Suchet Sapre
Period: 3
01/09/20

Journal 14

Throughout this past week, I outlined my plan on how I will utilize Machine Learning (ML) to bolster the generalizability of my current naive crash predictor. The first component of a future ML model that I would have to consider is the format of the features and labels of the dataset. For a given video, the features (x-data) would be the set X of frames and frame features. Here, each individual element, X_i, would represent the grayscale and down sampled i^{th}image frame along with the locations, velocities, and depths of the vehicles, pedestrians, etc. In this manner, the ML model would be given both the raw image frames themselves and more sophisticated information about the important attributes of the video itself. The length of $X$is simply the number of frames in a given video which in my case is 100.

Constructing the labels of this dash camera video dataset is slightly more complicated because I want to train the model to be able to *predict* vehicle crashes not simply detect them. In order to account for this, my method for creating the labels (y-data) of the dataset relies on offsetting the time of the crash. Specifically, consider a video V, in which the vehicle crash occurs at frame t_c. I will create a "frame offset amount" number $q$which represents how far into the future I want the model to predict into the future. For a 30 FPS video, if I wanted my model to predict the crashes 1 second into the future, I would set $q$to be 30. My labels will simply be a one-dimensional binary vector Y whose size is the number of frames in V. In this vector, elements $Y[0{:}t_c - q\ ]$ are all 0 and all elements $Y[t_c - q{:}\ \mathrm{len}(V)]$ are a 1.

By feeding the ML model each of these (X_i, Y_i)pairs sequentially into a Recurrent Neural Network (RNN) so that it will slowly be able to learn what types of frames indicate that a car crash will occur one second into the future. In the coming weeks, I will determine the training specifications needed to ensure that the model is able to correctly learn from the videos, namely, the loss functions, batch sizes, and training time I will use.