**Creating Video Capture Script**

| **Notes** | **Script / Code** |
| --- | --- |
| After looking into what causes choppy videos on the pi camera, I came across a forum that mentioned the intra-refresh period and provided the code on the right that runs a clear video - which worked  <https://forums.raspberrypi.com/viewtopic.php?t=245875> | raspivid -v -o video.h264 -t 20000 -g 1 |
| Following what I had learned with the intra-refresh period, I continued researching what causes the jumps in frames, and found the below article as to how it is best to set the frame rate for the videos.  <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7> |  |

**File Conversion Script**

| Script pictured here. This script sets the variables for the Pi camera, then in the convert function the camera records and then converts the .h264 to an .mp4 using the command line “MP4Box -add” command. This command is usually done in a CLI interface rather than code but works well here. |  |
| --- | --- |
| Showing completion of the script. |  |
| Evidence of the original video.h264 and then the newly converted video.mp4 next to it. |  |
| After reviewing the camera\_2 script and the camera\_3 script, I felt as if these could be combined and using some OOP principles, could be made more efficient for use later in our project.  I found that the mp4 video was not skipping frames but was sped up, so () was changed to fix this. |  |

**Articles from Research**

| Notes | Link |
| --- | --- |
| Has good information on what the intra-refresh period means for the h264 video format | <https://news.ycombinator.com/item?id=25971605> |
| This article contained the information regarding frame rates. While it mentions what needs to be done for just images, it examples shown worked well for videos as well. | <https://pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/> |
|  | <http://thezanshow.com/electronics-tutorials/raspberry-pi/tutorial-13> |