Chapter 6

# Vehicle Detection by CCTV

Vehicle detection is the essential technology to monitor traffic statuses such as traffic volume, travel speed, and traffic density in a city. Traditionally, a Loop-coil detector or ultrasonic detector has been used as the vehicle detector to monitor traffic states. However, the cost of installing such a vehicle detector is costly.

So, vehicle detection by a Closed-circuit television camera (CCTV) is one of the practical approaches. The performance of the camera has improved, and even inexpensive cameras can now shoot with high resolution. Also, identifying a vehicle from the video image became reliable because of the recent development of image processing technology. Besides, the image data compression technology and transfer technology have improved dramatically. Due to the above technological innovations, CCTV vehicle detection technology has become a very inexpensive and deployable technology.

This technology has already been introduced in many developed cities, but some problems could be solved when introduced in developing cities. One is how to detect mixed traffic, including small type vehicles such as an auto-rickshaw, a motorcycle, etc. Since those vehicles, it is challenging to observe each of these vehicles' conditions, as they often run as a group without following the lane. In developing countries, the traffic is highly heterogeneous and often densely populated. With the help of CCTV feed from Ahmedabad, the vehicle detection dataset with classes like motorcycle and auto-rickshaw, in addition to the classes like car, bus, truck, etc were annotated as part of the project. This collection and annotation of data specific to Indian roads will help boost the performance of the vehicle detection model because of the similarity of distributions of training data and testing data (data from Ahmedabad).

## Introduction

### 1: Case Study Field

The project's case study field is the west side of Ahmedabad city of Gujarat state in India, where it is one of the cities rapidly economic grow and heavy traffic congestion by transportation demand.

### 2: CCTV

Closed-circuit television (CCTV), also known as video surveillance, uses video monitoring cameras to transmit a signal to a specific place on a limited set of monitors. It differs from broadcast television in that the traffic signal is not openly transmitted, though it may employ point-to-point (P2P), point-to-multipoint (P2MP), or mesh wired, wireless links. Though almost all video cameras fit this definition, the term is most often applied to those used for surveillance in areas that may need monitoring, such as banks, stores, and other areas where security is needed. In this program, we installed a 360-degree direction local controlled movement with a 4K resolution camera as shown in Figure 6-1.

### 3: Vehicle Detection

In this section, programs were developed to draw reference lines and detect vehicles in each frame. The programs allow users to draw reference lines in the first frame of a video file, programs were developed to provide threshold values to sort out potential detected objects.

The outputs of a processed image are rectangular bounding boxes and the score of those boxes. The score means a confidence level for a detected object, i.e., how confident YOLO is that the box contains a valid object. This value lies in the range of 0 to 1. The higher the score value, the higher the confidence that an object is indeed an object of interest. The bounding boxes are processed using a non-maximal suppression method in YOLO. The controlling factor in this process is a threshold value (a factor to screen out bounding boxes with low confidence levels), which means that bounding boxes having a score value higher than threshold are accepted for further processing. Those that have values below threshold are

**Figure 6-1** Monitoring Camera, 360-degrees 4K Camera (photo by Zero Sum Ltd.)

eliminated. The sorted bounding boxes are sent to the tracker program to track detected vehicles.

### 4: Vehicle Tracking

In this step, the sorted bounding boxes are tracked frame by frame. For the tracking process, Kalman Box Tracker was used. In this research, programs were developed to draw a centerline on each bounding box. Since a video file consists of thousands of frames, an object must be tracked from one frame to another to determine its direction of movement. This tracker compares the current frame with the immediately previous frame using the pixel variance of the frame. When it finds a similarity in pixel values, it updates the object (i.e., bounding box) and memorizes it for consideration in the next frame. Then, it compares the updated frame to the next frame. The tracker titles each tracked bounding box by a numeric value such as 1, 2, 3, etc. In this way, the tracker tracks an object from frame to frame.

Rectangles are drawn around each tracked object on the computer screen. The tracked rectangles are used for vehicle counting.

### 5: Vehicle Counting

Vehicle counting plays a significant role in vehicle behavior analysis and traffic density detection for better traffic optimization. Accurate estimation of the number of vehicles on the road is an important endeavor in the intelligent transportation system (ITS). An effective measure of on-road vehicles can have a plethora of applications in transportation sciences including traffic management, signal control and on-street parking.

In the earlier days before the rise of machine learning, the process of vehicle counting was done manually. It was performed by a person standing by the roadside; using an electronic device to record the data using a tally sheet. In some cases, the person may do the counting by observing video footage captured by city cams or closed-circuit television (CCTV) placed above the road or highway. Although the manual method provides high accuracy, it requires an extensive amount of human resources. Besides, it tends to be error-prone, especially on severe traffic flow and multiple road lines. Therefore, manual calculations are usually performed with only a small sample of data, and the results are extrapolated for the whole year or season for long-term forecasts.

Existing sensor methods and the traditional image processing method have the problems of difficulty in installation, high cost, and low precision, which resulted in serious road damage, expensive information construction, and poor vehicle counting accuracy. Therefore, it is of great theoretical and practical significance to make full use of existing monitoring resources and apply the methods of deep learning and computer vision to study the video-based vehicle counting method for traffic monitoring and traffic optimization.

The vehicle counting system is made up of three main components: a detector, tracker and counter. The detector identifies vehicles in a given frame of video and returns a list of bounding boxes around the vehicles to the tracker. The tracker uses the bounding boxes to track the vehicles in subsequent frames. The detector is also used to update the trackers periodically to ensure that they are still tracking the vehicles correctly. The counter counts vehicles when they leave the frame or makes use of a counting line drawn across a road.

The counting method is based on the vehicle regional bounding box marks and the virtual reference line. This technique assumes that the vehicle movement is in a direction. For counting, each detected vehicle in the detection step is assigned with a unique label and tracked until it reaches the virtual line. In this work, we have used five different class labels, namely auto, car, motorcycle, bus and truck. And all these labels are categorized as vehicle objects and will be used in the counting system. After that, each vehicle position is checked whether it has crossed the horizontal reference line (red line) at the y-axis as drawn in figure. If it passes the line, then it will be counted as one.

## Experimental Results

Figure 6-2 is the output on the Paldi junction in Ahmedabad for an interval of 15 sec during the night conditions, red line is the reference line and vehicles are counted from both the directions and the value is updated at the left topmost corner.

Figure 6-3 is the output on the IIT Hyderabad main gate for an interval of 15 sec, the two red lines are the reference lines at both the directions and vehicles are counted from multiple directions and the value is updated at the left topmost corner.

Figure 6-4 shows the vehicle distribution for 20 min at Paldi junction during the peak interval of the day and also the pie chart explains the vehicle type distribution during that interval. Motorbikes contribute to the highest occupancy of 46% on the road followed by Auto which contribute 29% of the road traffic during the peak interval.

## Future Issue

Based on the results of this chapter, we are considering the deployment of our system to other cities as a future task. Secondly, it is necessary to deal with the night-time problem of systems using vehicle detection by image sensors. In the following, we first explain the possible problems in applying the method described in this chapter to other cities. Next, we present two methods that are generally considered to be effective in addressing this problem, based

**Figure 6-2** Vehicle counting on Paldi junction data (Ahmedabad) for reference line


**Figure 6-3** Vehicle counting on IITH data for multiple directions

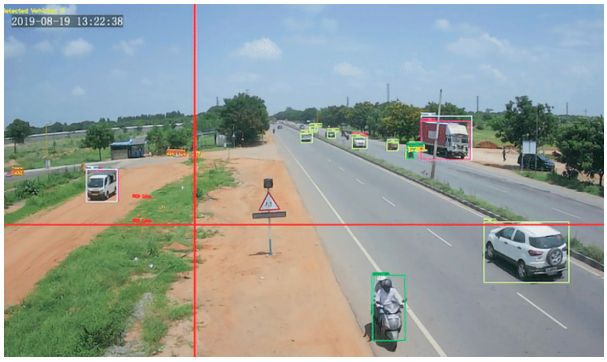
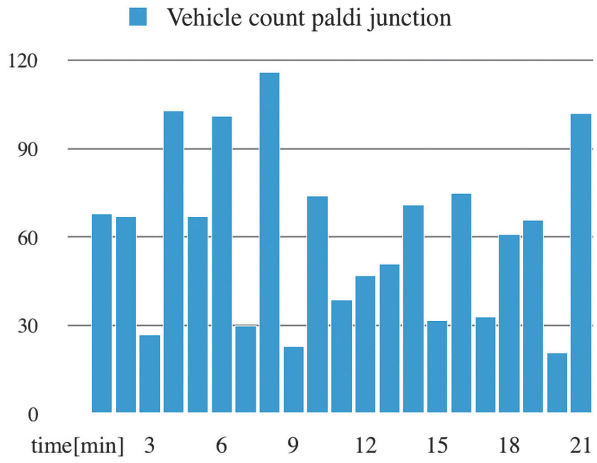



**Figure 6-4** Vehicle count distribution with multiple vehicle types at Paldi junction




on simple experimental results. Furthermore, we present the results of using ToF sensors as one of the solutions to the problem of vehicle detection at night by image sensors. Expansion to other cities

The performance of CNN is highly dependent on the quality and quantity of training data. Because of the specific appearance of the target environment, it is more important to prepare data that matches the target than to prepare general large-scale data. However, the cost of preparing enough training data is a major problem. There are two techniques, called Transfer Learning and Data augmentation, they may be useful in extending the results of this case study to other cities. These are approaches that can be applied to a variety of methods without major changes to the framework.

A simple experiment will show the effectiveness of these techniques.

## 1: Transfer Learning

In order to achieve the best performance in traffic image analysis and traffic measurement, it is useful to have a sufficient amount of training data in the target environment, but the cost of this task is prohibitive. To reduce this cost, we have tested the effectiveness of transfer learning for training a CNN.

Transfer learning is a method of efficiently and effectively training a model to fit a new task or data (domain) by reusing the results of training on another related task or data (domain). Typical transfer learning in DNN has two main approaches: one is to retrain the parameters of the whole model (Fine Tuning), and the other is to learn only the parameters of the output layer, which is newly prepared for a new task or data.

In this verification, we adopt M2Det [1], which has been proposed as a robust object detection model for small objects. M2Det consists of a backbone network, a multi-level feature pyramid network (MLFPN) and a prediction layer. M2Det is characterized by a multi-level feature pyramid (hierarchical structure of feature maps), which consists of multi-scale and multi-level feature maps.

By training M2Det on large-scale data, backbone network and MLFPN of M2Det, are expected to acquire feature extraction methods that are generally effective for object detection, including vehicle detection. Therefore, we first train M2Det on large scale data for generic object detection. Then, a vehicle detection model tuned to the target junction in India is built by transfer learning, where only the prediction layer, which is the output layer of M2Det, is trained using a small-scale data set of the target junction.

The M2Det model, which is trained on the COCO dataset [2] for generic object detection consisting of 80 classes, is transfer learned under two different conditions. One condition is trained using a large-scale dataset, the India driving dataset (IDD), and the other condition is trained using

**Table 6-1** Details of each data set

| | IDD | Paldi |
|---|---|---|
| Car | 54,258 | 328 |
| Motorbike | 63,121 | 556 |
| Auto | 32,280 | 748 |
| Bus | 9,723 | 156 |
| Truck | 15,305 | 40 |

our own small-scale dataset (Paldi). IDD [3] is a vehicle detection dataset provided by Indian Institute of Information Technology, Hyderabad. The Paldi dataset was annotated by us using 100 images from the video of the target Paldi junction. Table 6-1 shows the number of vehicles in each dataset.

The result of the vehicle counts for each training condition on a 30-minute video of the Paldi junction are shown in the Table 6-2. The out results are the vehicle count results output by the system. The correct results are the out results minus the incorrect counting.

The results suggest the effectiveness of using small data sets created in the target environment rather than large-scale data sets.

## 2: Data Augmentation

Data Augmentation is one of effective methods to increase object detection performance. It is almost impossible to manually collect enough numbers of image data to achieve object recognition engine, because the data scale often could be several thousand or more.

In outdoor object recognition like vehicle and pedestrian recognition, the “appearance” changes remarkably depending on the sunshine conditions, in addition to the movement of the object itself. It is costly and not realistic to learn all vehicle data corresponding to this change. For the issue, the data augmentation is applied. In it, artificially adding deformation according to the expected change to the seed image and inflating the training data is used as a seed of data acquired very coarsely for the change. We focused Yolo V3 [4], well known object detection and classification method, and apply it to detect auto-rickshaw with using the data augmentation. We compare the detection accuracy (recall and false positive rate) between learning auto-rickshaw with very few 250 samples and learning it by applying data augmentation, where rotated and brightness changed data is artificially generated up to 17,800 samples. Figure 6-5 show an example of the augmentation for the rickshaw image. Table 6-3 shows the

**Table 6-2** Comparison of measurement accuracy under different learning conditions

| | IDD | | | | | | Paldi | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Car | Motor-bike | Auto | Bus | Truck | Total | Car | Motor-bike | Auto | Bus | Truck | Total |
| Ground truth (A) | 320 | 976 | 575 | 90 | 26 | 1987 | 320 | 976 | 575 | 90 | 26 | 1987 |
| Out results (B) | 352 | 1321 | 405 | 71 | 59 | 2208 | 319 | 980 | 490 | 68 | 20 | 1877 |
| Correct results (C) | 286 | 877 | 369 | 59 | 18 | 1609 | 291 | 845 | 438 | 66 | 16 | 1656 |
| Recall [%] (C/A) | 89.4 | 89.9 | 64.2 | 65.6 | 69.2 | 81.0 | 90.9 | 86.6 | 76.2 | 73.3 | 61.5 | 83.3 |
| Precision [%] (C/B) | 81.3 | 66.4 | 91.1 | 83.1 | 30.5 | 72.9 | 91.2 | 86.2 | 89.4 | 97.1 | 80.0 | 88.2 |

**Figure 6-5** Data Augmentation Examples for the Rickshaw Image

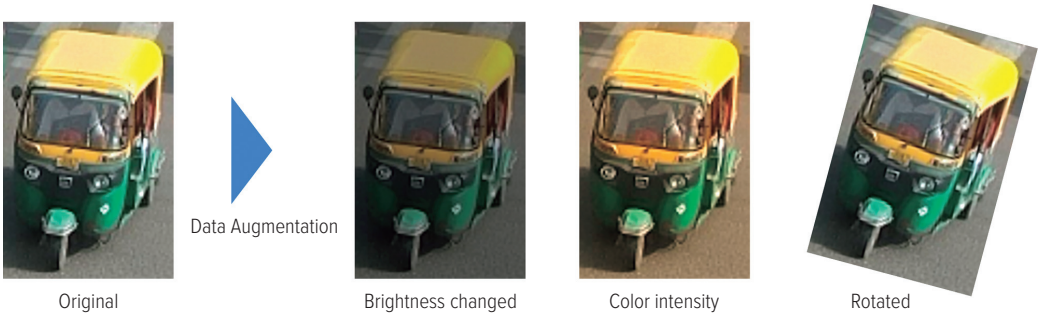



**Table 6-3** Comparison of Auto-Rickshaw Detection Performance

| | Recall Ratio (True Positive/Correct Ans.) | Error Ratio (Err./Detected) |
|---|---|---|
| ① Small Data (250) | 46% (38/82) | 2.5% (1/39) |
| ② Augmentation (17,800) | 70% (58/82) | 1.6% (1/59) |

**Figure 6-6** Example of Improvement for Auto-Rickshaw Detection


① Small Data ② Augmentation

**Figure 6-7** A New Detected Rickshaw Examples

result. As expected, the recall ratio increased from 46% to 70%, and the false positive rate decreased from 2.5% to 1.6% simultaneously. In this experiment, number of correct auto-rickshaw, should be detected, is 82. And, detected number for small data and augmented data are 38 and 58, respectively. At the same time, false detect number for both is 1, during detect number for small data and augmented data are 39 and 59, respectively.

Figure 6-6 shows the comparison of auto-rickshaw detection. In the Right Result Augmentation is used, and the bottom-left rickshaw is correctly detected. Figure 6-7 shows anew detected rickshaw examples with the augmentation. These rickshaws cannot be detected in small data set.

## 3: Adaptation to night lighting environment

Image-type vehicle detectors are affected by ambient light, e.g. headlights at night. We believe that an effective solution to this problem is the supplementary use of laser ranging image sensors (ToF sensors), which are not affected by lighting.

A Time-of-flight (ToF) sensor irradiates near-infrared pulse and measures the distance from the time it takes for the laser reflected from the object. Since the near-infrared pulse uses a coaxial optical system that irradiates and receives light at points, it is resistant to the effects of ambient light and can measure distance values day and night. The 3D sensor has a disturbance resistance of 200,000 lux or more for about 130,000 lux of direct sunlight in midsummer, and can be used outdoors at an operating temperature of −10℃ to +50℃.

Figure 6-8 shows the installation position and detection range assumed in this study. The installation position was on the shoulder, and the shooting direction was an incident angle of 90 degrees with respect to the vehicle running in the lane. The maximum detection range is 15 m. Calculations show that if the distance from the sensor to the vehicle is at least 2.1 m, it is possible to detect vehicles up to 70 km/h.

The background subtraction method of image processing can be used for vehicle detection. Figure 6-9 shows a captured image of the ToF sensor and an image of background subtraction. The distance information to the sensor is expressed in color

**Figure 6-8** Sensor Installation and Detection Range

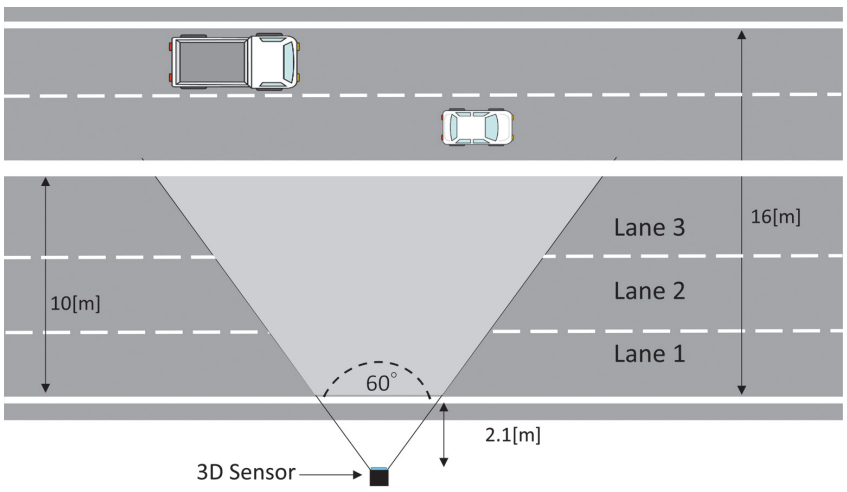


**Figure 6-9** Image of ToF Sensor and Background Subtraction Method

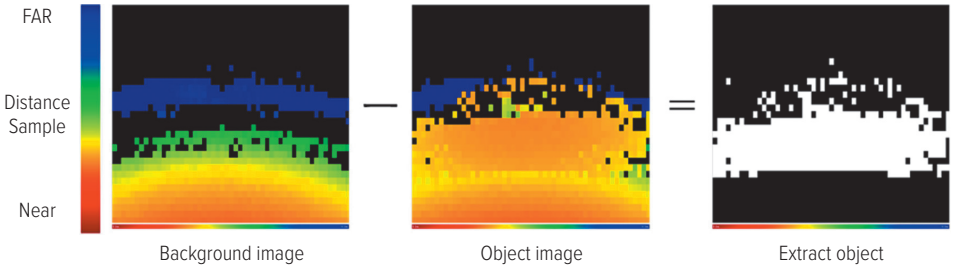
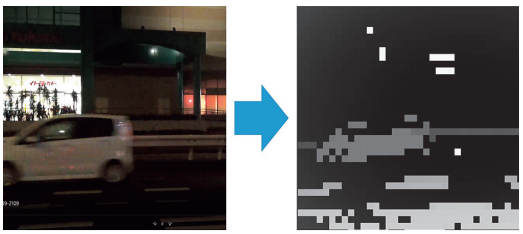



**Figure 6-10** Detection Result Sample of Lane 3




for the same two-dimensional image as the camera image. The closer the distance, the redder the color, and the farther the distance, the bluer the color. From the background image on the left, the moving object can be extracted as shown in the image on the right by taking the subtraction from the image containing the moving object in the middle. However, due to the characteristics of the sensor, the movement of the background is taken as noise, so we devised ways to eliminate noise, such as preparing multiple back-ground images. Here, the extracted moving object is to be determined whether a vehicle or not. The width of the moving object is calculated as the vehicle length, and those below the threshold are removed. We also check the ground contact conditions at the bottom of the moving object, because the tires are contacted with ground absolutely. A moving object is detected as a vehicle only when the vehicle length judgment and the ground contact condition are satisfied. The vehicle judgment is performed only in the first lane. And the ground contact condition cannot be confirmed in the second and third lane due to the distance is far. Therefore, it is deter-mined whether or not a vehicle based on the size of the moving object.

The vehicle detection performance at night (after 21:00) was tested on a one-way three-lane road in Japan. The results of the experiment showed that the detection accuracy of the ToF sensor was 89% on average. An example of detection in lane 3 is shown in Figure 6-10. In particular, the accuracy was 100% in the traffic measurement in the front lane, confirming the effectiveness of the sensors as vehicle detectors. The results show that the 3D sensor is a promising complement to the image sensor, which has a problem of accuracy degradation due to headlights.

## References

[1] Zhao, Q., T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling (2019) M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network, The Thirty-Third AAAI Conf. on Artificial Intelligence, pp. 9259–9266.
[2] Lin, Y.T., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár: Microsoft COCO: Common Objects in Context, In ECCV 2014, pp.740–755, 2014.
[3] Varma, G., A. Subramanian, A. Namboodiri, M. Chandraker, C. V. Jawahar: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments, IEEE Winter Conference. on Applications of Computer Vision, pp. 1743–1751, 2018.
[4] Redmon, J., and A., Farhadi; “YOLOv3: An Incremental Improvement”, arXiv:1804.02767, 2018.