Sanskriti Sharma, Rahul Shiv Chand, Kushagra Jain,

**Abstract:** The project aims at providing a user-friendly tool for real-time vehicular traffic analysis and road-asset management.

The first part of deals with video feed gathered through cameras mounted at appropriate heights (egg. under an overpass) It then goes on to:-

1. Detect the vehicles using Haar features.
2. Count the number of vehicles in every frame.
3. Plot the trajectory of each vehicle using Kalman filter and Hungarian algorithm
4. Compute each vehicle’s speed.

The second part aims to segment the images obtained from static or dynamic cameras into objects belonging to multiple classes ranging from cars, bikes to people and animals for asset-mapping. Our project uses an ensemble of Deep learning ConvNet models FCN and F-RCNN. The F-RCNN model was trained on VOC2012 image dataset. For classes not present in VOC2012 we used a modified fully convolutional model built on top of VGG-16. We combine the outputs of both ConvNet models to obtain our final segmentation. Our final segmented image consists of bounding boxes drawn on the top of the original image, with each bounding having a corresponding class and probability

**Keywords** Haar-features, Kalman Filter, Convolutional Neural Network (ConvNet), FCN.

**Introduction**

Roads and Highways form the backbone of any strong economy. Roads facilitate mobility, thus playing a key role in the socio-economic development of an economy. With the road network across the country becoming denser by the day, the planning and managing of roads has come to be of utmost importance. Road asset management is the process of planning, monitoring and managing road assets in a manner that is both cost effective and provides accurate information.

In this paper we propose a novel method which is both efficient and cost effective for road asset management, particularly in vehicular traffic analysis and asset-mapping of roads using Machine learning and deep learning models. To keep our vehicular traffic analysis as close to real time as possible we used a combination of Haar-features, Kalman Filter and Hungarian algorithm for detection, quantification and trajectory mapping.

In asset-mapping we used state of the art models for segmentation, our contribution included realizing that most state of the art models require a large carefully pre-processed annotated image dataset, such a dataset is hard to duplicate or reproduce for a new surrounding (in our case Indian Roads) , therefore to both maintain both accuracy and ease of training we propose using an ensemble of models, one base model (F-RCNN) to be used on objects invariant to location and a secondary easy to train model (FCN) on objects which are sensitive to surroundings.

**Data and Methods**

**Vehicular Traffic Analysis**

We obtain image frame from a fixed camera mounted at appropriate height. We used a combination of Haar Classifier and CNN.

Haar Classifier[1] is a machine learning based approach where a cascade function is trained from a lot of positive (images containing an object) and negative images (images not containing an object). It is then used to detect objects in other images. The data to train these images was obtained from Image-Net and personal sources. We trained two Harr-classifier to account for both 4-wheeled and 2-wheeled vehicles, both classifiers had 15 stages.

One of the problems with Haar is that it leads to detection of a lot of false positives, that is, detections in areas where there is not an object. This can be handled partially by modifying the Nearest Neighbours parameter, higher this parameter lower the number of bounding boxes that overlap. To make our implementation more robust and also to keep it real time we came up with a solution wherein only the regions where Haar classifier detects an object of interest, were passed to a neural net, this neural net was also trained on the same image dataset as the Haar classifier.

Heuristics about the size of the bounding boxes was also used to eliminate false positives, that is very small or very large bounding boxes were eliminated. Masking was used on the video feed to remove the unwanted regions from the frame. These regions are usually the sidewalks and other regions outside the road.

For trajectory plotting and speed detection we used a Kalman filter[2], it is a probabilistic filter used to predict future values based on previous estimates.

In the first frame, all the vehicles detected are assigned a Kalman filter instance with initial position as given by the Haar-classifier, initial velocity = an arbitrary value (here, 0).The filter instances in each frame is then used to predict the positions and velocities corresponding to every vehicle. To monitor the cars which have left the frame, the Kalman filter instances corresponding to cars which have not been detected for a particular number of frames(5) are erased.

Consider any two consecutive frames f1 and f2. We have (i)the bounding boxes corresponding to all vehicles in f1 (ii) the positions predicted by Kalman filter in f1 about the positions of all those vehicles in f2 and (iii) the bounding boxes corresponding to all of the vehicles in f2. These two sets of bounding boxes have no implicit association and the association has to be formed by certain heuristics.

To assign bounding boxes in f2 to their corresponding predicted values, we associate each bounding box with the closest predicted value. This is reasonable assumption since, in the time between two consecutive frames (usually 1/50th of a second), the vehicles don’t change their positions drastically. For this we use a modified implementation of Hungarian algorithm[3].

After determination of position, velocity can be calculated using adjacent frames, to transform the velocity from image frame to real world frame we used[4].

**Asset-Mapping**

In asset mapping our aim was to segment images obtained from either a static or a dynamic camera into different common objects (people, car, bike, steer sign, building, and animals). We used an ensemble of two different models **FCN and F-RCNN** keeping in mind both accuracy and ease of training.

**FCN (Fully Convolutional Network)**

Our first model was built on top of VGG 16[5], which is a 16 layer ConvNet model, VGG16 was used as a feature extractor to train our own smaller model. The dataset for training was obtained from Image-Net and screenshots of videos of Indian roads.

The dataset was divided with a proportion of 5:1 into training and validation.

For training each image in dataset was passed through a modified pre-trained VGG16 model to extract a 3d 512\*7\*7 feature map of the image, this feature map was used as an input for our own smaller model, our smaller model then classified the images into 11 different classes. To have no restriction on the size of the image, our new model was an **FCN,** which is a Fully Convolutional Network, consisting of no fully connected layers or pooling layers.

All the models used were both trained and built on keras using a TensorFlow backend. The idea behind use of an **FCN** was to facilitate easy and quick training, Obtaining or producing a custom dataset to train FCN is a relatively simpler task, since we only require different types of images to be stored in separate folders (e.g. pictures of people and cars) and minimal pre-processing. FCN can therefore be quickly adapted to a new environment.

**F-RCNN (Faster – Regional CNN)**

Though FCN is an easy to train model, its accuracy in terms of segmentation is at best moderate, therefore we used a second model F-RCNN[6], F-RCNN was the base model for the winning entry in Image-Net 2015 segmentation category. F-RCNN uses two neural networks one to propose regions in the image which might contain an object (RPN) and a second to classify these proposed regions.

We trained the F-RCNN model on the VOC2012[7] database, since training an F-RCNN from scratch is computationally expensive, we modified the model so as to decrease the time it took for training. In our modified implementation we use a pre-trained weights which were obtained by training the model for 1000 epochs.

In our training we kept the weights of the RPN (Regional proposal network) fixed and only the weights of our classifier network were trainable, this reduced the training time from 93 minutes to 8 minutes for one epoch.

**Ensembling both models**

Our F-RCNN was trained to detect cars, bikes, people and animals. Our FCN model was trained on classes not present in VOC2012 but are common sight on roads like Sign boards, traffic lights and buildings.

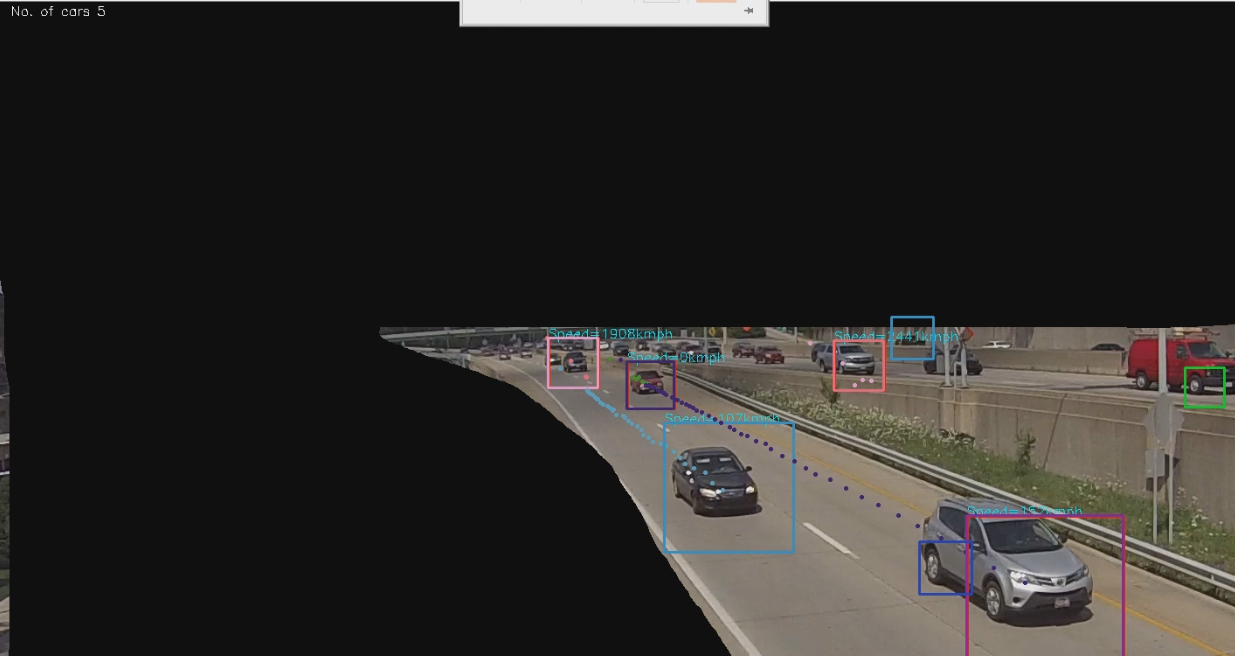
Since the output of an FCN by itself does not give us bounding box but rather a colour coded heat map of probabilities , therefore we obtained bounding boxes by masking individual colours and finding the contours made by them, overlapping boxes were removed using a fast non-maximum suppression algorithm. The final output had objects detected by both F-RCNN and FCN.

**Results and conclusion**

1. The output of our vehicular traffic analysis is shown below:

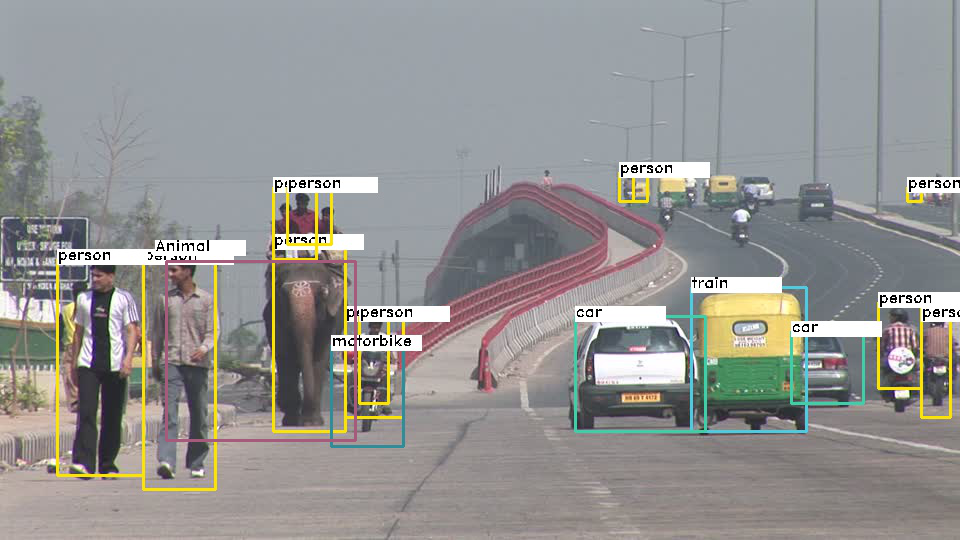


Original Video Frame



Output Video frame (The black portion is the masked portion of the video)

Each car has a trajectory and velocity associated with it, at the top the average count of vehicles over a period of time is shown.

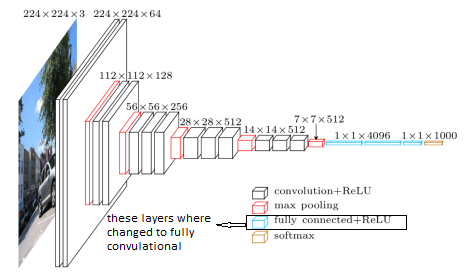
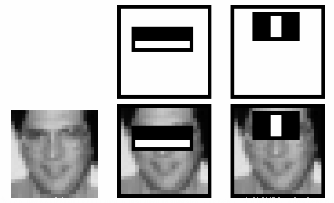
1. The output of our asset-mapping is shown below

Conclusion

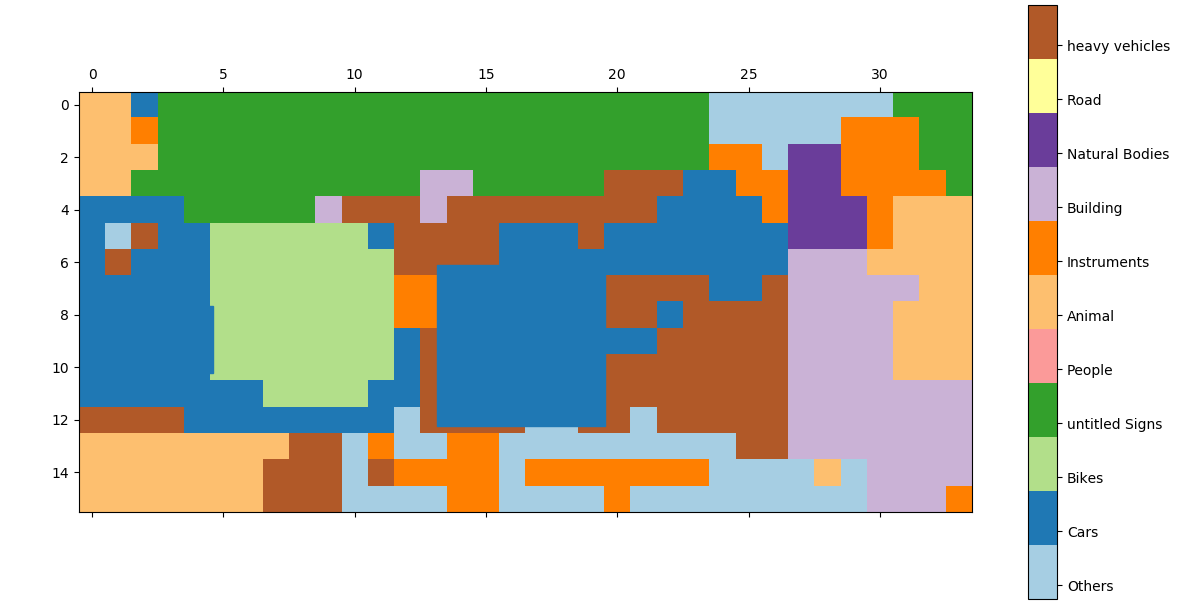
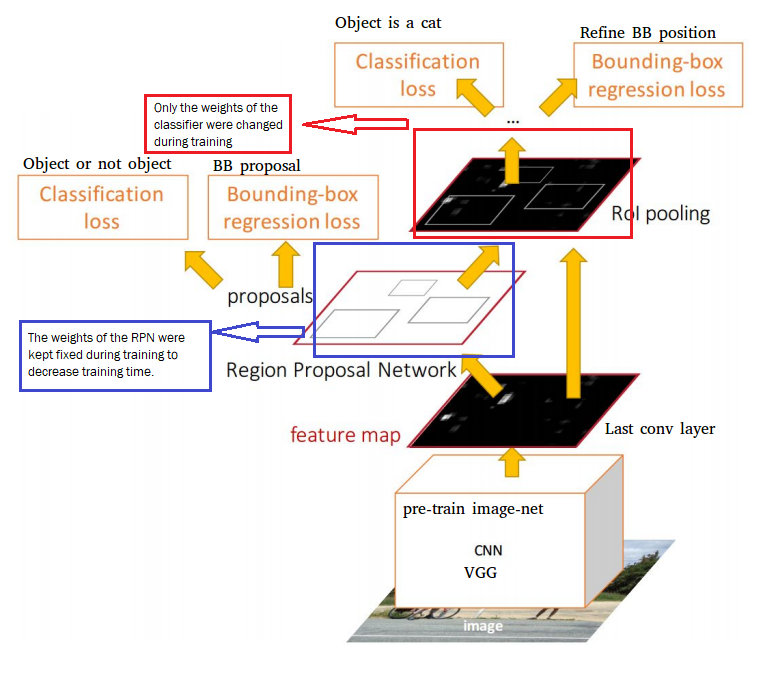
The vehicular traffic analysis project enables traffic analysis using only a camera mounted at an overpass, the above method is both robust and flexible to change in environment , using such analytical system can help in regulating and analysis traffic, the trajectory and vehicular velocity data obtained can further be used to identify vehicles breaking traffic and speed regulation rules.

The future scope of our segmentation project involves asset-management and creation of a geo-tagging system where with the help of a camera mounted on top of a car we can create a geo-tagged model of all objects present in a particular location. Also we hope this project encourages use of Deep learning in Road GIS.

**Diagram References**



**Fig1. Harr-features Fig2. Modified VGG16**



**Fig3. Modified F-RCNN model. Fig4. Output of FCN model.**

**References**

1. Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple. *Conference on Computer Vision and Pattern Recognition*

2. R. E. KALMAN , A New Approach to Linear Filtering and Prediction Problems.

3. Harold W. Kuhn The Hungarian Method for the Assignment Problem.

4. Bas, E. K., & Crisman, J. D. (1997). An easy to install camera calibration for traffic monitoring. IEEE. Retrieved from IEEE Website.

5. Simonyan, K., & Zisserman, A. (n.d.). VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE SCALE IMAGE RECOGNITION. ICLR 2015.

6. Ren, S., Kaiming He, R., & Girshick, J. S. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

7. *VOC Project*. (n.d.). Retrieved from http://host.robots.ox.ac.uk/pascal/VOC/voc2012