All Photographs and Video Taken By and Used with the Permission of Jonathan Clingan

I Robot

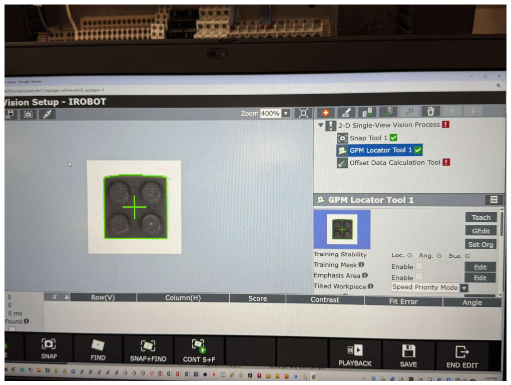
Evalynn Clingan

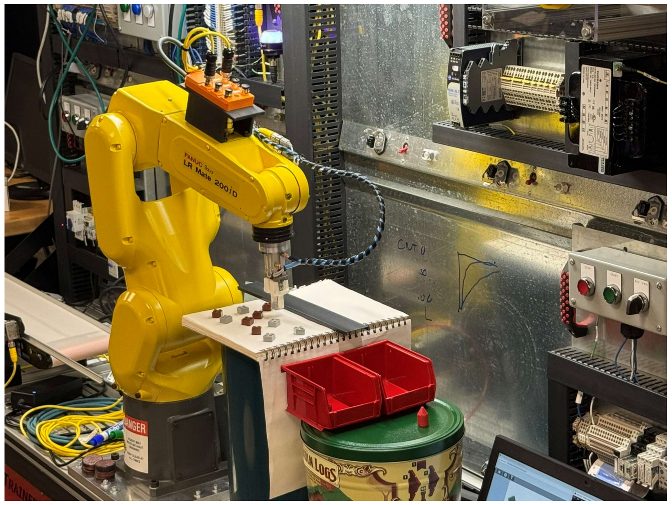
Purpose

To determine how accurate A robot with a grey scale camera is at sorting colored bricks.

To complete this task, the robot will be programmed to find and then sort the bricks into two different bins.

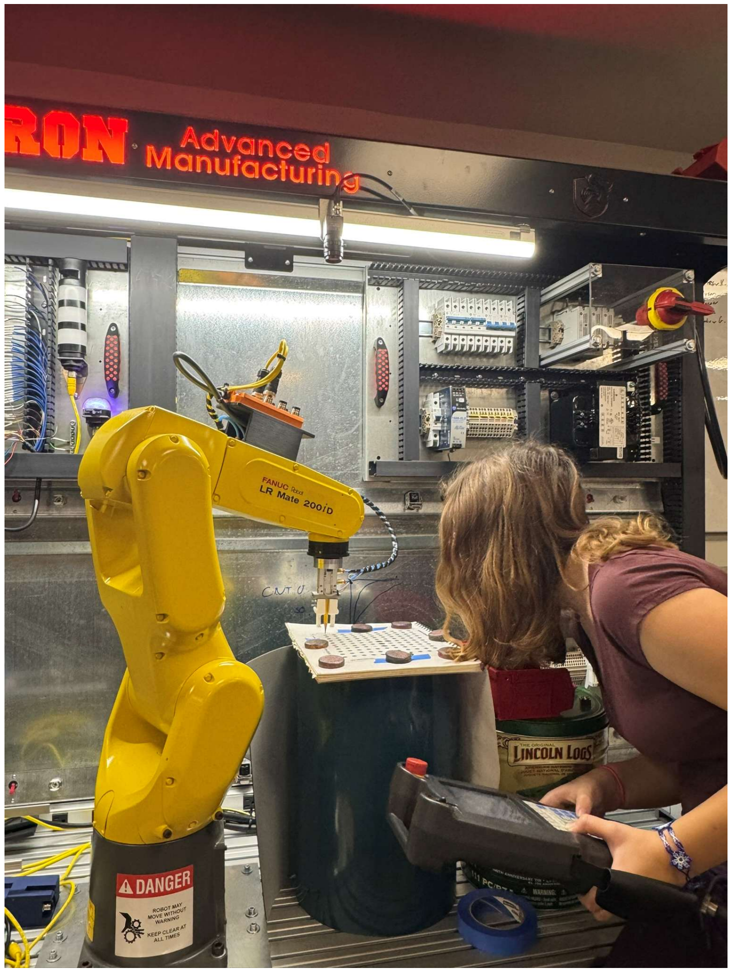
The experiment will use randomly place bricks all over the field of view for the robot to find.





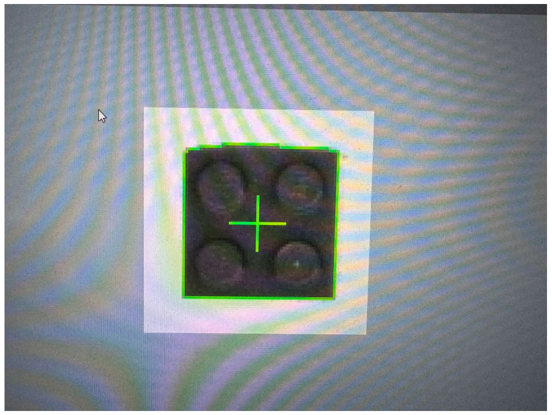
Hypothesis

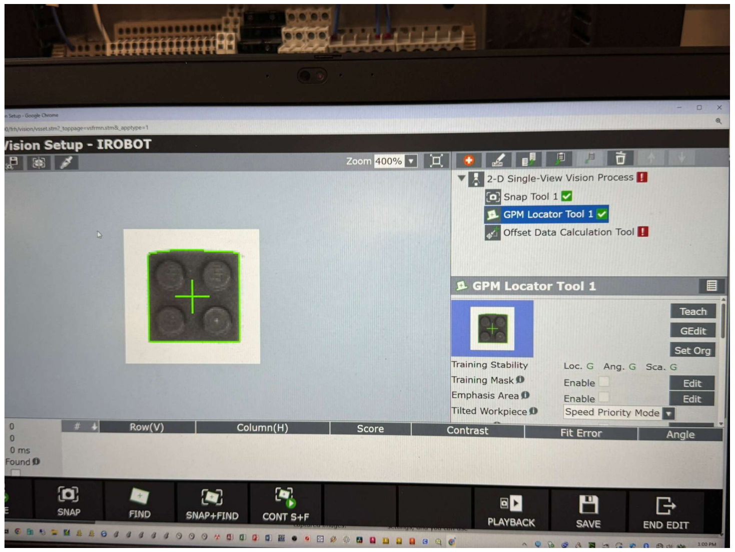
My hypothesis is that the robot will be able to correctly sort the bricks 70% of the time, the reason I think is robot's aren't accurate their repeatable doing the same thing over and over.



Set up

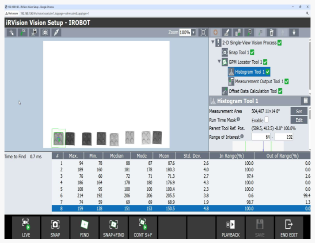
To set up for this experiment I spent nearly a whole day messing with the lights, try to eliminate any shadows as they could effect the robots ability to see clearly. I had to do this as even a slight shadow could throw the color off enough that a grey could be identified as a brown. Then I raised everything thing up on large bins so the robot could grab it easier.





Procedure

- The first step to do was to get the tool frame and user frame set up. To set up the tool frame was use to tell the robot where the gripper were and the user frame to tell it where the board it was working on was. I then started to work on the program. The first 2 lines of code told robot to use the taught user and tool frame. Then I taught it the master point which was a brick in the center of the board, along with the pounce point 50 millimeter up from the master point. Next was the home and drop of positions, home was where the robot would start and end the program and their were 2 drop off points for the 2 different colors of bricks.
- Stating on the camera the first thing I did was measure how far the camera was away from board. Next I used the vision camera calibration grid and I took a picture for the robot to see the dots on the grid however some of the dots would be blurred so I had to manually delete then so it would calibrate properly. After that I had to measure how tall the Lego pieces were: 11.37 mm. Next I taught the robot to find the bricks by taking a picture of one of them, I then added a histogram tool to the camera that record things such as the mean, median, and mode of the bricks. I then added a line in the program that would make it choose a random brick to pick up and sort, I also added a line of code if the robot saw a brick 125 or higher with would go into the grey bin if not the brown bin.


iRVision Vision Setup - IROBOT

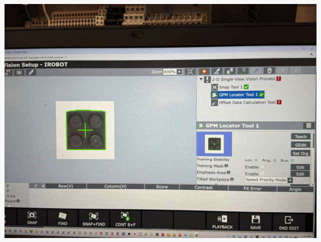
- 2-D Single-View Vision Process ✓
  - Snap Tool 1 ✓
  - GPM Locator Tool 1 ✓
    - Histogram Tool 1 ✓
    - Measurement Output Tool 1 ✓
  - Offset Data Calculation Tool ✓

Histogram Tool 1

Measurement Area	504,407 11×14 0°
Run-Time Mask	Enable
Parent Tool Ref. Pos.	(509.5, 412.5) -0.0° 100.0%
Range of Interest	64 - 192

Time to Find 0.7 ms

#	Max.	Min.	Median	Mode	Mean	Std. Dev.	In Range(%)	Out of Range(%)
1	94	78	88	87	87.6	2.6	100.0	0.0
2	189	160	181	178	180.3	4.0	100.0	0.0
3	76	60	72	71	71.3	2.7	97.4	2.6
4	186	164	178	180	176.9	4.3	100.0	0.0
5	108	95	100	100	100.4	2.3	100.0	0.0
6	214	192	206	206	205.5	3.8	0.6	99.4
7	74	59	69	69	68.9	1.9	98.7	1.3
8	159	128	151	153	150.5	4.8	100.0	0.0

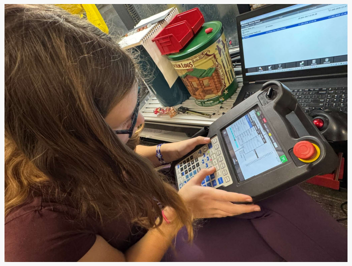

Vision Setup - IROBOT

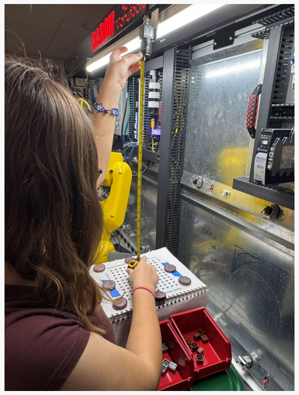
- 2-D Single-View Vision Process
  - Snap Tool 1 ✓
  - GPM Locator Tool 1 ✓
  - Offset Data Calculation Tool

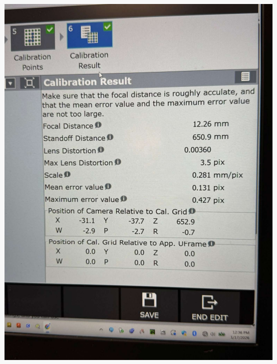
GPM Locator Tool 1

Training Stability	Loc. G Ang. G Sca. G
Training Mask	Enable
Emphasis Area	Enable
Tilted Workpiece	Speed Priority Mode

#	Row(V)	Column(H)	Score	Contrast	Fit Error	Angle







Calibration Result

Make sure that the focal distance is roughly acculate, and that the mean error value and the maximum error value are not too large.

Focal Distance	12.26 mm
Standoff Distance	650.9 mm
Lens Distortion	0.00360
Max Lens Distortion	3.5 pix
Scale	0.281 mm/pix
Mean error value	0.131 pix
Maximum error value	0.427 pix

Position of Camera Relative to Cal. Grid

X	-31.1	Y	-37.7	Z	652.9
W	-2.9	P	-2.7	R	-0.7

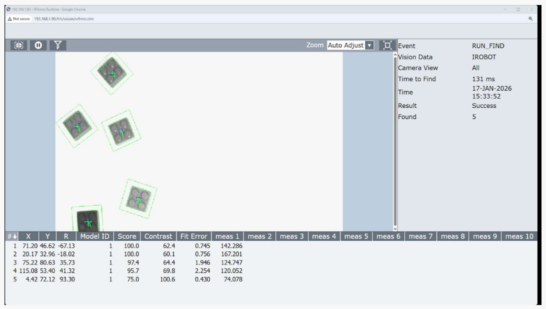
Position of Cal. Grid Relative to App. UFrame

X	0.0	Y	0.0	Z	0.0
W	0.0	P	0.0	R	0.0

Procedure Continued

For the first run of the programs I placed down 12 brown bricks randomly around the board coming up with new ways to put them every time running the program after putting them down. I did the same with the grey bricks putting 12 down randomly and then running the program and taking notes.

When running the final experiment I place 6 brown and grey bricks randomly on the board for the robot to sort, then recording the results after each of the 5 runs.

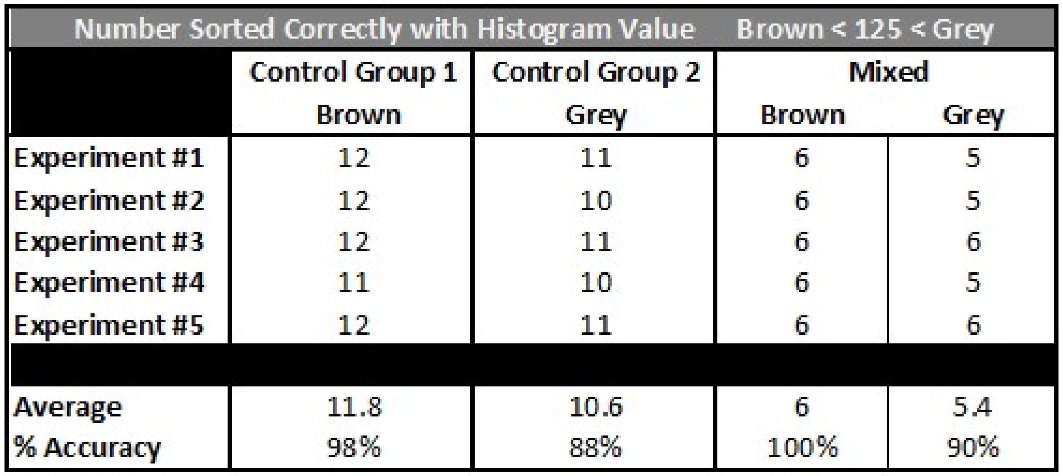


192.168.1.90 - iRVision Runtime - Google Chrome
Not secure 192.168.1.90/frh/vision/vrfrmn.stm
Zoom Auto Adjust

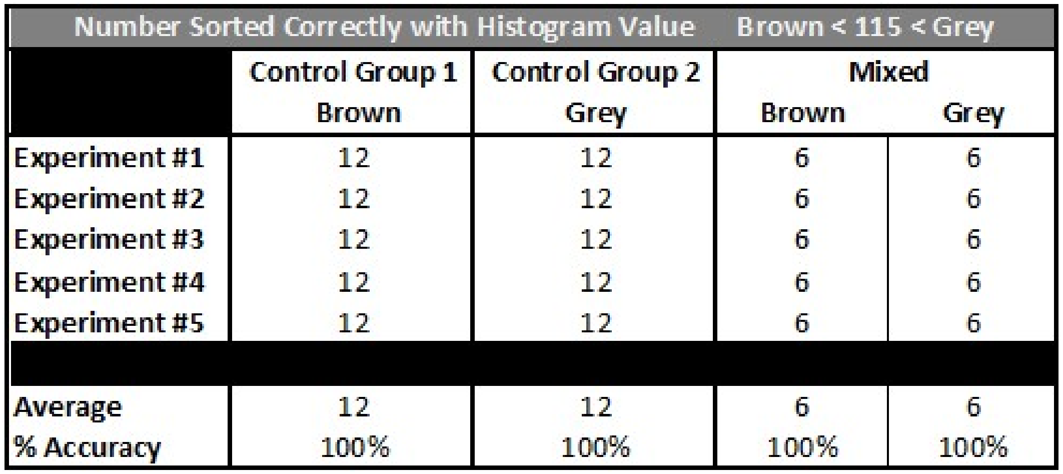
Event	RUN_FIND
Vision Data	IROBOT
Camera View	All
Time to Find	131 ms
Time	17-JAN-2026 15:33:52
Result	Success
Found	5

#	X	Y	R	Model ID	Score	Contrast	Fit Error	meas 1	meas 2	meas 3	meas 4	meas 5	meas 6	meas 7	meas 8	meas 9	meas 10
1	71.20	46.62	-67.13	1	100.0	62.4	0.745	142.286									
2	20.17	32.96	-18.02	1	100.0	60.1	0.756	167.201									
3	75.22	80.63	35.73	1	97.4	64.4	1.946	124.747									
4	115.08	53.40	41.32	1	95.7	69.8	2.254	120.052									
5	4.42	72.12	93.30	1	75.0	100.6	0.430	74.078									

Results



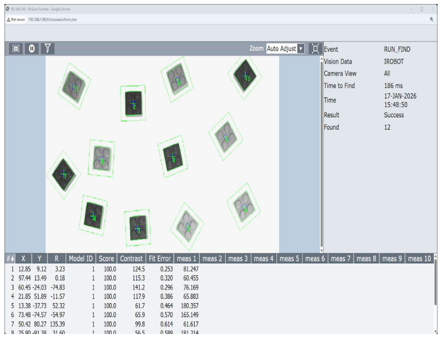
Number Sorted Correctly with Histogram Value			Brown < 125 < Grey	
	Control Group 1	Control Group 2	Mixed	
	Brown	Grey	Brown	Grey
Experiment #1	12	11	6	5
Experiment #2	12	10	6	5
Experiment #3	12	11	6	6
Experiment #4	11	10	6	5
Experiment #5	12	11	6	6
Average	11.8	10.6	6	5.4
% Accuracy	98%	88%	100%	90%



Number Sorted Correctly with Histogram Value			Brown < 115 < Grey	
	Control Group 1	Control Group 2	Mixed	
	Brown	Grey	Brown	Grey
Experiment #1	12	12	6	6
Experiment #2	12	12	6	6
Experiment #3	12	12	6	6
Experiment #4	12	12	6	6
Experiment #5	12	12	6	6
Average	12	12	6	6
% Accuracy	100%	100%	100%	100%

Conclusion

The experiment successfully demonstrated the accuracy of using a gray scale camera to sort color parts with a sorting accuracy of 90%. While it was highly effective at identifying brown bricks, 98% accurate in the control group and 100% accurate in the mixed trials, it consistently struggled with grey bricks, which fell to an accuracy as low as 88% in the control group and 90% in mixed trials. This indicated that the 125 threshold histogram value was too broad, causing the machine to misclassify some grey objects.
Analyzing the results, the robot was able to sort the parts with greater that 70% accuracy as defined in my hypothesis.
I was interested to see if changing the histogram value could improve the results. I modified the threshold to 115 after looking at many results of the Lego bricks and reran the experiments.
Following this adjustment, the robot's performance improved significantly, achieving a perfect 100% accuracy across all five experiments for both color groups and the mixed part trials. These results prove that while the initial setup was functional, fine-tuning the camera's histogram parameters to a specific 115 value threshold was the key to eliminating errors and achieving 100% accuracy.


192.168.1.90 - iRVision Runtime - Google Chrome
Not secure 192.168.1.90/frh/vision/vrfrmn.stm
Zoom
Auto Adjust

Event	RUN_FIND
Vision Data	IROBOT
Camera View	All
Time to Find	186 ms
Time	17-JAN-2026 15:48:50
Result	Success
Found	12

#	X	Y	R	Model ID	Score	Contrast	Fit Error	meas 1	meas 2	meas 3	meas 4	meas 5	meas 6	meas 7	meas 8	meas 9	meas 10
1	12.85	9.12	3.23	1	100.0	124.5	0.253	81.247									
2	97.44	13.49	0.18	1	100.0	115.3	0.320	60.455									
3	60.45	-24.03	-74.83	1	100.0	141.2	0.296	76.169									
4	21.85	51.89	-11.57	1	100.0	117.9	0.386	65.883									
5	13.38	-37.73	52.32	1	100.0	61.7	0.464	180.357									
6	73.48	-74.57	-54.97	1	100.0	65.9	0.570	165.149									
7	50.42	80.27	135.39	1	100.0	99.8	0.614	61.617									
8	25.90	-91.38	31.60	1	100.0	56.5	0.589	181.214									