CSCE 5214 - Software Development for Artificial Intelligence Group Proposal

A Deep Learning Model of Vehicle Detection System Using R-CNN

Team Members

Gokilavani Sagadevan - 11500587

Gummadi Nagasai- 11540905

Pujah Balasubramaniam - 11500588

Pepeti SaisriTeja - 11555656

Venkata Ram Sushen Rallabandi - 15547220

Abstract:

We would like to implement a decentralized vehicle detection system by using feature engineering and R-CNN. We planned to use feature engineering to filter out the features that We (or the machine) needed to consider after We downloaded a trusted dataset from a reputable website. As a Game changer, We would like to apply R-CNN to create a prediction model that can be applied to different color channels. Additionally, We should be able to develop histogram images of vehicles.

To implement, the first step would be to generate sliding windows to split the search area. One mandatory step is to convert the image into YCbCr color space to extract the search features. The trained R-CNN model prediction can then be applied to the features for categorization. If time permits, the project can also be extended to detect vehicle varieties by incrementing the features. The interface will be built using Python Programming. We can also generate our own dataset. But, it would be apt to work on a massive information set gathered from trusted online sources.

Business Objective:

The goal of this project is to detect the vehicle in a most efficient way rather than using CNN. Fast R-CNN is a well-known deep convolutional network-based approach for object recognition. Regional proposal and object recognition are two different aspects of the original rapid R-CNN.

Data specification:

The unsupervised Dataset has been taken in the form of 2.05 minutes video from the traffic video surveillance from a trusted site. The video consists of more than 1098 vehicle and Non-vehicular objects and those have been extracted using the crop_video code and have been processed.

In the data cleaning part, The crop_video code has played a vital part. It segregated the video into many parts to extract the picture from it. We particularly used a video consists of motorbikes, cars, trucks/buses and non-vehicular objects.

You can see in the output that more than 300 motor-bikes, 1400 cars, 89 heavy vehicles and 500 miscellaneous objects have been detected. By using wide varieties of data, the model has been trained on different object types to face the testing decently.

During the first of training, We found some errors in the cropping due the clarity of the video and it was cleared as we forwarded through.

Design and milestone:

The design of this method is nothing but the advanced form of CNN with precise object recognition.is one of the most comprehensive learning modes, regions with convolutional neural networks (R-CNN), combining rectangular regional suggestions with features of the convolutional neural network. R-CNN is a two-phase acquisition algorithm.

As we have 5 people in our project team, our milestones have been reached as follows:

Gummadi Naga Sai- I've found the proper dataset and preprocessed the data as needed. I also used Crop_video code to extract the images.This work has been completed on March 30th, 2022.

Gokilavani Sagadevan- As I received the preprocessed data, I started working on training the project on march 30th. I also created the Fast R-CNN network according to the data manner. As it was challenging to refer CNN and create the actual R-CNN for precise object recognition, This job has been completed on april 3rd.

Pujah Balsubramaniam- I've started adding the vehicular proposal functions for the training part. As different vehicle types have different features and characteristics, I've generated reports according to that. Thai has been done on april 10th.

Pepeti Saisriteja & venkata Ram Sushen Rallabandi- We both took care of the testing part. As the training has been done on the basic mandatory features, We developed the ability of the model during testing. It's completed by april 15th.

And, All the team members contributed to drafted their respective parts in the project report.

Goals of the project:

The aims are divided into three sub-aims:

1. **Vehicle and Non-Vehicle classification**: We regard localization as a regression problem that a slide-window detector can solve. Using CNN with five layers of convolutional neural networks and strides, we propose a sliding-window technique. The fixed-length feature vectors are extracted by CNN and then categorized as vehicle or non-vehicle.

2. **Supervised and Unsupervised Training**: We use the traditional method of supervised fine-tuning after unsupervised pre-training. When data is insufficient, supervised pre-training on a large auxiliary dataset (ILSVRC) followed by domain-specific fine-tuning on a small dataset is a successful strategy for CNN learning.

3. **Region of Interest (ROI) analysis**: In this Point, Region of Interest has to be found.To create a conv feature map, the CNN uses many convolutional and max pooling algorithms to process the entire image. A region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map for each object proposition.

The entire project is divided into checkpoints mentioned bellow.

Checkpoint 1: Data Cleaning

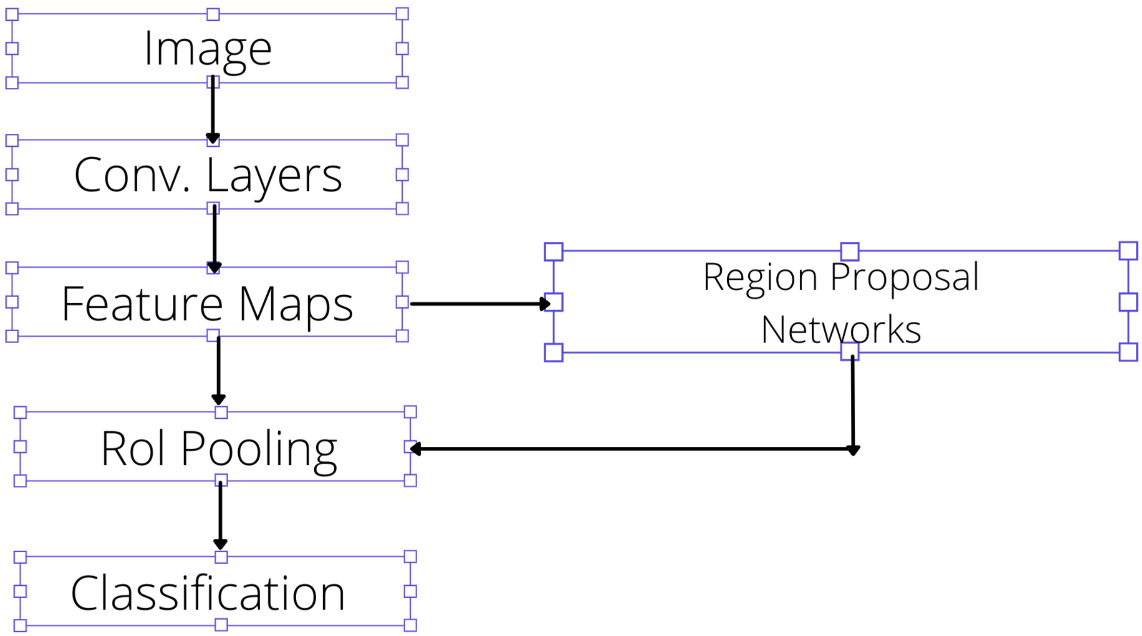
Checkpoint 2: Vehicle and Non-Vehicle Classification

Checkpoint 3: Region of Interest Analysis

Checkpoint 4: Training

Checkpoint 5: Vehicle feature proposal

Flowchart:



Results expecting from each checkpoint:

Checkpoint 1: Load all the Vehicular data received from the Road-side surveillance into the system. Data has been cleaned and formatted into a proper format as it is received as unsupervised data.

Checkpoint 2: Vehicle detection, unlike image classification, necessitates the location of the vehicle in the image. We utilized CNN to extract Fixed-length feature vectors that will be categorized into vehicular and non-vehicular objects in this step.

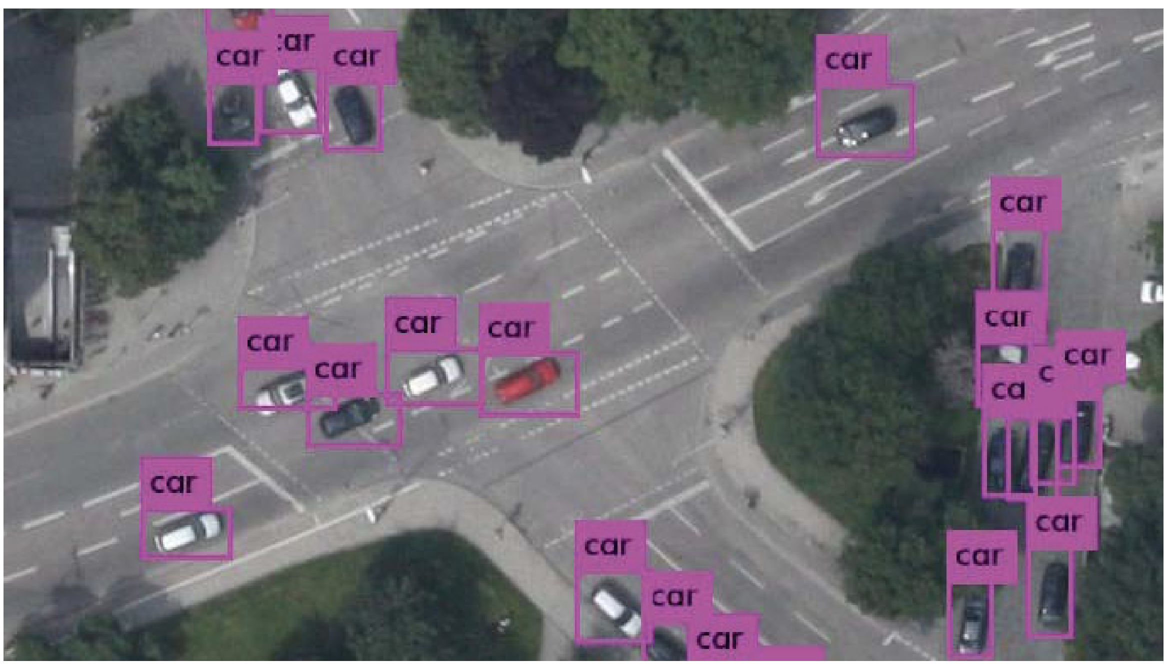
Checkpoint 3: Each feature vector is fed into a series of fully connected layers that branch out into two sibling output layers: one is a softmax layer that distinguishes the vehicle object from the background, and the other is a layer that outputs four real value numbers for the vehicle object that encodes the refined bounding-box positions of the object.

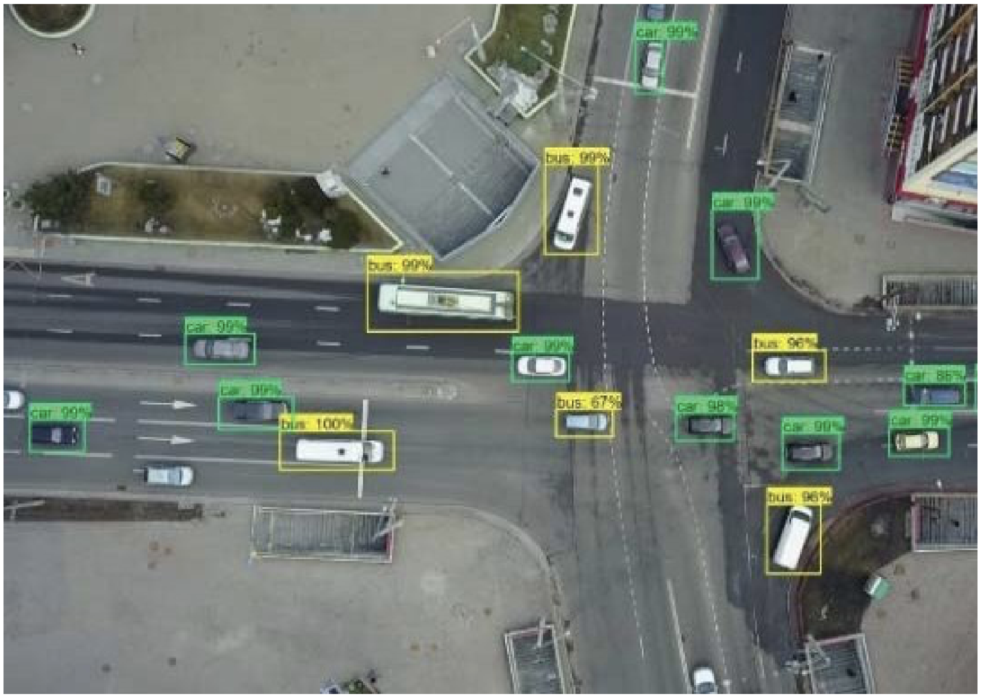
Checkpoint 4: Region of Interest has been constructed in this step to train the data.With a ground-truth bounding box regression offset ttv, each training RoI is categorized as vehicle or non-vehicle. This necessitates the use of the non-maximum regression procedure.

A single vehicle picture has been tested during the training session as follows:

Checkpoint 5:

Finally, the vehicle proposal removes superfluous region suggestions using non-maximum suppression and outputs the final result as below.





Future Enhancement:

As it is challenging to implement this project on real-time video, we used cro_video code to extract the images from stored videos. In the future, we plan to try this on a real-time video. In that way, we can neglect cropping the video.

Conclusion:

The paper introduces a simpler and faster way of making R-CNN faster to find and make local. CNN and the multiplicity of integration provide a representation of the ROI object for the following - fully integrating the layer to differentiate the ROI. We use pre-monitored training and network optimization to solve the problem of training data with insufficient label. In the test, we show that our approach can detect and locate vehicles with a variety of effective views.

Repository:

https://github.com/gokilavanis/Vehicular_Detection_using_R-CNN