**Week 1**

* Problem statement narrowed down on.
* Threat level factors considered : object orientation & angle, relative speed & displacement of the obstacle, (maybe the likelihood of erratic behaviour).
* The algorithm for classification poses the largest challenge in the detection of orientation angle.
* Initially experimented with the usage of stereo cameras to measure depth of the object and use that data to calculate angle of orientation.
* Currently exploring the usage of quaternions to directly extract angle of orientation from a rotation matrix obtained by processing pedestrian body landmarks (identified using mediapipe at the moment) due to latency issues in the usage of two simultaneously active cameras.
* Learned how to implement a work ethic for projects :
  + Maintain git repo.
  + Record decisions for paper writing.
  + Have hackathons to speed up coding.

**Week 2**

* Found the angle of orientation from quaternion but only from 0 to 180.
* Angle measurement taken from 0 degrees (left facing, user perspective) to 180 degrees (right facing, user perspective) but is only accurate from 30 to 160 degrees for reasons not ascertained yet.
* Working on a method to detect a face to determine if the subject is front or back facing.
* Ditched Haar cascades for the same due to failure in different lighting.
* Ditched face detection altogether in favour of direction of motion (potential edge case – people walking backwards for some reason).
* Ditched stereo approach and currently considering in frame speed, displacement from centre and percentage of Y axis covered by target pedestrian.

**Week 3**

* Hardcoded a path overlay.
* Implemented fuzzy logic to determine predicted path.
* Implemented a streaming average filter for processed outputs. The same was not done for raw inputs due to inaccurate predictions.
* Recorded all relevant readings onto a CSV file for data analysis and visualizations.
* Automated testing for stationary robot cases implemented.
* Automated data extraction and visualization.
* Streaming average gives outliers upon context switch from one pedestrian to another.
* Falling behind on implementing multithreading to detect multiple pedestrians at a time as well as replacing streaming average filter with Kalman filter. Will have to carry this over to the next week.