A Real-time Computer Vision System for Managing Traffic

Kushagra Mahajan
IIIT-Delhi
kushagra14055@iiitd.ac.in

Rohan Tiwari
IIIT-Delhi
rohan14157@iiitd.ac.in

Sakti Saurav
IIIT-Delhi
sakti14093@iiitd.ac.in

Abstract

The aim of this project is to design a computer vision based system to detect the number of vehicles at traffic signal using static cameras, so that the amount of time given to a signal can be varied depending on real time traffic density. The approach would involve isolating the lane under consideration from the traffic light camera and is followed by detection of cars. We will be using the HoG features, Part Models and Single Shot Detector (SSD) and comparing the result to attain greater accuracy. Our work also involves measuring vehicle speed using uncalibrated traffic cameras.

Introduction

As the problem of urban traffic congestion spreads, there is a pressing need for a traffic control system that manages traffic dynamically and efficiently. Nowadays, traffic problems are increasing because of the rapid increase in the number of vehicles and the limited resources provided by current infrastructures. The current way for managing traffic light uses timer for each phase. This timer is fixed and does not take into the traffic conditions at that moment. This leads to waste of both time and fuel. We propose a system for controlling the traffic light by detecting vehicle density in lanes. One way to do this is to use a systems of sensors installed along side roads to count the cars like magnetic loop detectors, which are buried underneath roads. But this is a cumbersome task and involves a higher cost of installation.

A much more efficient and cheaper way is to use cameras that will be installed alongside the traffic light that will provide live video feed to count the number of vehicles in real time. The method of dynamically changing the traffic light will try to take into account vehicle breakdown and accidents to help smoother movement of traffic in these situations.

1. Prior Work

The authors of [6] have worked on developing a feature based tracking approach for the task of tracking vehicles under congestion. Instead of tracking entire vehicles, vehicle sub-features are tracked to make the system robust to partial occlusion. The systems before this, broke down for congested traffic due to the problem of partial occlusion. Those systems worked well in free-flowing traffic only.

2.1 Algorithm: First they have rectified the image so that the lanes are parallel. Then feature based detection for tracking. Vehicle sub-features are detected and tracked in order to be insensitive to partial occlusion. Even if part of the vehicle is obscured due to congested traffic, some of the vehicle's sub-features should still remain visible. Corner features are the chosen sub-features since they can be reliably tracked. The purpose of the grouping module is to group together sub-features that come from the same vehicle

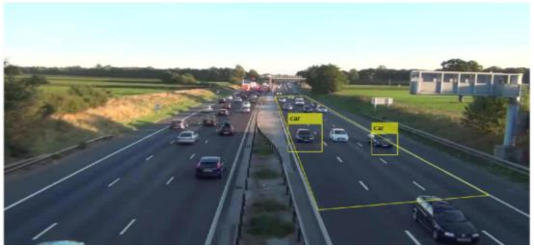
2.2 Result analysis:
It was evaluated considering:
1. True match. A one-to-one matching between a ground truth and a group 2. False negative. An unmatched ground truth. 3. Over Segmentation. A ground truth that matches more than one group. 4. False positive. An unmatched group. 5. Over grouping. A group that matches more than one ground truth.

2.3 Summary:
A vehicle detection and tracking system that is designed to operate in congested traffic. Instead of tracking entire vehicles, vehicle sub-features are tracked, which makes the system less sensitive to the problem of partial occlusion. In order to group sub-features that come from the same vehicle, the constraint of common motion over trajectory lifetimes is used.

2.4 Shortcomings:
Freeway images are not that much congested and also the level of congestion of traffic in the paper shows that. The density of cars at that time and the density of cars in today's world is much larger. It will still fail to count or take into account the space the car occupies which is completely occluded.

Our focus will also be to emulate the work of Dalal et al [2] on cars. The work uses supervised learning of HoG Features extracted from the MIT and INRIA Pedestrian database and classified using a linear SVM.

Girshick et al [2] propose Regional Convolutional Neural Networks in which they use Selective Search for region proposals and extract features from these regions using a CNN. Then linear SVMs are trained for each class to classify the positive region proposals.



HoG Based Car Detection

DPM Based Car Detection

2. Problem Statement

We propose a system for controlling the traffic light by detecting vehicle density in lanes. The approach would also involve isolating the lane under consideration from the traffic camera feed and is followed by detection of cars in the lane under consideration using HoG features extracted from the car images and Part Models for detecting cars and Deep Learning techniques (Single Shot Detector) and comparing results over multiple approaches. The assumption is that the cameras placed are static and give top view images of vehicles and have a large field of view. The system would also involve measuring vehicle speed using uncalibrated traffic cameras to detect instance of over speeding and rash driving.

3. Approach

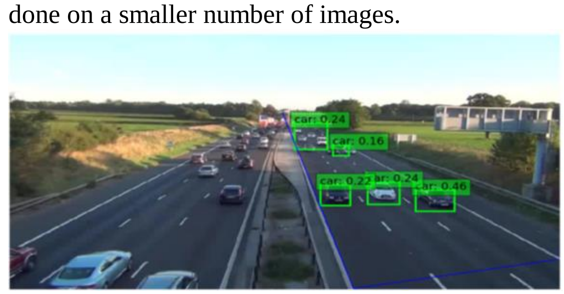
2.1 Car Detection:

2.1.1 **HoG Feature:** We initially started car detection using the HoG features extracted from the car images. In this work, HoG is used to provide the underlying image patch descriptor for matching scale invariant key points. We use Linear SVM as the classifier for learning whether the HoG feature representation is a car or not.

2.1.3 **Deformable Part Models (DPM):** It relies on new methods for discriminative training with partially labeled data. Our star models are defined by a coarse root filter that approximately covers an entire object and higher resolution part filters that cover smaller parts of the object. We have found that using higher resolution features for defining part filters is essential for obtaining high recognition performance. With this approach the part filters capture finer resolution features that are localized to greater accuracy when compared to the features captured by the root filter. Consider building a model for a face.

2.1.4 **Single Shot Detector (SSD):** We extend our car detection approach to deep models using a Single Shot Detector to see if any improvement in detection accuracy is possible using deep learning. We have successfully shown that our SSD model (pre-trained on the Pascal VOC dataset) is able to detect a larger number of vehicles than the HOG feature based model and the Deformable Part Model. To improve detection accuracy further, we have fine-tuned our model on the dataset created.

This was necessary since the Pascal VOC dataset on which SSD has been trained contains only large frontal or back view images of vehicles. However, we focus on images in which the vehicle is smaller and is captured aerially at an angle. Thus fine-tuning was essential and has shown considerably improved detection accuracy as well as the confidence of predictions as compared to the pre-trained model.For measuring the accuracy, we annotate the data only up to a fixed distance from the camera. The images of vehicles that are really far away are not annotated. This makes sure that the mAP accuracy gives a realistic number that can be compared among the different approaches. Also we show that the results improve when the fine-tuning is done on a larger dataset as compared to when it is done on a smaller number of images.



SSD based car detection

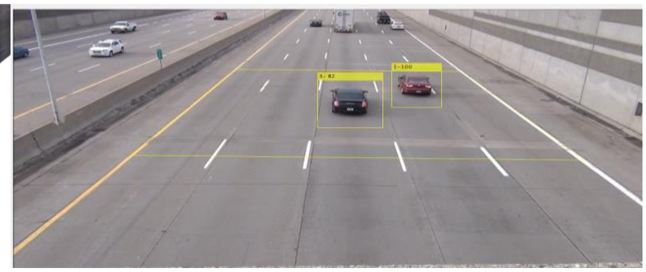
2.2 **Lane Detection:**

We intend to use Canny edge detector and the output is a binary map that shows where the contours of the image are located. Hough transform will be applied on top of this output to obtain better quality lines. The lane detection will help isolate the desired lane from the camera view. Then only the cars detected with bounding box in the desired lanes are counted for calculating the density.

2.3 **Tracking and Measurement of Vehicle Speed:**

We need to calculate the speed of a vehicles on an uncelebrated camera. We use metric rectification to get the image of the lane rectified up to scale and also calculate it's homography. We then calculate the length of a stripe on the lane and equate it to its actual length in meter. This helps us give a conversion between pixel distance and actual length. The distance between the centroid of

Metric Rectified Image



Speed of vehicles

tracked car over two consecutive frames is calculated after apply the homography. Tracking is done using Kalman Filter The conversion factor is multiplied to convert it into distance in meters. Distance travelled is multiplied by the fps of the video to get the speed of the vehicle.

4. Experimental Setup

4.1. The experimental setup consists of videos taken from camera placed on a traffic signal at a slight elevation. We will analyze the images taken by the camera periodically after fixed intervals (say 5 seconds) to determine the situation of the traffic at that point of time. These static images will be used to perform all the experiments and analysis. We will evaluate results on the images taken from foreign countries where the occlusion is less and then try to reciprocate the results on videos with greater density and occlusion.

4.2. We will use evaluations metrics like Mean Average Precision (mAP). In the bounding box detection we choose of threshold of 50% overlap of the detected box with the ground truth box as True Positive. True Positives (TP), True Negative (TN), False Positive (FP) and False Negative (FN), Detection Rate (TP / (TP + FN)) to detect the accuracy of our detection model.

4.3. Number of cars that were correctly detected, not detected, or areas falsely detected as cars and thus be a good measure of our performance. In the demo, we plan to evaluate our approaches on videos by taking frames after fixed time intervals and finding the density of traffic.

8. Dataset

We initially reported results using the pre-trained models on the Pascal VOC dataset. We have annotated about 200 real world aerial images of road traffic using Matlab Computer Vision annotation tool. Each image has about 5-6 bounding box annotations of vehicles of various sizes. We have fine-tuned our models on this dataset created to generate better results than those obtained using pre-trained models on Pascal VOC dataset. For measuring the accuracy, we annotate the data only upto a fixed distance from the camera. The images of vehicles that are really far away are not annotated. This makes sure that the mAP accuracy gives a realistic number that can be compared among the different approaches.

8. Results

We compare the results of the 3 methods: HoG Features, DPM and SSD in terms of the calculating density of cars. The results were tested by comparing mAP values of all the methods over multiple videos of varying traffic conditions. It was observed that SSD gave the best performance followed by Part Model and HoG features. The results are summarized below in the table.

Model	Accuracy
HOG	34.46%
Deformable Parts Model	41.80%
Single Shot Detector(pretrained model)	57.22%
Single Shot Detector (finetuned in 100 images)	64.97%
Single Shot Detector (finetuned in 200 images)	69.56%

We also improved the results by fine-tuning the SSD over the 100 and 200 annotated images. The results showed that the results improved significantly and close to the state of the art values of 81% of SSD.The speed calculation on Highway traffic has results also produced values lie in the range of regular traffic date observed on highways. The tracker was also able to detect stationary vehicles over frames to detect possible vehicle breakdown.

[7]

Fine Tuned SSD

5. Contribution

5.1. Kushagra Mahajan
Deep Models – Single Shot Detector (SSD) and Fine Tuning of the SSD

5.2. Rohan Tiwari
Lane Detection, Deformable Part Models, Vehicle Speed Calculation

5.3. Sakti Saurav
HoG Features, Vehicle Tracking and Vehicle Speed Calculation

6. References

[1] Beymer, David, et al. "A real-time computer vision system for measuring traffic parameters." Computer Vision and Pattern Recognition, 1997. Proceedings. 1997 IEEE Computer Society Conference on. IEEE, 1997.

[2] Dalal, Navneet and Triggs, Bill: Histograms of Oriented Gradients for Human Detection Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)

[3] R Girshick, J Donahue, T Darrell, J Malik: Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2014)

[4] Han, Feng, et al. "A two-stage approach to people and vehicle detection with hog-based svm." Performance Metrics for Intelligent Systems 2006 Workshop. 2006.

[5] Swapnil R. Dhawad, Mrs. Rajashri R. Itkarkar:”Car Detecting Method using high Resolution images.” International Journal on Recent and Innovation Trends in Computing and Communication

[6] Dubská, Markéta, Adam Herout, and Jakub Sochor. "Automatic Camera Calibration for Traffic Understanding." BMVC. 2014