Journal of Industrial Information Technology and Application
Vol.9 No. 3
ISSN(Online): 2586-0852

VRSMY9: Lightweight Deep Learning-Based Traffic Surveillance with YOLOv9

Sang Suh[*)], and Bilal Mushtaq

Department of Computer Science East Texas A&M University, U.S.A.

Abstract: Overspeeding is a major cause of fatal accidents and car crashes around the world. Drivers that aren't timely held accountable for their recklessness become a threat for themselves and for those sharing the road with them. Numerous researchers have presented machine learning and deep learning-based object detection and recognition techniques that aim to accurately detect vehicles but have come across hurdles such as varying illumination and weather conditions, overlapping vehicle images in complex traffic situations, the need for powerful GPU based computers at surveillance, and slow detection speeds. The proposed work VRSMY9 powered by YOLOv9 and EasyOCR focuses on enhancing the overall detection speed in real-time without compromising too much on the detection accuracy whilst being lightweight enough for computing devices that are deprived of powerful GPUs. The system detects vehicles, estimates their travelling speeds, detects the license plates of the overspeeding vehicles resulting in license plate extraction of all the violators, and then keeps a stored record of them. Results show that the system is highly precise and has decent accuracy with a high detection speed but isn't accurate enough when handling challenging images. The scope of this work is extended but not limited to autonomous vehicles, unmanned aerial vehicles, intelligent transportation systems, robotics, intelligent AI agents, domestic security, healthcare and all other areas where high speed accurate detection is needed on an embedded smart device that may not be computationally super powerful.

Keywords; YOLOv9, EasyOCR, Vehicle Detection, License Plate Recognition, Speed Estimation, Real-time Traffic Surveillance

[*]Corresponding author: Sang.Suh@etamu.edu
Received: May. 7. 2025 Accepted: Jul. 16. 2025 Published: Sep. 31. 2025

Cite this paper as : Sang Suh, and Bilal Mushtaq (2025) "VRSMY9: Lightweight Deep Learning-Based Traffic Surveillance with YOLOv9", Journal of Industrial Information Technology and Application, Vol. 9. No. 3, pp. 1130 – 1145

1. Introductionssss

Computer Vision is the domain of Artificial Intelligence (AI) that focuses on enabling computers to be able to understand images and videos and extract vital information that could be further processed to make decisions and/or predictions. Researchers have worked extensively in this domain, and have used various statistical and probabilistic models, in addition to complex mathematical techniques to address vehicle detection and classification problems. Even though they have seen considerable success, there are still some significant hurdles that need to be overcome to effectively solve this problem.

Traditional methods such as Blob analysis [1][2] and Haar like feature classification [3][4][5][6] techniques aren't robust enough to perform in low lightning conditions. Support Vector Machines and Principal Component Analysis [7], and Convolutional Neural Networks [8][9][10] fail to perform in congested traffic settings, while ResNet [11] and Fast R-CNN [12] aren't able provide a fast detection speed due to their deep layered network size. 3D modeling techniques perform best when distinguishing among vehicles of varying shapes and sizes but are not scalable enough to work with large commercial datasets [13] [14] [15].

To address these challenges, we propose a framework based on YOLOv9 [16] detection model and EasyOCR [17] feature extraction model that aims to deliver faster vehicle detection and license plate extraction in real-time traffic scenarios without compromising significantly on the accuracy, along with the ability to estimate the travelling speeds of the respective vehicles so that the violators could be penalized and brought to account. The hallmark of this system is its ability to operate on systems that are deprived of the luxury of having powerful GPUs at their disposal all the time, making it a practical solution for the existing commercial traffic infrastructure.

The model was trained on a dataset containing diverse sets of vehicle images containing varying angles, different weather and lightning conditions, and various license plate designs, for a 1000 epoch. The experimental results show that the framework is highly accurate in avoiding false positives but missed some valid detections, making the overall true positive

score decent. Even though the detection accuracy is impressive in general cases, the performance drops a bit when handling difficult cases. Moreover, the framework successfully carries out speed estimation of the vehicles in motion and can differentiate the overspeeding ones from the vehicles that are found to be moving within the designated speed limit.

2. Literature Review

Salvi [1] put forward an automated vehicle counting system used in traffic surveillance that utilizes a five-stage algorithm. The steps include background subtraction, blob detection, blob analysis, blob tracking, and vehicle counting, enabling the system to be quite efficient in providing real-time traffic surveillance. Ramalingam and Varsani [2] carried out an in-depth comparative analytical study on algorithms used for effective detection and tracking of vehicles. The blob analysis, optical flow, and foreground detection algorithms are tested and compared using multiple video sequences in varying levels of difficulty. Wei et al [3] presented the idea of using the combination of Haar and HOG features to design a multi-vehicle detection algorithm. It takes advantage of the commendable descriptive ability of the HOG feature and the extraction of the prospect ROI using Haar Features. Furthermore, the extraction of the precise target is carried out using SVM making the entire algorithm efficient in complex urban scenarios. Hasan et al [4] presented a comprehensive analysis in which a comparison of Haar cascade classifiers and Blob analysis was carried out based on their detection and classification performance on multiple datasets.

Han et al [5] put forward the technique of robust detection of vehicles using Haar-like features using a two-step custom algorithm. Moreover, the paper suggests that the removal of the shadows of vehicles and accurate calculation of the vehicle boundary is essential for efficient detection. ElKerdawy et al [6] presented a framework that uses a stationary camera for the detection and tracking of vehicles. Vehicles are detected using Haar-like features and the framework can calculate statistical information such as average traffic flow and speed that contribute to the model's overall accuracy. Wen et al [7] proposed a Haar-like feature selection and classification technique for vehicle detection via AdaBoost. Furthermore, a normalization algorithm is designed according to the specific features to reduce the inter-class difference thereby increasing the inter-class variability.

Convolutional Neural Networks (CNN) based methods have recently been regarded as one of the most efficient and accurate in carrying out object detection tasks. The CNN deep

learning algorithms have made a special mark in the domain of Computer Vision and Intelligent Transportation Systems (ITS), particularly in applications that require vehicle detection. He et al [8] compared the accuracy results of the CNN model with multiple feature extracting and classifying models such as HSV color histogram and direction of controllable characteristics model (for important features extraction) combined with SVM classifier and a large-scale sparse learning model and concluded that the deep learning algorithm proved to be far more superior to the rest. Li et al [9] designed a fully convolutional neural network to detect and localize objects as 3D boxes from range scan data using a Velodyne 64E Lidar sensor and were able to predict 3D boxes using 2D CNN. Chen et al [10] worked on an improved convolutional neural network with the baseline upon Faster R-CNN and SSD for fast and accurate detection of highway vehicles. They used k-means to cluster the dataset to learn prior information, used feature concatenation to extract rich features, and detected various features from vehicles of different sizes to enhance overall detection.

ResNet or Residual Neural Network is state-of-the-art and one of the most popular Deep learning techniques that uses Convolutional Neural Network (CNN) as its baseline network. It makes the use of identity mapping and stacking of multiple layers to achieve high accuracy in classification and localization procedures. Jung et al [11] proposed a vehicle localization and classification model that was based on ResNet and used original traffic surveillance recordings. The images consisted of real-life traffic scenarios in varying weather and illumination conditions. A ResNet consisting of 18 layers with ReLU was used to perform the classification task while a Region based Fully Convolutional Neural Network (R-FCN) was applied to carry out the localization. Moreover, they proposed the use of joint fine-tuning combined with DropCNN to improve the algorithm's overall accuracy. Results showed that the model outperformed well-reputed state-of-the-art algorithms in terms of mAP and other performance parameters. Fast R-CNN put forward by Girshick [12] is a revolutionary single-stage training algorithm that can refine spatial locations and recognize object proposals at the same time. It is considerably faster than R-CNN and surprisingly more accurate than it as well.

Feris et al [18] suggested the searching of vehicles in surveillance videos based on particular semantic attributes (such as dimensions, color, size, direction of motion et cetera). They used multi-view cameras to extract important features based on motion let classifiers that were then stored in databases for vehicle classification. Moreover, they trained the motion-let detectors in a way that all the training samples are resized to the same aspect

ratio so that various kinds of vehicles with discrepancies among their sizes and shapes could be dealt with effectively.

Guo et al [19] proposed an approach that uses coarse-to-fine structured feature embedding for the purpose of carrying out Vehicle Re-Identification. They implemented coarse to fine ranking loss term via multiple techniques: Modeling of vehicle classification loss with cross entropy loss, Application of Coarse grained ranking loss to enhance the distinction among vehicles of different models and to store the few variations among vehicles of the same model, Construction of Fine-grained ranking loss term to clearly discriminate between the different images of the same model vehicles and finally, to achieve intra-class compactness, they introduced a Pair-wise loss term. Furthermore, they presented the "Vehicle-1M" Vehicle Re-Identification dataset consisting of 1 million vehicle images collected in various surveillance scenarios. Fang et al [20] worked on a model that utilized feature maps to detect and recognize the unique parts of any vehicle that makes it different from the rest.

Hence, there is a need for a system that can efficiently carry out detection, speed estimation and license plate feature extraction of vehicles, while being lightweight enough to be implemented on the existing traffic infrastructure so that accurate practical implementation is possible.

3. Methodology

The proposed system initially uses a camera to take real-time traffic videos as input where the focus is on the moving vehicles. YOLOv9 is utilized for carrying out vehicle detection in real-time. Among all the vehicles in the flowing traffic, the system then aims to find out which of them are travelling over the speed limit. For this purpose, an effective method to determine the speed of each moving vehicle is used where two designated points are set in our frame of reference and it is calculated how much time a vehicle takes to move from point A to point B. This simple yet effective method returns the speed of each detected vehicle thereby letting us compare their respective speeds to the predetermined threshold i.e. speed limit. In case the speed is below the limit, the system ignores that vehicle but if the speed is over the limit, the system then passes it on to the next steps. All the over speeding detected vehicles are then passed on to the next section where the system uses the Optical Character Recognition (OCR) method to extract the features from the respective license plates of all the over speeding vehicles. This enables the system to carry out vehicle

recognition based on the contents of the license plate of the respective vehicle, resulting in further action to be taken by authorities to address the noncompliance.

Unlike past works, our system does not rely on a computationally expensive deep layered network that would take its sweet time to detect a vehicle and then more time to extract the important features for carrying out vehicle recognition. Because even though computationally expensive deep layered networks boast high accuracy for detection and recognition tasks, they aren't feasible to be implemented in real time applications where fast detection is necessary, without compromising considerably on the accuracy of detection and feature extraction. Furthermore, the other novelty of our approach is the simplicity of the overall network. We propose a network that is not only fast and accurate but can be hosted on medium to lightweight devices, after the training and testing is completed. Furthermore, our system aims to overcome the most occurring issues that the past systems haven't been able to tackle such as varying lightning conditions, complex traffic scenarios, varying vehicle types, overlapping vehicle images, scalability with different datasets, etc.

3.1 VRSMY9 System Architecture

Figure 1 gives a bird's eye view of how the architecture of the proposed VRSMY9 system looks like. The system takes real-time traffic video feed as input from multiple sources, depending on the application, and initially carries out vehicle detection using YOLOv9 by forming bounding boxes around every vehicle that can be seen passing through in the frame. Then, each vehicle's speed is estimated based on how much time it takes for it to travel from one point in the frame to another designated point. After that, the estimated speed is compared with the set threshold to determine whether it is within the speed limit or not. As Figure 1 further illustrates, license plate detection is then carried out using YOLOv9 for all the vehicles that seem to be travelling over the speed limit and then their license plate contents are extracted using EasyOCR to trace the owner of the vehicle.

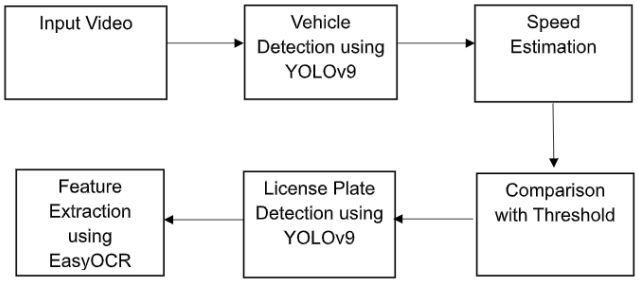

Figure 1. VRSMY9 System Architecture

3.2 YOLOv9

YOLOv9 proposed by Wang et al [16] introduces groundbreaking improvements in real-time object detection, particularly through Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). PGI is an innovative feature designed to address the issue of information bottlenecks, ensuring that crucial data is retained throughout deep network layers. This helps produce more reliable gradients, leading to precise model updates and enhancing overall detection accuracy. GELAN is a key architectural improvement that optimizes parameter usage and boosts computational efficiency. Its flexible structure allows for the seamless integration of different computational blocks, making the model adaptable to various applications without compromising on speed or accuracy. YOLOv9's design tackles the challenge of information loss in deep neural networks, improving the model's learning capacity and detection performance. These innovations enhance both accuracy and efficiency, making it a great model to use in relatively lighter weight applications where high accuracy is required.

3.3 EasyOCR

Optical Character Recognition (OCR) [17] is a technology that converts text within images, such as scanned documents or photos, into machine-readable text. The process begins with an image that contains text, which serves as the input for the OCR system.

In the pre-processing step, the input image is prepared for text detection. Pre-processing includes resizing the image, converting it to grayscale, and removing noise. These

transformations improve the system's ability to accurately identify text regions. After pre-processing, the image passes to a deep learning model specifically designed to detect areas within an image that contain text highlighting individual characters and groups of characters. The model produces bounding boxes around text regions, which are then passed to the next stage.

The detected text regions are further refined and prepared for text recognition. This involves adjusting the bounding boxes and normalizing the regions to enhance recognition accuracy. The refined text regions are then fed into a recognition model. After recognition, the Greedy Decoder interprets the model's predictions to produce readable text and selects the most probable characters to form words.

In the post-processing step, the decoded text undergoes final processing to enhance readability and correctness. This involves correcting errors, filtering out unwanted characters, and formatting the output for consistency. The final recognized text is then presented in machine-readable format.

4. Implementation and Results

During training, we used a dataset [21] that contained a plethora of images containing vehicles with license plates in varying traffic and lightning conditions, appearing in a variety of angles. After the training, the model was able to produce a confidence score for each prediction of the license plates, implying the likelihood of that segment of the image being a license plate.

Even though the models used to build this system were pretrained, they were further trained for a 1000 epoch with the dataset to be finetuned for this application. As a result, the classification model scored a perfect 100% whilst identifying "background" of an image and never took it as a vehicle or license plate for that matter. As far as identifying license plates is concerned, the model was able to correctly identify 80% of the license plates, whilst misclassifying the rest of the 20%.

Figure 2. shows a confusion matrix that highlights all the true positives, true negatives, false positives, and false negatives.

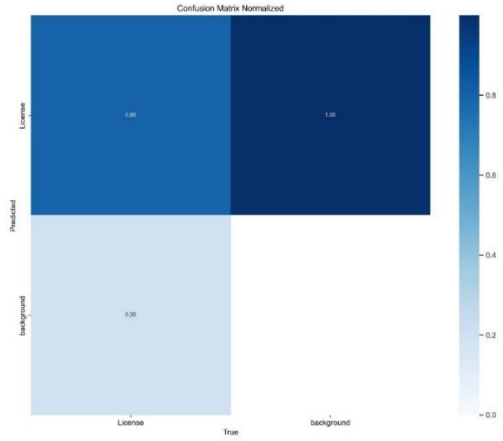

Figure 2. Normalized Confusion Matrix

The accuracy of this system can be inferred from its precision, recall, and mean average precision (mAP) metrics, as shown in Figure 3:

Precision (0.75–0.95): Indicates the model is highly accurate in avoiding false positives, meaning when it predicts a detection, it is correct most of the time.

Recall (0.6–0.8): Shows the system moderately captures true positives, suggesting it misses some valid detections.

mAP@50 (0.7–0.85): Represents strong overall detection accuracy at a commonly used IoU threshold (50%).

mAP@50-95 (0.25–0.40): Highlights stricter evaluation across multiple IoU thresholds, where the system's performance drops, suggesting challenges in precise localization and handling difficult cases.

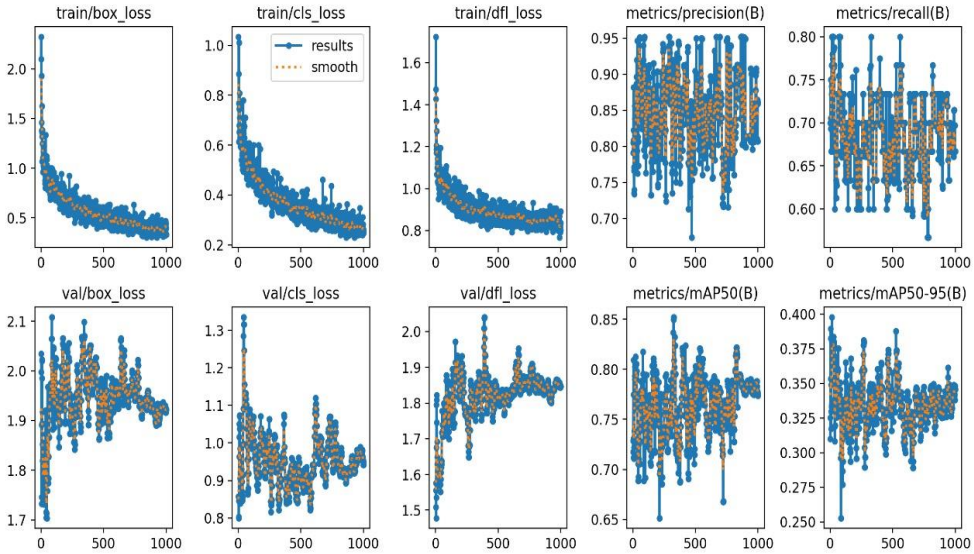

Figure 3. Performance metrics of the VRSMY9 system after 1000 epoch

As Figure 3. shows, the training metrics indicate consistent improvement, with the box loss starting at ~2.5 and stabilizing below 0.5, reflecting better bounding box predictions. Similarly, the classification loss decreases from 1.0 to ~0.2, showing enhanced object classification, while the distribution focal loss stabilizes around 0.8, signifying improved localization of bounding boxes. In contrast, the validation metrics exhibit slightly higher and fluctuating losses, with the box loss stabilizing between 1.8 to 2.0, and classification and focal losses showing a broader range. These trends suggest possible overfitting to the training data or a more challenging validation set.

To test the system in real time, we set the speed limit at 10 mph and utilized a live traffic surveillance video to see how the system performs in a live scenario. Ideally, the system should detect every vehicle passing in the frame and determine its speed. To estimate the speed, we used a simple technique by marking two lines in the frame that are 6m apart from each other. Based on the time it takes for each vehicle to travel from one point/line to the other, dividing it by the time the vehicle takes would yield its estimated speed. Hence, every vehicle travelling within the speed limit would have its respective speed reported in green. On the other hand, those vehicles travelling over the speed limit would have their respective speeds reported in red, thereby subjecting them to go through the license plate detection phase. Every over speeding vehicle's license plate is first detected and then extracted from

the video feed and then saved in a separate folder, where the vehicle's image and its license plate contents are recorded. Figure 4 shows the results of a vehicle travelling within the speed limit.

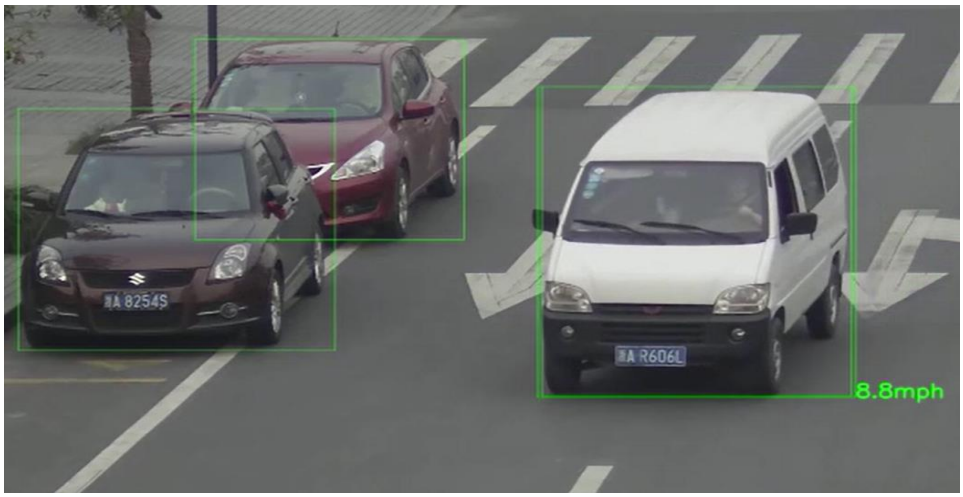

Figure 4. Vehicle travelling within the speed limit using VRSMY9

Notice in Figure 4 that both the vehicles parked on the side of the road are also detected by the system, even though they aren't moving at all. This is because every vehicle within the frame, whether it is stationary or moving, will be detected by the system automatically.

Figures 5 and 6 show instances of vehicles travelling over the speed limit. Once the system detects that they are travelling over the speed limit, it detects the license plate, takes a snapshot of the plate and stores it with its contents in the specified folder keeping a record of all the over speeding vehicles.

Figure 5. VRSMY9 Vehicle travelling over the speed limit

Figure 6. VRSMY9 Vehicle travelling over the speed limit

5. Comparative Analysis and Discussion:

Table 1. highlights the comparison of the most renowned techniques with our proposed work.

Technique(s) Used	Contributions	Shortcomings
Blob Analysis [1] [2]	Cost effective compared to traditional methods	Not effective in low lightning, bad weather and high traffic density
Haar like features classification [3] [4] [5] [6]	Better detection results in complex and dense traffic conditions	Lack of robustness and unable to detect accurately in low lightning conditions
SVM and PCA [7]	Improvement in detection time	Not effective whilst detecting real-time congested traffic
Convolutional Neural Network (CNN) [8] [9] [10]	More effective than traditional methods when handling various vehicle shapes and sizes	Relatively lower classification accuracy in adverse traffic conditions
ResNet [11]	Deep layers produce better classification results	Computationally expensive
Fast R-CNN [12]	Faster processing speed compared to its predecessors	Needs large amounts of data and isn't fast enough for real-time applications
3D modeling and 3D Bounding Boxes [13] [14] [15]	Most accurate and effective in distinguishing between closely related vehicle models and variations	Lack of scalability when working with large datasets
Proposed VRSMY9	Lightweight model with an improvement in detection and feature extraction speed	Not accurate enough when handling challenging images

Table 1. Comparison of related works with the proposed VRSMY9

6. Conclusion and Future Works

This paper presents a framework based on YOLOv9 object detection and classification model, and EasyOCR feature extraction model. The proposed VRSMY9 system addresses the need for a real time vehicle detection and speed estimation system that is lightweight and accurate enough to be implemented on devices that might not have powerful GPUs at their disposal. Furthermore, the system aims to deliver consistent performance in varied weather conditions and complex traffic scenarios. To achieve this, the model was trained on a dataset containing diverse set of vehicle images containing varying angles, different weather and lightning conditions, and various license plate designs, for a 1000 epoch. The experimental results show the framework is highly accurate in avoiding false positives but missed some valid detections, making the overall true positive score decent. Even though

the detection accuracy is impressive in general cases, the performance drops a bit when handling difficult cases. Moreover, the framework successfully carries out speed estimation of the vehicles in motion and can differentiate the overspeeding ones from the vehicles that are found to be moving within the designated speed limit.

The biggest challenge that we came across was finding the right data with enough images that weren't distorted and had visible license plates. Another lesson learned was that speed estimation powered by computer vision isn't as effective compared to utilizing radar sensors for estimating speeds. Even though our aim to achieve a faster detection speed without compromising too much on accuracy was achieved to some extent, the model didn't showcase a high enough accuracy as we would have liked when handling difficult cases such as overlapping vehicle images and unclear license plates. We hope to take this research as a step forward in the right direction so that one day we can achieve the unmet objectives in their entirety as well.

This framework serves as an important steppingstone for further research in the fields of autonomous vehicles, unmanned aerial vehicles, intelligent transportation systems, robotics, intelligent AI agents, domestic security, healthcare and various other walks of life, where computationally inexpensive systems are to be built without compromising on accuracy and performance.

References

[1] G. Salvi. "An Automated Vehicle Counting System Based on Blob Analysis for Traffic Surveillance". In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), (2012), p.1

[2] S. Ramalingam, and V. Varsani. "Vehicle detection for traffic flow analysis". In IEEE International Carnahan Conference on Security Technology (ICCST), (2016), p.1-8

[3] Y. Wei, Q. Tian, J. Guo, W. Huang and J. Cao. "Multi-vehicle detection algorithm through combining Harr and HOG features". In Mathematics and Computers in Simulation Vol: 155, Elsevier, (2019), p.130-145

[4] Y. Hasan, M. U. Arif, A. Asif and R. H. Raza. "Comparative Analysis of Vehicle Detection in Urban Traffic Environment Using Haar Cascaded Classifiers and Blob Statistics". In 2016 Future Technologies Conference (FTC), (2016), p.547-552

[5] S. Han, Y. Han and H. Hahn. “Vehicle Detection Method using Haar-like Feature on Real Time System”. In International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol: 3, No: 11, (2009).

[6] S. ElKerdawy, R. Sayed, and M. ElHelw. “Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking”. In First Iberian Robotics Conference. Advances in Intelligent Systems and Computing, vol: 252. Springer, (2013).

[7] X. Wen, L. Shao, W. Fang and Y. Xue. “Efficient Feature Selection and Classification for Vehicle Detection”. In IEEE transactions on circuits and systems for video technology, (2015)-03, Vol.25 (3), p.508-517

[8] D. He, C. Lang, S. Feng, X. Du and C. Zhang, “Vehicle Detection and Classification Based on Convolutional Neural Network”, In Proceedings of the 7th International Conference on Internet Multimedia Computing and Service, (2015), p.1-5.

[9] X. Li, M. Ye, M. Fu, P. Xu, and T. Li. “Domain Adaption of Vehicle Detector based on Convolutional Neural Networks”. In International Journal of Control, Automation, and Systems, (2015), 13(4), pp.1020-1031

[10] L. Chen, F. Ye, Y. Ruan, H. Fan and Q. Chen, “An algorithm for highway vehicle detection based on convolutional neural network”. In EURASIP Journal on Image and Video Processing, (2018), Vol. 1, p.1-7

[11] H. Jung, M.Choi, J. Jung, J. Lee S. Kwon and W. Y. Jung. “ResNet-based Vehicle Classification and Localization in Traffic Surveillance Systems”. In IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2017), p.934-940

[12] R. Girshick. “Fast R-CNN”. In IEEE International Conference on Computer Vision (ICCV), 2015, p.1440-1448

[13] Y-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. “Jointly Optimizing 3D Model Fitting and Fine Grained Classification”. In ECCV, (2014).

[14]] J. Sochor, A. Herout and J. Havel. “BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition.” In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, p.3006-301

[15] J. Sochor, J. Spanhel and A. Herout. “BoxCars: Improving Fine-Grained Recognition of Vehicles using 3D Bounding Boxes in Traffic Surveillance”. In IEEE Transactions on Intelligent Transportation Systems, 2019, Vol.20 (1), p.97-108

[16] C.Y. Wang, I.H. Yeh, and H.Y.M. Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”. Available at https://arxiv.org/abs/2402.13616, (2024).

[17] JaidedAI, “EasyOCR”. Available at: https://github.com/JaidedAI/EasyOCR, (2020).

[18] R. Feris, B. Siddique, Y. Zhai, J. Petterson, L. Brown and S. Pankati. “Attribute-based Vehicle Search in Crowded Surveillance Videos”. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, p.1-8

[19] H. Guo, C. Zhao, Z. Liu, J.Wang and H, Lu. “Learning Coarse-to-Fine Structured Feature Embedding for Vehicle Re-Identification”. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[20] J. Fang, Y. Zhou, Y. Yu and S. Du. “Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture”. In IEEE Transactions on Intelligent Transportation Systems, 2017, Vol.18 (7), p.1782-179

[21] F. Elmenshaii, “Large-License-Plate-Detection-Dataset”, Available at: https://www.kaggle.com/datasets/fareselmenshawii/large-license-plate-dataset , 2023