Exercise #6 – Extended Lab Optimization and Demo

DUE: AS INDICATED on Canvas

Please provide all code with build and run instructions, PowerPoint (with optional audio presentation), and encoded video demonstrating example use of your CV application.

Exercise #6 Requirements:

1) [20 points] Present your work on Lab #5 and #6 in the form of a video demonstrating the MV/ML features (frame capture and re-encode to MPEG) and as a PowerPoint (providing technical details) – if neither apply, then provide a detailed write-up that includes screen captures and example images. Explain challenges you have overcome or still face, focused on how well your application works in terms of interactive performance or if applicable in terms of whether it is accurate (can in fact detect and recognize objects) and whether it can run continuously in real-time. This should be a good technical overview of how you used OpenCV, the hardware used and how you designed your application.

2) [30 points] Analyze the accuracy and the continuous processing capability of your system in terms of frame rates, correct/incorrect object segmentation and recognition, and or improvement in stability or estimation of depth (commonly used methods include ROC, PR, and F-measure). The accuracy analysis will vary substantially based on the project you selected, but you should refer back to your Lab #5 goals and test how well you meet them as well as documenting frame rate, resolution, and duration of operation limits and capabilities.

3) [50 points] Turn in your completed application so it can be built on Ubuntu Linux 12.04 LTS with makefile so your project can easily be rebuilt by someone else on the same platform you used. If you application uses a camera, it should work the ASUS Xtion camera or a Logitech C200/C270 (or equivalent) and Ubuntu Linux 12.04 LTS or common Linux distributions found on R-Pi or NVIDIA Tegra. Ideally you should demonstrate this works by capturing transformed frames to disk (e.g. 3000 for 100 seconds of 30Hz video – can be JPEG) and then encoding them into an MPEG (which can satisfy requirements for technical reporting as well – above), but if this is not feasible, make sure I can build, run and use your application with the ASUS Xtion or Logitech cameras on Ubuntu Linux or embedded Linux.

Grading Rubric

[20 points] Video presentation or annotated video showing results (also referenced in report):

[10 pts] Annotated and processed video showing results for MV/ML work (e.g. bounding boxes, tracking overlay of centroid for targets of interest, etc.)____

[10 pts] Written or video technical overview and summary of challenges

____

[30 points] Accuracy analysis and verification methods, code, and discussion (also referenced in report):

[10 pts] Test case and verification methods and results

____

[10 pts] Performance analysis in terms of frame rates, run times, throughput, latency, etc. and either embedded performance or analysis and discussion of how to embed given challenge video performance

____

[10 pts] Segmentation, detection, tracking, and/or classification/recognition performance in terms of ROC, PR, or F-measure or similar TP, FP, TN, FN statistical analysis

____

4) [50 points] Write a report that includes [16 total pages including cover page, but not appendices or code]. The report should be composed of results from your work in Lab Exercise #5 and work above, but documented in complete form as follows:

[1 page] **Cover Page (list all group members clearly)**

[1 paragraph] **Introduction**

[1 page] **Functional (capability) Requirements**

[1 page] **Machine Vision and Machine Learning Requirements**

[N pages] **Functional Design Overview and Diagrams**

[N pages] **MV/ML Analysis and Design with TP, FP, TN, FN Diagrams such as ROC, PR, F-measure, both measured and expected based upon theory; Also, you should have some embedded performance analysis if you embed on Jetson or Raspberry Pi in terms of CPU, I/O, memory use and processing latency**

[N pages] **Proof-of-Concept with Example Output and Tests Completed** - **(Must contain final annotated MPEG)**

[1 paragraph] **Conclusion**

[1 page] **Formal References (and Attributions to Anyone who helped not on the team)**

[N pages] **Appendices with results, code and supporting material** (to stay in page bounds)