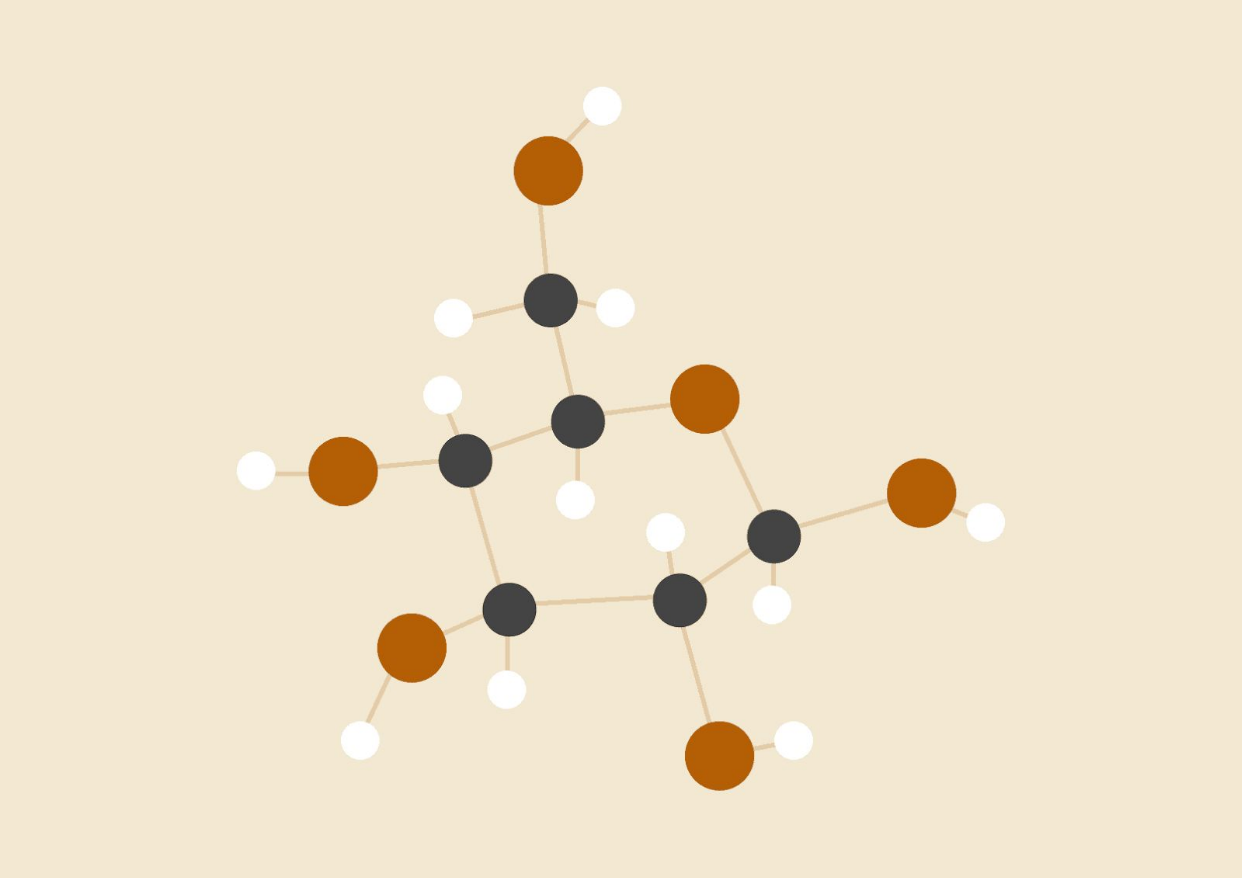
# CS 354 PROJECT REPORT

*Under guidance of Dr. Aruna Tiwari*



## By

Niranjan Joshi 160001026<br>
Kanishkar J 160001028

## INTRODUCTION

The project is about object detection- vehicles and traffic lights using image classifier and sliding window algorithm.

## NETWORK ARCHITECTURE

- ➢ (conv1): Conv2d(3, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
- ➢ (conv2): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
- ➢ (pool): MaxPool2d(kernel_size=8, stride=8, padding=0, dilation=1, ceil_mode=False)
- ➢ (dropout1): Dropout(p=0.25)
- ➢ (conv3): Conv2d(10, 128, kernel_size=(8, 8), stride=(1, 1))
- ➢ (dropout2): Dropout(p=0.5)
- ➢ (conv4): Conv2d(128, 1, kernel_size=(1, 1), stride=(1, 1))

## PIPELINE

- Binary image classifier using Convolutional Neural Networks.
  - The CNN classifies 64x64 closely cropped images into vehicles and non-vehicles.
  - The **network is fully convolutional** and the dense layers are implemented as 1x1 convolutions.
  - The final output of the model is 1x1x1 image or pixel value.
- The classifier is then run of the images to be tested. The output of this is an heatmap image.
- The heatmap is then processed by applying threshold so as to remove false positives. The resulting image contains many bounding boxes.
- Rectangles are then grouped to get one bounding box for producing final output.

## DATA

- Total Samples: 17760
  - Non-Vehicles: 8968
  - Vehicles: 8792

- Split
    - Training: 10656
    - Cross Validation: 3552
    - Testing: 3552

## TRAINING

- Device : CUDA (GPU)
- Batch Size : 64
- Optimizer : ADAM
- Epochs : 20
- Time Taken: 90.2096 s
- Training Accuracy: 99.0897
- Validation Accuracy: 98.5642
- Testing Accuracy: 99.0428

## OUTPUT OF THE PROJECT

- Detect cars from multiple images and draw bounding box around them.
- Detect Cars from video files and produce video with bounding box around the cars in each frame.
- Average Time 1 image : 0.7732 s

(Samples included in the repository)

## LIBRARIES USED

1. Pytorch
2. OpenCV
3. Python Image Library
4. Matplotlib

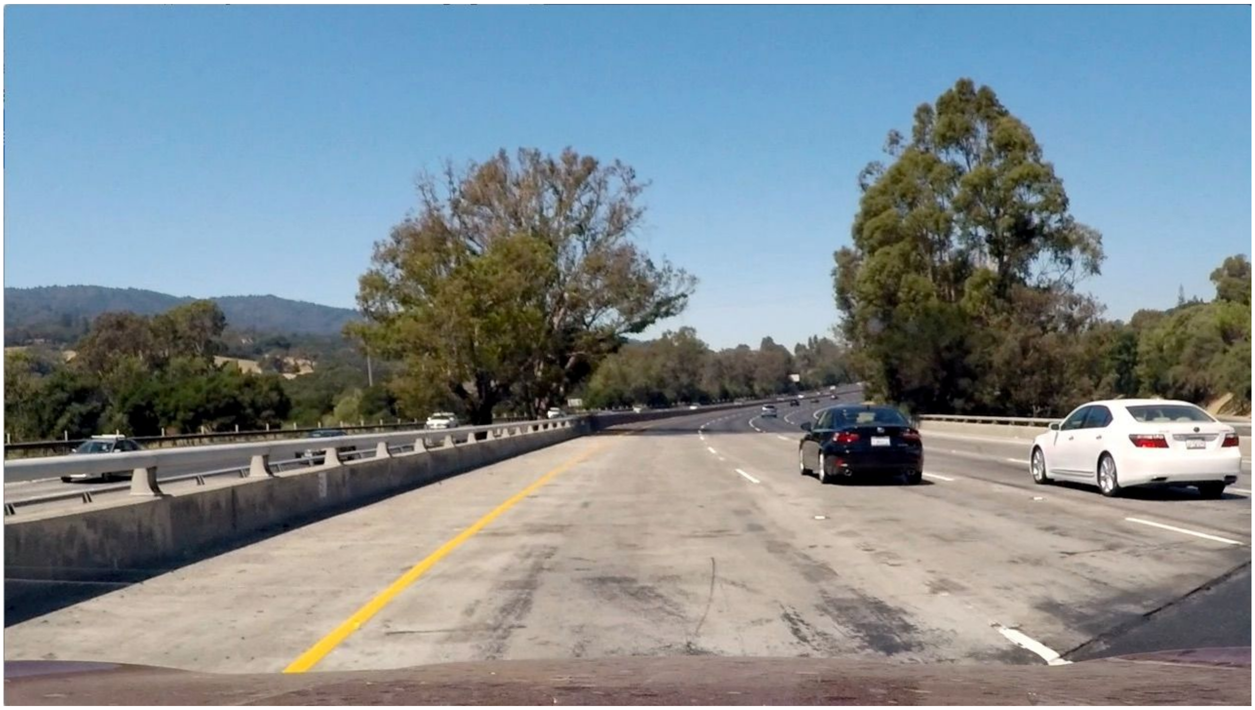
## BUILD AND RUN INSTRUCTIONS

**Project link :** https://github.com/Niranjan-J/CI_Project

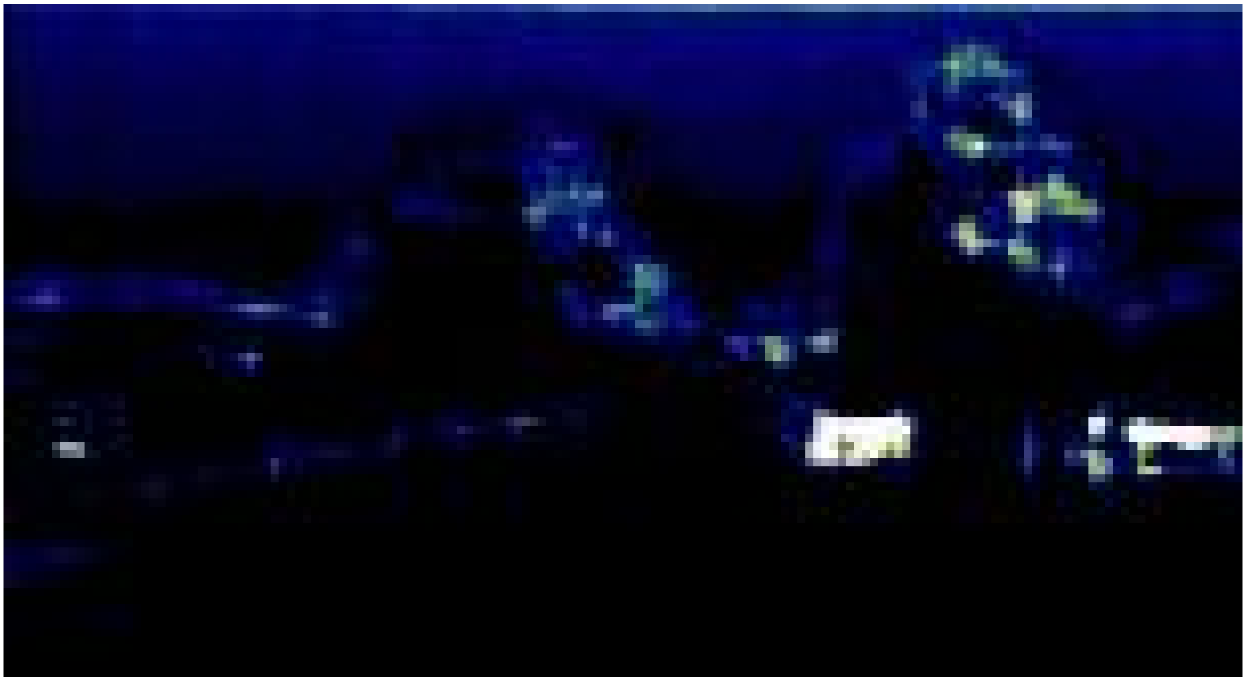
1. Download data zip files and code files.
2. Extract vehicles and non-vehicles into Data/ .
3. Open console in the root project folder, and run the following commands :
   a. python3 ./preprocess.py
   b. python3 ./train.py
   c. python3 ./sw_algorithm.py

## RESULTS

**Sample Test Image :**

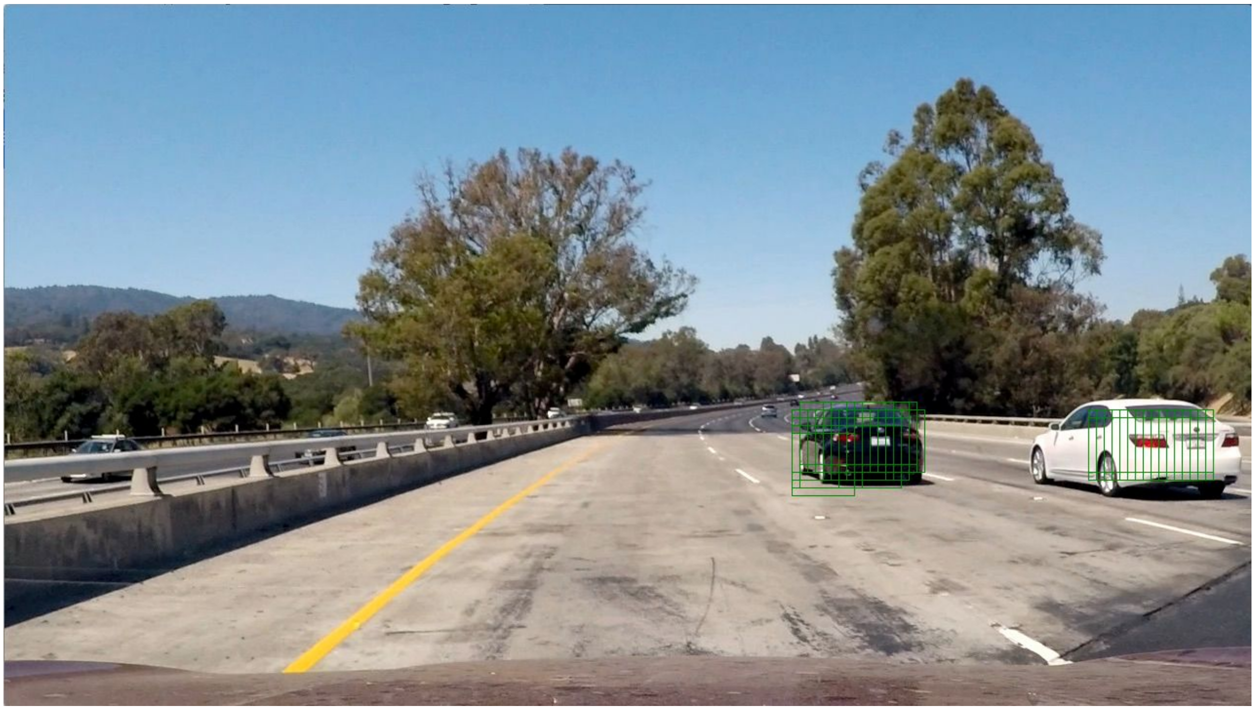


**Heatmap without threshold :**

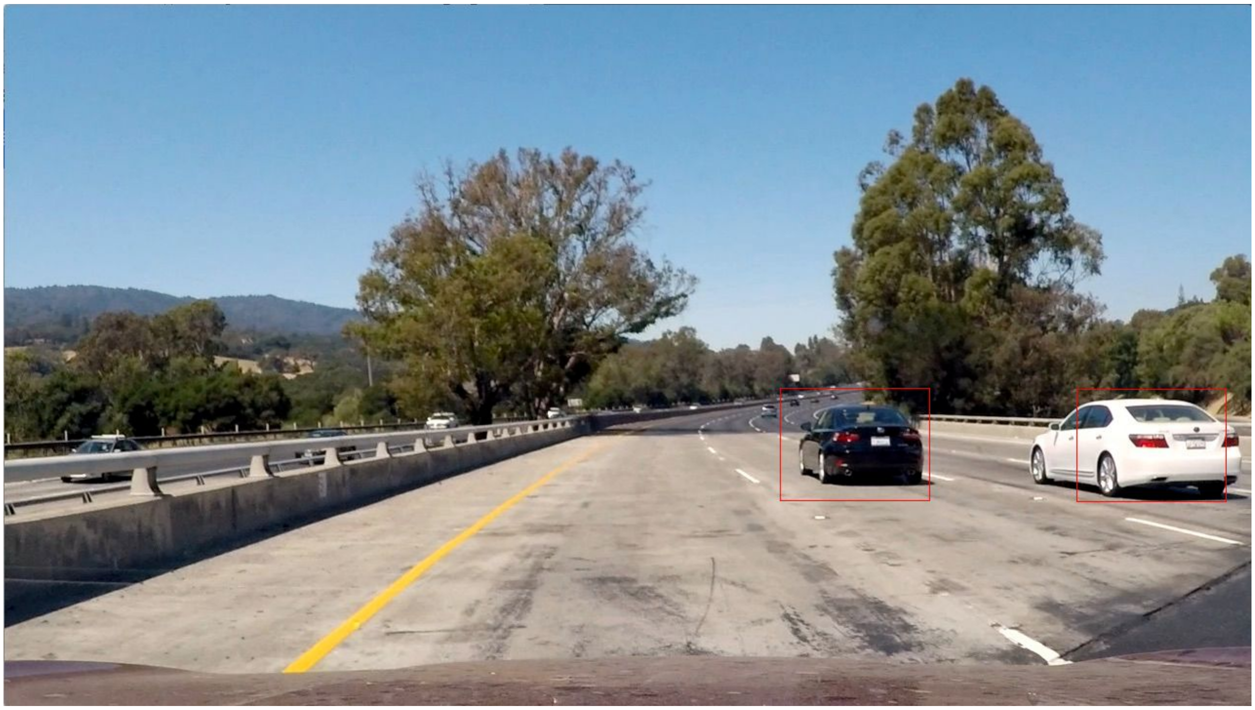


Heatmap with threshold :

Image with multiple bounding boxes :



**Final Output :**



## CONCLUSION

The fully convolutional network speeds up the sliding window phase which would otherwise require few seconds. The bounding boxes given by sliding window algorithm are good approximations though not exactly accurate.

## REFERENCES

1. https://www.youtube.com/watch?v=XdsmlBGOK-k&t=594s