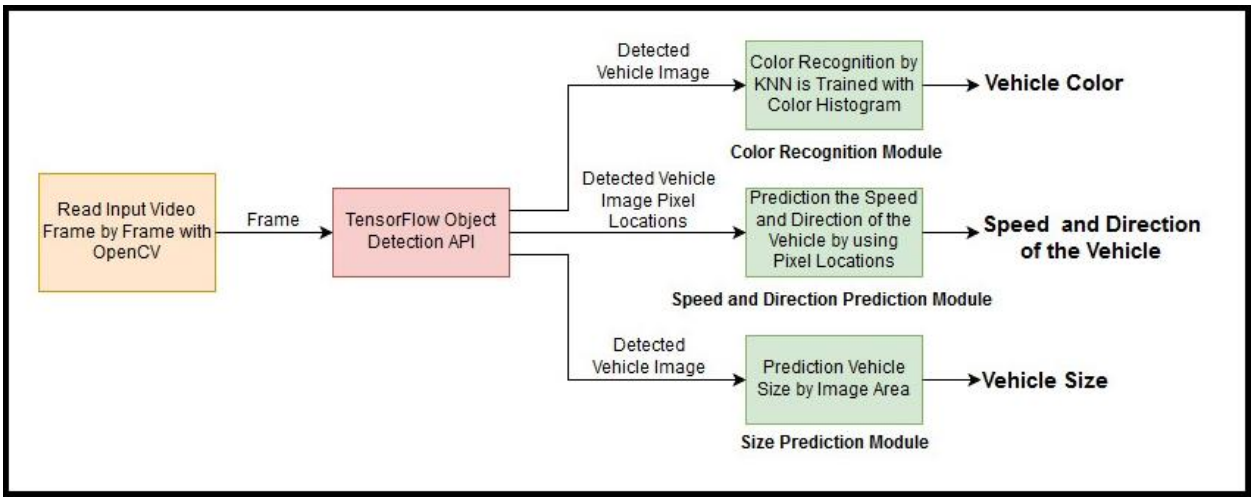
Aircraft Landing at Runway

General Capabilities of This Project

- Recognition of approximate vehicle color
- Detection of vehicle's direction of travel
- Prediction the speed of the vehicle
- Prediction of approximate vehicle size
- The images of detected vehicles are cropped from video frame and they are saved as new images under "detected vehicles" folder path
- The program gives a .csv file as an output (traffic_measurement.csv) which includes "Vehicle Type/Size", " Vehicle Color", " Vehicle Movement Direction", " Vehicle Speed (km/h)" rows, after the end of the process for the source video file.
- Detecting Tail Number or any Text from the cropped detected image under "detected vehicles" folder path

Theory

System Architecture



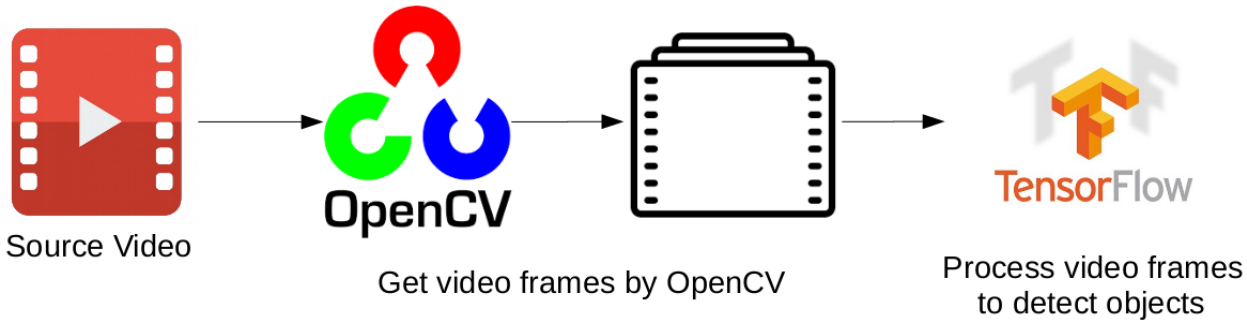
- Vehicle detection and classification have been developed using TensorFlow Object Detection API
- Vehicle speed prediction has been developed using OpenCV via image pixel manipulation and calculation

- Vehicle color prediction has been developed using OpenCV via K-Nearest Neighbors Machine Learning Classification Algorithm is Trained Color Histogram Features

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.

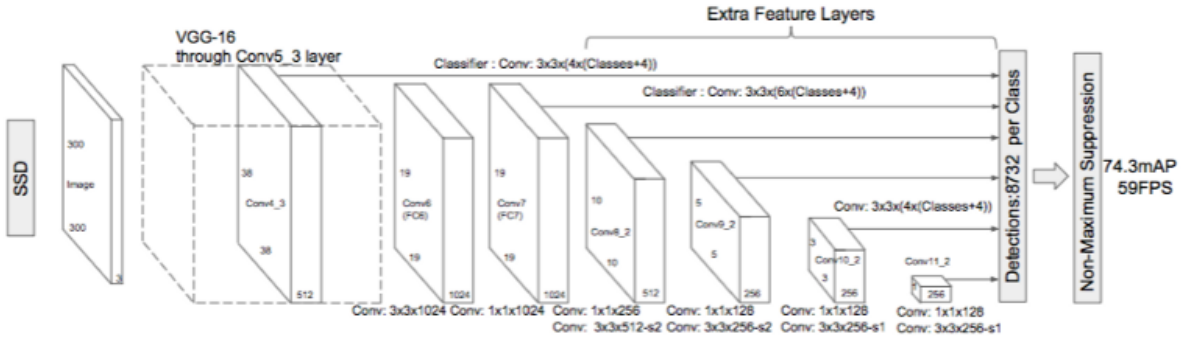
OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Tracker



Source video is read frame by frame with OpenCV. Each frames is processed by "SSD with Mobile net" model is developed on TensorFlow. This is a loop that continue working till reaching end of the video. The main pipeline of the tracker is given at the above Figure.

Model



By default an "SSD with Mobile net" model is used in this project. We must update this with a model that can detect airplanes instead of cars. We will do this when we are able to record videos of the airplane

Project Demo

Demo is in the video provided names “demo.mp4”

Installation

1) *Python and pip*

Python 3.6+

2) OpenCV

OpenCV is a must for this project to run

3) TensorFlow

Install TensorFlow by invoking one of the following commands:

```
$ pip3 install tensorflow==1.14.0     # Python 3.n; CPU support (no GPU support)
$ pip3 install tensorflow-gpu==1.14.0 # Python 3.n; GPU support
```

4) TensorFlow Object Detection API

The Model used to detect airplanes in the video

5) Tesseract OCR

1. You need to have Tesseract OCR installed on your computer.
   get it from here. https://github.com/UB-Mannheim/tesseract/wiki
   Download the suitable version.
2. Add Tesseract path to your System Environment. i.e. Edit system variables.
3. Run `pip install pytesseract` and `pip install tesseract`
4. **Add this line to your python script every time**
   pytesseract.pytesseract.tesseract_cmd = 'C:/OCR/Tesseract-OCR/tesseract.exe' # your path may be different

After completing these 5 installation steps that are given at above, you can test the project by this command:

```
python3 vehicle_detection_main.py
```

Code

The code starts with some essential imports of python, some libraries that we will be using to perform all the tasks

The following is the explanation for the function “object_detection_function”

After setting the name of our input video(it can also be a webcam or a camera video) we will start by importing the models we have downloaded to detect the presence of a vehicle in the video.

After that we will set a ROI Line where we will calculate the speed of the detected object. This line can be set anywhere on the screen according to our video

When the detected object reaches the ROI line set by us, it will detect and calculate the speed of the detected object by calculating the frames of the object while on the ROI.

Also note that at this point, a cropped image of the detected object will be taken which will be named “vechicle1.png” and will be used by our OCR function

The following is the explanation for the function “OCR”

In our OCR function, we will start detecting the tail number of our detected object. We will take the cropped image of our detected object and run the OCR on it and get the text as output and show it to the user.