

Report 3

Jiachen Tian

Objectives achieved this week

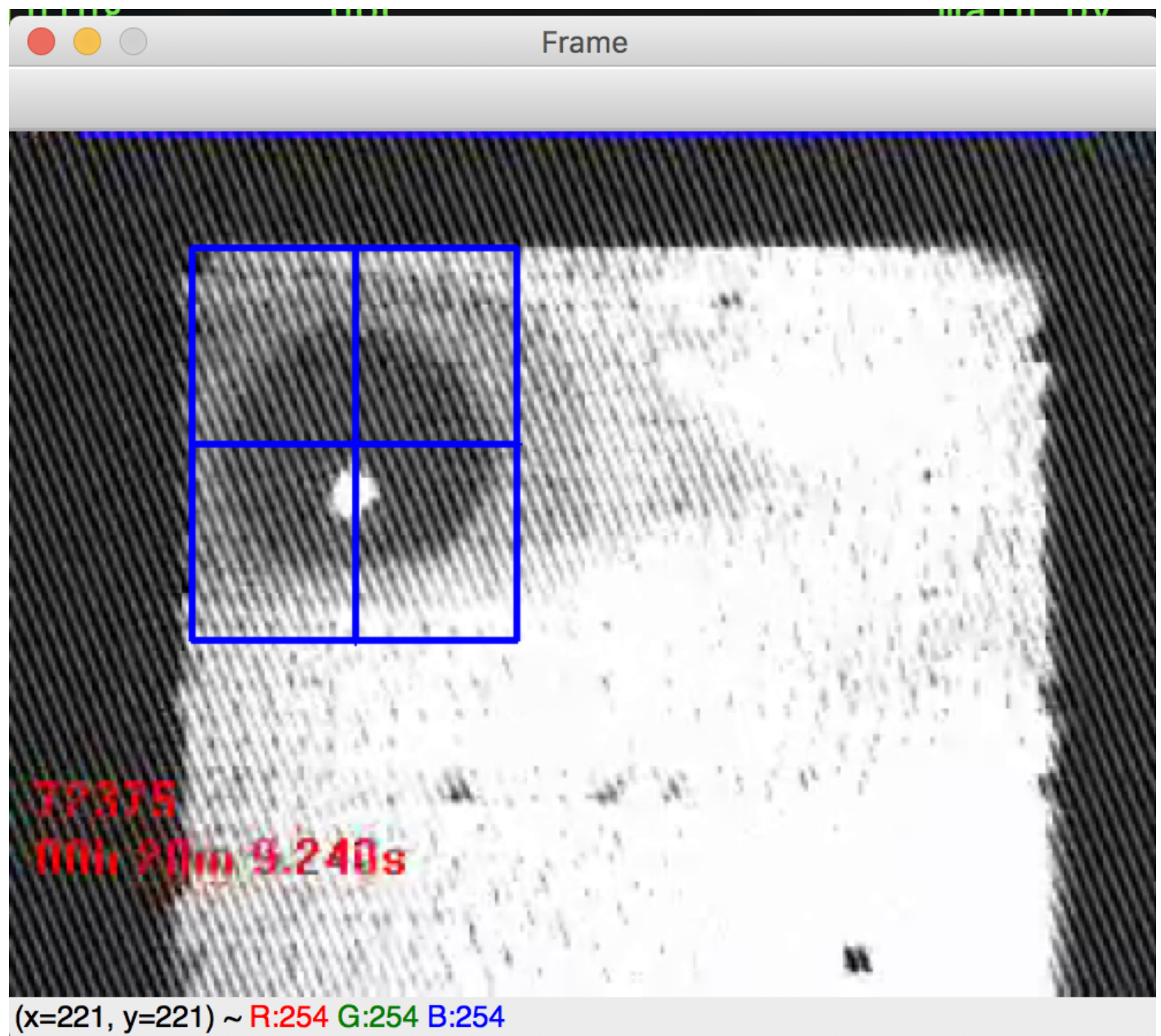
- Analyzing the data file that contains `cue(center)`, `vgs(picture cue)`, `dly(center)`, and `mgs(empty cue)`.
- Implementing the machine learning filter to start developing the tracking system.
- Plotting and analyzing the data retrieved from the machine learning algorithm.

Objectives for next week

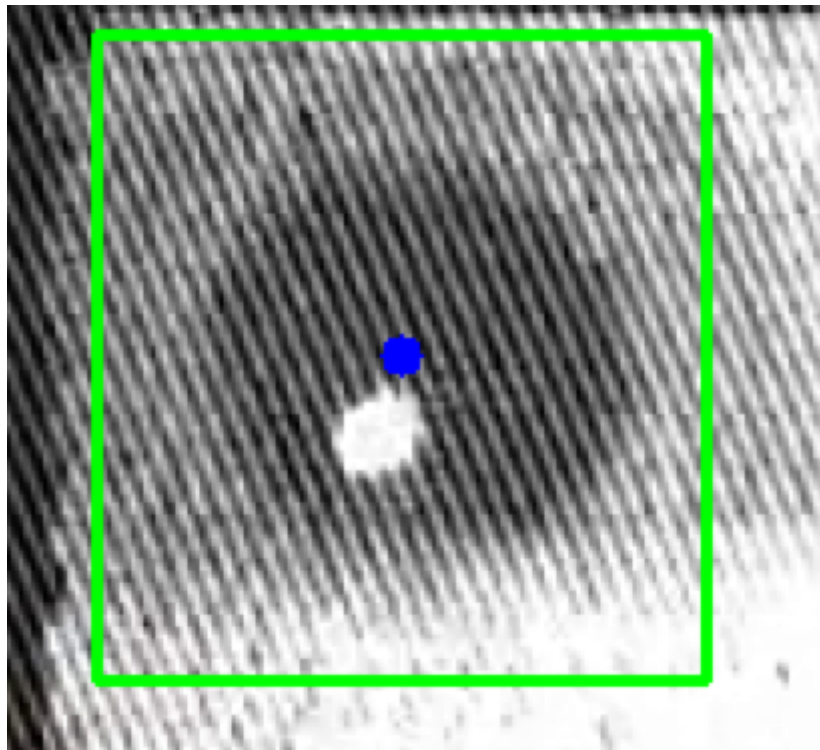
- Link the Machine learning filter to the user interface,
- Implement the filter for potential improvements.
- explore Hough Transform.

Results Demo

Blue box shown in the image is user selected space used for KCF analysis.



Green box shown in the image is auto-tracker generated by machine learning algorithm(KCF filter) that tracks the pupil location.

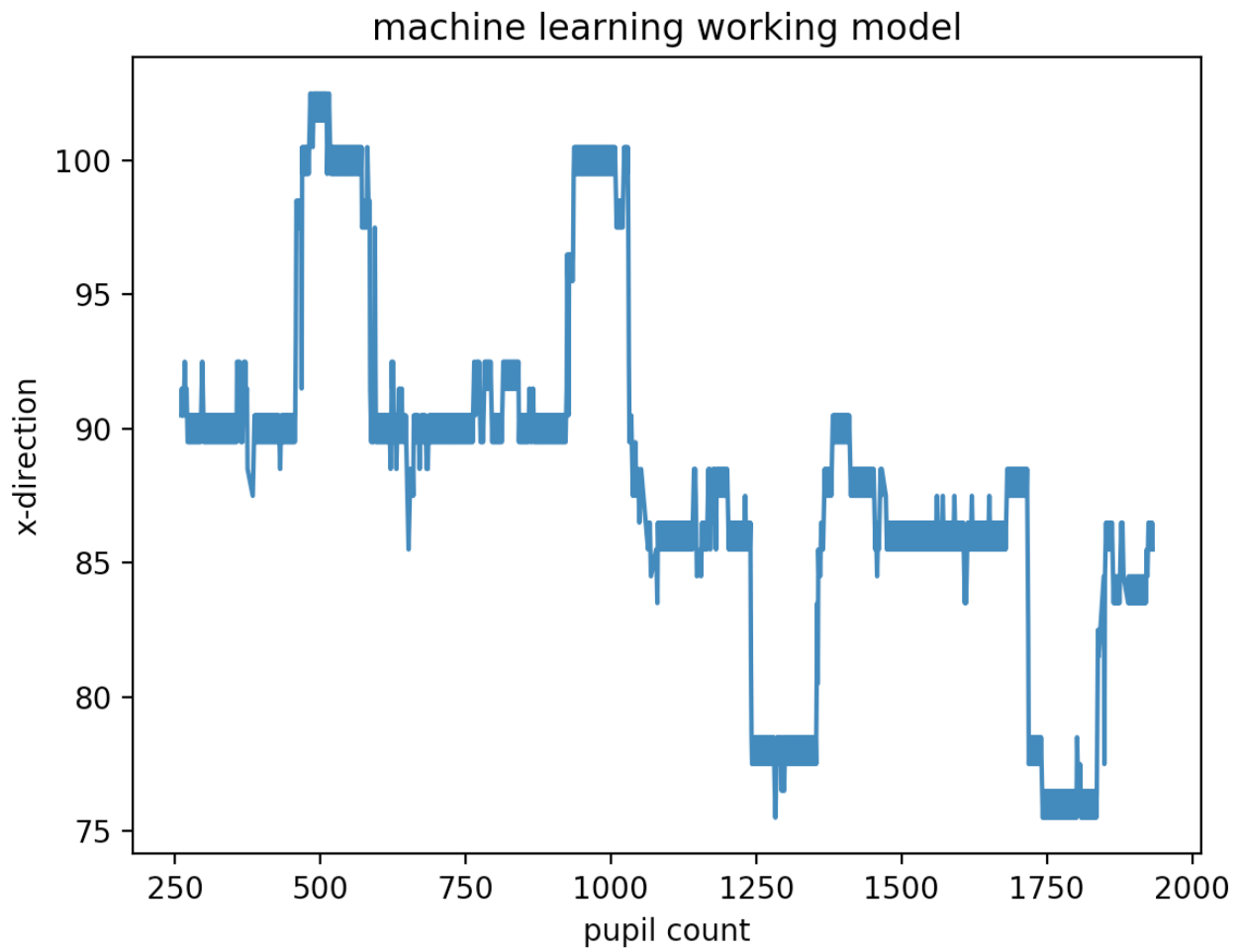


Annotation on the bottom left that shows speed(fps), blinking, and tracker type.



The tracking result.

Figure 1



x=1144.68

y=92.3268

Explanation

Commands to run the tracker

```
-cd Machine_Learning/  
-python3 object_tracker.py --video ../input/run1.mov --tracker kcf  
-Ps: Make sure there is a run1.mov video file inside input directory and all the necessary bags installed.  
(After the program is running)~  
-k: Choose big area.  
-s: Choose tracking area.  
-t: Plot Graph.  
-q: Exit.
```

Graph analysis

-as shown above in the annotation section, the KCF tracker can run up to 51 fps, which reflects its efficiency in pupil tracking. However, as shown in the graph, although the tracker successfully retrieves the pupil location, the precision still requires some refinements.

Conclusion

-Even though machine algorithm alone cannot get the precise result of pupil location, it could at least narrow down the search area for other more precise but slower algorithm -- Hough Transform and RANSAC -- to boost up the efficiency of the tracker as a whole