ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCED VEHICLE DETECTION USING POOLING BASED DENSE-YOLO MODEL

SRIHARSHA VIKRUTHI[1], DR. MARUTHAVANAN. ARCHANA[2], DR. RAMA CHAITHANYA TANGUTURI[3]

[1]Research Scholar, Department of Computer Science & Engineering, Annamalai University, Chidabaram, Tamil Nadu-608001, India.
[2]Assistant Professor, Department of Information Technology, Annamalai University, Chidabaram, Tamil Nadu-608001, India.
[3]Professor, Department of Computer Science & Engineering, PACE Institute of Technology and Sciences, Andhra Pradesh-523272, India.

E-mail: [1]sriharsha.vikruthi@gmail.com , [2]archana.aucse@gmail.com, [3]trchaitanya@gmail.com

ABSTRACT

The field of transportation management is seeing a rise in the importance of intelligent transportation detection and counting. Yet, detecting them remains difficult because of the wide range in vehicle sizes, which has a direct impact on the precision of vehicle counts. In this research, we present a vehicle counting and detection system that uses computer vision to solve this problem. The Intelligent Transportation System (ITS) has been developed through the enhanced optimization tool which belongs to the conventional time prediction framework, normally used in industry. The non-parametric approach has been developed with the help of machine learning approach and enhances the computational power. Huge historical information is used to generate the accuracy through traffic pattern with non-parametric approach. Additionally, this approach needs strong dataset, system for training and testing. The main problem with the current concept is that it takes too long to detect vehicles using machine vision. In order to alleviate traffic and safety concerns, the proposed model centres on precise vehicle recognition for usage in the intelligent transport system. The core components of the technique, which include convolution layers, pooling layers, and fully-connection layers, provide it exceptional robustness and allow it to efficiently complete the task of vehicle detection. The proposed pooling based dense- You Only Look Once (YOLO) model is constructed to enhance the accuracy in vehicle detection and eliminating the vanishing gradient issue. The pooling strategy is used to pool and update the multi-scale region features for enhanced detection. The loss function is established with MSE and the performance analysis shows that the proposed technique is performed well on UA-DETRAC dataset.

Keywords: *Intelligent Transportation System, YOLO, accuracy, vanishing gradient, UA-DETRAC dataset.*

1. INTRODUCTION

The intelligent transportation system has been focused on the usage of latest information technology to implement the resource allocation strategy for enhancing the public service capacities [1]. The ITS framework has the organization of the traffic prediction and enhances the resource utilization. The framework can observe the total amount of potential vehicle based communication network using cloud storage [2]. The main application of traffic prediction has the most accuracy and efficiency for prediction using time series framework, normally utilized in industry [3]. The conventional methods are not able to predict the complex issues that will cause the unsolvable problems. The deep neural network technique has been implemented for processing the prediction with enhanced outputs. The recurrent neural network has been utilized that will share the values within the time period but couldn't manage the long term issues through the existing data [4].

Extracting target features is the initial step in conventional machine learning approaches, and this is typically done using techniques like the histogram of oriented gradient (HOG) [5], the scale invariant feature transform (SIFT) [6], or the local binary pattern (LBP) [7]. Once the features have been retrieved, they are put into a pretrained classifier, such as a support vector machine (SVM) [8] or an iterator of Ada Boost (AB) [9]. Since data size is always a constraint in identification tasks, the extracted features may resist with generalization capacity, making it hard to achieve precise recognition in the real world.

In subsequent years, optimizations to the network led to the development of methods such as SPP-Net [14], Fast RCNN [15], Faster RCNN [16], and YOLO [17]. A new technology called YOLO employs a deep convolutional neural network pooling to detect targets in real time (CNN). However, deep convolutional neural networks are able to efficiently overcome the challenges created by the ever-changing appearance of cars since they are invariant to geometric change, deformation, and lighting, unlike classic feature extraction algorithms. As a result, the feature description can be built adaptively under the training data, demonstrating increased adaptability and generalizability.

The vehicle detection techniques according to the computer vision which has been applied in several applications like medical diagnosis, security. The detection techniques have the issues of reduced accuracy, high amount of execution time. The feature extraction techniques with LBP [5] and HOG [6] have been implemented for object detection combined machine learning functionality. The Deformable Parts Model [7] has been produced the improved accuracy through the enhanced object detection functionality according to the machine learning concepts and these kinds of techniques are required the concept of scanning with the sliding window concept. The Convolutional neural networks [8] have been implemented to provide object images may be classified through the feature extraction functionality with more accurately. Features are used to detect the vehicle and vehicle detection model is shown in Figure 1.

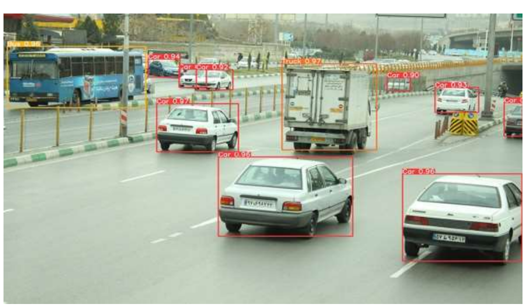


Fig 1: Vehicle Detection Model

Currently, the deep learning based techniques have been widely used for object detection to produce the desired results. The inefficient detection through the sliding window is the main issue for object detection, the enhanced methodology of R-CNN [9] has been implemented to provide the solution for the inefficiency. The related techniques of CNN model are R-FCN [10], R-FCN-3000 [11], RRPN [12] which has the enhanced accuracy for object detection. Moreover it is very hard to produce real-time detection through these kinds of techniques, so the region and subsequent based feature extraction are improved effectively. This paper proposes deep learning based non-parametric framework that produces enhanced accuracy, and computational cost. The pooling process used in vehicle detection process is shown in Figure 2.

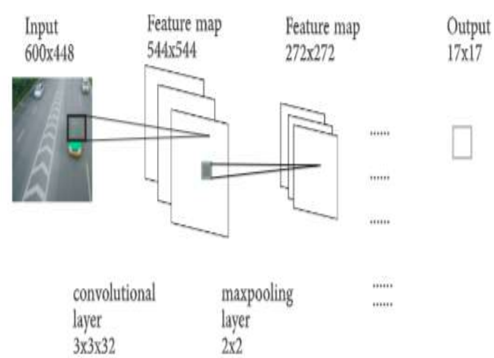


Fig 2: Vehicle Detection Using Pooling Process

Modern transportation has been drastically altered by the integration of information and communication technology into vehicle cabins and the underlying infrastructure. These aids improve traffic flow by shortening commute times and decreasing congestion; they also help police identify and ticket offenders; they provide drivers with assistance; they lessen the likelihood of accidents; and they lessen the severity of any damage that may occur as a result of an accident. These applications place requirements on the system as a whole, calling for trustworthy, specialised hardware and quick, efficient communication. As a result of its low price, simple maintenance, and capacity to record high quality images of the traffic scene, camera-based video surveillance is the foundation of most traffic management systems. In this way, information may be shared and gathered between vehicles, as well as between vehicles and transportation infrastructure, to improve both the safety and comfort of road trips. However, in typical situations, these systems can be relied upon. In reality, under certain conditions they might not function well.

The improved framework is created using a thick connection strategy, which also improves detection accuracy. A pooling framework is created to pool and update the regional characteristics on reduced location error when recognising microscopic items. Faster and more precise categorization is possible thanks to a loss function that is kept constant regardless of where it is used. Different performance evaluations using hyper parameters and priority values have been applied to the UA-DETRAC dataset.

2. RELATED WORKS

The YOLO object detection network was proposed by Lu et al. [1]. It views object identification as a regression problem and employs an end-to-end framework to make direct predictions about categories and locations. Guo et al. [3] developed an enhanced version the dubbed YOLO9000. This new version included anchor boxes to aid the detecting head in predicting the target box, as well as batch normalisation (BN) to mitigate overfitting. The most recent iteration of the YOLO object detection algorithm, YOLOv5, replaces the original backbone with CSP-DarkNet and incorporates data augmentation techniques like mosaic to greatly increase the algorithm's accuracy and performance.

Using convolutional neural networks, Kong et al. [4] were able to build vehicle-like areas from the extracted features of different layers in the backbone and pool the features of the shallow and deep layers, which aids in the detection of small objects. To counteract the feature loss introduced by deep convolutional neural networks, Ali et al. [6] implemented cross-layer skip connections. Since Srivastava et al. [7] shown that the state-of-the-art region proposal network (RPN) was ineffective for detecting small objects, the RPN network was employed to do so, with the addition of the fast R-CNN enhancement. The aforementioned scientists have examined network architectures extensively. The preceding techniques aren't very good at improving the vehicle detection method of high-resolution photos because of the convolutional neural network's stringent limitation of the input size.

Adaptive neural network, was proposed by Bouguettaya et al.[8], splits the last layer into many networks in order to extract characteristics at different scales. In comparison to more conventional approaches, it is far more effective. To perform vehicle detection, Cao et al. [9] enhanced CNN and developed a unified multi-scale deep CNN (MS-CNN), which was split into two sub-networks: the object proposals network and the detection network. The outcomes demonstrated significant enhancements in precision together with markedly better memory and calculation. The MS-CNN is capable of performing detection at a rate of 18 frames per second. The models increases the delay levels and consumes more time for processing that need to be reduced.

Faster R-CNN has been successfully applied to vehicle detection proposed by Laopracha et al. [11]. Faster R-CNN, VGG16, and ResNet-152 were coupled to recognise vehicles; the result was good accuracy, but the detection time was too long to meet the needs of real-time traffic monitoring. In most cases, deep learning-based algorithms are too sluggish to be considered real-time. There is a need for better accuracy in detection as well as in the ability to generate. Pan et al. [12] introduced the end-to-end object identification method YOLO by converting direct object detection to regression in order to enhance the speed and precision of region-based object detection methods. Boukerche, et al. [14] proposed the YOLOv2 object detection model to dramatically enhance the speed of object recognition while maintaining detection accuracy.

The object detection process has been improved with the long short-term memory (LSTM) framework that the architecture has specific gate to control the data. It is utilized to generate the spatial attention framework to predict the objects through the longitudinal velocity. The multiple model is generated the social LSTM framework through the trajectory model for predicting the objects. The encoder related neural network framework has the enhanced procedure for object detection in efficient way. The initial regression related technique YOLOv2 [14] is constructed to predict the coordinates using the bounding box and perform the classification through the neural network concept. The main issue of YOLO model is to detect the tiny objects in real-time classification. Another model SSA has been introduced the object detection using reference boxes; the multi-scale feature extraction is used for producing the enhanced accuracy of object detection.

The YOLO model has been used the Darknet framework [15] but the performance is low for providing the feature extraction and couldn't utilize the multi-scale feature extraction concept. Consequently, the ResNet [16] has been employed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

with DSSD [17] framework to produce better accuracy. The complex network has been involved to reduce the detection speed, so that the DesNet [18] has been employed to get enhanced accuracy and speed. The backbone network has been involved the specific role in vehicle detection through the feature detection through the feature extractor as the accuracy is related with the performance.

The VGG network has been employed as a full of Convolutional kernels for enhancing the network through Faster R-CNN framework. Moreover, the enhancement of the huge Convolutional layers, the communication within the Convolutional layers have been exposed the vanishing gradient. The ResNet has been strengthened for reducing the vanishing gradient related issues. The DenseNet [19] through the dense connection framework of layers has been introduced for producing more accuracy while completing the image recognition task.

The multipath framework with Xception [20] network has been designed for producing the backbone network as the enhanced R-CNN model. Additionally, the MobileNet has been compressed the network for producing highest speed. The ShuffleNet [21] has been utilized to construct the backbone network for detecting the vehicles. The fusion based feature maps have been improved the accuracy through the multi-scale detection strategy as the fusion data from dissimilar scales. The pooled maps with arbitrary features have been maintained the feature vectors of a constant size while the CNN model didn't maintain the size of input images for multi-scale feature extraction.

The FPN [22] model has been implemented the deep CNN framework to enhance the multi-scale detection accuracy as the PPN [23] model has been introduced to avoid the conventional framework for maintaining the minimized computational complexity for fast detection. These techniques have been the independent values on dissimilar feature maps through the fusion based multi-scale feature extraction while using the common features from dissimilar layers. Moreover, these techniques didn't utilize the common region features on dissimilar scales from the Convolutional layers as detecting the tiny objects in more accurate manner through high region feature concept.

3. PROPOSED WORK

The proposed dense-YOLO contains the convolution blocks; a dense connection with pooling block and max-pooling layers with object detection has been demonstrated in Fig. 3. The object detection is constructed to get the output with more accuracy. The proposed framework has the convolution block which minimizes the feature maps by 1/32 of the given input image and the total amount of feature maps are enhanced by 512 through the feature extraction.

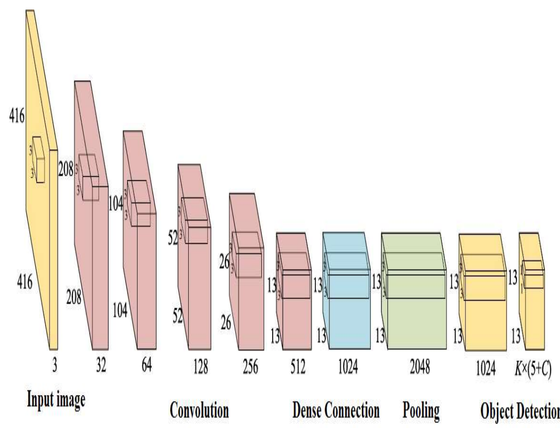


Fig. 3 Dense-YOLO Framework

The dense connection has the components generated by 1 x 1 and 3 x 3 connected layers through the enhancement of feature maps that have been generated as 256, 512. The total amount of feature maps as the output which is minimized by the filters of 3 x 3 x 1024. The dense connection framework is demonstrated in Fig. 4.

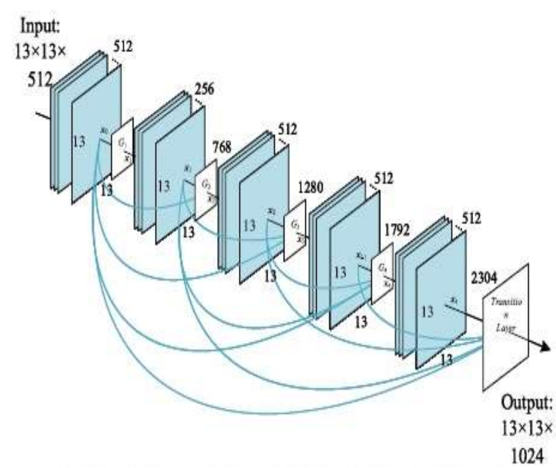


Fig. 4 Dense Connection

The pooling block contains different kinds of max pooling layers for implementing the feature extraction and pooling convergence. The input feature maps have been reduced the window size as 1024 to 512. The sliding window sizes have been pooled to get the output of feature extraction as the object detection has the reconstructed feature maps with high resolution which is demonstrated in Fig. 5.

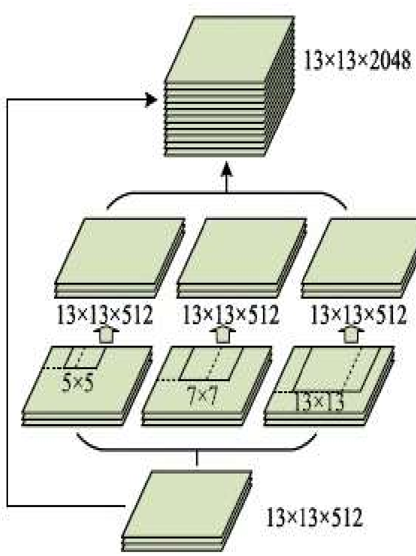


Fig. 5 Pooling

Initially, performing the pre-processing procedure has been implemented through the over fitting function for enhancing the objection detection. The generation of the anchor boxes has been identified with the Intersection over Union (IoU) function within the bounding boxes of the training samples and it is computed in Eq. (1).

$$distance_c^b = 1 - IoU_c^b \quad (1)$$

Where b demonstrates the bounding box, c denotes the centroid value. The parameters of training have been initialized to implement the CNN model. The loss function is computed through the summation of squared errors while performing the classification. The dense-YOLO model is constructed for performing the object classification process. Each boundary box is demonstrated in Eq. (2).

$$bb = [bb_a, bb_b, bb_w, bb_h, bb_c]^T \quad (2)$$

Where (bb_a, bb_b) is the coordinates for the bounding box, bb_w is the width, bb_h is the height, and bb_c is the confidence value for the bounding box. The sigmoid function is used for every bounding box and the ground truth is computed in Eq. (3).

$$gb = [gb_a, gb_b, gb_w, gb_h, gb_c]^T \quad (3)$$

The output classification for every box is demonstrated in Eq. (4).

$$Cl = [Cl_1, Cl_2, \dots \dots, Cl_l]^T \quad (4)$$

The predicted probability is computed using Eq. (5).

$$Pp = Pp(Cl_l)_{l \in C} \quad (5)$$

The loss function is framed with CNN by training with the coordinate values of squared error function for classifying object detection. The cross entropy value is used for representing the classification while loss value has been maintained by training model and the loss function is computed in Eq. (6).

$$LF(x, Cl) = \lambda_n \sum_{i=0}^{n} (gb_n - Cl_n)^2 \quad (6)$$

The prediction value of every grid into several bounding boxes has the anchor values which produces the instability. The stability has been improved with the training model; the computation of the loss value within the bounding boxes couldn't produce the improved predictions. The prior box value is computed in Eq. (7).

$$Po = [Po_a, Po_b, Po_w, Po_h]^T \quad (7)$$

The accuracy of object detection using the proposed technique utilizing the loss function with the dataset. The convergence rate is computed through the epoch values and maintaining the network model for enhancing the accuracy and convergence while training the dataset. The YOLOv2 technique has been constructed with segregating the image into grids, every grid discovers the bounding boxes and the confidence is computed in Eq. (8).

$$Bc(Cl_i|obj) * Bc(obj) * IoU_p^t = Bc(Cl_i) * IoU_p^t \quad (8)$$

Where $Bc(obj) * IoU_p^t$ is the bounding box confidence value which consists of the objects, IoU_p^t is the IoU value within the ground truth and the prediction. Hence, the value of prediction has the encoded value using tensor parameters. The dense-YOLO is constructed to extract the features using the Convolutional framework which is same as the neural network concept. The relationship within the m^{th} and $(m-1)^{th}$ layer is computed in Eq. (9).

$$a^m = fn(b^m) = fn(a^{m-1} * wt^{m-1} + bias^m) \quad (9)$$

Where a^m is the input of the m^{th} layer, the activation function is demonstrated as $fn(b^m)$. The intermediate variable is denoted as (b^m), wt^m is the weight value of the m^{th} layer, $bias^m$ is the bias value and $*$ demonstrates the convolution operation. The propagation based gradient function is computed in Eq. (10).

$$\psi^{m-1} = \frac{\partial M}{\partial b^{m-1}} \quad (10)$$

The loss function is propagated with the derivative function and weight parameters to produce the output as better accuracy.

Algorithm 1 – Object detection

Step 1: Segregate the input image as grids.
Step 2: Implement CNN model to complete the feature extraction
Step 3: Construct the class values using the predicted box values
Step 4: Identify the bounding boxes through the threshold value.
Step 5: If the value of the bounding box is greater than the threshold value then it contains the object else doesn't contain the object.
Step 6: Discover the object according to the probability value.
Step 7: Implement the suppression process by removing the repeated boxes.
Step 8: Compute the object detection as the output

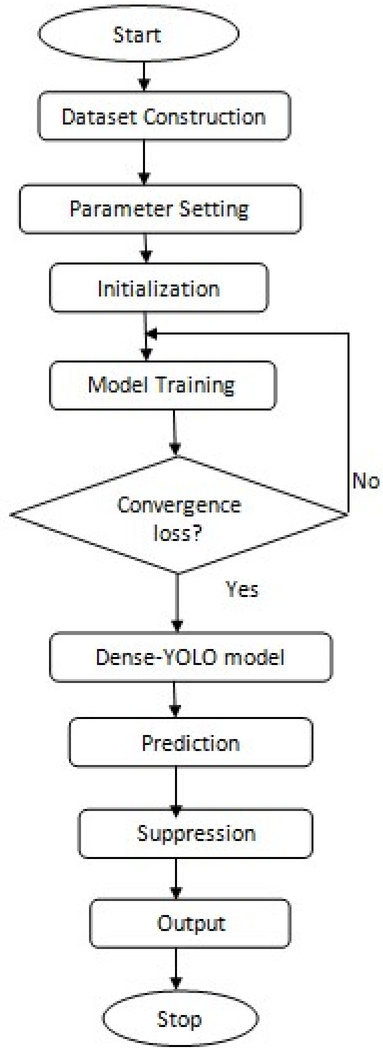


Fig. 6 Object Detection Procedure

The YOLO related object detection is fully based on the multi-scale detection that the several feature maps have been extracted through several layers with the fusion features are extracted through dissimilar layers in the network. The small objects are extracted the feature maps through the highest layers. The back propagation process has been used to enhance the accuracy of classification and minimizes the entire training time. This process has been framed to diminish the repeated training procedure while implementation. The dissimilar detectors are gathered at the similar period which is directly proportional to the real-time traffic flow. The fine tuning components into the entire network have been allowed to perform reusability of the trained model.

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 2 – Back propagation process

Input: Group of untrained components, dataset, threshold value, epochs.
Output: trained components
Step 1: Train the basic model with the dataset and the components of untrained models.
Step 2: Copy the parameter values into the model.
Step 3: Test the model with the predicted error of the dataset.
Step 4: If the predicted value is greater than the threshold then update the model with back propagation process through the epochs.
Step 5: If step 4 is not satisfied then go to step 2.
Step 6: Return the trained models from the back propagation model.

The execution time for performing the back propagation process with the proposed technique, a model is trained through the dataset that enhanced the accuracy with the proposed framework. The computational complexity is the correlation with the different layers and the hidden units in every layer that needs more training and predicting period. The time cost should be high if the system requires generating a model for every detector and training every model on a specific dataset to confirm the entire system's prediction accuracy. Additionally, the refinement layer has been used to restrict the overfitting risk whenever implementing extra training loops to enhance the accuracy with limited time.

4. PERFORMANCE EVALUATION

The Darkflow coding is used with the Python background and constructed as the YOLO model. The enhanced dense-YOLO model is constructed for object detection with the modified parameters which contains the convolution procedure and layers. The proposed model has no fully connected layers that the input images processed through the Convolutional layers with ReLU activation function. The kernel size is the important parameter that the filter values from the initial Convolutional layer to the final layer. The filter in the Convolutional window has the stride value with width and height values, the batch normalization process have been initialized through the activation function. Each class has the filter value and total amount of layers by adding one by one that the bounding box consists of the anchor boxes for detecting the objects from input images. The proposed technique is compared with the relevant techniques of R-CNN [9], LSTM [13], and YOLOv2 [14].

The UA-DETRAC dataset [24] is used for conducting the performance evaluation that 82.500 vehicle images from the real-time cameras which belongs to perform the testing, training and validation. The proposed dense-YOLO model has the ReLU parameter with the loss function which has the same framework as the YOLO model. The loss function is used to conduct the experiments through the hyper parameters and the total amount of priority values is also increased. The adaptive estimation values are implemented to improve the network weights. The GPU memory is used as the batch size that will be very high and 145 epochs have been trained as the initial learning rate of 0.0010 related with the dataset and the experiments. The accuracy has been measured using mAP parameter and the speed parameter in fps which is demonstrated in Fig. 7.

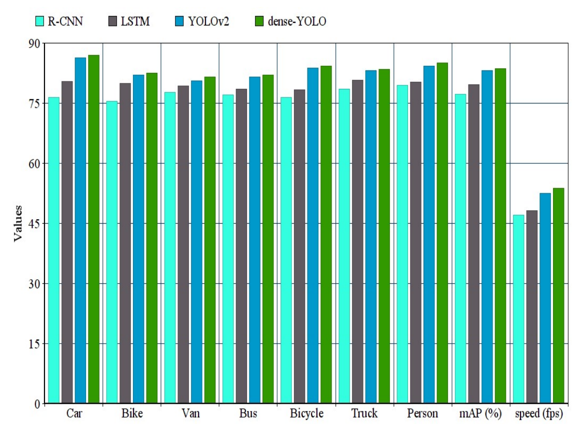


Fig. 7 Vehicle Detection Accuracy

The data smooth procedure is utilized for computing the moving average function with limited window size. The data preprocessing is computed in Eq. (11).

$$\delta_{std} = \frac{(\delta - \delta_{min})}{(\delta_{max} - \delta_{min})} \quad (11)$$

Where δ is the original data, δ_{min} is the minimum value, δ_{max} is the maximum value and δ_{std} is the scaling data in normalization. The scaled value is computed in Eq. (12).

$$\delta_{scaled} = \delta_{std}\ x\ (\delta_{max} - \delta_{min}) + \delta_{min} \quad (12)$$

The normalization maintains the record values lies within the range of 0 to 1, it is the best model for

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

coverage. The Mean Squared Error is computed in Eq. (13)

$$MSE = \frac{1}{x}\sum_{i=1}^{x}|\gamma_i - \hat{\gamma}_i| \qquad (13)$$

Where $\hat{\gamma}_i$ is the prediction result and γ_i is the traffic flow value, the root mean squared error is computed in Eq. (14).

$$RMSE = \sqrt{\frac{1}{x}\sum_{i=1}^{x}(\gamma_i - \hat{\gamma}_i)^2} \qquad (14)$$

RMSE performs better in huge data that will use to ensure the performance on unstable circumstances with huge amount of time. R^2 is ensured to identify the accuracy of the system for maintaining the horizontal comparison, it is computed in Eq. (15).

$$R^2 = 1 - \frac{\sum_{i=1}^{x}(\hat{\gamma}_i - \gamma_i)^2}{\sum_{i=1}^{x}(\overline{\gamma}_i - \gamma_i)^2} \qquad (15)$$

The performance comparison for the proposed technique is compared with the related technique of the parameters like MSE, $RMSE$, and R^2 which is demonstrated in Fig. 8.

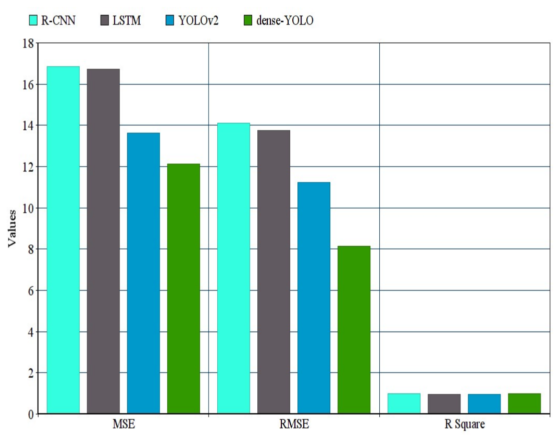


Fig. 8 Performance Comparison

The system's training and prediction time could be utilized to identify the exact efficiency of the model. Every model has been trained on the dataset; the method has to complete the prediction of the traffic flow within the specific interval. By performing the random stuffing, 70% of the dataset has been trained by the cross validation technique for avoiding the deviation in Fig. 9.

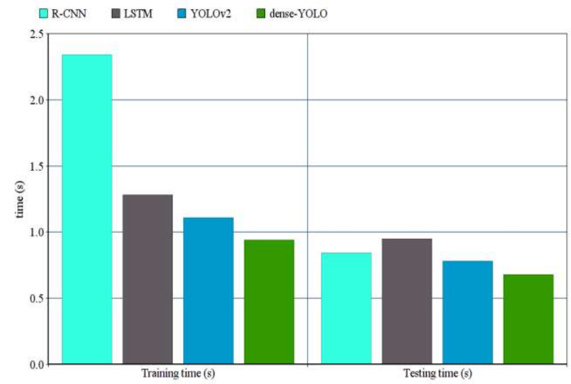


Fig. 9 Training And Testing Time

Fig. 10 demonstrates the training time for the various techniques for the dataset that can be computed in 3 different levels at the normal stage, dense connection stage and the back propagation stage. The convergence procedure of the proposed technique is very higher than the basic structure, needs more information for training it. Additionally, if the training information is required for more epochs, this model may over fit the dataset can influence the accuracy of the system.

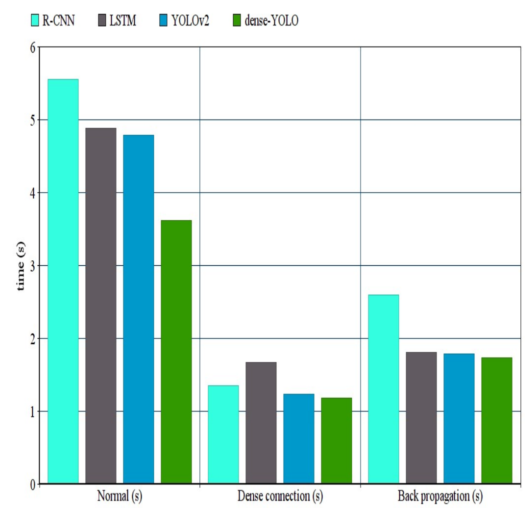
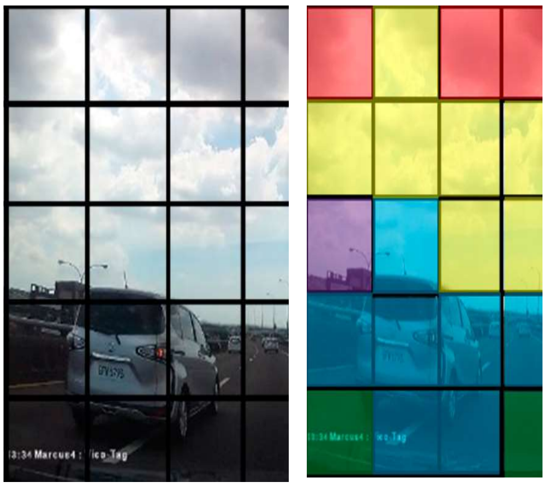


Fig. 10 Training Time For 3 Different Levels

The dense-YOLO framework for object detection is implemented as detecting the object according to the training of the proposed object detection technique. The location is detected and the object class for the input image has been carried through the Convolutional neural network. The input image is segregated into 5 x 5 grid cells while the bounding box prediction procedure has the specific confidence value. The centre value of the bounding box has the confidence value of IoU that the probability of the object detection. The repeated bounding boxes have been removed through the suppression technique which has the detection procedure of the image location. The vehicle detection process through the bounding box and suppression technique is demonstrated in Fig. 11 and the entire vehicle detection process through the dataset is illustrated in Fig. 12.



Input image with 5 x 5 grids

Mapping with specific classes

Fig. 11 Vehicle Detection Process

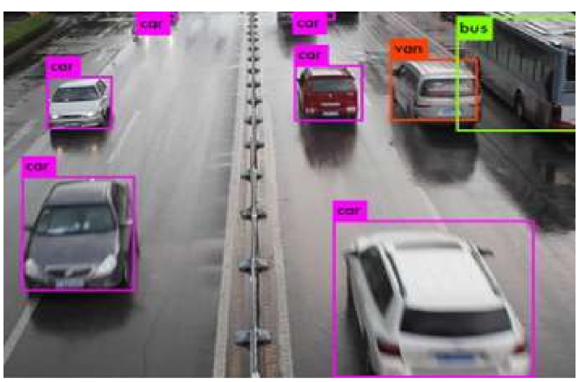


Fig. 12 Vehicle Detection Using Dense-YOLO On UA-DETRAC Dataset

5. CONCLUSION

The ability to recognise vehicles is a crucial component of many object detection-based intelligent transportation system technologies. The research goal is to identify the type of vehicle in an image or video. In this research, a new model for vehicle identification called dense-YOLO, which addresses some of the shortcomings of previous methods is proposed in this paper. The benefits of additional decision-making in a wide range of complex traffic situations can be fully exploited with the help of the updated YOLO algorithm, which features quick recognition speed, good precision, and a good detection effect. The dense-YOLO vehicle detection method is used to increase detection precision without incurring performance penalties in a real-time environment. In order to solve the gradient issue and enhance backbone network related identification, the suggested method makes use of a thick connection structure of the layers. We provide a loss function and a more efficient pooling architecture to merge data from various scales in order to rapidly train models. Detection accuracy of the suggested method is found to be higher than that of comparable methods based on performance tests on the UA-DETRAC dataset. In future, image processing techniques need to be strictly applied for better feature extraction and feature dimensionality reduction can be applied to reduce the time complexity levels.

References

[1]. C. Lu, H. Zhang, and M. Chen, "Realization of high-quality development of transportation in the new era," *China Journal of Highway and Transport*, vol. 34, no. 6, p. 1, 2021.View at: Google Scholar

[2]. R. Li and C. Wang, "Department of advanced traffic management systems," *Journal of Tsinghua University(Science and Technology)*, pp. 1–7, 2021.View at: Google Scholar

[3]. G. Guo, Y. Xu, T. Xu, D. Li, Y. Wang, and W. Yuan, "A survey of connected shared vehicle-road cooperative intelligent transportation systems," *Control and Decision*, vol. 34, no. 11, pp. 2375–2389, 2019.View at: Google Scholar

[4]. X. Kong, J. Zhang, L. Deng, and K. Liu, "Research advances on vehicle parameter identification based on machine vision," *China Journal of Highway and Transport*, vol. 34, pp. 13–30, 2021.View at: Google Scholar

[5]. A.Tian, X. Cai, W. Chen, W. Luo, and Y. Yin, "Vehicle illegal parking detection and evidence collection system on uav," *Measurement & Control Technology*, vol. 40, pp. 67–74, 2021.View at: Google Scholar

[6]. A.S. Ali, "Traffic management for drones flying in the city," *International Journal of*

Critical Infrastructure Protection, vol. 26, article 100310, 2019.

[7]. S. Srivastava, S. Narayan, and S. Mittal, "A survey of deep learning techniques for vehicle detection from uav images," *Journal of Systems Architecture, page*, vol. 117, p. 102152, 2021.View at: Publisher Site | Google Scholar

[8]. A.Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from uav imagery with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021.View at: Publisher Site | Google Scholar

[9]. M. Cao, H. Ji, Z. Gao, and T. Mei, "Vehicle detection in remote sensing images using deep neural networks and multi-task learning," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 797–804, 2020.

[10]. Guo, E.; Bai, L.; Zhang, Y.; Han, J. Vehicle Detection Based on Superpixel and Improved HOG in Aerial Images. In Proceedings of the International Conference on Image and Graphics, Shanghai, China, 13–15 September 2017; pp. 362–373.

[11]. Laopracha, N.; Sunat, K. Comparative Study of Computational Time that HOG-Based Features Used for Vehicle Detection. In Proceedings of the International Conference on Computing and Information Technology, Helsinki, Finland, 21–23 August 2017; pp. 275–284.

[12]. Pan, C.; Sun, M.; Yan, Z. The Study on Vehicle Detection Based on DPM in Traffic Scenes. In Proceedings of the International Conference on Frontier Computing, Tokyo, Japan, 13–15 July 2016; pp. 19–27

[13]. D. Zhong, P. Sun, A. Boukerche, Empirical study and analysis of the impact of traffic flow control at road intersections on vehicle energy consumption, in: Proceedings of the 18th ACM Symposium on Mobility Management and Wireless Access, MobiWac '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 21–28.

[14]. Boukerche, Y. Tao, P. Sun, Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems, Comput. Netw. 182 (2020) 107484.

[15]. Thompson, S.; Horiuchi, T.; Kagami, S. A probabilistic model of human motion and navigation intent for mobile robot path planning. In Proceedings of the IEEE Xplore, 2009 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 10–12 February 2009; pp. 663–668.

[16]. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 961–971.

[17]. S. Agarwal, J.O.D. Terrail, and F. Jurie, "Recent advances in object detection in the age of deep convolutional neural networks,"arXiv: 1807.04606 , Sep. 2018.

[18]. J. Zhang , K. Huang , Y. Yu , T. Tan , Boosted local structured HOG-LBP for object localization, in: 2011 IEEE Computer Vision and Pattern Recognition (CVPR), CO, USA, Colorado Springs, 2011, pp. 1393–1400.

[19]. P.F. Felzenszwalb , R.B. Girshick , D. McAllester , D. Ramanan , Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (Sep. 2010) 1627–1645.

[20]. A.Krizhevsky , I. Sutskever , G.E. Hinton , ImageNet classification with deep convolutional neural networks, in: International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[21]. R. Girshick , J. Donahue , T. Darrell , J. Malik , Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580–587.

[22]. J. Dai , Y. Li , K. He , J. Sun , R-FCN: object detection via region-based fully convolutional networks, Neural Inf. Process. Syst. (2016) 379–387.

[23]. Singh , H. Li , A. Sharma , L.S. Davis , R-FCN-30 0 0 at 30fps: decoupling detection and classification, in: 2018 IEEE Conference on Computer Vision and

Pattern Recognition (CVPR), Salt Lake City, Utah, USA, 2018, pp. 1081–1090.

[24]. R. Nabati and H. Qi, "RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 3093-3097, doi: 10.1109/ICIP.2019.8803392.

[25]. SStaudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv 2019, arXiv:1909.09586.

[26]. Trieu, T.H. Trieu2018darkflow: Darkflow, on GitHub Repository. 2018. Available online: https://github.com/thtrieu/darkflow

[27]. C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A.C. Berg, "DSSD : deconvolutional single shot detector,"arXiv: 170106659 Cs,Jan.2017.

[28]. P. Zhou , B. Ni , C. Geng , J. Hu , Y. Xu , Scale-Transferrable object detection, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah, USA, 2018, pp. 528–537.

[29]. G. Huang , Z. Liu , L. van der Maaten , K.Q. Weinberger , Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2261–2269.

[30]. F. Chollet , Xception: deep learning with depthwise separable convolutions, in: Computer Vision and Pattern Recognition, 2017, pp. 1800–1807.

[31]. X. Zhang , X. Zhou , M. Lin , J. Sun , ShuffleNet: an extremely efficient convolutional neural network for mobile devices, in: Computer Vision and Pattern Recognition, 2018, pp. 6 84 8–6 856.

[32]. Zeng Dan , et al. , Proposal pyramid networks for fast face detection, Inf. Sci. 4 95 (2019) 136–149.