# Files

* Main.py: Main file which creates full output with vehicle tracking, lane/line creation
* Dashcam.py: Stripped down version of main.py intended for moving cameras. Removes lanes/lines functionality and doesn’t track objects between frames
* Config.py: Control hyperparameters, model option and input/output filepaths
* Setup\_video.py: Code for drawing lanes/lines on new video
* Tracking.py: Algorithm for assigning vehicle unique IDs and tracking each vehicle between frames
* My\_utils.py: All code and functions needed that aren’t related to tracking. Drawing text on frame, custom classes, list of objects of interest, plotting number of cars
* Model\_setup.py: Creates standardised get\_model\_output function using chosen model framework so pytorch and opencv both run identically in main.py

## Config.py

* MODEL\_TYPE: choose pytorch or cv2
* weightsPath, configPath: stored weights and config for cv2 model
* preDefinedConfidence: Model confidence required for vehicle to be counted. Only for cv2
* preDefinedThreshold: NMS threshold. Only for cv2
* RUN\_EVERY: model uses every RUN\_EVERYth frame
* FRAMES\_BEFORE\_CURRENT: Vehicles are sometimes missed in a single frame. The tracking algorithm searches through the previous FRAMES\_BEFORE\_CURRENT frames for each vehicle, taking the most recent sighting
* inputWidth, inputHeight: frame is resized for model. Must be a squre. Only for cv2
* BATCH\_SIZE: only for pytorch

## Main.py

Initial setup:

* Checks if lanes and lines for the chosen video name already exist in video\_configs. If not, calls setup\_video.py to do so and stores results
* Sets up video stream and initialises writer to save output video
* Starts looping through video

Loop over frames:

* Creates list of frames of length BATCH\_SIZE. If BATCH\_SIZE = 3, RUN\_EVERY = 4, it will collect frames 4, 8, 12 and discard the rest
* Gets lists of potential bounding boxes for objects, model confidences and predicted class for each box for each frame
* Loops through boxes, confidences, classes for each frame
  + Draws lines and lanes on frame
  + Runs tracking algorithm. Uses previous detections in all\_detections to track vehicles and identify new vehicles. Idxs represent the indices of the bounding boxes to be kept after running non-maximum suppression on the boxes
  + Detects line crossings and vehicles in each lane
  + Writes text on frame
  + Stores frame with writer
* Updates messy stats\_df frame and saves it
* Displays final frame of the batch and updates counting graph

Saves objects, stats, history of run