**Write-up Document**

* Exploratory Data Analysis:
  + Displaying 10 images with color coding for each class(more details in Exploratory Data Analysis.ipynb )
  + **A picture containing text, road

    Description automatically generated**
  + Similar images are displayed for validation and test sets too
  + Additional analysis of class distribution:
    - Training:

Text

Description automatically generated

* + - Validation

Graphical user interface, text, application

Description automatically generated

* + - Testing

Graphical user interface, text, application

Description automatically generated

* Explore Image augmentation
* First run
  + Training

**Graphical user interface, chart

Description automatically generated**

* + Eval

**Text

Description automatically generated**

* Augmentation applied
  + random\_adjust\_hue : This is done so that vehicles are detected not based on their colors anad conversely all cars should be deteced as cars irrespective of its color

**A picture containing text, road, outdoor, sign

Description automatically generated**

* + adjust\_gamma :  Gamma value changes difference between light and dark areas(icreasing gamma would make darker area darker and light areas lighter)

**Graphical user interface

Description automatically generated**

* + random\_jitter\_boxes : this changes box bounds by a value and would help in generalization

**A picture containing text, outdoor, street, way

Description automatically generated**

* Final Run
  + Training
    - Loss is smaller with all augmentations mentioned above
    - Algorithm has started to pick up objects in image as seen in testing
    - Low classification and localization loss
    - Slow decreasing of loss towards optimal(not yet there)

**Chart

Description automatically generated**

* + Validation

**A screenshot of a computer

Description automatically generated with medium confidence**

Other Modifications of tensorflow obejct detection API explored include:

1. Changing optimizer from momentum optimizer to RMS optimizer
2. Changing learning rate schedule from cosine decay to exponential decay
3. increasing batch size is no longer possible with memory usage overhead
4. changing achor\_generator values
5. IOU threshold for NMS
6. using drop out and augmentation to avoid overfitting