Quick Fix to Image Processing Errors

The image processing script can sometimes run into errors due to changes in light conditions, changes in the shade of the objects or any other physical quantity.

The below mentioned fix is only if the program doesn’t run after recalibrating multiple times.

The below mentioned changes must be enacted only if the Image Processing error keeps occurring as the below changes make changes at a fundamental level of the code.

# Notes:

1. Opening a default terminal: Press Alt+ Ctrl+T
2. Opening a terminal in the said folder: right click anywhere inside the folder and click on ‘Open in terminal’.
3. Opening the file explorer : Press Alt+ Ctrl+F
4. Application folder: On desktop, “**Gui-Voice**” folder exists inside a folder named “**Brabo**”

# Step 1:

Navigate to the location of the application folder and go into the **Scripts** folder.

# Step 2:

After going to the Scripts folder open the **pops.py** file. The remaining procedure involves changing the file contents. Taking a back up of this file is strongly recommended.

# Step 3:

In **pops.py**  find the following lines **#cv2.imshow()** and **#cv2.waitKey(0) .** The Lines are commented. Uncomment all the instances of the said lines by removing the **#** symbol preceding the statements. Save the file (Ctrl+S).

# Step 4:

Now open another instance of the application folder. There open a terminal (right click and select open terminal) and type the following command. **python controller.py -t 1 .** If all the cell elements (camera, light and robot) are connected, the command will execute only the image processing part of the program.

# Step 5:

The first image will show the image taken by the camera. By using the scroll gesture on the track pad, zoom into all the objects and note their RGB values.

# Step 6:

Go to the folder **Scripts/dbdp** where you will find a database named calib.db. Open a terminal here and execute the command **sqlite3.** This opens the sqlite3 program within the terminal.

# Step 7:

In the sqlite3 terminal type the following commands **.open calib.db** after typing this command press enter. No visible changes will be observed in the terminal, now after doing the same type **SELECT \* FROM calX; .** This command will display a list of the database entries.

# Step 8:

Compare the earlier noted RGB values to the Values in the database. If the values are different but within a range of +/- 10 Proceed to **Step 10**  else **Step 9**.

# Step 9:

Recalibrate the system using the recalibrate procedure on the GUI. And try again.

# Step 10:

If the range is within 10 then proceed to reopen the **pops.py** file

# Step 11:

In this file within the first few lines find the following variable declarations

**calibwidth** #NOTE: WIDTH FOR INDEX COLOR

**boundwidth** #NOTE: WIDTH FOR COLORS IN IMAGE-PROCESSING

**calibthreshold** #NOTE: THRESHOLD VALUE FOR INDEX COLOR

**boundthreshold** #NOTE: THRESHOLD VALUE FOR OTHER COLOR

the calib prefixed variables are for the **index color** (purple) and the bound prefixed variables are for **other colors**.

# Step 12:

Modify the width value in unit increments (Final value shouldn’t be more than 40) and the threshold value in unit decrements ( Final value shouldn’t be less than 5), and every time retry the entire process untill all objects are detected by following **Step 4**.

NOTE: If the problem still persists then use this manual calibrate document and try this procedure again.