**Original Milestone 4 Goals:**
Code and presentation should be done. At this point, the code should do the following: have a GUI that allows the user to select a video/images to process with processing options, detect relevant objects in the provided file(s), determine the velocity and apply the per-object motion blur effect accordingly, and show a preview of the output and the option to save it.

NOTE: For some reason cloning from the Github repo won't download Model.pt properly. For the code to run properly, please download Model.pt directly from the Github repo or else it won't work

**Progress:**

Presentation/Report:
Report and video are both finished and ready to submit.

Blurring:
Blurring has been completed, so that the blurred boxes are easier to see. In other words, the blurring effect has been increased.

Integration:
We have connected all parts together, this means that our backend (which works locally, i.e. not hosted) can take an input video, and pass the input to the machine learning section. This will generate the boxes and ids that will be passed to the image manipulation, which will return a newly generated video, which is our end product. This video can then be downloaded by a link on the webpage.

Testing:
Video returned is at a higher fps as expected, and blur effect on boxes is very noticeable. Takes roughly 20 minutes to modify a 24 fps, 8 second video.

Conclusion:
All parts are working as intended with each other, and does what we expect.

**Changes:**
-Once the video is finished running through the model and returned, it doesn't display, it only leaves a link so that the video can be downloaded.

-Blurring was changed to have less blending/merging so that the blurring effect was emphasized more in video.

-If frame A and frame B had nothing to blur in between, another frame A would be inserted in between. This would prevent certain portions of the video which had nothing to blur to play faster, since the fps had gone up.

**Challenges/bottlenecks:**

-Rotation of a limb still doesn’t work particularly well with blurring, whereas translation works a lot better.

-Testing was slow, as well as finding and fixing bugs in the code since one successful run on a short video can take upwards of 8 minutes.

-Splitting work between all 4 members at this point was difficult, as most parts had been completed and only needed one or two people to integrate together. Others would simply work on the report/video/milestone.

- By wrapping the processed video file in a Django File object, the path url (a member variable of the file class) is unable to be displayed in a video through html. Therefore it is only available as a link

-Hosting externally on Google Cloud and Heroku ran into a lot of issues, and storing model and loaded images was very difficult to implement

**Team Member Progress:**

Henry:
Helped fine tune and debug the project. Worked a lot on the final report. Created the diagram of components on the final report. Edited most of the video. Animated parts of the video to better convey concepts.

Nancy:
Contributed with the report and video.

William:
Changed blurring, so that there was less blending and more blurring per frame. This makes the blurring more pronounced per frame. Also added feature so that when no blurring is done, the original frame is repeated. This makes it so that parts of the video with no blurring don’t play faster than the original. Did report/video on image manipulation section.

Ivan:
Cleaned up the code, took out useless code that was commented. Also was able to wrap the processed video in a Django file object so I can display the downloadable url of a local file to a server’s side.