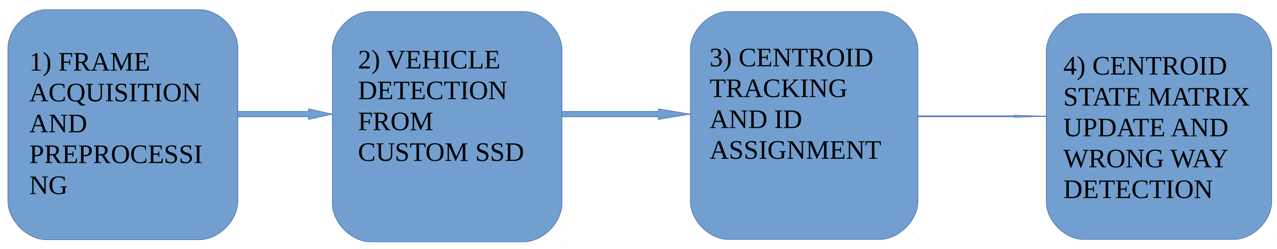
IDENTIFY ROAD FLOW DIRECTION & WRONG WAY DETECTION USING DEEP LEARNING & OBJECT TRACKING

SYSTEM ARCHITECTURE:


1) FRAME ACQUISITION AND PREPROCESSING
2) VEHICLE DETECTION FROM CUSTOM SSD
3) CENTROID TRACKING AND ID ASSIGNMENT
4) CENTROID STATE MATRIX UPDATE AND WRONG WAY DETECTION

The proposed system is a straightforward implementation of custom SSD based vehicle detector followed by centroid tracking and updating state matrix of vehicle for every frame.

In stage1, the frames are processed from the video with standard OpenCV utilities and preprocessing like BGR2RGB, frame resizing and float32 conversion are done to prepare the frame to be fed into the model.

In stage2, the preprocessed frame is fed to the model. The custom model by design outputs 123120 boxes for possible locations of vehicle in that image which are postprocessed by confidence thresholding and Non-Maximum Suppression/IoU thresholding of same class. The model is designed to be lightweight for faster experimentation and prototyping , more explanation in model summary.

In stage3,The centroid tracking is done from the coordinates obtained from stage 2 vehicle detector. The detected vehicle centroids are registered with respective objectID and updated by computing euclidean distance between the successive frames. The centroids are de-registered if not detected for consecutive 3 frames.

In stage4, a state matrix is maintained for each ObjectID.The state values are updated by how the Y coordinates or height coordinates of centroid moves from inital position. If state of centroid is non-negative, the centroid will be monitored for 10 frames the centroid will turn Orange/Yellow in output window. If the state is negative for more than 10 frames , the centroid will be flagged as moving in wrong direction and bounding box turns Red .

MODEL SUMMARY:

ARCHITECTURE: CUSTOMIZED SINGLE SHOT DETECTOR

INPUT SHAPE: (BATCH,384,640,3)

OUTPUT SHAPE: (BATCH,123120,14)
14 refers to num_classes+1+4 coords+4 anchors +4 variances
123120 refers to num anchors predicted for image

POSTPROCESSING: IOU THRESHOLD & NMS

DECODED BBOX COORDINATES: XMIN,YMIN,XMAX,YMAX

MODEL PARAMS / MEMORY SIZE / FPS : 202,876 / 1.2MB / 6-8 FPS

TESTED HARDWARE: CORE I5 7th GEN . 8GB RAM . NO GPU USED

RESOURCES:

The centroid tracking algorithm(Stage3) is implemented from https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/ . Apart from this all other software resources are from my personal tech stack . The training is done on Colaboratory K80 GPU. Yolomark is mode for annotation

Train Data Frames: NVR_ch1_main_20200207123000_20200207130000.asf
Test Data Frames: NVR_ch1_main_20200207140000_20200207143000.asf

Link to video frames :

Train 449 frames: https://drive.google.com/file/d/1yjjgkL0fnKHCp-2y6jTcjlgP9ploUxPX/view?usp=sharing

Test 80 frames: https://drive.google.com/file/d/1-WZLDFljEmS7tg2xXKQ-j3aUSFcU4c6T/view?usp=sharing

KNOWN ISSUES:

The centroid tracking algorithm assigns ObjectID of one vehicle to another if they are very close to each other leading to confusion in tracking leading to False Positives in wrongway flag assignment .

The model produces false positives & multiple bounding boxes for very heavy vehicles. This can be attributed to sparse amount of heavy vehicles in training data.

ToDo:

Optimise centroid tracking algorithm by assigning it only in certain area say within the height and weight of the vehicle box.

More training data is required to create a robust vehicle detector. Experiment with more data augmentations and semi-supervised labelling techniques.