Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 1, 2024, 127-136.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Iranian Vehicle Images Dataset for Object Detection Algorithm

Pouria Maleki[1] , Abbas Ramazani[1*], Hassan khotanlou[2] and Sina Ojaghi[3]

1.Department of Electrical Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran.
2.Department of Computer Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran.
3.School of Computer and Electrical Engineering, University of Tehran, Tehran, Iran.

Article Info

Article History:
Received 21 January 2024
Revised 07 February 2024
Accepted 12 March 2024

DOI:10.22044/jadm.2024.13858.2501

Keywords:
Object Detection, YOLOv8s, Vehicle Dataset, Deep Neural Network.

**Corresponding author: a.ramazani@basu.ac.ir (A. Ramazani).*

Abstract

Providing a dataset[1] with a suitable volume and high accuracy for training deep neural networks is considered to be one of the basic requirements in that a suitable dataset in terms of the number and quality of images and labeling accuracy can have a great impact on the output accuracy of the trained network. The dataset presented in this article contains 3000 images downloaded from online Iranian car sales companies, including Divar and Bama sites, which are manually labeled in three classes: car, truck, and bus. The labels are in the form of 5765 bounding boxes, which characterize the vehicles in the image with high accuracy, ultimately resulting in a unique dataset that is made available for public use. The YOLOv8s algorithm, trained on this dataset, achieves an impressive final precision of 91.7% for validation images. The Mean Average Precision (mAP) at a 50% threshold is recorded at 92.6%. This precision is considered suitable for city vehicle detection networks. Notably, when comparing the YOLOv8s algorithm trained with this dataset to YOLOv8s trained with the COCO dataset, there is a remarkable 10% increase in mAP at 50% and an approximately 22% improvement in the mAP range of 50% to 95%.

1. Introduction

Due to the increase in population in recent years, intelligent transportation systems (ITS) need more accurate and newer tools for traffic control. With the progress of computer vision techniques, reducing hardware costs, providing faster (real-time) response, simpler installation and maintenance, and the existence of more comprehensive information in the traffic monitoring method by using video image processing, this method has been more popular among researchers [1].

Having a suitable and accurate dataset is crucial in training a deep neural network to identify desired objects in images. In specific applications like vehicle identification, a local dataset proves invaluable for researchers in developing algorithms that teach effectively and yield acceptable results [2].

In this article, numerous vehicle images from Iran were collected from online vehicle marketplaces. Human experts at the artificial intelligence school of Bu Ali Sina University classified them into three categories: car, bus, and truck, achieving high labeling accuracy.

Prior to this study, researchers utilized public datasets mentioned in the dataset introduction section to train vehicle identification networks. However, this approach led to ineffective algorithms in the Iranian context. The diversity and high labeling accuracy of the dataset presented in this article, combined with training the YOLOv8s algorithm, resulted in promising outcomes. In training the YOLOv8s algorithm with the dataset

[1] Dataset is available in address: https://github.com/pouria-maleki/Iranian-Vehicle-images-dataset-for-detection

introduced in this article and comparing it to the version trained with the COCO dataset [3], a significant 7% increase in average precision across three classes was observed. Furthermore, the Mean Average Precision (mAP) at a 50% threshold improved by 10%, and the mAP across the 50%-90% threshold range showed a substantial 20% increase.

Our study focuses on developing a specialized dataset for Iranian vehicles and optimizing it for object detection with YOLOv8s. While our theoretical contributions may seem modest, the true value lies in addressing practical challenges and filling a critical gap in the field. The dataset's uniqueness lies in its meticulous curation from diverse sources, accurately reflecting Iranian vehicular scenarios. Demonstrating the YOLOv8s algorithm's efficacy, the dataset significantly improves object detection accuracy in Iranian urban traffic. Exceptional precision and mAP metrics highlight its practical value for training deep neural networks in intelligent transportation systems. Despite not introducing groundbreaking algorithms, our manuscript makes a substantial contribution through its specialized dataset, emphasizing the importance of domain-specific data in advancing intelligent transportation research.

2. Object Detection

Regarding the problem of classifying vehicles in one image, there can be several different classes in one image (for example, human, car, etc.), so the position of the objects must be found first and then classified. In such cases, object detection algorithm is used. Figure 1 shows a general diagram of the object detection algorithm.

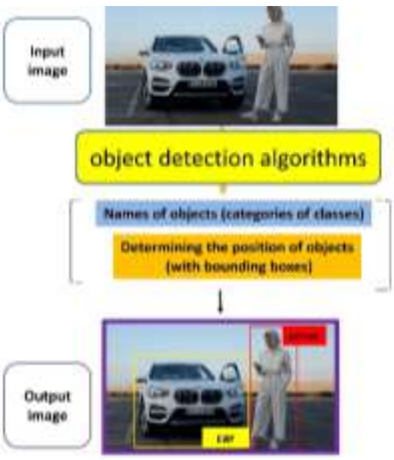

Figure 1. Object detection algorithm.

Normally, modern object detection algorithms can be divided into one-stage and two-stage methods. Although two-stage methods are more accurate in identifying objects, one-stage methods are much faster in this field. Two-stage methods perform the object detection process in two stages: Region Proposal Network (RPN) and Detection head (DH). Algorithms of R-CNN series are of two-stage type and have high popularity and power [4]. In these methods, first a large set of proposals or boxes are produced for each image, then each proposal is changed to a specific size and provided to CNN networks for feature extraction [5]. Finally, a classifier is used to specify these generated boxes, which is a relatively complex path that has low speed and is difficult to optimize, because each of these components must be trained separately [6]. In R-CNN, there were 2000 region proposals for each image, and the image classification algorithm calculated a separate feature map for each initial region proposal, which was a costly process. In this regard, Ross Girshick proposed a method called Fast R-CNN, which significantly increased the speed of object detection. The main idea in this method was that instead of 2000 feature maps as the initial output of the network, a single feature mAP is calculated for the entire image, and finally, for each area proposal, a region of interest (RoI), extracts a pooling layer, a feature vector of constant length from the feature map [7].

Each feature vector is then used for object classification and recognition. Another introduced algorithm for object recognition was a method that worked with the help of the Faster R-CNN algorithm. In this algorithm, the main idea was that both parts (region proposal calculation and image classification) can use the same feature map and thus share the computational burden. In this method, a convolutional neural network is used to prepare a feature mAP of the image, which simultaneously trains the region proposal network and image classification. Because of this joint process, the speed of object detection decreases significantly [6]. Another method from this group is called Mask R-CNN, introduced in 2017, which has a faster approach to object detection. In this algorithm, objects are detected by performing four steps. First, with the help of a convolution network (Resnet 50 or Resnet 101) a feature mAP is obtained from each image, and then in the second step, with the help of the RPN network, it extracts the proposed regions that are likely to have objects in them, and finally in the third step, the identified regions are classified and the correction of the proposed bounding box is done. In the fourth step, a convolution network is increased with the help of defined masks to the size of the boxes surrounding the proposed regions and provide the final masks for each object [8].

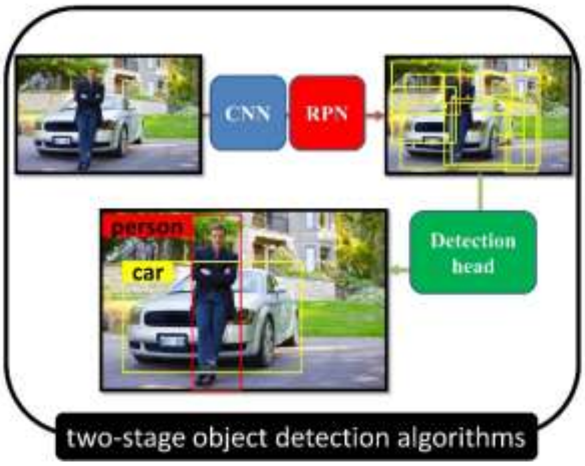

Figure 2. Two-stage detection of objects.

one-stage methods were proposed to solve the challenge of low speed in two-stage methods. In one-stage methods, the RPN block in two-stage methods is removed, so in this category, according to Figure 3, there is no more RPN part in single-stage object recognition, and the CNN output feature map goes directly to DH. The famous YOLO network is the founder of one-stage methods, which is still very popular among researchers [9].

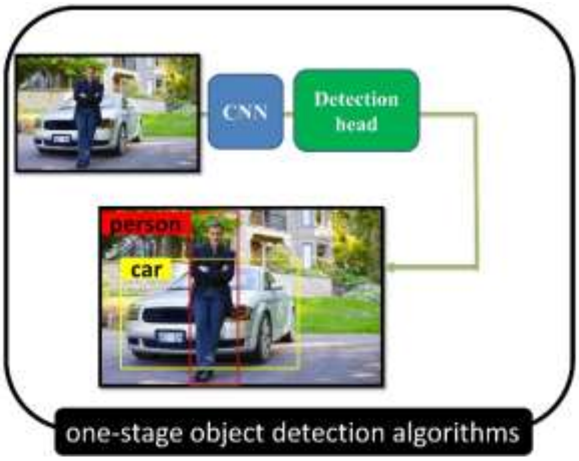

Figure 3. Single-stage detection of objects.

Therefore, to improve the speed and accuracy of region-based object detection methods, Redmon et al. converted direct object detection to regression and proposed the YOLO object detection method. In the initial version of Yolo, the Darknet trained network was used, which of course faced problems in detecting small objects [9]. In the second version of the YOLO algorithm, the Darknet-19 network was used with 19 convolution layers and 5 max pooling layers, and the innovation used in this version is to use the k-means clustering algorithm to select six anchor boxes of different sizes according to the training dataset [10]. In the third version of Yolo algorithm, 53 convolution layers are used for detection and more than 53 layers for final detection, which includes about 106 layers in total.

The innovation used in this version of the algorithm is the use of residual blocks, which has made this version relatively outperform the previous versions [11]. In the fourth version, the YOLO algorithm is used in the GPU model of CSPDarknet53 and to improve the performance of network. The innovations of this version are divided into BOF and BOS, among which the most important ones include cases such as the use of the mish activation function, changes in the cost function, unique data augmentation methods, and so on [12].

YOLOv5 is highlighted for its implementation using the PyTorch library, resulting in significantly reduced training time compared to YOLOv4 while maintaining similar object detection performance. YOLOv6, developed by the Meituan team, introduced architectural improvements and hardware-aware designs. YOLOv7 reduces parameters and computations while achieving high performance [13-15].

YOLOv8s underscores the growing misuse of drones, prompting the need for robust detection technology. The paper proposes a generalized real-time flying object detection model using YOLOv8s. The methodology involves training on a diverse dataset and transferring weights for refinement. The study showcases the effectiveness of YOLOv8s in detecting aerial objects, positioning it as a potential state-of-the-art solution [3].

Detecting vehicles in urban environments has more challenges than detecting objects in non-urban environments (roads and highways) because detecting objects in urban environments comes with many challenges, including the effects of various factors, such as more congestion in urban environments compared to road environments, the presence of different fixed and moving objects in the background of images and various environmental changes, etc. so that the percentage of object detection by detection algorithms in urban environments ranges from 60 to 86%, but in suburban environments, they have reached numbers higher than 90% [1].

There are many articles on different versions of the YOLO algorithm or its upgraded and personalized versions used for vehicle detection, each of which has been upgraded for a specific purpose; for example, vehicle speed monitoring [16,17]; supervision of traffic violations [18]; vehicle tracking [17,19]; classification of vehicles [17, 20-23]; vehicle counting system on streets and highways [17]; parking point detection from the car's point of view for parking attendants [24];

monitoring the parking place [24], detecting cars in special environmental conditions [25,26], and detecting emergency vehicles in snowy weather [27,28] for which there should be a suitable dataset for training the network. Among the famous datasets used in this field, we can refer to the KITTI dataset [29], which contains images of highways and ordinary roads, generally used for training networks to detect and track objects for automatic driving. Another famous dataset in this field is the Stanford Car dataset [30], which contains 19,618 categories of vehicles that include the brands, models, and years of production of cars. Still another famous dataset in this field is the Comprehensive Cars Dataset [31], which contains 27,618 images including maximum vehicle speed, number of doors, number of seats, and car type.
The next dataset is BIT-Vehicle, which provides 9,850 images of vehicles captured from above by surveillance cameras, including SUV, sedan, minivan, truck, bus, and minibus categories [32]. The next dataset is the UA_DETRAC dataset, which contains 10 hours of footage captured with a Cannon EOS 550D camera at 24 different locations in Beijing and Tianjin, China. Videos are recorded at 25 frames per second with a resolution of 540 x 960 pixels and ultimately contain 8,250 vehicles [33].
Several existing datasets for Iranian cars predominantly focus on classification algorithms, particularly centering on domestically produced vehicles. Notably, the SIVD [34] dataset, comprising 29 classes and 36,705 images, prioritizes the development of a classification network for Iranian vehicle recognition. However, it lacks labeled images for identifying specific cars. Another dataset, the IRVD [35] dataset, emphasizes the preparation of large-scale datasets for each country, introducing a new standard dataset for popular Iranian vehicles, focusing on vehicle classification and license plate recognition. In contrast, our dataset, presented in this article, offers a unique approach. It is meticulously curated from online car sales platforms and urban intersections, encompassing all categories of cars—whether domestically produced, assembled, or imported. This diversity positions our dataset as a valuable resource for training car recognition algorithms.

3.Dataset

In this article, a database consisting of images of common vehicles in Iran from websites and companies selling vehicles in Iran (Divar site [36], Bama site [37]) as well as traffic recorded images from several intersections has been collected and used in the city of Hamedan, Iran, which has provided a unique dataset for training the traffic detection network of Iranian vehicles in urban roads, which has not been provided in any format so far. In order to label images, there are different methods and software. In this paper, a web-based software called Roboflow is used, which provides special features for labeling and data management. The collected dataset is labeled in three classes: car, bus and truck. Figure 4 shows an example of each labeled image from each class.

Figure 4. Image of classes.

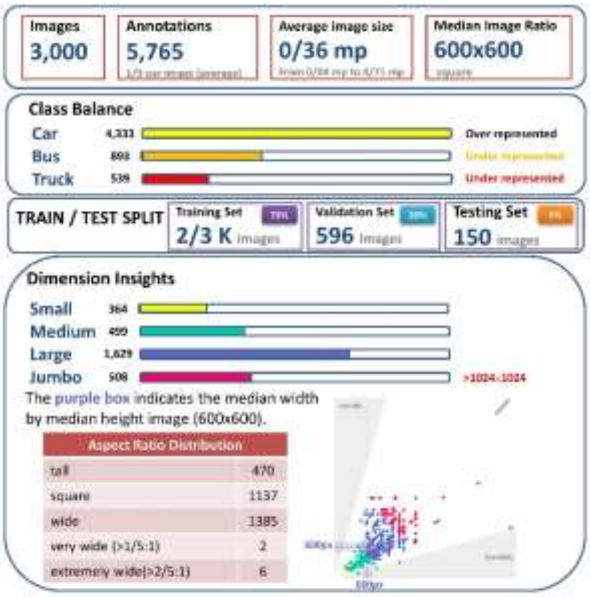

Figure 5. Dataset parameters.

The total number of images in the dataset is 3000, which includes 5,765 labels of all classes. The average size of the dataset images is 0.36 mega pixels and the average dimensions of the images are 600 x 600 pixels, which are suitable for the input of deep neural networks. The distribution of labels in the dataset images for different classes and some other parameters is shown in Figure 5, and as can be seen, the number of samples in the bus and

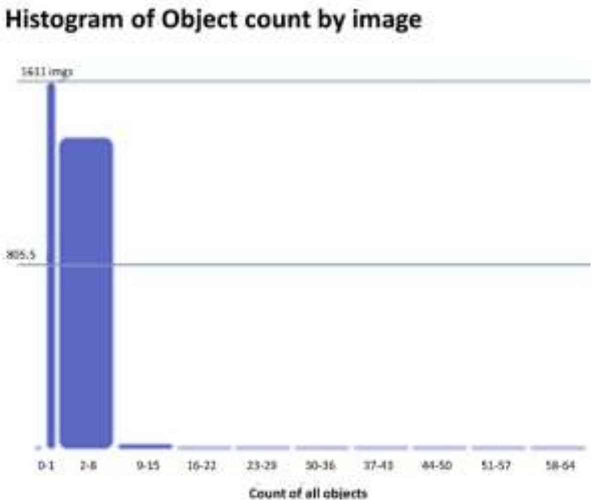

Figure 6. Dispersion of vehicle counts in dataset.

Figure 7. An example of different directions of a vehicle in the dataset.

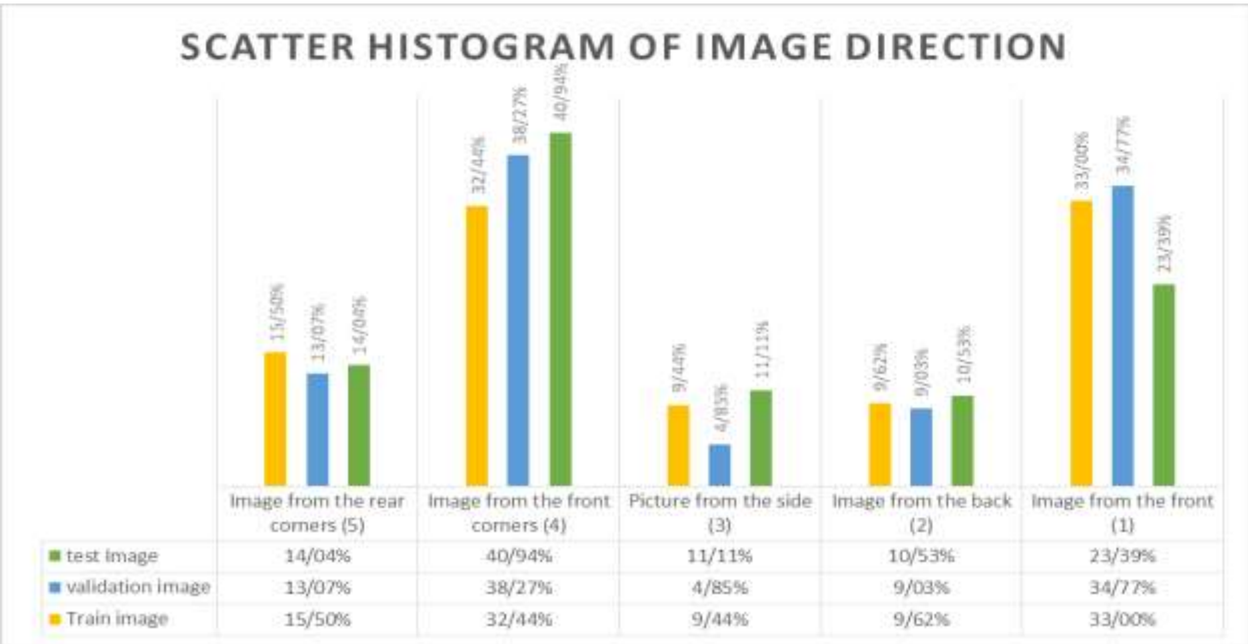

Figure 8. **Scatter histogram of image direction.**

truck class is much less than that of the car class, which should be taken into account when training the network.

The histogram or dispersion of the number of objects (vehicles) in the dataset images is shown in Figure 6. As can be seen, the dataset images are generally single objects or "two to eight objects" and the images with a large number of objects in the dataset are very rarely seen.

The images downloaded from online car sale sites have been selected from the site's latest ads and the most popular items for sale (the most popular models), because this choice makes the selected images the best representative for images of vehicles in use in Iran.

As the pictures were taken by the sellers of the vehicles and the buyers have to see all the directions of the vehicle when buying, generally there are images from different directions of the same vehicle. An example of these images taken from different directions of a vehicle can be seen in Figure 7, and this issue can help a lot to learn object detection networks.

The following histogram in Figure 8 shows the dataset images scattered in 3 categories: Training data, Validation data, and Test data. To divide the directions, Figure 9 is used, which includes the front direction, the side, the back, the front corners and the rear corners, and as it is known, the dataset has a good richness in terms of the variety of the directions of the images, and it is a good representative for network training and evaluation in all directions.

Figure 9. The method of dividing the directions of images

Figure 10. Different versions of vehicles produced by Iran Khodro and Saipa

Iran has 2 main vehicle sale factories, namely Iran Khodro and Saipa, each having several best-selling products, which includes the majority of vehicles currently in use in Iran. The dataset collected in this article also has suitable samples of vehicles produced by these two factories. For example, one of the best-selling products of the Saipa factory is Pride, and one of the best-selling products of the Iran Khodro factory is the assembly cars of the French Peugeot company with modifications, each having different versions. Figure 10 shows the existence of different versions and different images of these products. The dataset shows that the situation is the same for other products of these two factories, and considering that the images were collected from the two main car sale sites in Iran, other popular brands in Iran also have good representatives in dataset images.

Next, in order to train and evaluate the network, the dataset is randomly divided into three categories: training data, validation data, and a category completely outside of this collection is prepared for the final test of the network, called test data.

In order to improve the training of YOLO algorithm and the training time, the images have been converted to 800 x 800 and due to the problem of imbalance in the dataset (caused by the large difference in the number of samples from each class in the dataset), the data augmentation method has been used. This approach includes the following techniques: turning the images from left to right and top to bottom, rotating the image to a random value, cutting the image to a random size, erasing randomly, mixing images, mosaic data augmentation, etc. An example of the images produced after data augmentation is shown in Figure 11, indicating the types of data augmetation used in this article on the images [38-41].

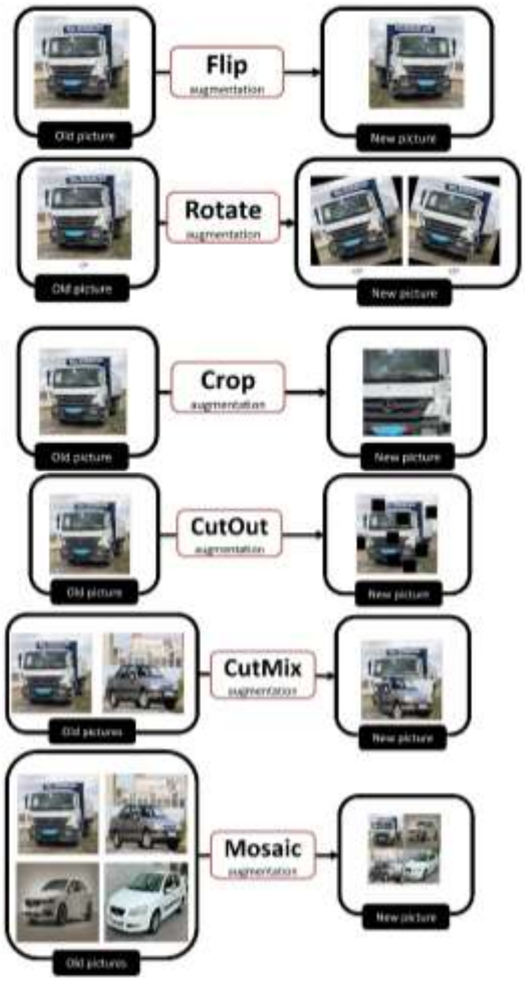

Figure 11. Types of augmentation on the images

4. Experimental setups and results

In order to train the algorithm, TCI TELSA P-100 graphics card is used in this article, which has about 16 GB of memory.

Table 1. Comparison of YOLOv8s Algorithm Outputs: Trained with Dataset from this Article vs. COCO Dataset

class	percision		recalling		Map for threshold 50 %		Map for threshold 50%-90%	
	COCO Dataset	Our Dataset	COCO Dataset	Our Dataset	COCO Dataset	Our Dataset	COCO Dataset	Our Dataset
For all classes	85.9%	91.7%	82.1%	86.6%	82.43%	92.6 %	48.5%	70.2 %
car	88.4%	90.3 %	81.6%	83.1 %	77.4%	90.7 %	49.3%	64.1 %
bus	89.2%	95 %	84.4%	90.6 %	85.3%	96.3 %	50.8%	74.9 %
truck	80.1%	89.9 %	80.3%	86.1 %	84.6%	90.8 %	45.6%	71.5 %

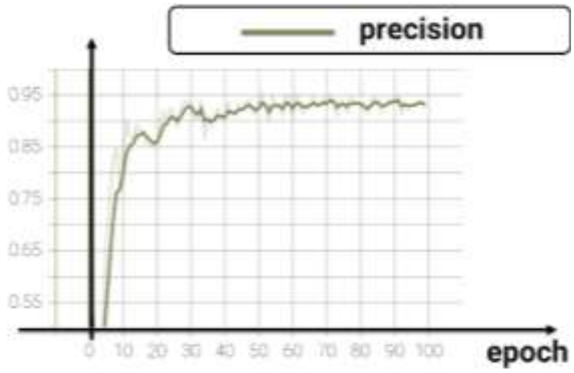

Figure 12. Precision during network training.

Figure 13. Validation box loss during network training.

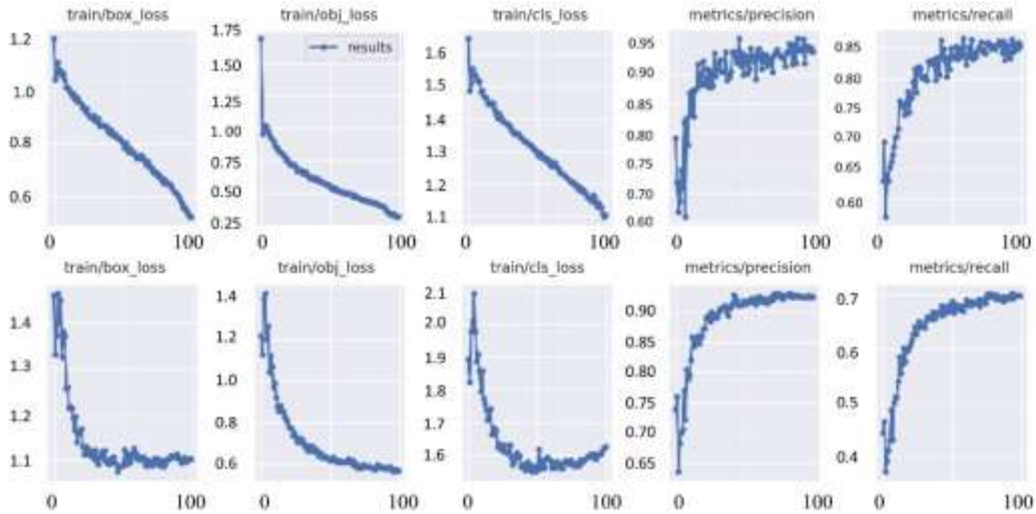

Figure 14. Plots of box loss, objectness loss, classification loss, precision, recall and mean average precision (mAP) over the training epochs for the training and validation set.

The total number of epochs considered for training is 100 and the learning rate is set downward from the initial value of 0.01 to the final value of 0.001. To examine the results obtained after training the first network, we evaluated the percision of the network. As seen in Figure 12, the percision of the network has increased and almost at the end of the training, a constant trend has taken place. It can be said that due to the constant trend of increasing percision at the end of training and not increasing percision, the selected point for ending network training is the right point. In the following, loss reduction has been investigated. As can be seen in Figure 13, the loss has a completely downward trend, which shows that the training method of the network has been properly selected and performed. Of course, to make a definitive statement, more investigation is needed on other network performance parameters, which will be discussed further. The network demonstrates an impressive overall precision of 91.7%, coupled with an outstanding mAP of 92.6% at a 50% threshold. Notably, the Bus class achieves the highest performance with an mAP of 96.3%, potentially attributed to inherent differences in the visual characteristics of buses compared to other vehicle classes.

Comprehensive results and criteria are detailed in Table **1**, while the graphical representation in Figure 14 vividly illustrates the model's enhanced performance across various metrics during both training and validation phases. Furthermore, to emphasize the substantial progress, we compare the YOLOv8 algorithm trained with the COCO dataset. Despite the absence of specific classes (car, bus, and truck) in the COCO dataset, our model, when tested on the dataset presented in this article, In the mAP for a 50% threshold, there is approximately a 10% improvement, and for the mAP across the 50%-90% threshold range, the improvement is around 22%, as reflected in the results presented in Table **1**. Figure 15 and Figure 16 are examples of the output of the algorithm images after training the network. It can be said that the network has been able to detect the location of objects and their class with high percision.

Figure 15. Images showing the performance of detecting by using Yolov8s algorithm.

Figure 16. The result images from the processing of the traffic control camera installed in Be'sat Boulevard, Hamadan, Iran by the algorithm developed in this article.

According to the obtained results, it can be said that the dataset introduced in this article, which has been used for training the network, has a suitable quality and quantity, and the trained network can also cope well with the detection of vehicles in Iran in order to identify traffic. The YOLOv8 algorithm used in this article is one of the emerging cases in the field of object detection, which we see its progress and increase of use every day. To train a deep neural network as well as possible, above all, the existence of a good dataset can directly affect the performance of the system.

5. Result and conclusion

In future, the artificial intelligence transportation system will have a special place in the traffic control of cities so that they will gradually become an essential factor in the process of developing new infrastructure and systems for the efficiency of transportation systems. Also, with the emergence of smart vehicles and the ability to connect these devices with smart agents, there will definitely be new ways to develop transportation systems that can replace classic control methods with smart control techniques, but the lack of local datasets and the difficulty of collecting and labeling datasets has been a barrier to develop these systems and better training and usability of the networks. In this research, in order to identify Iranian vehicles, a new and optimal dataset has been introduced with the help of a single-stage method (YOLO algorithm) in order to detect Iranian vehicles for urban traffic control in the city of Hamedan, Iran. The dataset used in this article is completely unique, which has been collected, modified and labeled by the researchers of this article, and it can be very useful for future work, made available to the public for this purpose. One advantage of the smart methods used in this article is the good ability of the algorithm to adapt to different environments, and it can be said that in future better efficiency and performance can be achieved by adjusting the network parameters.

References

[1] Buch, N., S.A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920-939, 2011.

[2] Yang, Z. and L.S. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image and Vision Computing*, vol. 69, pp. 143-154, 2018.

[3] Reis, D., Kupec, J., Hong, J., & Daoudi, A., "Real-Time Flying Object Detection with YOLOv8," arXiv preprint arXiv:2305.09972, 2023.

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

[5] Du, J., "Understanding of object detection based on CNN family and YOLO," in *Journal of Physics: Conference Series, IOP Publishing*, 2018.

[6] Ren, S., He, K., Girshick, R., & Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[7] Girshick, R., "Fast R-CNN," *in Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[8] He, K., Gkioxari, G., Dollár, P., & Girshick, R., "Mask R-CNN," *in Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961-2969, 2017.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[10] Redmon, J. and A. Farhadi, "YOLO9000: better, faster, stronger*," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] Redmon, J. and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv*:1804.02767, 2018.

[12] Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv*:2004.10934, 2020.

[13] Nasehi, M., Ashourian, M., & Emami, H. "Vehicle Type, Color and Speed Detection Implementation by Integrating VGG Neural Network and YOLO algorithm utilizing Raspberry Pi Hardware, " *Journal of AI and Data Mining*, pp.579-588, 2022.

[14] Team, M., "YOLOv6: A fast and accurate target detection framework is open source," June 23, 2022. [Online].Available: https://github.com/meituan/YOLOv6.

[15] Wang, C.Y., Bochkovskiy, A., Liao, H.Y., "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2023, pp. 7464-7475.

[16] Asgarian Dehkordi, R., and H. Khosravi. "Vehicle type recognition based on dimension estimation and bag of word classification." *Journal of AI and Data Mining*, vol. 8, no. 3, pp. 427-438, 2020.

[17] Oltean, G., Florea, C., Orghidan, R., & Oltean, V., "Towards real time vehicle counting using YOLO-tiny and fast motion estimation," *in 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME) IEEE*, Cluj-Napoca, Romania, October 2019, pp. 240-243.

[18] Rahman, Z., A.M. Ami, and M.A. Ullah, "A real-time wrong-way vehicle detection based on YOLO and centroid tracking," in 2020 IEEE Region 10 Symposium (TENSYMP), IEEE, Dhaka, Bangladesh, 2020, pp. 916-920.

[19] Al-qaness, M. A., Abbasi, A. A., Fan, H., Ibrahim, R. A., Alsamhi, S. H., & Hawbani, A., "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, no. 2, pp. 211-230, 2021.

[20] Kim, J.-a., J.-Y. Sung, and S.-h. Park, "Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," *in 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), IEEE*, Seoul, Korea (South) 2020, pp. 1-4.

[21] Zhu, E., M. Xu, and D.C. Pi, "Vehicle Type Recognition Algorithm Based on Improved Network in Network," *Complexity*, pp. 1-10, 2021.

[22] Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B., "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, p. 4272, 2018.

[23] Gholamalinejad, H., and Hossein Khosravi. "Irvd: A large-scale dataset for classification of iranian vehicles in urban streets." *Journal of AI and Data Mining*, vol.9, no. 1, pp. 1-9, 2021.

[24] Carrasco, D. P., Rashwan, H. A., García, M. Á., & Puig, D., "T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks," *IEEE Access*, vol. 11, pp. 22430-22440, 2021.

[25] Miao, Y., Liu, F., Hou, T., Liu, L., & Liu, Y., "A nighttime vehicle detection method based on YOLO v3," *in 2020 Chinese Automation Congress (CAC), IEEE*, Shanghai, China, November 2020, pp. 6617-6621.

[26] Huang, S., Y. He, and X.-a. Chen, "M-YOLO: A Nighttime Vehicle Detection Method Combining MobileNet v2 and YOLO v3," *in Journal of Physics: Conference Series, IOP Publishing*, Shanghai, China, 2021, pp. 6617-6621.

[27] Goel, S., Baghel, A., Srivastava, A., Tyagi, A., & Nagrath, P., "Detection of emergency vehicles using modified YOLO algorithm," *in Intelligent Communication, Control and Devices: Proceedings of ICICCD 2018, Springer Singapore*, 2020, pp. 671-687.

[28] Baghel, A., Srivastava, A., Tyagi, A., Goel, S., & Nagrath, P., "Analysis of Ex-YOLO algorithm with other real-time algorithms for emergency vehicle detection," *in Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019), Springer Singapore*, 2020, pp. 607-618.

[29] Geiger, A., P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," *in 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE*, Providence, RI, USA, 2012, pp. 3354-3361.

[30] Krause, J., Gebru, T., Deng, J., Li, L. J., & Fei-Fei, L., "Learning features and parts for fine-grained recognition," *in 2014 22nd International Conference on Pattern Recognition, IEEE*, Stockholm, Sweden, August 2014, pp. 26-33.

[31] Yang, L., Luo, P., Chang Loy, C., & Tang, X., "A large-scale car dataset for fine-grained categorization and verification," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3973-3981.

[32] Dong, Z., Wu, Y., Pei, M., & Jia, Y., "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247-2256, 2015.

[33] Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M. C., Qi, H., ... & Lyu, S., "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[34] Siahkali, F., Alavi, S. A., & Masouleh, M. T., "SIVD: Dataset of Iranian Vehicles for Real-Time Multi-Camera Video Tracking and Recognition," *in 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), IEEE*, Behshahr, Iran, December 2022, pp. 1-7.

[35] Gholamalinejad, H., & Khosravi, H., "Irvd: A large-scale dataset for classification of Iranian vehicles in urban streets," *Journal of AI and Data Mining*, vol. 9, no. 1, pp. 1-9, 2021.

[36] Divar, https://divar.ir/, 2021.

[37] Bama, https://bama.ir/.

[38] Shorten, C. and T.M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1-48, 2019.

[39] Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T. Y., Shlens, J., & Le, Q. V., "Learning data augmentation strategies for object detection," *in Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020, pp. 566-583.

[[40] Kaur, P., B.S. Khehra, and E.B.S. Mavi, "Data augmentation for object detection: A review," *in 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE*, Lansing, MI, USA, 2021, pp. 537-543.

[41] Shin, H.-C., K.-I. Lee, and C.-E. Lee, "Data augmentation method of object detection for deep learning in maritime image," *in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE*, Busan, Korea (South), 2020, pp. 463-466.

Dataset is available in address: https://github.com/pouria-maleki/Iranian-Vehicle-images-dataset-for-detection

مجموعه داده تصاویر خودرو های ایرانی برای الگوریتم تشخیص اشیاء

پوریا ملکی[۱]، عباس رمضانی[۱،*] ، حسن ختن لو[۲] و سینا اجاقی[۳]

[۱] گروه مهندسی برق، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران.

[۲] گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران.

[۳] دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران.

ارسال ۲۰۲۴/۰۱/۲۱؛ بازنگری ۲۰۲۴/۰۲/۰۷؛ پذیرش ۲۰۲۴/۰۳/۱۲

چکیده:

ارائه یک مجموعه داده با حجم مناسب و دقت بالا برای آموزش شبکه‌های عصبی عمیق یکی از الزامات اساسی تلقی می شود که در آن یک مجموعه داده مناسب از نظر تعداد و کیفیت تصاویر و دقت برچسب گذاری می‌تواند تاثیر زیادی در دقت خروجی شبکه داشته باشد. مجموعه داده ارائه شده در این مقاله شامل ۳۰۰۰ تصویر دانلود شده از شرکت‌های فروش آنلاین خودرو ایران از جمله سایت دیوار و باما می‌باشد که به صورت دستی در سه کلاس خودرو، کامیون و اتوبوس برچسب گذاری شده‌اند. برچسب‌ها به شکل ۵۷۶۵ جعبه محدودکننده هستند که خودروهای موجود در تصویر را با دقت بالا مشخص می‌کنند و در نهایت منجر به ایجاد یک مجموعه داده منحصر به فرد می‌شود که برای استفاده عمومی در دسترس قرار می‌گیرد. الگوریتم YOLOv8s که بر روی این مجموعه داده آموزش داده شده است، به دقت نهایی موثر ۹۱٫۷٪ برای تصاویر اعتبار سنجی دست می‌یابد. میانگین دقت (mAP) در آستانه ۵۰٪ در ۹۲٫۶٪ ثبت شده است. این دقت برای شبکه‌های تشخیص وسایل نقلیه شهری مناسب در نظر گرفته می شود. قابل ذکر است، هنگام مقایسه الگوریتم YOLOv8s آموزش دیده با این مجموعه داده با YOLOv8s آموزش دیده با مجموعه داده COCO، افزایش قابل توجه ۱۰٪ در میانگین دقت با آستانه ۵۰٪ و تقریباً ۲۲٪ بهبود در محدوده ۵۰٪ تا ۹۵٪ وجود دارد.

کلمات کلیدی: تشخیص شیی ، YOLOv8s ، مجموعه داده خودرو، شبکه عصبی عمیق.