Real-time Classification and Counting of Vehicles from CCTV Videos for Traffic Surveillance Applications

T. Indu[1], Y. Shivani[1], Akhila Reddy[1], Dr. S. Pradeep[2]

[1]UG Student, Department of Computer Science and Engineering, Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

[2]Asssociate Professor & Head, Department of CSE-IoT, Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

ABSTRACT

Traffic Analysis has been a problem that city planners have dealt with for years. Smarter ways are being developed to analyze traffic and streamline the process. Analysis of traffic may account for the number of vehicles in an area per arbitrary time and the class of vehicles. People have designed such mechanisms for decades now but most of them involve use of sensors to detect the vehicles i.e., a couple of proximity sensors to calculate the direction of the moving vehicle and to keep the vehicle count. Even though over time these systems have matured and are highly effective, they are not very budget friendly. The problem is such systems require maintenance and periodic calibration. Therefore, this project has proposed a vision-based vehicle counting and classification system. The system involves capturing of frames from the video to perform background subtraction in order detect and count the vehicles using Gaussian Mixture Model (GMM) background subtraction then it classifies the vehicles by comparing the contour areas to the assumed values. The substantial contribution of the work is the comparison of two classification methods. Classification has been implemented using Contour Comparison (CC) as well as Bag of Features (BoF) method.

Keywords: Traffic analysis, vehicle counting and classification, Gaussian Mixture Model (GMM), Bag of Features (BoF).

1. INTRODUCTION

Nowadays countries and governments require reliable and low-cost systems for traffic automation and vehicle theft controlling. The enormous increase in the vehicles on roads and highways, the Increasing congestion and problems associated with existing traffic detectors have motivated the development of new vehicle detection technologies. Computer vision systems are the most common choice, but several issues must be solved to perform the classification successfully. Identifying and track moving objects or vehicles in real-time that appears in different kinds of roads, by an intelligent vision system, is important to many areas of research and technological applications. Extracting useful information such as traffic density, object speed, drivers' behavior, and vehicle types from these camera systems has become critical. Manual analysis is now inapplicable. The development of intelligent systems that are able of extract traffic density and vehicle classification information from traffic surveillance systems is crucial for traffic management. Otherwise, surveillance systems are also important in driver assistance applications because a vision system allows the detection and classification of vehicles that appear in a captured scene.

An image is a visual representation of something. In information technology, the term has several usages. An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics. Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focuses particularly on images. DIP focuses on developing a computer system that can perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms and gives an image as an output. It allows much wider range of algorithms to be applied to the input image and can avoid problems such as build-up of noise and signal distortion during processing. An image can be classified into the following three types.

A binary image is one that consists of pixels that can have one of exactly two colours, usually black and white. Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit—i.e., a 0 or 1. Grey is an intermediate colour between black and white. It is a neutral colour or achromatic colour, meaning literally that it is a colour "without colour" because it can be composed of black and white. It is the colour of a cloud-covered sky, of ash and of lead. A (digital) colour image is a digital image that includes colour information for each pixel. The process is environmentally friendly since it does not require chemical processing. Digital imaging is also frequently used to help document and record historical, scientific, and personal life events. In this paper, a vision-based system for detection, tracking and classification of moving vehicles is described. Four different potential vehicles groups can be identified, but the proposed software is flexible related to the number of groups that can be classified.

2. LITERATURE SURVEY

Alpatov et al. considered road situation analysis tasks for traffic control and ensuring safety. The following image processing algorithms are proposed: vehicle detection and counting algorithm, road marking detection algorithm. The algorithms are designed to process images obtained from a stationary camera. The developed vehicle detection and counting algorithm was implemented and tested also on an embedded platform of smart cameras. Song et al. proposed a vision-based vehicle detection and counting system. A new high-definition highway vehicle dataset with a total of 57,290 annotated instances in 11,129 images is published in this study. Compared with the existing public datasets, the proposed dataset contains annotated tiny objects in the image, which provided the complete data foundation for vehicle detection based on deep learning.

Neupane et al. created a training dataset of nearly 30,000 samples from existing cameras with seven classes of vehicles. To tackle P2, this trained and applied transfer learning-based fine-tuning on several state-of-the-art YOLO (You Only Look Once) networks. For P3, this work proposed a multi-vehicle tracking algorithm that obtains the per-lane count, classification, and speed of vehicles in real time.

Lin et al. presented a real-time traffic monitoring system based on a virtual detection zone, Gaussian mixture model (GMM), and YOLO to increase the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. In this study, the Montevideo Audio and Video Dataset (MAVD), the GARM Road-Traffic Monitoring data set (GRAM-RTM), and our collection data sets are used to verify the proposed method. Chauhan et al. used the state-of-the-art Convolutional Neural Network (CNN) based object detection models and train them for multiple vehicle classes using data from Delhi roads. This work gets upto 75% MAP on an 80-20 train-test split using 5562 video frames from four different locations. As robust network connectivity is scarce in developing regions for continuous video transmissions from the road to cloud servers, this work also evaluated the latency, energy and hardware cost of embedded implementations of our CNN model-based inferences. Arinaldi et al. presented a traffic video analysis system based on computer vision techniques. The system is designed to automatically gather important statistics for policy makers and regulators in an automated fashion. These statistics include vehicle counting, vehicle type classification, estimation of vehicle speed from video and lane usage monitoring. The core of such system is the detection and classification of vehicles in traffic videos. This work implemented two models for this purpose, first is a MoG + SVM system and the second is based on Faster RCNN, a recently popular deep learning architecture for detection of objects in images.

Gomaa et al. presented an efficient real-time approach for the detection and counting of moving vehicles based on YOLOv2 and features point motion analysis. The work is based on synchronous vehicle features detection and tracking to achieve accurate counting results. The proposed strategy works in two phases; the first one is vehicle detection and the second is the counting of moving vehicles. For initial object detection, this work has utilized state-of-the-art faster deep learning object detection algorithm YOLOv2 before refining them using K-means clustering and KLT tracker. Then an efficient approach is introduced using temporal information of the detection and tracking feature points between the framesets to assign each vehicle label with their corresponding trajectories and truly counted it. Oltean et al. proposed an approach for real time vehicle counting by using Tiny YOLO for detection and fast motion estimation for tracking. This application is running in Ubuntu with GPU processing, and the next step is to test it on low-budget devices, as Jetson Nano. Experimental results showed that this approach achieved high accuracy at real time speed (33.5 FPS) on real traffic videos.

Pico et al. proposed the implementation of a low-cost system to identify and classify vehicles using an Embedded ARM based platform (ODROID XU-4) with Ubuntu operating system. The algorithms used are based on the Open-source library (Intel OpenCV) and implemented in Python programming language. The experimentation carried out proved that the efficiency of the algorithm implemented was 95.35%, but it can be improved by increasing the training sample. Tituana et al. reviewed different previous works developed in this area and identifies the technological methods and tools used in those works; in addition, this work also presented the trends in this area. The most relevant articles were reviewed, and the results were summarized in tables and figures. Trends in the used methods are discussed in each section of the present work.

Khan et al. aimed of this work is that a cost-effective vision-based vehicle counting and classification system that is mainly implemented in OpenCV utilising Python programming and some methods of image processing. Balid et al. reported on the development and implementation of a novel smart wireless sensor for traffic monitoring. Computationally efficient and reliable algorithms for vehicle detection, speed and length estimation, classification, and time-synchronization were fully developed, integrated, and evaluated. Comprehensive system evaluation and extensive data analysis were performed to tune and validate the system for a reliable and robust operation.

Jahan et al. presented convolutional neural network for classifying four types of common vehicle in our country. Vehicle classification plays a vital role of various application such as surveillance security system, traffic control system. This work addressed these issues and fixed an aim to find a solution to reduce road accident due to traffic related cases. To classify the vehicle, this work used two methods feature extraction and classification. These two methods can straight forwardly be performed by convolutional neural network. Butt et al. proposed a convolutional neural network-based vehicle classification system to improve robustness of vehicle classification in real-time applications. This work presented a vehicle dataset comprising of 10,000 images categorized into six-common vehicle classes considering adverse illuminous conditions to achieve robustness in real-time vehicle classification systems. Initially, pretrained AlexNet, GoogleNet, Inception-v3, VGG, and

ResNet are fine-tuned on self-constructed vehicle dataset to evaluate their performance in terms of accuracy and convergence. Based on better performance, ResNet architecture is further improved by adding a new classification block in the network.

Gonzalez et al. showed a vision-based system to detect, track, count and classify moving vehicles, on any kind of road. The data acquisition system consists of a HD-RGB camera placed on the road, while the information processing is performed by clustering and classification algorithms. The system obtained an efficiency score over the 95 percent in test cases, as well, the correct classification of 85 percent of the test objects.

3. PROPOSED SYSTEM

The system could be used for detection, recognition and tracking of the vehicles in the video frames and then classify the detected vehicles according to their size in three different classes.

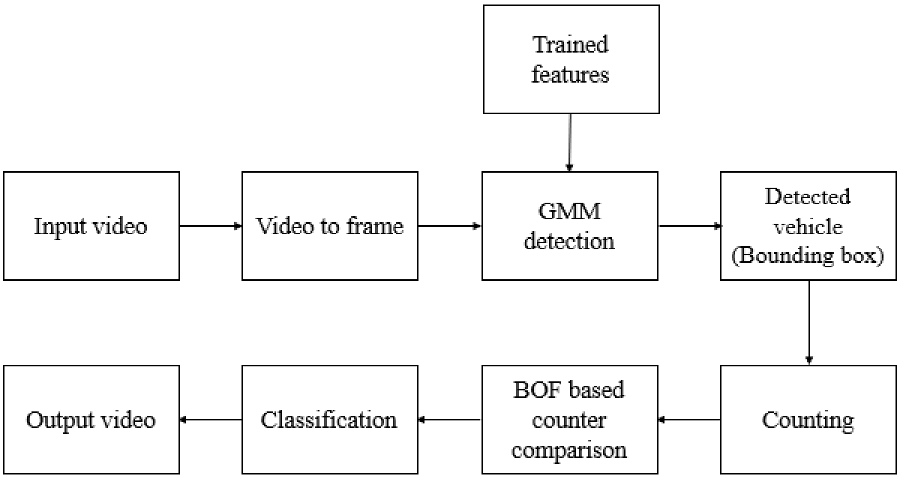


Fig. 1: Block diagram of proposed system.

The proposed system is based on three modules which are background learning, foreground extraction and vehicle classification as shown in Background subtraction is a classical approach to obtain the foreground image or in other words to detect the moving objects.

3.1 Gaussian Mixture Modelling (GMM)

At its simplest, GMM is also a type of clustering algorithm. As its name implies, each cluster is modelled according to a different Gaussian distribution. This flexible and probabilistic approach to modelling the data means that rather than having hard assignments into clusters like k-means, we have soft assignments. This means that each data point could have been generated by any of the distributions with a corresponding probability. In effect, each distribution has some ‘responsibility’ for generating a particular data point.

How can we estimate this type of model? Well, one thing we could do is to introduce a latent variable γ (gamma) for each data point. This assumes that each data point was generated by using some information about the latent variable γ. In other words, it tells us which Gaussian generated a particular data point. In practice, however, we do not observe these latent variables, so we need to estimate them. How do we do this? Well, luckily for us there is already an algorithm to use in cases like these, the Expectation Maximization (EM) Algorithm and this is what we will discuss next.

The EM algorithm: The EM algorithm consists of two steps, an E-step or Expectation step and M-step or Maximization step. Let’s say we have some latent variables γ (which are unobserved and denoted by the vector Z below) and our data points X. Our goal is to maximize the marginal likelihood of X given our parameters (denoted by the vector θ). Essentially, we can find the marginal distribution as the joint of X and Z and sum over all Z’s (sum rule of probability).

$$ln\, p(X|\Theta) = ln\left\{\Sigma_z p(X, Z|\Theta)\right\}$$

The above equation often results in a complicated function that is hard to maximize. What we can do in this case is to use Jensens Inequality to construct a lower bound function which is much easier to optimize. If we optimize this by minimizing the KL divergence (gap) between the two distributions, we can approximate the original function. This process is illustrated in Figure 1 below. I have also provided a video link above which shows a derivation of KL divergence for those of you who want a more rigorous mathematical explanation.

To estimate our model essentially, we only need to carry out two steps. In the first step (E-step) we want to estimate the posterior distribution of our latent variables γ conditional on our weights (π) means (μ)and covariance (Σ) of our Gaussians. We can then move to the second step (M-step) and use γ to maximise the likelihood with respect to our parameters θ. This process is repeated until the algorithm converges (loss function doesn't change).

Background Learning Module

This is the first module in the system whose main purpose is to learn about the background in a sense that how it is different from the foreground. Furthermore, as proposed system works on a video feed, this module extracts the frames from it and learns about the background. In a traffic scene captured with a static camera installed on the roadside, the moving objects can be considered as the foreground and static objects as the background. Image processing algorithms are used to learn about the background using the above-mentioned technique.

3.2 Vehicle Detection and Counting

The third and the last module in the proposed system is classification. After applying foreground extraction module, proper contours are acquired, Features of these contours such as centroid. Aspect ratio, area, size and solidity are extracted and are used for the classification of the vehicles. This module consists of three steps, background subtraction, image enhancement and foreground extraction. Background is subtracted so that foreground objects are visible. This is done usually by static pixels of static objects to binary 0. After background subtraction image enhancement techniques such as noise filtering, dilation and erosion are used to get proper contours of the foreground objects. The result obtained from this module is the foreground.

Region of Interest selection: In the very first frame of the video, I define a ROI by drawing a close line on the image. The goal is to recognize that ROI in a later frame, but that ROI is not a salient vehicle. It is just a part of a vehicle, and it can deform, rotate, translate and even not be fully in the frame.

Vehicle Detection: Active strategy to choose a search window for vehicle detection using an image context was proposed GMM framework to capture the vehicle by sequential actions with top-down attention. It has achieved satisfactory performance on vehicle detection benchmark, by sequentially refining the bounding boxes. Proposed a sequential search strategy to detect visual vehicles in images, where the detection model was trained by proposed a deep RL framework to select a proper action to capture a vehicle in an image.

Vehicle Counting: In this module detected vehicles will be counted and these counted results will be updated frequently based on vehicle detection, results will be printed streaming video using OpenCV.

3.3 Bag of Features model

The bag of visual features (BOF) model is one of the most important concepts in all of computer vision. We use the bag of visual words model to classify the contents of an image. It’s used to build highly scalable (not to mention, accurate) tracking systems. We even use the bag of visual words model when classifying texture via textons. As the name implies, the “bag of visual words” concept is actually taken from the “bag of words” model from the field of Information Retrieval (i.e., text-based search engines) and text analysis.

The general idea in the bag of words model is to represent “documents” (i.e., webpages, Word files, etc.) as a collection of important keypoints while totally disregarding the order the words appear in. Documents that share many the same keywords, again, regardless of the order the keywords appear in, are considered to be relevant to each other. Furthermore, since we are totally disregarding the order of the words in the document, we call this representation a “bag of words” rather than a “list of words” or “array of words”: Treating a document as a “bag of words” allows us to efficiently analyze and compare documents since we do not have to store any information regarding the order and locality of words to each other — we simply count the number of times a word appears in a document, and then use the frequency counts of each word as a method to quantify the document. In computer vision, we can apply the same concept — only now instead of working with keywords, our “words” are now image patches and their associated feature vectors:

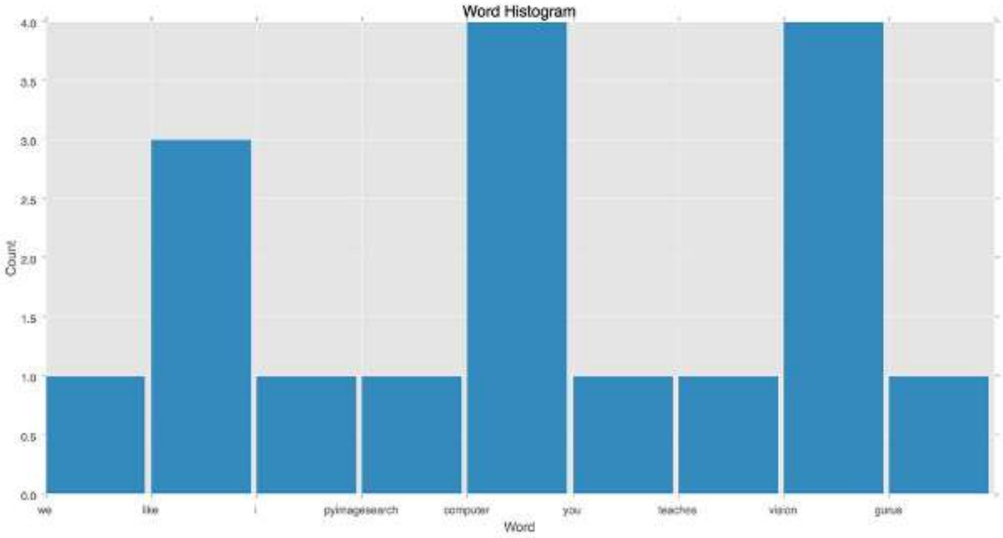


Fig. 2: An example of taking a blob of text and converting it into a word histogram.

Given our dictionary of possible visual words, we can then quantify and represent an image as a histogram which simply counts the number of times each visual word appears. This histogram is our actual bag of visual words. Building a bag of visual words can be broken down into a three-step process:

Step #1: Feature extraction

The first step in building a bag of visual words is to perform feature extraction by extracting descriptors from each image in our dataset.

Feature extraction can be accomplished in a variety of ways including: detecting keypoints and extracting SIFT features from salient regions of our images; applying a grid at regularly spaced intervals (i.e., the Dense keypoint detector) and extracting another form of local invariant descriptor; or we could even extract mean RGB values from random locations in the images.

The point here is that for each image inputted, we receive multiple feature vectors out:

Figure 5: When constructing a bag of visual words, our first step is to apply feature extraction, where we extract multiple feature vectors per image.

Step #2: Dictionary/Vocabulary construction

Now that we have extracted feature vectors from each image in our dataset, we need to construct our vocabulary of possible visual words.

Vocabulary construction is normally accomplished via the k-means clustering algorithm where we cluster the feature vectors obtained from Step #1.

The resulting cluster centers (i.e., centroids) are treated as our dictionary of visual words.

Step #3: Vector quantization

Given an arbitrary image (whether from our original dataset or not), we can quantify and abstractly represent the image using our bag of visual words model by applying the following process:

Extract feature vectors from the image in the same manner as Step #1 above.

For each extracted feature vector, compute its nearest neighbor in the dictionary created in Step #2 — this is normally accomplished using the Euclidean Distance.

Take the set of nearest neighbor labels and build a histogram of length k (the number of clusters generated from k-means), where the i‘th value in the histogram is the frequency of the i‘th visual word. This process in modeling an object by its distribution of prototype vectors is commonly called vector quantization.

3.4 Classification

One of the compelling features of our network is its simplicity: the classifier is simply replaced by a mask generation layer without any smoothness prior or convolution structure. However, it needs to be trained with a huge amount of training data: vehicles of different sizes need to occur at almost every location.

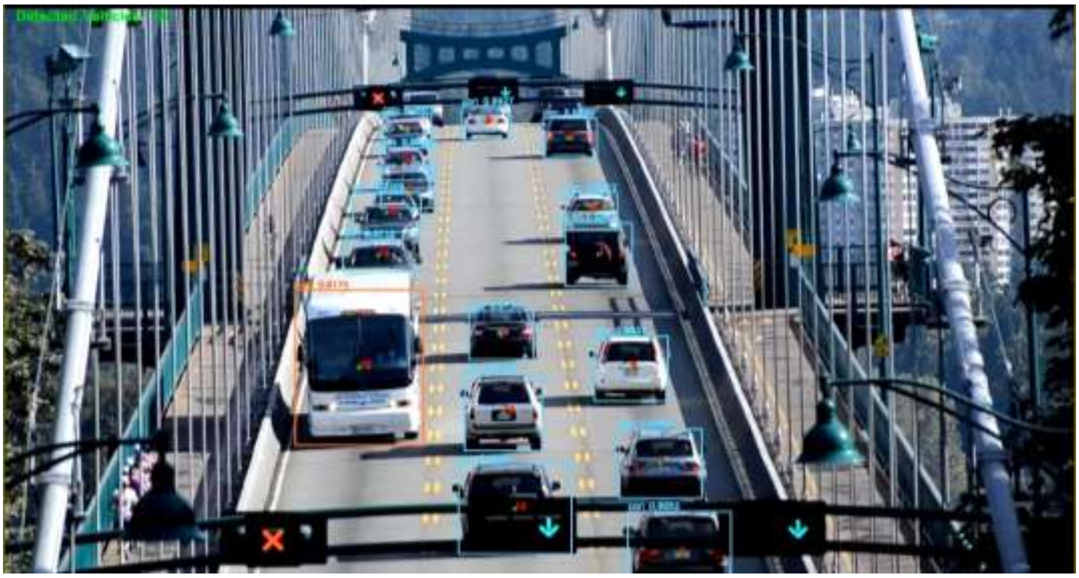
Visual tracking solves the problem of finding the position of the target in a new frame from the current position. The proposed tracker dynamically pursues the target by sequential actions controlled by the GMM. The GMM predicts the action to chase the target moving from the position in the previous frame. The bounding box is moved by the predicted action from the previous position, and then, the next action is sequentially predicted from the moved position. By repeating this process over the test sequence, we solve the vehicle tracking problem. The GMM is pretrained by SL as well as RL. During actual tracking, online adaptation is conducted.

The GMM is designed to generate actions to find the location and the size of the target vehicle in a new frame. The GMM learns the policy that selects the optimal actions to track the target from the state of its current position. In the GMM, the policy network is designed, in which the input is an image patch cropped at the position of the previous state and the output is the probability distribution of actions, including translation and scale changes. This action selecting process has fewer searching steps than sliding window or candidate sampling approaches. In addition, since our method can precisely localize the target by selecting actions, post processing, such as bounding box regression, is not necessary.

3.5 Advantages of proposed system

- Detection of multiple moving vehicles in a video sequence.
- Tracking of the detected vehicles.
- Identification of Vehicle types.
- Counting the total number of vehicles passing in videos.

4. RESULTS AND DISCUSSION



5.CONCLUSION

This project has the purpose of a vision-based vehicle counting and classification system. The system involved capturing of frames from the video to perform background subtraction in order detect and count the vehicles using Gaussian Mixture Model (GMM) background subtraction then it classifies the vehicles by comparing the contour areas to the assumed values. The substantial contribution of the work is the comparison of two classification methods. Classification has been implemented using Contour Comparison (CC) as well as Bag of Features (BoF) method.

5.1 Future scope

One of the limitations of the system is that it is not efficient at detection of occlusion of the vehicles which affects the accuracy of the counting as well as classification. This problem could be solved by introducing the second level feature classification such as the classification on the bases of color. Another limitation of the current system is that it needs human supervision for defining the region of interest. The user must define an imaginary line where centroid of the contours intersects for the counting of vehicles hence the accuracy is dependent on the judgment of the human supervisor. Furthermore, the camera angle also affects the system hence camera calibration techniques could be used for the detection of the lane for the better view of the road and increasing the efficiency. The system is not capable of detection of vehicles in the night as it needs the foreground objects to be visible for extraction of contour properties as well as features for the classification using SIFT features. The system could also be improved for better accuracy using the more sophisticated image segmentation and artificial intelligence operations.

REFERENCES

[1] Alpatov, Boris & Babayan, Pavel & Ershov, Maksim. (2018). Vehicle detection and counting system for real-time traffic surveillance. 1-4. 10.1109/MECO.2018.8406017.

[2] Song, H., Liang, H., Li, H. et al. Vision-based vehicle detection and counting system using deep learning in highway scenes. Eur. Transp. Res. Rev. 11, 51 (2019). https://doi.org/10.1186/s12544-019-0390-4.

[3] Neupane, Bipul et al. “Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network.” Sensors (Basel, Switzerland) vol. 22,10 3813. 18 May. 2022, doi:10.3390/s22103813.

[4] C. J Lin, Shiou-Yun Jeng, Hong-Wei Lioa, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO", Mathematical Problems in Engineering, vol. 2021, Article ID 1577614, 10 pages, 2021. https://doi.org/10.1155/2021/1577614.

[5] M. S. Chauhan, A. Singh, M. Khemka, A. Prateek, and Rijurekha Sen. 2019. Embedded CNN based vehicle classification and counting in non-laned road traffic. In Proceedings of the Tenth International Conference on Information and Communication Technologies and Development (ICTD '19). Association for Computing Machinery, New York, NY, USA, Article 5, 1–11. https://doi.org/10.1145/3287098.3287118.

[6] A. Arinaldi, J. A. Pradana, A. A. Gurusinga, “Detection and classification of vehicles for traffic video analytics”, Procedia Computer Science, Volume 144, 2018, Pages 259-268, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.10.527.

[7] Gomaa, A., Minematsu, T., Abdelwahab, M.M. et al. Faster CNN-based vehicle detection and counting strategy for fixed camera scenes. Multimed Tools Appl 81, 25443–25471 (2022). https://doi.org/10.1007/s11042-022-12370-9.

[8] G. Oltean, C. Florea, R. Orghidan and V. Oltean, "Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation," 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), 2019, pp. 240-243, doi: 10.1109/SIITME47687.2019.8990708.

[9] L. C. Pico and D. S. Benítez, "A Low-Cost Real-Time Embedded Vehicle Counting and Classification System for Traffic Management Applications," 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), 2018, pp. 1-6, doi: 10.1109/ColComCon.2018.8466734.

[10] D. E. V. Tituana, S. G. Yoo and R. O. Andrade, "Vehicle Counting using Computer Vision: A Survey," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), 2022, pp. 1-8, doi: 10.1109/I2CT54291.2022.9824432.

[11] A. Khan, A., Sabeenian, R.S., Janani, A.S., Akash, P. (2022). Vehicle Classification and Counting from Surveillance Camera Using Computer Vision. In: Suma, V., Baig, Z., K.

Shanmugam, S., Lorenz, P. (eds) Inventive Systems and Control. Lecture Notes in Networks and Systems, vol 436. Springer, Singapore. https://doi.org/10.1007/978-981-19-1012-8_31.

[12] W. Balid, H. Tafish and H. H. Refai, "Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 6, pp. 1784-1794, June 2018, doi: 10.1109/TITS.2017.2741507.

[13] N. Jahan, S. Islam and M. F. A. Foysal, "Real-Time Vehicle Classification Using CNN," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225623.

[14] M. A. Butt, A. M. Khattak, S. Shafique, B. Hayat, S. Abid, Ki-Il Kim, M. W. Ayub, A. Sajid, A. Adnan, "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems", Complexity, vol. 2021, Article ID 6644861, 11 pages, 2021. https://doi.org/10.1155/2021/6644861.

[15] P. Gonzalez, Raul & Nuño-Maganda, Marco Aurelio. (2014). Computer vision based real-time vehicle tracking and classification system. Midwest Symposium on Circuits and Systems. 679-682. 10.1109/MWSCAS.2014.6908506