Vehicle Detection in Traffic Monitoring with Machine Learning

Based on Big Data and Cloud

[1]Patel Parin,[2]Gayatry Pandi (Jain)
[1]Research Scholar,[2]Head of Post-Graduation Department
[1,2]Computer Engineering,
[1]L.J Institute of Engineering & Technology, Ahmedabad, Gujarat,India

Abstract:Traffic Monitoring is a challenging task on crowded roads. Traffic Monitoring procedures are manual, expensive, time consuming and involve human operators. Large-scale storage and analysis of video streams were not possible due to limited availability. However, it is now possible to implement object detection and tracking, behavioral analysis of traffic patterns, number plate recognition and surveillance on video streams produced by traffic monitoring. In Big data, video datasets are so large that typical database systems are not able to store and analyses the datasets. Storing and Processing big volumes of data requires Scalability, Fault Tolerance and Availability. Thus, Big Data and Cloud computing are two compatible concepts as cloud enables big data for traffic monitoring. In this paper, we proposed vehicle detection for traffic monitoring. We have applied Support Vector Machine (SVM) machine learning algorithm for detecting vehicles.

Index Terms - Machine Learning, Big Data Analytics, Traffic Video Monitoring, Video Analysis, Vehicle Detection

I. INTRODUCTION

Traffic jams, congestion, and accidents on city roads is a common problem in most major cities across the world. Traffic monitoring systems have the capability to capture and transmit number of vehicles that pass through an intersection as a function of time intervals, and average speed of vehicles. Most cities have digital video cameras installed in hundreds of locations primarily for monitoring crime and terrorist activities. They generate video data per day at the scale of terabytes. Issues involved include efficient and secure transmission and storage, processing and feature extraction, storage and retrieval of features, and performing analytics on feature data. Analytics reveal traffic patterns keyed to geographic location and time intervals, congestion and accident reports.

Traffic Monitoring can capture, process, store, analyze, and retrieve video data at scale. Also detects and tracks individual vehicles in the video frames and computes total number of vehicles that have passed through an intersection over a time interval. It also computes the speed of individual vehicles and average speed of vehicles. In real-time additional functionality of the system includes suggesting alternative routes to commuters when congestion spotted on roadways.

Fig 1. Block Diagram of Traffic Monitoring System [4]

Traffic Monitoring System used for video analysis process as shown in the Fig 1. After splitting the video into chunks, vehicles are detected from the chunks. Vehicle detection is not a trivial task and is performed in two stages. [8]

- First, **Haar classifiers** are used for pre-detecting vehicles in the video frames. This is a pre-processing step.
- Second, **Support Vector Machine (SVM**) is use to accurately detect the presence of vehicles. This is also referred as post-processing step.

Purpose of this research is to build a robust and high throughput cloud computing based solution for automatic analysis of video streams coming from traffic monitoring cameras and recorded in a cloud based storage. The term video analytics refers to the processing and analysis of video streams using computing resources [3].

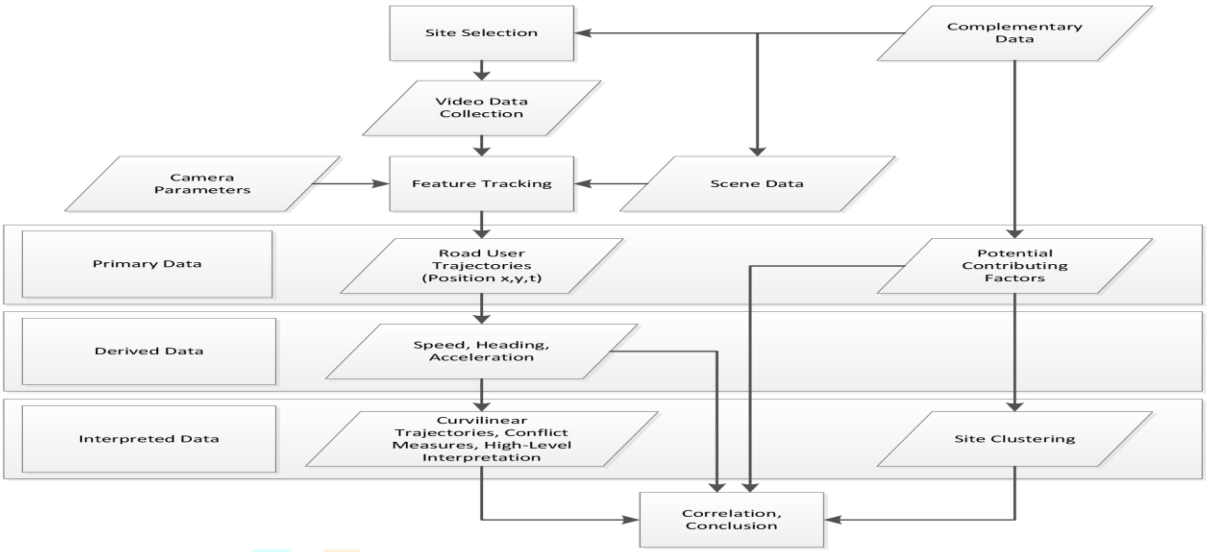


Fig 2. Flow Diagram of Video Data Processing [3]

Traffic video analysis aims at using a number of techniques to achieve better traffic and road safety, control congestion and provide immediate care for accident victims. The system alerts nearby hospitals and highway rescue teams when accidents occur. It also detects road congestion and broadcasts alternative route information to relevant computers. Flow Diagram of video data processing is shown in the Fig 2. Here data is divided mainly 3 parts Primary Data, Derived Data and Interpreted Data [3].
Primary Data is collection of video data with feature tracking and camera parameters. Using Primary Data we can analyze road user trajectories with position x, y, z. Derived Data is derived from Primary Data, Using this data we can estimate speed and acceleration of the vehicles. Interpreted Data is use for clustering and correlation of the Derived Data. Therefore, we follow this process for video processing [3].

II. MACHINE LEARNING APPLIED TO BIG DATA

This Machine Learning process is useful in Big Data for Data Analysis with deciding steps. Here, all steps are given by Rubens Zimbres and applied for Descriptive, Predictive and Prescriptive analysis. Predictive analysis is useful in training dataset for real-time data and pattern recognition with particular simulation [9].

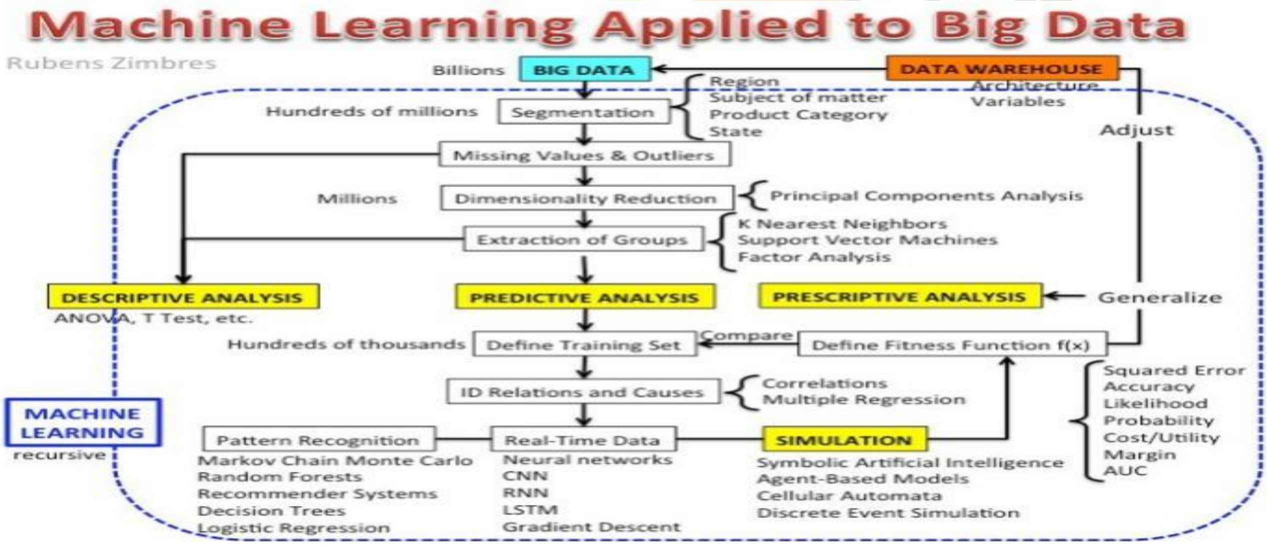


Fig 3. Machine Learning Applied to Big Data [9]

- Descriptive analysis is used to describe the basic features of the data and arranged tables and graphs.
- Predictive analysis is the practice of extracting information from existing data in order to determine patterns and predict future outcomes and trends. Predictive analysis does not tell you what will happen in the future.
- Prescriptive analysis is combination of Descriptive and Predictive analysis. Prescriptive analysis extended beyond Predictive analysis by specifying both actions necessary to achieve predicated outcomes, and the interrelated effects of each decision.

Here, Shown in the Fig. 3 All possible machine learning tasks are here we can use any one for our data.

III. VEHICLE DETECTION USING TRAFFIC MONITORING

In Traffic Monitoring, Vehicle detection is major part of traffic monitoring it is very useful for traffic management and use for finding stolen vehicles and crime related vehicle detect by this process [4].
Traffic Surveillance video metadata can be represented by a collection of vehicle metadata, we design the metadata structure (See Table 1). It contains many basic attributes of vehicle. (e.g. location, license plate number, vehicle type) . Other attributes, such as velocity, direction and duration can be inferred from these basic attributes, and thus are omitted in this structure [4].
This type of data we can analyses using machine learning and Big Data:

Table 1: The Vehicle Metadata Structure [4]

Attribute Name	Type	Description
Location	String	The geohash of camera coordinate
Trajectory	pixel location[]	Recording object's position per second in one camera's view
Vehicle Type	Short	4 types are pre-defined: car, bus, minibus, truck
Color	Short	15 kinds of colors are predefined
License Plate Number	String	Vehicle's license plate number
Vehicle Image	byte[]	The image of this vehicle
License Plate Image	byte[]	The image of vehicle license plate

Therefore, all details given in the table are required for detecting the vehicle. We can analysis using machine-learning algorithm through some decided steps given below. Vehicle Detection is performed a steps specified in the Fig 4.

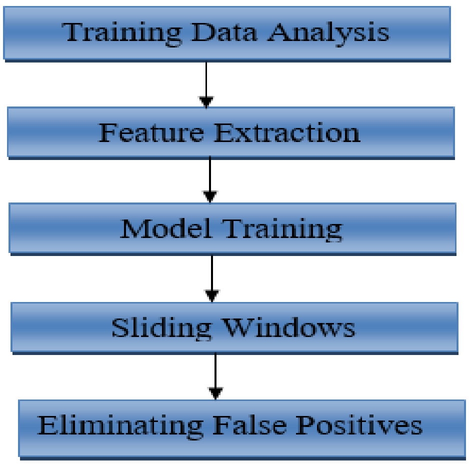


Fig 4. Vehicle Detection Process [10]

Training Data Analysis: Different images of vehicles from different angles and non-vehicle image [10].
Feature Extraction: Detect vehicle on image. We identify features, which uniquely represent a vehicle. For increasing accuracy rate, we will be using Histogram of Gradient (HOG) [10].

- The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.
- Scikit-image python library provides us with the necessary API (Application Programming Interface) for calculating HOG feature.

Model Training: We will using Linear Support Vector Machine (LSVM). It is supervised learning model which will be able to classify whether something is vehicle or not after we train it [10].
Sliding Window: Prediction model will be applied in a special technique called Sliding Window. Increase the robustness of this approach we will adding multiple grids. Which will be traversed by the prediction model. We are doing this since vehicles can appear on the image in various sizes depending on its location on the road [10].

- Window Sizes from 128 X 128 for area closer to the vehicle and 64 X 64 for area further away from the vehicle. Windows overlap is set to 65%.

Eliminative False Positive: Improve the accuracy of the final output and trying for multiple vehicles. This approach is equivalent to creating a heat map [10].

IV. PROCEDURAL WORK

Procedural work is used in OpenCV using video for Traffic Flow Vehicle Detection for Detecting vehicle image with color, speed, Direction and size of vehicle shown in Fig 5.When we use a video in OpenCV Python that is divided video in to some frames and after that TensorFlow Object Detection API apply on this frames. The TensorFlow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. There are three modules: [5]

Color Recognition Module: This module is detected vehicle using color recognition by KNN trained with color histogram for identify vehicle color in the frame.

Speed and Direction Prediction Module: This module is detected vehicle image pixel locations for prediction the speed and direction of the vehicle by pixel locations.

Size Prediction Module: This module detected vehicle image with prediction vehicle size by image area.

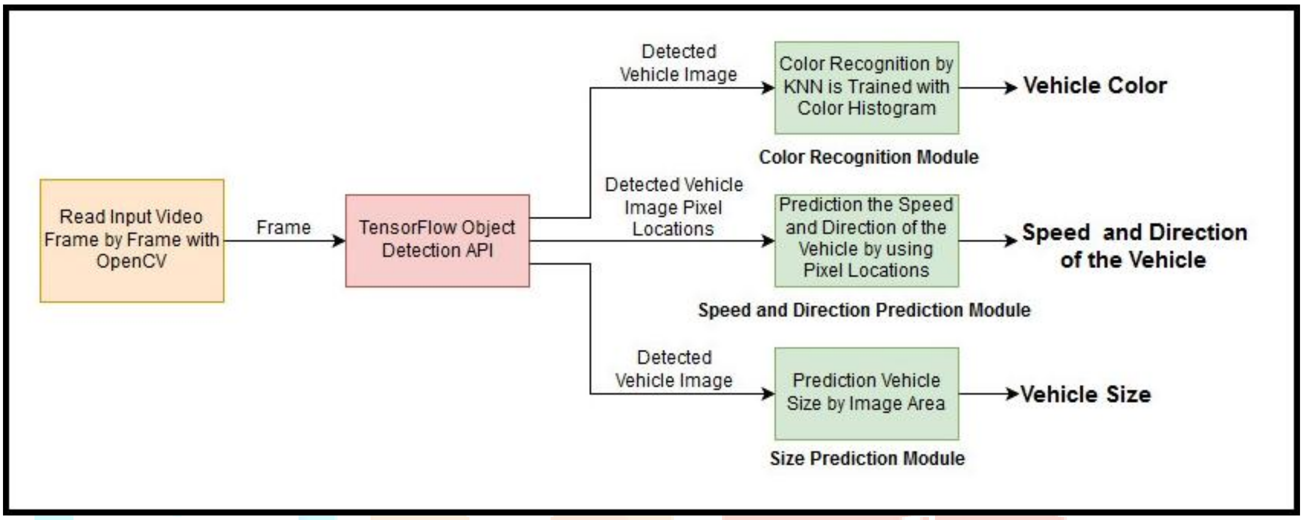


Fig 5. Procedural Work for Experiments [5]

All steps are using with SVM (Support Vector Machine) Algorithm:

```
feature params=
{
'color model' : 'yuv',                    # hls, hsv, yuv, ycrcb
'bounding box size':64,          # 64 pixels x 64 pixel image
'number of orientations': 11,    # 6 – 12
'pixels per cell': 16,                    # 8, 16
'cells per block': 2,                     # 1, 2
'do transform sqrt': True
}
src = FeatureSourcer(feature params, temo frame)
cls = BinaryClassifier(svc, scaler)
slider = Slider(source = src, classifier = cls, increment = 8)
this heatmap = HeatMap(frame = temp frame, thresh = 30, memory = 40)

window sizes = 80, 120, 150, 180
strip positions = 410, 390, 380, 380

def pipeline(this frame):

for sz, pos in zip(window_sizes, strip_positions):
        bounding boxes = slider.locate(frame = this_frame, window size = sz,
                window position = pos)
        this_heatmap.update(bounding_boxes)

heatmap, threshold_map, labeled_map = this_heatmap.get()
labeled_frame = this_heatmap.draw(thus_frame)

return labeled frame
```

This Algorithm is use for all feature parameters like color model, bounding box size, binary classifier, heat map. We can change parameter as per our analysis. In addition, we change sliding window size as per our vehicle classification and strip position is use for increasing accuracy in the frame. Heat map is use for creating bounding boxes for locating detected vehicles. End of the coding we have a labeled frame as a output.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Steps of Implementation:

Step 1: Creating frames from video streams.

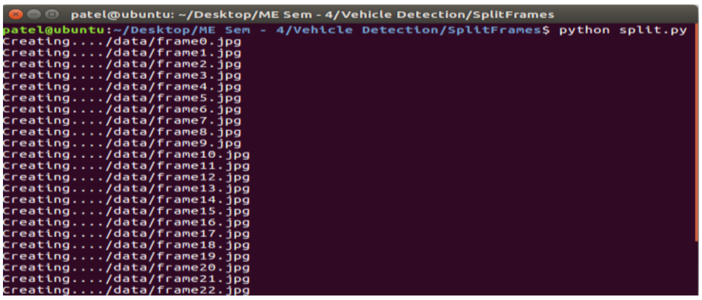


Fig 6. Creating Frames

Fig 7. Frames

Different frames are divided from video stream shown in the Fig 7. And that execution process is shown in Fig 6.

Step 2: Apply a color transform and append binned color feature as well as histogram of color to the HOG feature vector.

This all color model is useful in HOG feature vector:

RGB (red, green, and blue) refers to a system for representing the colors to be used on a computer display. Red, green, and blue can be combined in various proportions to obtain any color in the visible spectrum [9].

HSL (Hue, Saturation, and Lightness) and HSV (Hue, Saturation, Value) are two alternative representations of the RGB color model.Hue is the color portion of the color model, and is expressed as a number from 0 to 360 degrees. Saturation is the amount of gray in the color, from 0 to 100 percent. A faded effect can be had from reducing the saturation toward zero to introduce more gray. Value works in conjunction with saturation and describes the brightness or intensity of the color, from 0-100 percent, where 0 is completely black and 100 is the brightest and reveals the most color [9].

The Y′UV model defines a color space in terms of one Luma (Y′) and two Chrominance (UV) components [9].

YCrCbformat Luminance information is stored as a single component(Y) and Chrominance information is stored as two-color difference components (Cr & Cb). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value [9].

Table 2. HOG Feature Vector [9]

colorspace	orient	pix_per_cell	cell_per_block	hog_channel	Accuracy	Time to run
RGB	8	8	1	All	0.934	64.2
RGB	6	8	1	All	0.940	62.0
HSV	8	8	1	All	0.970	65.6
HSV	6	8	1	All	0.958	63.4
HSV	9	8	1	All	0.965	67.5
YCrCb	8	8	1	All	0.978	65.8
HSV (V)	8	8	1	2	0.939	187.4
HSV (H)	8	8	1	0	0.907	28.0
HLS (L)	8	8	1	1	0.939	28.0
LUV (L)	8	8	1	0	0.937	29.0
YcrCb (Y)	8	8	1	0	0.946	27.2
YUV (Y)	8	8	1	0	0.940	30.0
YUV	8	8	1	All	0.975	65.3
LUV	8	8	1	All	0.946	66.5
YcrCb	11	8	1	All	0.985	79.8

Step 3: Perform a HOG feature extraction on a labeled training set of frames.

Using RGB color space Extraction of Histograms of Color features:

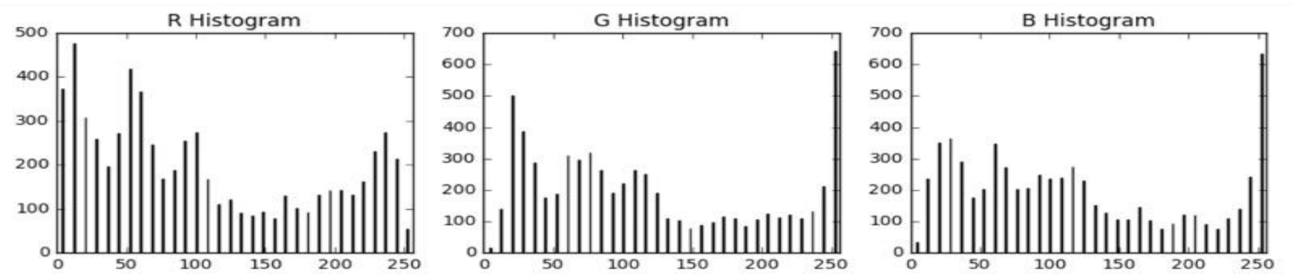


Fig 8. HOG Color Features [9]

Step 4: Apply to train a SVM Classifier.

The next step is to train a classifier. It receives the cars / non-cars data transformed with HOG detector, and returns if the sample is or is not a car.

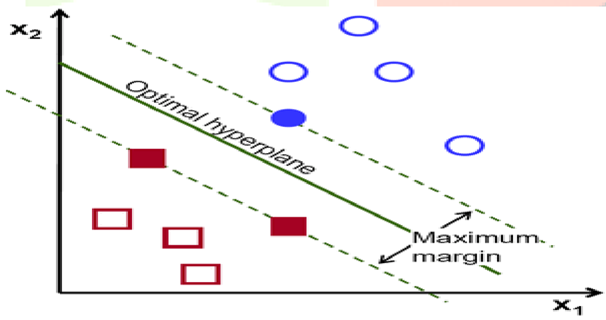


Fig 9. SVM Classifier [7]

I used a **Support Vector Machine Classifier (SVC),** with linear kernel, based on function SVM from Scikit-learn. In Second Step, Identify vehicle in particular frame. In Fig 9 x1 and x2 are the vehicle position coordinator for calculating margin for identify a moving vehicle type [7].

Fig 10. (a) Not found any vehicle in this Frame

Fig 10. (b) Two cars find in this frame

Using SVC and HOG features, we identify vehicle in the different frames. Here, Fig 10. (a) is not find any vehicle in whole frame and Fig 10.(b) is find two cars in the frame.

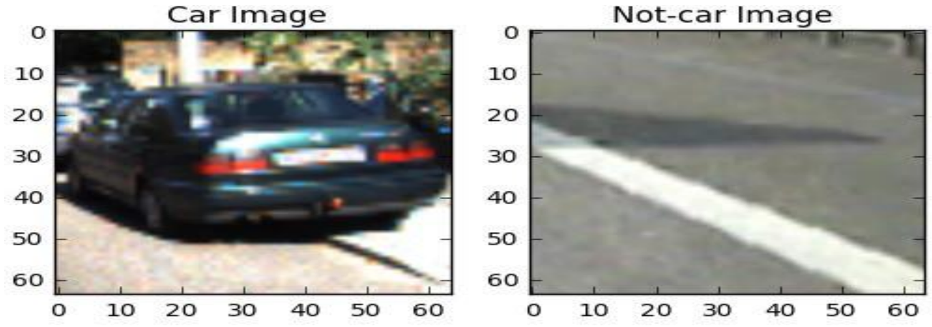
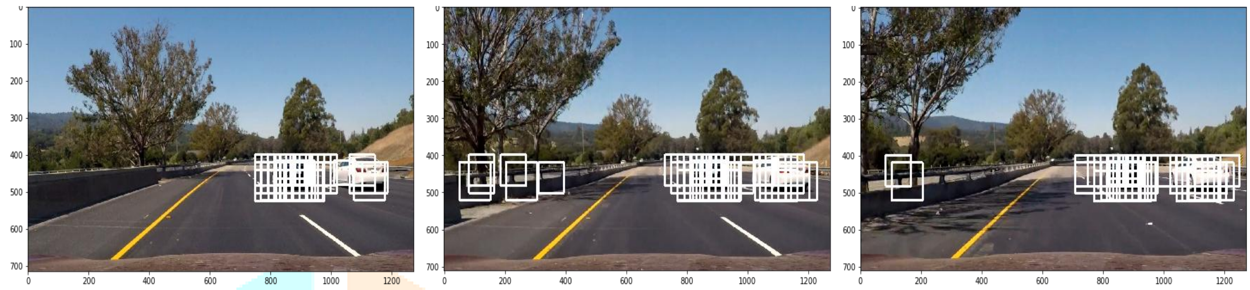


Fig 11. Vehicle Identify in Frame

Step 5: Use a Sliding window technique & the trained classifier to search for vehicle in frames.



(a) (b) (c)

Fig 12. Sliding Window

In Fig 12. Sliding window is trying to find multiple vehicles using this technique. Fig 12. (a) is find vehicle in only single lane and Fig 12. (b) and (c) are finding vehicle in multiple lanes.

Step 6: Run a detection pipeline on the video stream and generate a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.

Sliding window is useful for detecting a vehicle is shown in Fig 13. A **heat map** is a graphical representation of data where the individual values (vehicle) contained in a 2D matrix is represented as colors. Shown in Fig 14.

Fig 13. Detected Vehicles using Sliding Windows

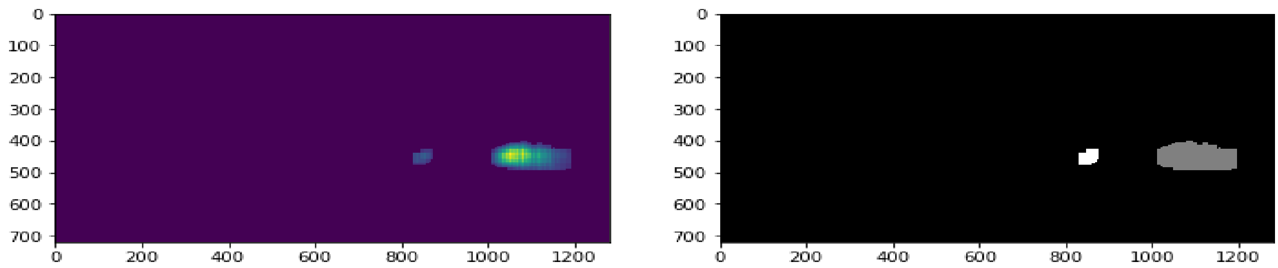
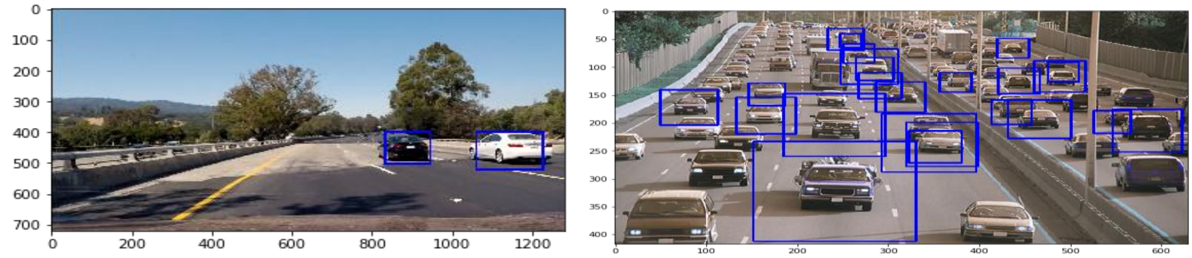


Fig 14. Heat Map

Step 7: Estimate a bounding box for detected vehicle.

Result of detecting a vehicle in shown in the Fig 15 (a) and (b) using different video streams.



(a) (b)

Fig 15. Vehicle Detection in different Video Streams

Detected Vehicle with color, speed and lane status:

(a) (b)

Fig 16. Detected Vehicle with Color, Speed and Lane Status

In this Fig 15 (a), we can see the detected vehicle with their color label and count detected vehicle their camera space. Region of Interest (ROI) is line of camera area and Fig 15 (b) is identify the lane of detected vehicle with their direction and off center.

We also detect multiple vehicles like bus, car, truck all vehicles detected by SVM algorithm with HOG feature. In Fig 16 Multiple, vehicles are detected in video frames

Multiple Vehicle Detected in Frame:

(a) (b)

Fig 17. Multiple Vehicle Detection

V. CONCLUSION AND FUTURE PROSPECT

An analytical solution is provided by us for road traffic video using OpenCV Python. We have demonstrated the scalability of the system. Here we use SVM classifier in entire process and generate all features. A deep learning approach would allow the classifier to learn some condition with different scaling. The advantage of SVM classifier is that it's quite fast to train, doesn't require big volume of data like a deep neural network and it's easy to implement. In future, the system performance will be analyzed by incorporating more nodes. Live analysis of video data is a task operating on a stream of data. OpenCV is intended for batch processing of large volumes of data. To support real time stream computing, We plan to work on enhancing high-level event recognition and prediction as well as classifying vehicles. We will also investigate and validate the relationship between collision probability and safety. There are many open security problems like storage and high security issues with confidential data, which need to be addressed in the cloud. We also use a deep neural network for any future work on vehicle detection because it would be a lot more accurate and would not have issues like in SVM. Such an approach would also be potentially faster in real time processing.

REFERENCES

[1.] Xiaomeng Zhao, Huadong Ma, and Haitao Zhang "HVPI: Extending Hadoop to Support Video Analytics Applications" IEEE - 2015 8th International Conference on Cloud Computing, pp. 789 – 796, 2015.

[2.] Tariq Abdullah, Ashiq Anjum, M Fahim Tariq, Yusuf Baltaci , Nikos Antonopoulos"Traffic Monitoring Using Video Analytics in Cloud " 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 39 – 48, 2014.

[3.] Zhou Lin, Li Zhen, ChenYingmei, Tan Yuqin "The Video Monitoring System Based on Big Data Processing" IEEE - 2014 7th International Conference on Intelligent Computation Technology and Automation, pp. 865 – 868, 2014.

[4.] Xiaomeng Zhao, Huadong Ma, Haitao Zhang, Yi Tang, Guangping Fu "Metadata Extraction and Correction for Large-Scale Traffic Surveillance Videos" 2014 IEEE International Conference on Big Data, pp. 412 – 420, 2014.

[5.] Mazhar Hameed, Hiba Khalid, Dr. Farooque Azam "Big Data: Mathematical Topology Video Data Analytics using Superimposed Learning" IEEE- 2015 International Conference on Machine Learning, pp. 133 – 140, 2015.

[6.] Dongbo Zhang, YanfangShou,Jianmin Xu"The Modeling of Big Traffic Data Processing Based on Cloud Computing" 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 2394 – 2399, June 12-15, 2016.

[7.] Jing Li, Xuquan Lian, Qiang Wu, Jiande Sun, "Real-time Video Copy Detection Based on Hadoop" 6th international conference on information science and Technology, pp. 492 – 497, May 6 – 8, 2016

[8.] Vaithilingam Anantha Natarajan, Subbaiyan Jothilakshmi, Venkat N Gudivada, "Scalable Traffic Video Analytics using Hadoop MapReduce", The First International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2015.

[9.] Porn-anan Raktrakulthum, Chayakorn Netramai, "Vehicle Classification in Congested Traffic Based on 3D Point Cloud Using SVM and KNN" 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand.

[10.]Ling Hu and Qiang Ni, Senior Member, "IoT-Driven Automated Object Detection Algorithm for Urban Surveillance Systems in Smart Cities", 2016 IEEE.