	97%	98.7%	97.5%
Average precision value	0.971	0.971	0.991	0.973
Average time	2.15	2.81	20.83	2.67
IoU threshold	-	-	0.69	0.67

Table 1 depicts the constructive differentiation between the models based on their parameters concerned. The results obtained according to the proposed method are tabulated.

As can be seen, the speed of implementation of YOLO D-NET has been drastically improved and the detection results with the intended vehicle class gives comparatively better accuracy and performance than the traditional models.

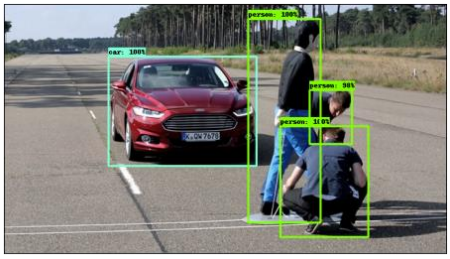
Instead of using the traditional model, the modified dilated layers shows significant improvement in the overall performance of the model.

Fig. 7 provides two different images of the detected vehicles and persons by the major two models considered, YOLO ResNet and YOLO D-NET. Considering image 1 YOLO ResNet gives an accuracy of 0.98 and YOLO D-NET gives 0.975 and for image 2 YOLO ResNet shows an accuracy of 0.985 and YOLO D-NET shows 0.98 respectively. Also IoU box for (a) and (c) is comparatively better than (b) and (d) which depicts that YOLO D-NET is capable of heterogeneous multi-object detection with good IoU threshold.

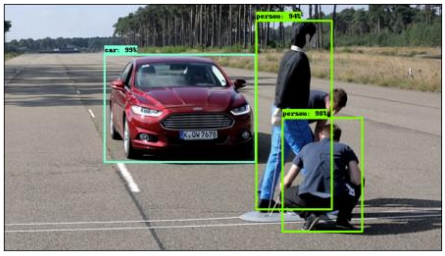
The visualizing cognitive parameters like the score, accuracy, precision and average time taken for the models to predict the outputs are denoted by the graphs picturized in Fig. 8. It is evident that YOLO RESNET model consumes a lot of training time than the other models even though the accuracy is high.

SSD was created with the goal of outperforming the state-of-the-art Faster R-CNN network in terms of mAP. One of the major drawbacks of SSD is that it has default fixed bounding boxes, even if the precision of prediction is high.

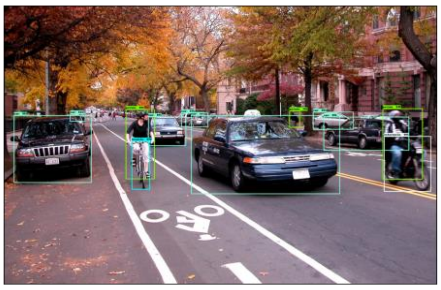
Faster RCNN provides a region of interest for convolution, while YOLO performs detection and classification simultaneously. The Faster RCNN also provides end-to-end instruction, but the measures are



(a)



(b)



(c)

(d)

Figure. 7 Accurate Detection in Image 1 and 2: (a) and (c) YOLO D-NET; (b) and (d) YOLO ResNet

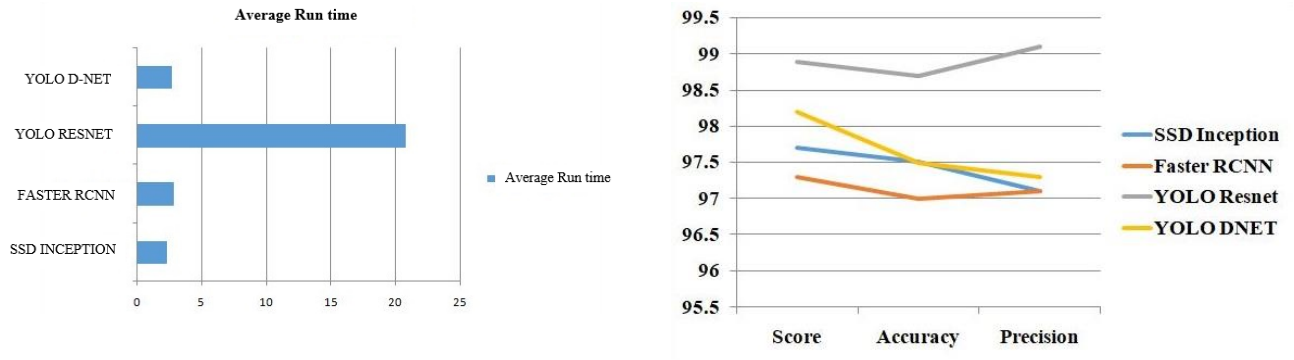


Figure. 8 Comparison of visualizing cognitive parameters of the four models: (a) SSD Inception [20], (b) Faster R-CNN [21], (c) YOLO RESNET [22], and (d) YOLO D-NET (Proposed Model)

significantly more involved. Since it is totally end-to-end educated, YOLO is a cleaner way of doing object detection when compared with other previous models under comparison.

YOLO ResNet is used primarily as a feature extractor for detecting multi-object in the image frames, but it is an inherent of traditional YOLO model.

The proposed YOLO D-NET model is a novel architecture as it helps in vehicle detection on images and videos based on a convolutional dilated network based on pre-defined YOLO custom weights. Thus the proposed method is superior when compared with other models under comparison in terms of speed, accuracy and precision. It is evident that YOLO ResNet model consumes a lot of training time than the other models even though the accuracy is high. The proposed YOLO D-NET model outputs fairly with a decent accuracy and precision and with less training time.

5. Conclusion

Our proposed model aims a fast and real-time detection of the vehicles on-board. The two significant takeaways of this work are that the model 3 (YOLO RESNET) achieves higher quantification with respect to time is 10x times substantial to the proposed YOLO D-NET model. Also, the other models taken into consideration (SSD and Faster-RCNN) does not fair well, as detection of small-scale objects, low-level features like edges and patches are not achieved in limited time. The significant proposed model, YOLO D-NET is found to perform fairly well in terms of accuracy (97.5%), average precision (97.3%), IoU and mainly computation time (2.67s). The IoU threshold reached around 0.67 which indicates the well-performedness of the proposed heterogeneous multi-object detection approach. Our model could detect vehicle images at all angles which finds its applications in vehicle tracking, machine vision, along with number plate recognition and car indicator detection which includes our future scope. In order to increase and improve vehicle detection accuracy in future studies, we intend to achieve higher results and accuracy by collecting more varied dataset images of vehicles and improving the dilated neural network architecture used.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision, and project administration, have been done by 2nd author.

References

[1] Y. Liu., H. Wang, Y. Xiang, and P. L. Lu, “An approach of real-time vehicle detection based on improved Adaboost algorithm and frame differencing rule”, *Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)*, Vol. 41, pp. 379-382, 2013.

[2] W. L. Wang, and Q. U. Shi-ru, “A Vehicle Overtake Accessorial Navigation System Based on Monocular Vision”, *Journal of Image and Graphics*, pp. 21, 2008.

[3] G. H. Chen, X. X. Pan, Z. H. Hou, and J. C. M. C. Faw Faway, “Preceding vehicle detection algorithm based on lane recognition and multi-characteristics”, *Sci. Technol. Eng*, Vol. 16, pp. 245-250, 2016.

[4] J. Zeng, Y. Ren, and L. Zheng, “Research on vehicle detection based on radar and machine

vision information fusion", *Chongqing Automot. Eng. Soc*, Vol. 6, pp. 18-23, 2017.

[5] Q. Xu, F. Gao, and G. Xu, "An algorithm for front-vehicle detection based on Haar-like feature", *Automot. Eng*, Vol. 35, No. 4, pp. 381-384, 2013.

[6] J. Zhang, L. Zhang, and Z. Liu, "Approach to front vehicle detection and tracking based on multiple features", *Jisuanji Gongcheng yu Yingyong (Computer Engineering and Applications)*, Vol. 47, No. 5, 2011.

[7] B. Yang, S. Zhang, Y. Tian, and B. Li, "Front-vehicle detection in video images based on temporal and spatial characteristics", *Sensors*, Vol. 19, No. 7, pp.1728, 2019.

[8] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification", In: *Proc. of IEEE conference on computer vision and pattern recognition*, pp. 3973-3981, 2015.

[9] C. Pan, M. Sun, and Z. Yan, "The study on vehicle detection based on DPM in traffic scenes", In: *Proc. of International Conference on Frontier Computing Springer, Singapore*, pp. 19-27, 2016.

[10] Y. Xiaojiao, G. Jing, X. Kai, and W. Na, "An approach of front vehicle detection based on Haar-like features and AdaBoost algorithm" *Microcomputer & Its Applications*, pp. 13, 2017.

[11] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification", *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 9, No. 4, pp. 356-367, 2016.

[12] G. V. Konoplich, E. O. Putin, and A. A Filchenkov, "Application of deep learning to the problem of vehicle detection in UAV images", In: *Proc. of XIX IEEE International Conference on Soft Computing and Measurements (SCM)*, pp. 4-6, 2016.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", In: *Proc. of IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014.

[14] A. P. Psyllos, C. N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, pp. 322-328, 2010.

[15] X. Wang, W. Zhang, X. Wu, L. Xiao, Y. Qian, and Z. Fang, "Real-time vehicle type classification with deep convolutional neural networks", *Journal of Real-Time Image Processing*, Vol. 16, No. 1, pp. 5-14, 2019.

[16] H. J. Lee, I. Ullah, W. Wan, Y. Gao, and Z. Fang, "Real-time vehicle make and model recognition with the residual SqueezeNet architecture", *Sensors*, Vol. 19, No. 5, pp. 982, 2019.

[17] C. Kwan, D. Gribben, B. Chou, B. Budavari, J. Larkin, A. Rangamani, T. Tran, J. Zhang, and R. Etienne-Cummings, "Real-Time and Deep Learning Based Vehicle Detection and Classification Using Pixel-wise Exposure Measurements", *Electronics*, Vol. 9, No. 6, pp. 1014, 2020.

[18] Q. Shan, Z. Li, J. Jia, and C.K. Tang, "Fast image/video upsampling", *ACM Transactions on Graphics (TOG)*, Vol. 27, No. 5, pp. 1-7, 2008.

[19] Y. Lin and J. Wu, "A Novel Multichannel Dilated Convolution Neural Network for Human Activity Recognition", *Mathematical Problems in Engineering*, 2020.

[20] C. Ning, H. Zhou, Y. Song, and J. Tang, "Inception single shot multibox detector for object detection", In: *Proc. of IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 549-554, 2017.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2016.

[22] Z. Lu, J. Lu, Q. Ge, and T. Zhan, "Multi-object Detection Method based on YOLO and ResNet Hybrid Networks", In: *Proc. of IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 827-832, 2019.