Project Report

Kelvin (Chang Chun Hei)

This project mainly aims to improve the 3-D block building robot by Ho Yin Chau from Hong Kong Polytechic University

The result can be seen by running main.cmd. Note that running will take a while for cropping all the images. According to instructions created by Ho Yin Chau, main.cmd does everything from 2.3 to 3.2

First, new python script image\_cropping\_for\_training.py will cut images of blocks captured from robot hand to a bounding box of the face being detected. Note: Images after taken will have to be cropped via previous image\_crop.py script before used.

image\_cropping\_for\_training.py takes all input images from Images file, inside which there is a folder for each different face of each block with respective images. (if necessary, each orientation could be a separate folder) For testing sake, bounding boxes will be saved in “python\_scripts/TEMP\_DO\_NOT\_TOUCH/photos with bounding box” results used in next part will be saved in “python\_scripts/TEMP\_DO\_NOT\_TOUCH/photos after cutting and resizing” (Currently, this process takes ~ 3-10 seconds per photo, may be different depending on the computer)

Here is the algorithm for actual bounding box finding:

- The algorithm starts at a particular parameter for Canny Edge detection

- using the edge detection tool, edges are found and then using Hough on the edge detected image, we find lines on the image

- Then, all parallel pairs of lines are found

- each 2 pairs of parallel lines are tested for the following:

- if they intersect (or are close enough on their ends)

- if they are perpendicular

- if the side length of the four sided shaped formed from the 4 lines seems to be appropriate

- if all the above is true for some pair, then that pair will be accepted

- the bounding rectangle will then be calculated and returned

- else, if there isn’t any pair satisfying the above, the algorithm will then rerun itself from the start at a lower parameter, so and so on.

Next, for training the model, training\_main.py will take the photos in “python\_scripts/TEMP\_DO\_NOT\_TOUCH/photos after cutting and resizing” and train with model.fit, a different evaluation function is used and a callback has been implemented to stop the training after there is no apparent change. Testing code has been implemented and is in comments of the code. Models will then be saved in TEMP\_DO\_NOT\_TOUCH/models with the current date and time as its title.

Please note that this is a completely different training process from before, it is much more efficient but slightly less accurate (about a minute vs hour, for ~ 90% accuracy), it is suggested that the evaluation function should be changed for better accuracy.

Other scripts implemented:

- Canny\_Edge\_parameter\_setting.py is used to see the effect of different parameters on photos of Canny edge detection.

- All other scripts are used in main\_prev.cmd, in attempt to recreate the behaviour of the project by Ho Yin Chau. For exact behaviour please see his report.

Suggested improvement:

- Evaluation function of training

- Canny parameters settings may be changed for quicker performance (currently looping a lot)

- Connecting work done to robot

- refine robot’s cube detection algorithm (not face)