Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.024812
Article

Deep Neural Network Based Vehicle Detection and Classification of Aerial Images

Sandeep Kumar[1], Arpit Jain[2,*], Shilpa Rani[3], Hammam Alshazly[4], Sahar Ahmed Idris[5] and Sami Bourouis[6]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
[2]Faculty of Engineering & Computer Sciences, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India
[3]Department of Computer Science and Engineering, Neil Gogte Institute of Technology, Hyderabad, India
[4]Department of Computer Science, Faculty of Computers and Information, South Valley University, Qena, 83523, Egypt
[5]College of Industrial Engineering, King Khalid University, Abha, Saudi Arabia
[6]Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
*Corresponding Author: Arpit Jain. Email: dr.jainarpit@gmail.com
Received: 01 November 2021; Accepted: 22 December 2021

Abstract: The detection of the objects in the ariel image has a significant impact on the field of parking space management, traffic management activities and surveillance systems. Traditional vehicle detection algorithms have some limitations as these algorithms are not working with the complex background and with the small size of object in bigger scenes. It is observed that researchers are facing numerous problems in vehicle detection and classification, i.e., complicated background, the vehicle's modest size, other objects with similar visual appearances are not correctly addressed. A robust algorithm for vehicle detection and classification has been proposed to overcome the limitation of existing techniques in this research work. We propose an algorithm based on Convolutional Neural Network (CNN) to detect the vehicle and classify it into light and heavy vehicles. The performance of this approach was evaluated using a variety of benchmark datasets, including VEDAI, VIVID, UC Merced Land Use, and the Self database. To validate the results, various performance parameters such as accuracy, precision, recall, error, and F1-Score were calculated. The results suggest that the proposed technique has a higher detection rate, which is approximately 92.06% on the VEDAI dataset, 95.73% on the VIVID dataset, 90.17% on the UC Merced Land dataset, and 96.16% on the Self dataset.

Keywords: Vehicle detection; vehicle recognition; neural network; LSTM; YOLOv3

1 Introduction

Nowadays, computer vision is a trending technology. It is highly in demand in the security and surveillance industry, self-driven cars, entertainment applications, etc. This surge in popularity of computer vision is mainly due to the emergence of state-of-the-art deep learning technologies that can solve computer vision tasks with very high accuracy, something which was considered unachievable a decade back. As a consequence, deep learning models have become the preferable approaches to improve

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the performance of various computer vision tasks such as object detection [1–5], biometric detection and recognition [6–8], and more recently in detecting abnormalities in medical images [9–11]. Computer vision is used to provide the human intelligence and understanding to the computers. Digital image are used as an input to the computer and later machines can identify the objects in image.

Vehicle detection has a significant role in computer vision based on deep learning and machine learning. Vehicle detection and classification are important research area in the field of vision and computing applications. Due to the alarming improvement in areal imaging, vehicle tracking on highways, vehicle detection concerns many researchers but due to the different shapes and sizes of vehicles and different resolutions of satellite images, detection is always challenging.

The global population is growing at a rapid pace, and this trend is expected to continue and as a consequence the transportation problem is increasing. Therefore managing the transportation system efficiently is a difficult task for every country. Intelligent tracking systems can quickly and correctly distinguish each individual car using hardware such as a camera. Many traffic lights, traffic signs, and traffic police were deployed in all the traffic-prone areas. However, these methods are not sufficient alone. It is difficult to manage traffic, accidents, and other related issues with old methods. New trends and technology are deeply required to manage the transportation system. Many researchers worked in this field continuously for many years and invented object detection and object tracking system to utilize automated camera surveillance to produce data that can give meaning to a decision-making process.

The vehicle detection system helps manage traffic flow on the roads, prevent accidents, and monitor traffic crimes and violations. Vehicle detection systems are considered very significant for monitoring traffic and controlling highway security. Nowadays, traffic surveillance cameras are installed on highways with colossal traffic video footage and this video footage could be used for analysis purposes. Generally, the viewing angle will be different or the camera position would be distant from the road, or the object's size could be considered small. In all these situations, it is not an easy task to detect the vehicles effectively and classify them further. Considering this, we propose a model with a reliable deep neural network architecture for detecting light and heavy vehicles in a given input image. We conducted extensive experiments on several benchmark datasets and considering various performance evaluation metrics. We also compare our proposed algorithm with state-of-the-art methods using similar preprocessing procedures. The results show that our proposed method has a high level of accuracy and better performance than existing methods.

2 Literature Work

Automatic detection of vehicle is widely used in many traffic management systems and vehicle information systems. This area attracted the attention of many researches in the last decade. Researchers have applied various approaches for the detection of vehicles. But still it is difficult to get the required accuracy and the gap is already exists. Therefore, many researchers are focusing on this problem. In Tab. 1 we summarize the work which has been done for detecting of vehicles from aerial images and highlight their findings.

The recent studies focus on the deep learning based vehicle detection methods due to their outstanding performance. However, these methods have many limitations especially when the objects are very small. Moreover, training deep neural networks requires a high computation cost which makes this task more difficult and time consuming. In this study, our main aim is to introduce a novel approach to detect the vehicle and classify it into light and heavy vehicles. In the proposed method Convolutional Neural Network (CNN) is combined with Long Short Term Memory (LSTM) as a CNN is unable to remember the previous output and considers only the current input. However, LSTM has a unique structure and it is more reliable when extracting the features in-depth. Therefore, we propose a hybrid deep learning-based approach and combine YOLO-V3 with LSTM.

Table 1: A summary of existing object detection methods

Authors & year	Methodology	Database	Remarks
Zhu et al. [1], 2019	Improved YOLOv3	VIVID & NPU	Precision = 95.0% & 97.4% Recall = 97.4% & 97.0% F1-Score = 96.2% & 97.2%
Boyuk et al. [2], 2020	Faster R-CNN & YOLOv3	Own	mAP = 0.528
Xianghui et al. [3], 2019	ISPDM + YOLOv3	UAV	Precision = 84.15%, Recall = 79.8% F1-Score = 81.91%
Hasan et al. [4], 2018	YOLO	Own	Accuracy 84.49%
Wang et al. [12], 2018	Multiscale Fusion	-	-
Li et al. [13], 2020	CNN + Outlier-Aware Non-Maximum Suppression	Self + UAVDT	Recall = 92.05%, Precision = 97.52%, F1-Score = 94.94%
Xu et al. [14], 2018	YOLOv2	UAV	mAP = 76.2%, FPS = 26
Javadi et al. [15], 2020	DarkNet-53, DenseNet-201	Self	Recall = 92.46% Precision = 96.43% F1-Score = 95.72% Mean IoU = 81
Lin et al. [16], 2020	Faster RCNN + ReLU	VAID DLR-MVDA KIT-AIS COWC	mAP = 89.3%, Precision = 94% Recall = 94%, F1-Score = 96%
Xu et al. [17], 2019	YOLOv3	VEDAI	Precision = 89.6%, Recall = 91.5%, F1-Score = 90.59%

3 Proposed Work

The main focus of this work is to develop a novel methodology which can detect heavy and light vehicles from the input image. The flow chart is shown in Fig. 1. The proposed model will be designed and simulated using Python tools. The detailed description of the proposed method is given below:

3.1 Pre-Processing

First, background has to be eliminated from the given input during the pre-processing step. Background Subtractor Mixture of Gaussians (MOG2) is used to remove the image's background. This technique is unique in that it chooses an appropriate Gaussian distribution for each pixel, with the pixel values providing the image's background information. This method aids in the adaptation of luminance so that color remains for longer periods of time in order to obtain more information, and this class also enables parallel computation.

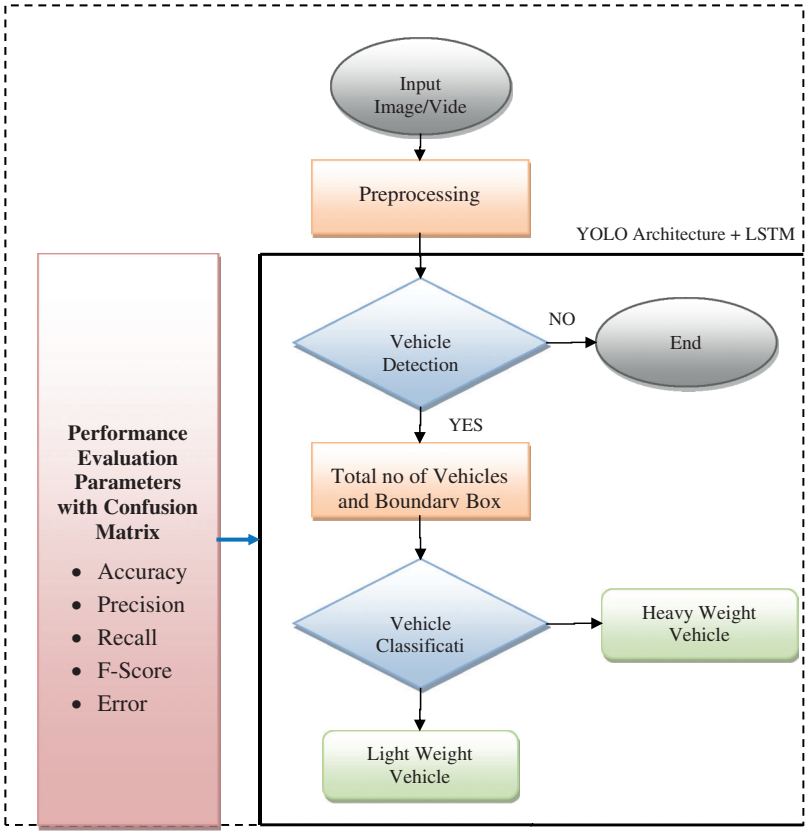

Figure 1: Flow chart of the proposed work

The sample results of background subtraction on sample image are shown in the Fig. 2. After successful background subtraction the next step is foreground subtraction. In the foreground extraction all the pixel values becomes zero except for the target object. This will help in minimizing the number of parameters which are further used in the deep neural network. Therefore feature are only extracted from the foreground image rather than extracting the features from the whole image. This will reduce the complexity of the model and improves the accuracy as well.

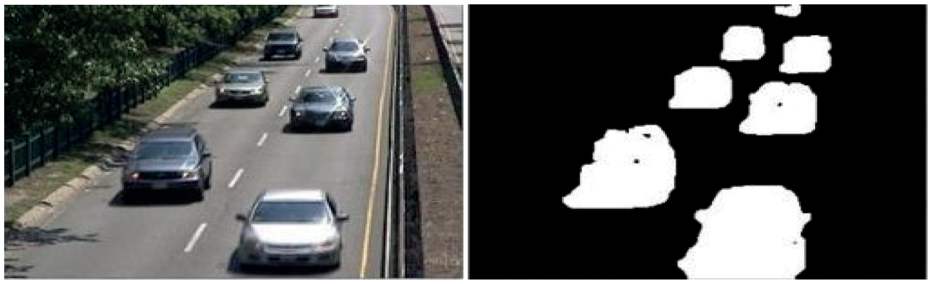

Figure 2: Pre-processing of proposed work

3.2 YOLOv3

YOLO-V3 (You Only Look Once) [5] is an object detection algorithm which is used to identify the objects from the image, video or live feeds. YOLO uses the features of deep neural networks to detect the objects.

In YOLO, the object prediction is performed using a convolutional layer which uses 1 × 1 convolutions. In the proposed method, the features which are extracted during the pre-processing are now fed into the YOLO-V3 network. This input image passes through different convolution layers, batch normalization, activation and other layers described in the following subsections.

3.2.1 Convolution Layer

To extract the key information from the given image, a fully convolutional layer is used as in the DarkNet. DarkNet originally has 53 layers network. For object detection 53 more layers are stacked with a total of 106 fully convolutional layers.

3.2.2 Residual Block

The primary job of a residual block is to extract the feature from the given image. Architectural diagram of a residual block is shown in the Fig. 3. Generally residual connection has two main branches. One is a series of convolution, batch normalization, and Rectified Linear Unit (ReLU) activation. The second branch is an identity mapping that connects the input to the block with the output of the first branch. When deep neural networks are implemented then residual or skip connections help us to avoid overfitting. In the architecture diagram of YOLO-V3, 1x, 2x, 3x is mentioned which signifies that a particular block has been repeated those many number of times in the architecture. The repetitions of the blocks make the total convolution layers 53. Every block is connected to the residual block which is connected to the output of the previous block. DarkNet has a total of five downsamplings stages, with each downsampling halving the size of the feature map. The feature map will be extracted in the final three down samplings, and it will then predict the various classifications. There is no max-pooling here therefore down sampling of filter maps are required.

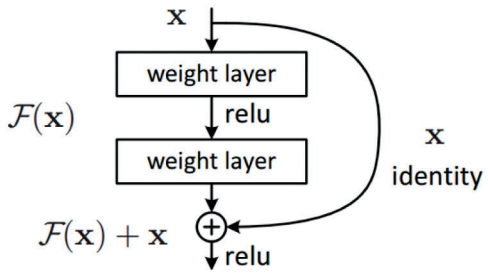

Figure 3: Architecture of residual block [18]

According to the DarkNet architecture the term 3 × 3/2 is used which means a 3 × 3 convolution with a stride of 2. If the input image is of size 256 × 256, then the stride of 2 makes the input size half and the new image size will be 128 × 128.

3.2.3 Batch Normalization

We used batch normalization to increase the speed of training and combat overfitting. Each layer in a CNN has corresponding inputs associated with it. During the training process, this input gets modified randomly. We use batch normalization to reduce this randomness to further propagate from the current layer to the next layer. This is achieved through a normalization step that regulates each layer's inputs' mean and variance values.

3.2.4 Leaky ReLU

Instead of taking a ReLU activations, we used the leaky ReLU [19] to challenge the proposed method's dying ReLU problem. As the segmented portion is too small, the neurons of few layers are pushed on to an

inactive state. As the learning rate is too low in the proposed methodology, most neurons are getting stuck in the dead state. This is further decreasing the model performance. Hence, we used leaky ReLU for the model. It will allow a slight gradient when the unit is inactive. The leaky ReLU can be denoted as:

$$f(x) = \begin{cases} I & if\ I > 0 \\ 0.02 \times I & otherwise \end{cases}$$

3.4 Object Detection and Bounding Box

You Only Look Once [20] is a fully convolutional network and each feature vector would be fed into the Fully Connected (FC) layer sequence as shown in Fig. 4. The most salient feature of YOLO-V3 is that it makes detections at three different scales i.e., 13×13, 26×26 and 52×52 grids.

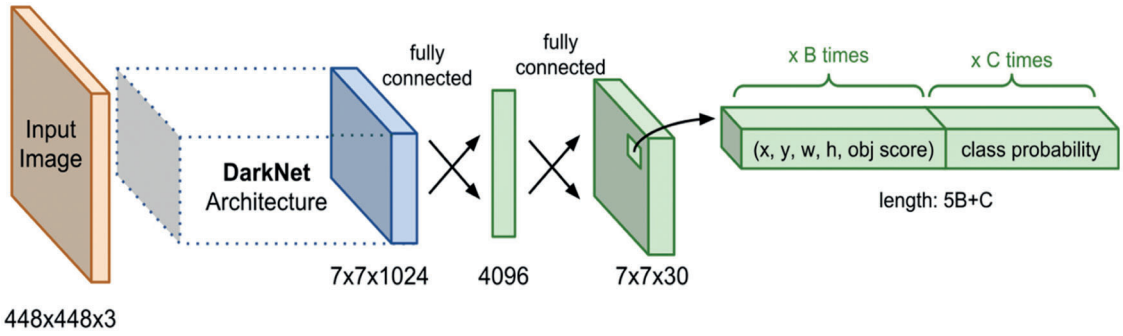

Figure 4: Schematic diagram of YOLO3

Total nine anchor boxes will appear in the given image i.e., 3 belongs to large objects, next 3 belongs to medium objects and last 3 belongs to small objects. Soft max layer produces the probability of k object classes and another output layer produces the four real-valued numbers for every k object class. Each set of these four real-valued numbers is used to find the bounding box position for each k class object. This layer helps to extract the box-specific information from the image and it would be feed into the final classification model of the network. The probability P_0 is the probability of the object means the objectness score, [tx, ty tw th] represent the coordinates of the boxes, and P_1, P_2, P_3…….P_N represent the class probability.

In the YOLO-V3 a total of 10,647 boxes for a single image $\{(13 \times 13 \times 3) + (26 \times 26 \times 3) + (52 \times 52 \times 3) = 10647\}$. The majorities of the boxes in the image are false positives. YOLO uses the Non-Maximum Suppression (NMS) technique to eliminate overlapping boxes and lower level confidence score boxes.

3.5 Convolutional LSTM

CNN is unable to remember the previous output and it only considers the current input. However, an LSTM has a unique structure and it is more reliable in extracting the features in-depth. In the proposed methodology, a hybrid deep learning-based approach is implemented to combine YOLO-V3 with LSTM. For feature extraction, the ConvLSTM and Convolution layer are combined. Here, the ConvLSTM is added with 16 filters. Convolutional LSTM has the ability to remember the previous inputs and dynamics between the features extracted from YOLO-V3 can be learned. The resultant data that flows from the ConvLSTM keeps the same input dimension, making it different from traditional LSTM. When training, the input images are resized into a size of 416×416 by default. While varying the learning rate of the model, the loss suddenly starts going up at some point and when the loss value reaches up to 0.xxx, than

no more changes occur in the output. So, we recommend to stop training and the loss result is as shown in Fig. 5. While training of the model, once the loss is stable then we stop training and we go with the testing step.

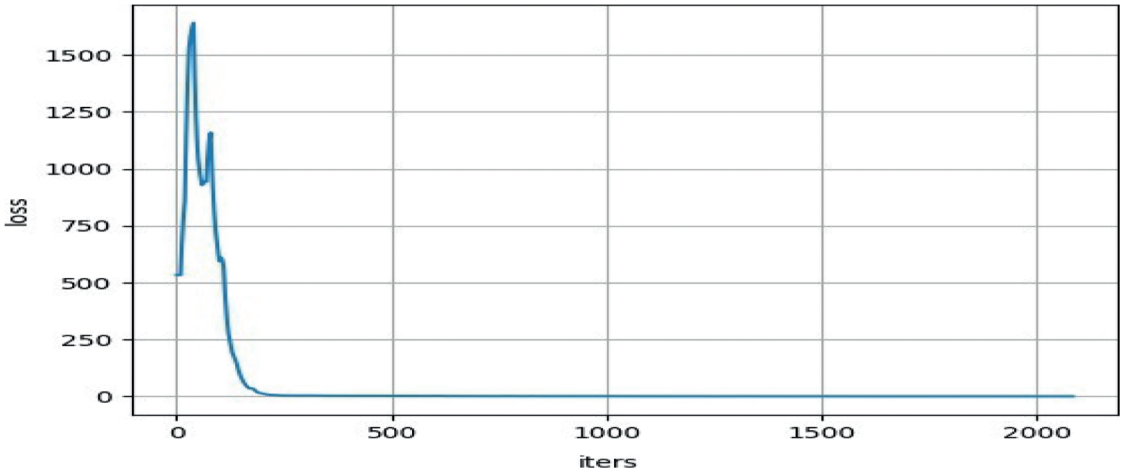

Figure 5: Loss curve of network while training

When performing classification of vehicle using YOLO architecture, parameters such as number of epochs, learning rate, dropout, and batch size are considered. The entire description of these parameters are given in Tab. 2. The overall performance of the proposed module is discussed in the results and analysis section.

Table 2: Details of the YOLO architecture

Parameter name	Vehicle classification
Epocs	35
Learning rate	0.001
Droupout	0.35
Batch size	128
Stride	2

4 Results and Analysis

To identify the efficiency of the proposed methodology, we performed the experiments on various datasets. We conducted experiments on VIVID dataset [21], VEDAI dataset [22], UC Merced Land Use dataset [23] and Self dataset. All the experiments were performed on a computer with a GPU and 8GB of unified memory. The implementation of the algorithms is validated through 5-fold cross validation technique.

4.1 Evaluation Parameters

To evaluate the performance of the proposed work, standard evaluation metrics such as accuracy, precision, recall, error and F1-score are used. These metrics are defined mathematically in the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{2}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{3}$$

$$\text{Error (E)} = 1 - \text{Accuracy} \tag{4}$$

$$F1 - Score = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where, TP stands for true positive, TN stands for true negative, FP stands for false-positive, and FN stands for false-negative.

4.2 Comparison with Existing Methodology

To make the environment user friendly for the user, a Graphical User Interface (GUI) is prepared and the sample image of GUI is shown in the Fig. 6. Proposed work is evaluated on 4 different datasets and according to the experimental analysis, the proposed methodology is able to detect the vehicle from given images. It shows the accuracy and efficiency of the proposed work. The outputs of the vehicle detection on different datasets are illustrated in Figs. 7–9.

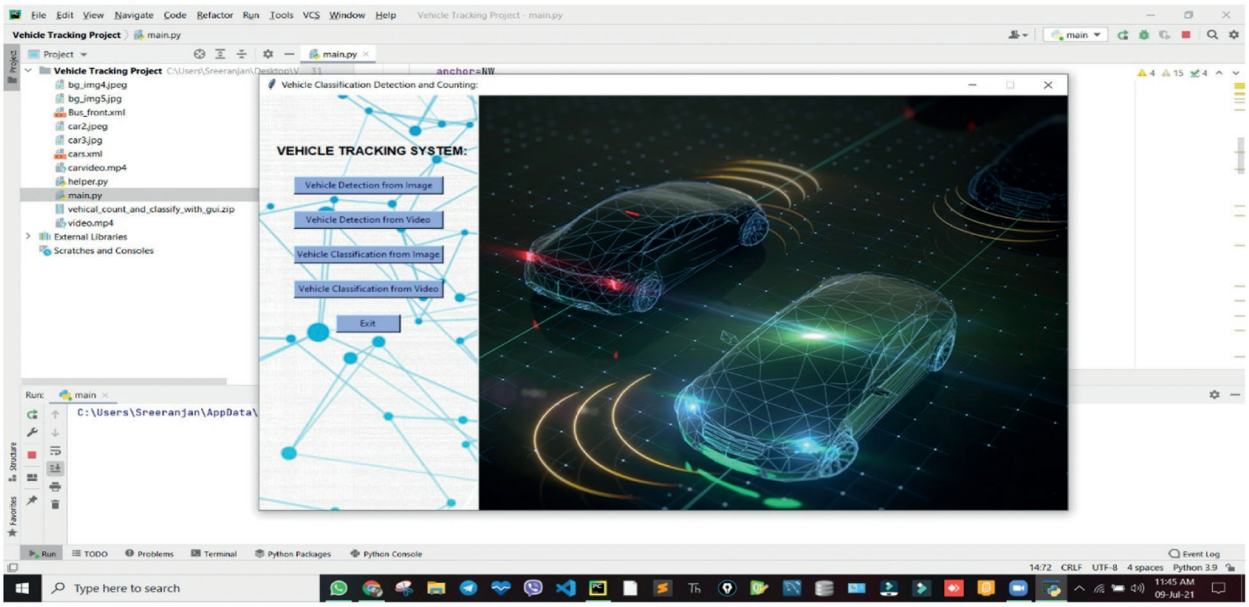

Figure 6: Overall GUI of the proposed work

The proposed model's performance is compared to that of existing state-of-the-art models. These comparisons are done on the basis of light vehicle and heavy vehicle classification. On VIVID datasets, Tab. 3 demonstrates a comparison of the proposed work with some well-known approaches.

(a) (b)

Figure 7: Output of proposed work on: (a) VIVID database, and (b) VEDAI database

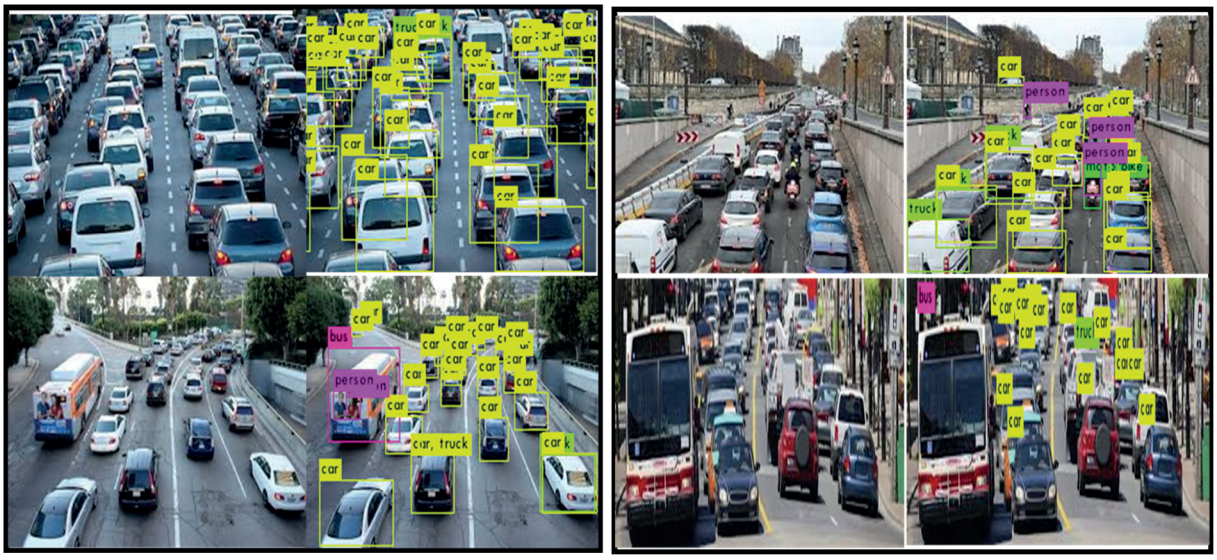

Figure 8: Output of proposed work on Self database

Tab. 4 shows the comparative analysis of the proposed work with the existing methods on VEDAI datasets. According to the experimental analysis the accuracy, Recall and F1-score of the proposed method is better than existing methods on VEDAI datasets. Error rate is also calculated and it is 7.94 and it is comparatively high from the VIVID dataset.

Figure 9: Output of proposed work on the UC Merced Land Use dataset

Table 3: Comparison of proposed work on VIVID database

Authors & year	Database	Results
Xunxun et al. [24], 2019	VIVID	Precision = 94.3%, Recall = 97% F1-Score = 96.1%
Yan et al. [25], 2017	VIVID	Accuracy = 72.75%
Proposed work	VIVID	Accuracy = 95.73%, Precision = 96.77%, Recall = 96.49%, F1-Score = 97.24%, Error = 4.31

Table 4: Comparison of proposed work on VEDAI database

Authors & year	Database	Results
Xu et al. [17], 2019	VEDAI	Precision = 89.6%, Recall = 91.5% F1-Score = 90.5%
Piao et al. [26], 2019	VEDAI	Accuracy = 88.1%, Precision = 79.6% F1-Score = 89.2%
Sommer et al. [27], 2018	VEDAI	Precision = 97.4%
Ajay et al. [28], 2017	VEDAI	Accuracy = 90.55%
Proposed work	VEDAI	Accuracy = 92.06%, Precision = 93.19%, Recall = 92.6%, F1-Score = 93.38%, Error = 7.94

Tab. 5 shows the comparative analysis of the proposed work with the existing methods on the Self dataset. According to the experimental analysis the Precision, Recall and F1-score of the proposed

method is better than existing methods on the Self dataset. The error rate is calculated and it is 3.94, which is comparatively lower than the error rate obtained for the VIVID and VEDAI datasets.

Table 5: Comparison of proposed work on Self database

Authors & year	Database	Results
Li et al. [29], 2019	Self	Precision = 84.15%, Recall Rate = 79.80% F1-Score = 81.91%
Ichim et al. [30], 2018	Self	Accuracy = 98.8%
Hamsa et al. [31], 2018	Self	Precision = 87.5%, Accuracy = 94.7%
Ponce et al. [32], 2019	Self	Accuracy = 75%
Ram et al. [33], 2016	Self	Precision = 77.8%, Recall = 77.3% F-Score = 77.5%
Proposed work	Self	Accuracy = 96.16%, Precision = 91.47% Recall = 93.70%, F1-Score = 94.38%, Error = 3.94

On the UC Merced Land Use dataset, the proposed model is examined. Tab. 6 shows the parameters that are used in the analysis. According to the experimental analysis, the proposed method accuracy is 90.17%, precision is 91.38%, recall is 91.73%, error is 9.83 and F1-score is 90.10%. As per the analysis, the error rate of the proposed work on UC Merced Land Use dataset is comparatively higher than the other datasets. Example of vehicles detection from the considered dataset is depicted in Fig. 9.

Table 6: Output of proposed work on UC merced land use dataset

Database	Results
UC merced land use dataset	Accuracy = 90.17%, Precision = 91.38%, Recall = 91.73%, F1-Score = 90.10%, Error = 9.83%

5 Conclusion

In the sphere of parking space management, traffic control activities, and surveillance systems, the detection of objects in an aerial image is critical. Traditional and deep learning based methods have limitation in extracting the important features from the image and the computation cost is also high. It is observed that still, the number of issues that come in vehicle detection and classification, i.e., the vehicle's small size, other items with similar visual appearances, distance, and other factors all contribute to the overall complexity of the scene are not correctly addressed. A robust algorithm for vehicle detection and classification has been proposed to overcome the limitation of existing techniques in this research work. Our proposed method improves the overall results of evaluation parameters on various publicly available standard databases, i.e., VEDAI, VIVID, UC Merced Land Use and Self datasets. The results are not much accurate for real time classification. Therefore this challenge will be considered for future work to improve the effectiveness of the proposed work.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through a Research Group Program under Grant RGP.2/53/42. They

would also like to thank the support from Taif University Researchers Supporting Project (TURSP-2020/26), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Zhang and X. Zhu, "Vehicle Detection in the aerial infrared images via an improved YOLOv3 network," in *Proc. of 4th IEEE Int. Conf. on Signal and Image Processing (ICSIP)*, Wuxi, China, pp. 372–376, 2019.

[2] M. Böyük, R. Duvar and O. Urhan, "Deep learning based vehicle detection with images taken from unmanned Air vehicle," in *Proc. of IEEE Int. Conf. on Innovations in Intelligent Systems and Applications Conf. (ASYU)*, Istanbul, Turkey, pp. 1–4, 2020.

[3] X. Li and X. Li, "Robust vehicle detection in aerial images based on image spatial pyramid detection model," in *Proc. of 4th IEEE Int. Conf. on Advanced Robotics and Mechatronics (ICARM)*, Toyonaka, Japan, pp. 850–855, 2019.

[4] H. Sarıbaş, H. Çevıkalp and S. Kahvecıoğlu, "Car detection in images taken from unmanned aerial vehicles," in *Proc. of 26th IEEE Int. Conf. on Signal Processing and Communications Applications Conf. (SIU)*, Izmir, Turkey, pp. 1–4, 2018.

[5] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Berlin/Heidelberg, Germany, pp. 1804–2767, 2018.

[6] H. Alshazly, C. Linse, E. Barth and T. Martinetz, "Deep convolutional neural networks for unconstrained ear recognition," *IEEE Access*, vol. 8, pp. 170295–170310, 2020.

[7] H. Alshazly, C. Linse, E. Barth and T. Martinetz, "Ensembles of deep learning models and transfer learning for ear recognition," *Sensors*, vol. 19, no. 19, pp. 4139, 2019.

[8] H. Alshazly, C. Linse, E. Barth, S. A. Idris and T. Martinetz, "Towards explainable ear recognition systems using deep residual networks," *IEEE Access*, vol. 9, pp. 122254–122273, 2021.

[9] H. Kaushik, D. Singh, M. Kaur, H. Alshazly, A. Zaguia *et al.,* "Diabetic retinopathy diagnosis from fundus images using stacked generalization of deep models," *IEEE Access*, vol. 9, pp. 108276–108292, 2021.

[10] H. Alshazly, C. Linse, M. Abdalla, E. Barth and T. Martinetz, "COVID-Nets: Deep CNN architectures for detecting COVID-19 using chest CT scans," *PeerJ Computer Science*, vol. 7, pp. e655, 2021. https://doi.org/10.7717/peerj-cs.655.

[11] H. Alshazly, C. Linse, E. Barth and T. Martinetz, "Explainable covid-19 detection using chest CT scans and deep learning," *Sensors*, vol. 21, no. 2, pp. 455, 2021.

[12] W. Weihua, W. Peizao and N. Zhaodong, "A Real-time detection algorithm for unmanned aerial vehicle target in infrared search system," in *Proc. of IEEE Int. Conf. on Signal Processing, Communications and Computing (ICSPCC)*, Qingdao, China, pp. 1–5, 2018.

[13] X. Li, X. Li and H. Pan, "Multi-scale vehicle detection in high-resolution aerial images with context information," *IEEE Access*, vol. 8, pp. 208643–208657, 2020.

[14] Z. Xu, H. Shi, N. Li, C. Xiang and H. Zhou, "Vehicle detection under UAV based on optimal dense yolo method," in *Proc. of 5th IEEE Int. Conf. on Systems and Informatics (ICSAI)*, Nanjing, China, pp. 407–411, 2018.

[15] S. Javadi, M. Dahl and M. I. Pettersson, "Vehicle detection in aerial images based on 3D depth maps and deep neural networks," *IEEE Access*, vol. 9, pp. 8381–8391, 2021.

[16] H. -Y. Lin, K. Tu and C. Li, "VAID: An aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212209–212219, 2020.

[17] B. Xu, B. Wang and Y. Gu, "Vehicle detection in aerial images using modified YOLO," in *Proc. of 19th IEEE Int. Conf. on Communication Technology (ICCT)*, Xi'an, China, pp. 1669–1672, 2019.

[18] K. He and X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada., USA, pp. 770–778, 2016.

[19] B. Xu, N. Wang, T. Chen and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015. https://arxiv.org/abs/1505.00853.

[20] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada., USA, pp. 779–788, 2016.

[21] R. Collins, X. Zhou and S. K. Teh, "An open source tracking testbed and evaluation web site," in *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, Colorado, USA, vol. 2, no. 6, pp. 35, 2005.

[22] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2014.

[23] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. of the 18th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (ACM GIS)*, California, USA, pp. 270–279, 2010.

[24] X. Zhang and X. Zhu, "Vehicle detection in the aerial infrared images via an improved yolov3 network," in *Proc. of 4th IEEE Int. Conf. on Signal and Image Processing*, Wuxi, China. pp. 372–376, 2019.

[25] L. Yan, J. Gong, D. Chen and S. Zhang, "A fast vehicle detection method by UAV using region feature gradient," in *Proc. of IEEE Int. Conf. on Unmanned Systems (ICUS)*, Beijing, China, pp. 384–388, 2017.

[26] Z. Piao, B. Zhao, L. Tang, W. Tang, S. Zhou *et al.,* "VDetor: An effective and efficient neural network for vehicle detection in aerial image," in *Proc. of IEEE Int. Conf. on Signal, Information and Data Processing (ICSIDP)*, Chongqing, China, pp. 1–4, 2019.

[27] L. Sommer, N. Schmidt, A. Schumann and J. Beyerer, "Search area reduction fast-RCNN for fast vehicle detection in large aerial imagery," in *Proc. of 25th IEEE Int. Conf. on Image Processing (ICIP)*, Athens, Greece, pp. 3054–3058, 2018.

[28] A. Ajay, V. Sowmya and K. P. Soman, "Vehicle detection in aerial imagery using eigen features," in *Proc. of IEEE Int. Conf. on Communication and Signal Processing (ICCSP)*, Chennai, India, pp. 1620–1624, 2017.

[29] X. Li and X. Li, "Robust vehicle detection in aerial images based on image spatial pyramid detection model," in *Proc. of 4th IEEE Int. Conf. on Advanced Robotics and Mechatronics (ICARM)*, Toyonaka, Japan, pp. 850–855, 2019.

[30] L. Ichim and D. Popescu, "Road detection and segmentation from aerial images using a CNN based system," in *Proc. of 41st Int. Conf. on Telecommunications and Signal Processing (TSP)*, Athens, Greece, pp. 1–5, 2018.

[31] S. Hamsa, A. Panthakkan, S. Al-Mansoori and H. Alahamed, "Automatic vehicle detection from aerial images using cascaded support vector machine and Gaussian mixture model," in *Proc. of Int. Conf. on Signal Processing and Information Security (ICSPIS)*, Dubai, United Arab Emirates, pp. 1–4, 2018.

[32] G. R. V. Ponce, K. Bhimani, J. A. Prakosa and M. A. B. Alvarez, "Pattern recognition through digital image processing for unmanned aerial vehicles," in *Proc. of 26th Int. Conf. on Electronics, Electrical Engineering and Computing (INTERCON)*, Lima, Peru, pp. 1–4, 2019.

[33] S. Ram and J. J. Rodriguez, "Vehicle detection in aerial images using multiscale structure enhancement and symmetry," in *Proc. of IEEE Int. Conf. on Image Processing (ICIP)*, Phoenix, Arizona, USA, pp. 3817–3821, 2016.