Activity 4:
FEATURE EXTRACTION FROM IMAGES USING IMAGEJ

Genesis Vertudez – 202003099

App Physics 157 - Computational Analysis and Modeling in Physics

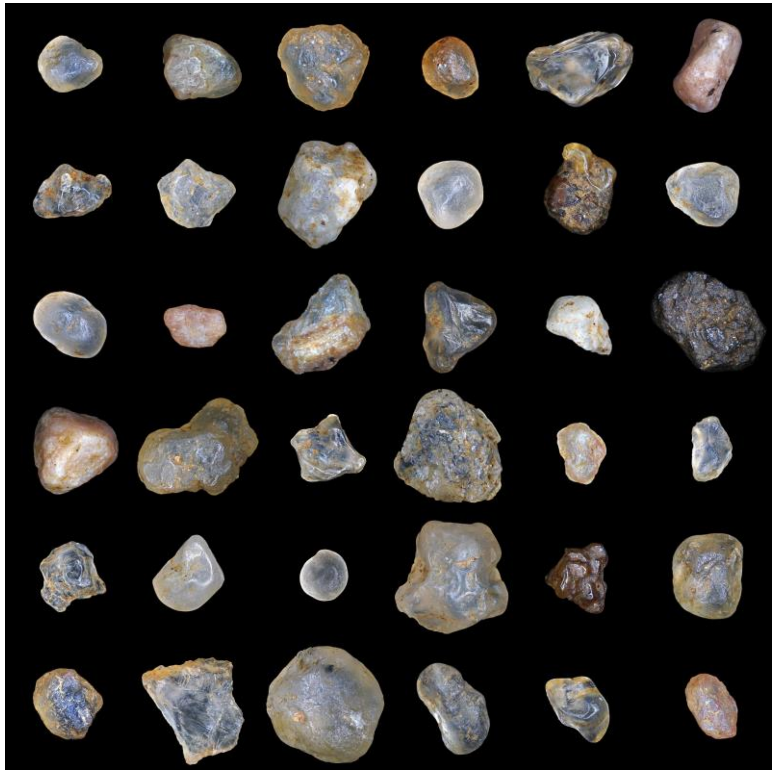
Submitted to Dr. Maricor Soriano; Mx. Rene Principe Jr.

OBJECTIVES

- Use ImageJ to manipulate images and extract their features
- Apply threshold to grayscale histogram to separate objects from background

RESULTS & ANALYSIS

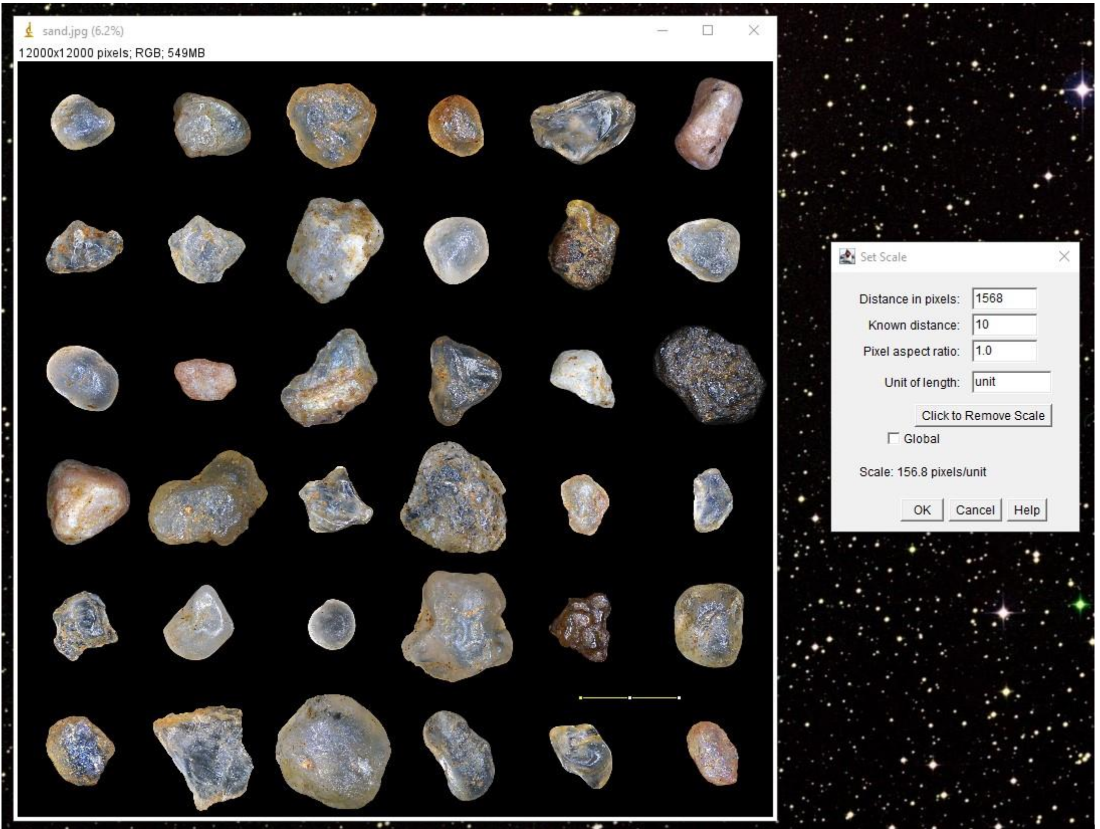
- Image of sand particles



Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Setting scale



sand.jpg (6.2%)

12000x12000 pixels; RGB; 549MB

Set Scale

Distance in pixels: 1568

Known distance: 10

Pixel aspect ratio: 1.0

Unit of length: unit

□ Global

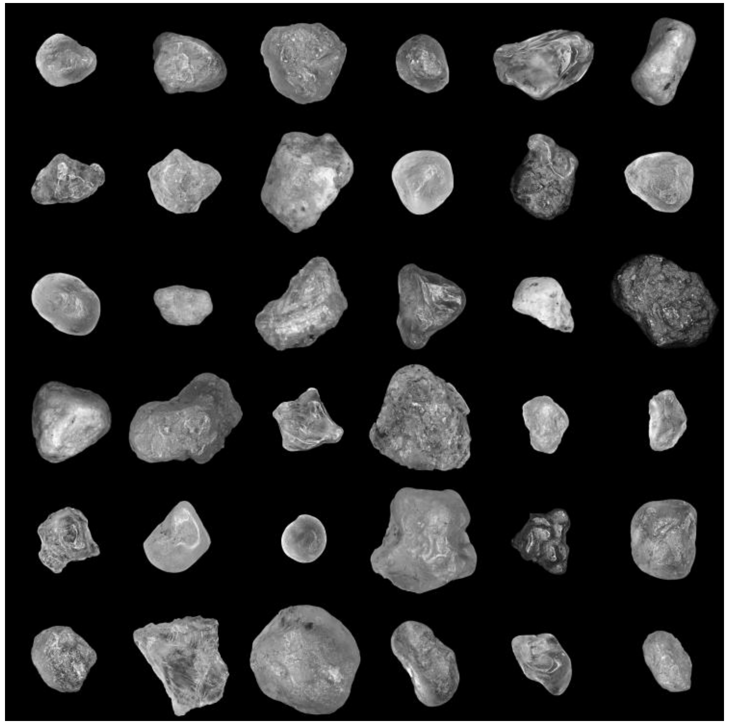
Scale: 156.8 pixels/unit

- I just set an arbitrary scale as instructed. I used 10 units for about the same distance in the cell example.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Converting to 8-bit

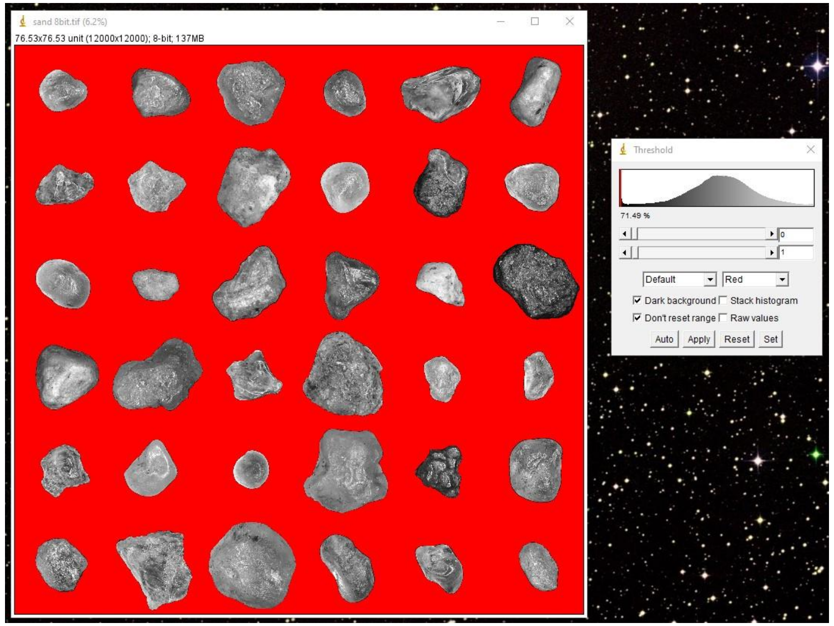


- Applying threshold only works on grayscale images, so we convert it first.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Applying threshold and binarizing



sand 8bit.tif (6.2%)

76.53x76.53 unit (12000x12000); 8-bit; 137MB

Threshold

71.49 %

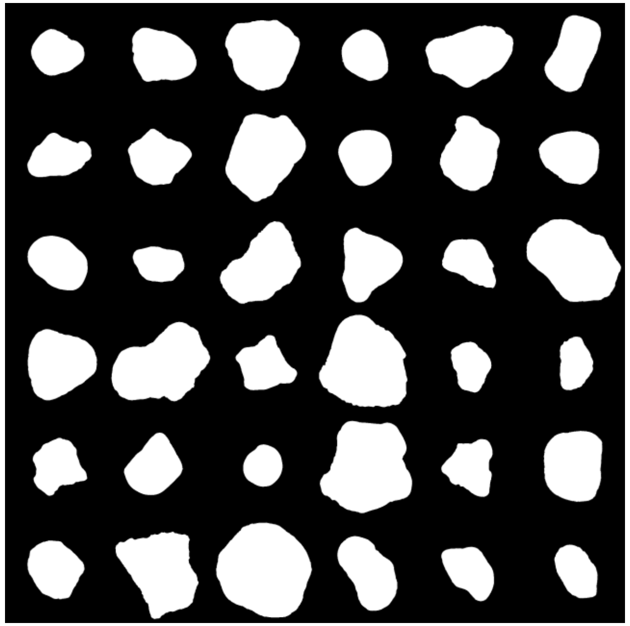
0

1

Default ▾ Red ▾

☑ Dark background □ Stack histogram

☑ Don't reset range □ Raw values

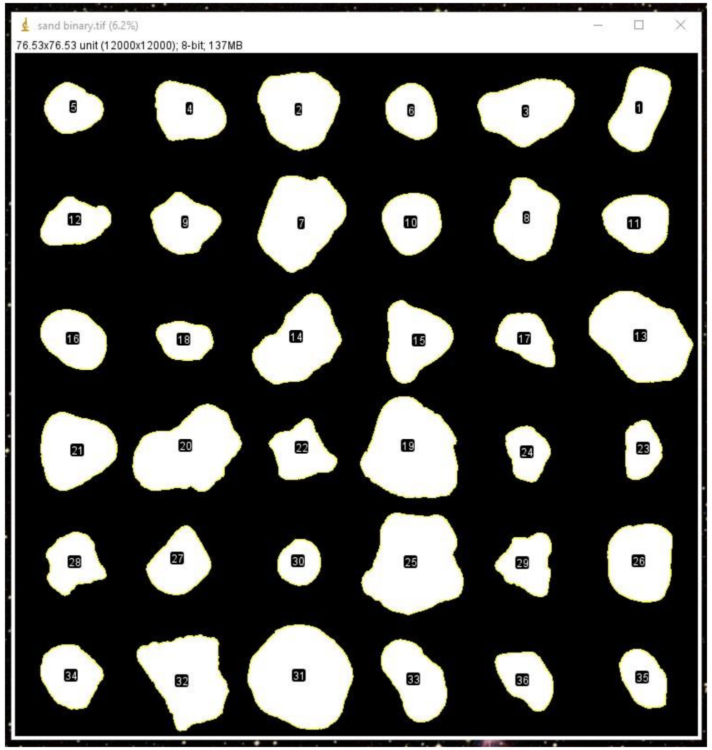


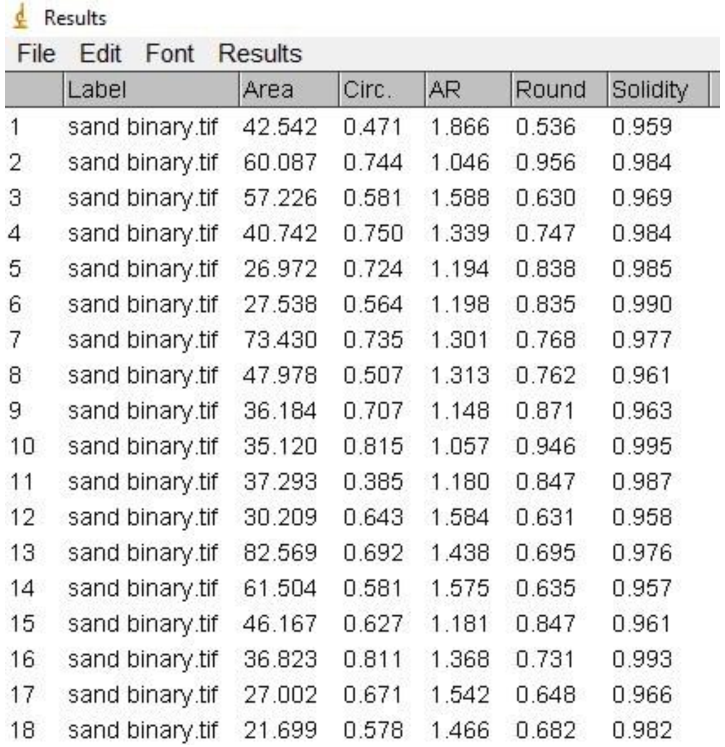
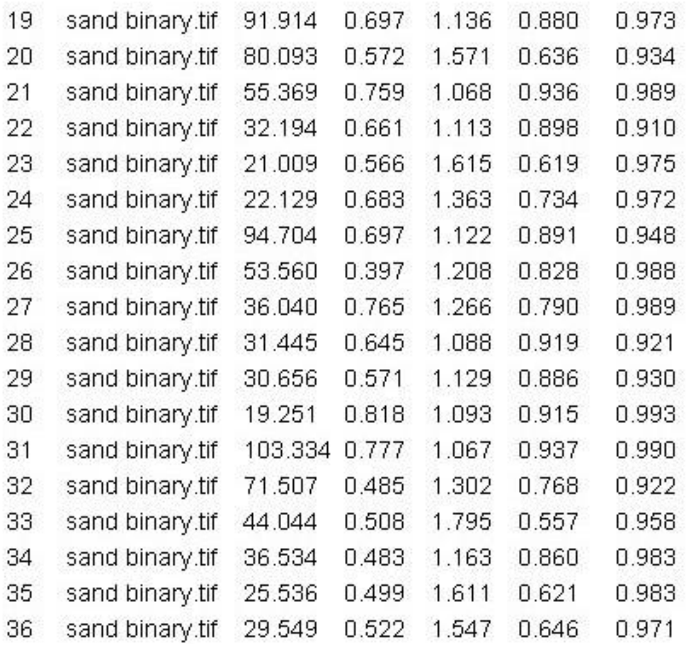
- Since the background is black, it is easy to pick it out the histogram.
- The image is then binarized to make sure that there are only two values for easier extraction of geometric features.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Analyze particles results



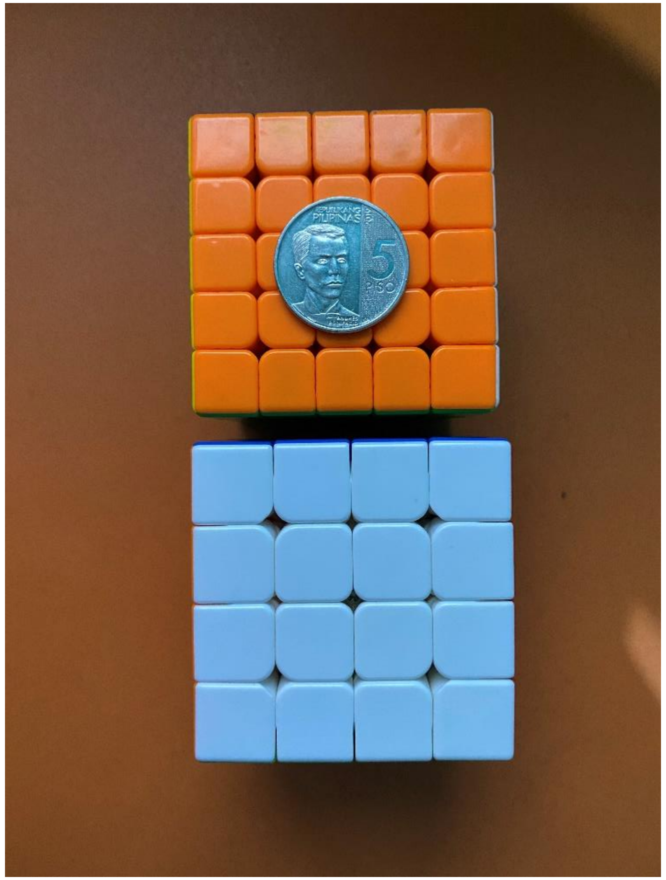


Results
File Edit Font Results

	Label	Area	Circ.	AR	Round	Solidity
1	sand binary.tif	42.542	0.471	1.866	0.536	0.959
2	sand binary.tif	60.087	0.744	1.046	0.956	0.984
3	sand binary.tif	57.226	0.581	1.588	0.630	0.969
4	sand binary.tif	40.742	0.750	1.339	0.747	0.984
5	sand binary.tif	26.972	0.724	1.194	0.838	0.985
6	sand binary.tif	27.538	0.564	1.198	0.835	0.990
7	sand binary.tif	73.430	0.735	1.301	0.768	0.977
8	sand binary.tif	47.978	0.507	1.313	0.762	0.961
9	sand binary.tif	36.184	0.707	1.148	0.871	0.963
10	sand binary.tif	35.120	0.815	1.057	0.946	0.995
11	sand binary.tif	37.293	0.385	1.180	0.847	0.987
12	sand binary.tif	30.209	0.643	1.584	0.631	0.958
13	sand binary.tif	82.569	0.692	1.438	0.695	0.976
14	sand binary.tif	61.504	0.581	1.575	0.635	0.957
15	sand binary.tif	46.167	0.627	1.181	0.847	0.961
16	sand binary.tif	36.823	0.811	1.368	0.731	0.993
17	sand binary.tif	27.002	0.671	1.542	0.648	0.966
18	sand binary.tif	21.699	0.578	1.466	0.682	0.982
19	sand binary.tif	91.914	0.697	1.136	0.880	0.973
20	sand binary.tif	80.093	0.572	1.571	0.636	0.934
21	sand binary.tif	55.369	0.759	1.068	0.936	0.989
22	sand binary.tif	32.194	0.661	1.113	0.898	0.910
23	sand binary.tif	21.009	0.566	1.615	0.619	0.975
24	sand binary.tif	22.129	0.683	1.363	0.734	0.972
25	sand binary.tif	94.704	0.697	1.122	0.891	0.948
26	sand binary.tif	53.560	0.397	1.208	0.828	0.988
27	sand binary.tif	36.040	0.765	1.266	0.790	0.989
28	sand binary.tif	31.445	0.645	1.088	0.919	0.921
29	sand binary.tif	30.656	0.571	1.129	0.886	0.930
30	sand binary.tif	19.251	0.818	1.093	0.915	0.993
31	sand binary.tif	103.334	0.777	1.067	0.937	0.990
32	sand binary.tif	71.507	0.485	1.302	0.768	0.922
33	sand binary.tif	44.044	0.508	1.795	0.557	0.958
34	sand binary.tif	36.534	0.483	1.163	0.860	0.983
35	sand binary.tif	25.536	0.499	1.611	0.621	0.983
36	sand binary.tif	29.549	0.522	1.547	0.646	0.971

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

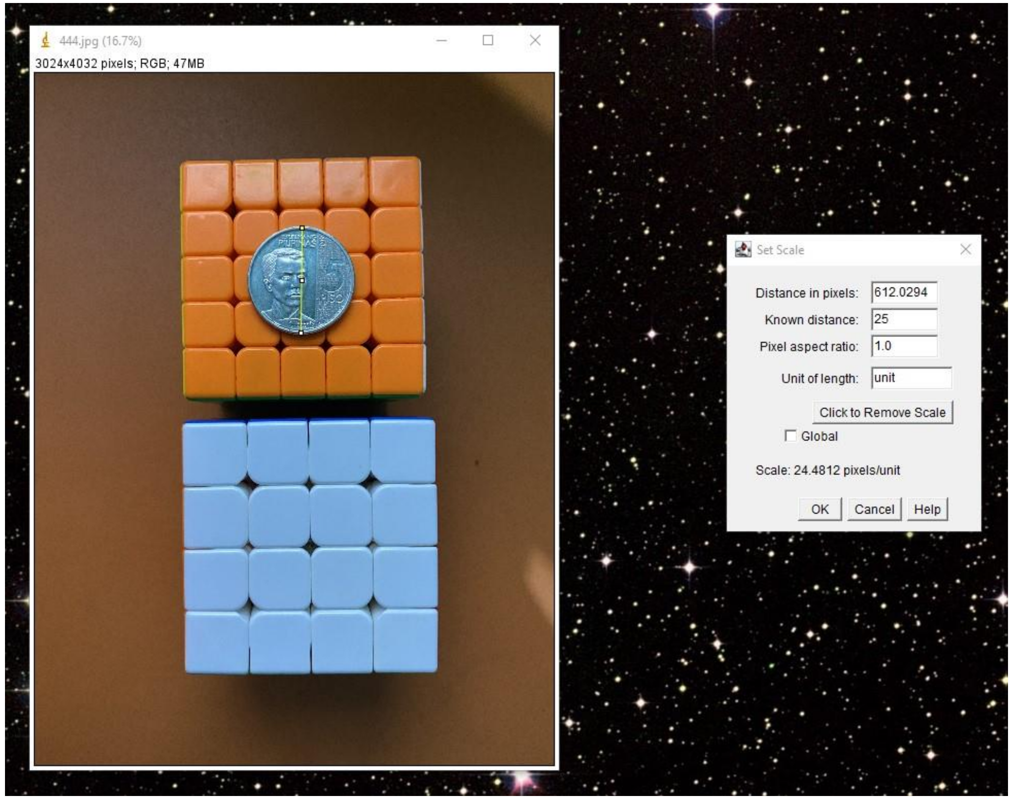
- Image of a 4x4x4 Rubik’s cube



Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Setting scale



444.jpg (16.7%)

3024x4032 pixels; RGB; 47MB

Set Scale

Distance in pixels: 612.0294

Known distance: 25

Pixel aspect ratio: 1.0

Unit of length: unit

□ Global

Scale: 24.4812 pixels/unit

- I used a 5 peso coin on top of a 5x5x5 cube which is about the same height as a 4x4x4 (if I just put it on the table, the height difference affects the scaling) for scaling. It is about 25 mm.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

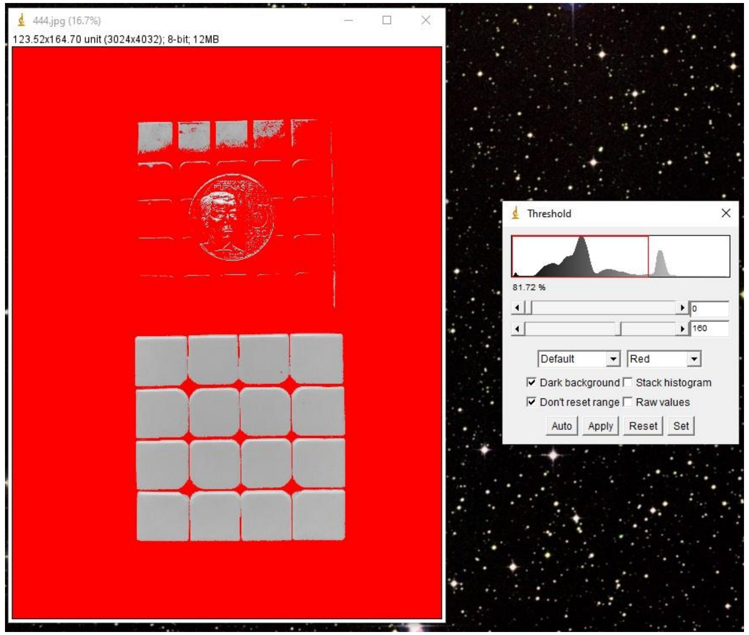
- Converting to 8-bit

- Applying threshold only works on grayscale images, so we convert it first.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Applying threshold and binarizing



444.jpg (16.7%)

123.52x164.70 unit (3024x4032); 8-bit; 12MB

Threshold

81.72 %

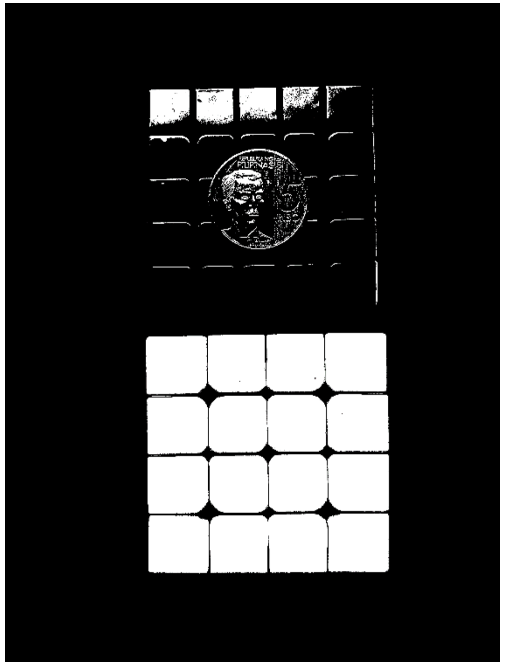
0

160

Default Red

☑ Dark background □ Stack histogram

☑ Don't reset range □ Raw values

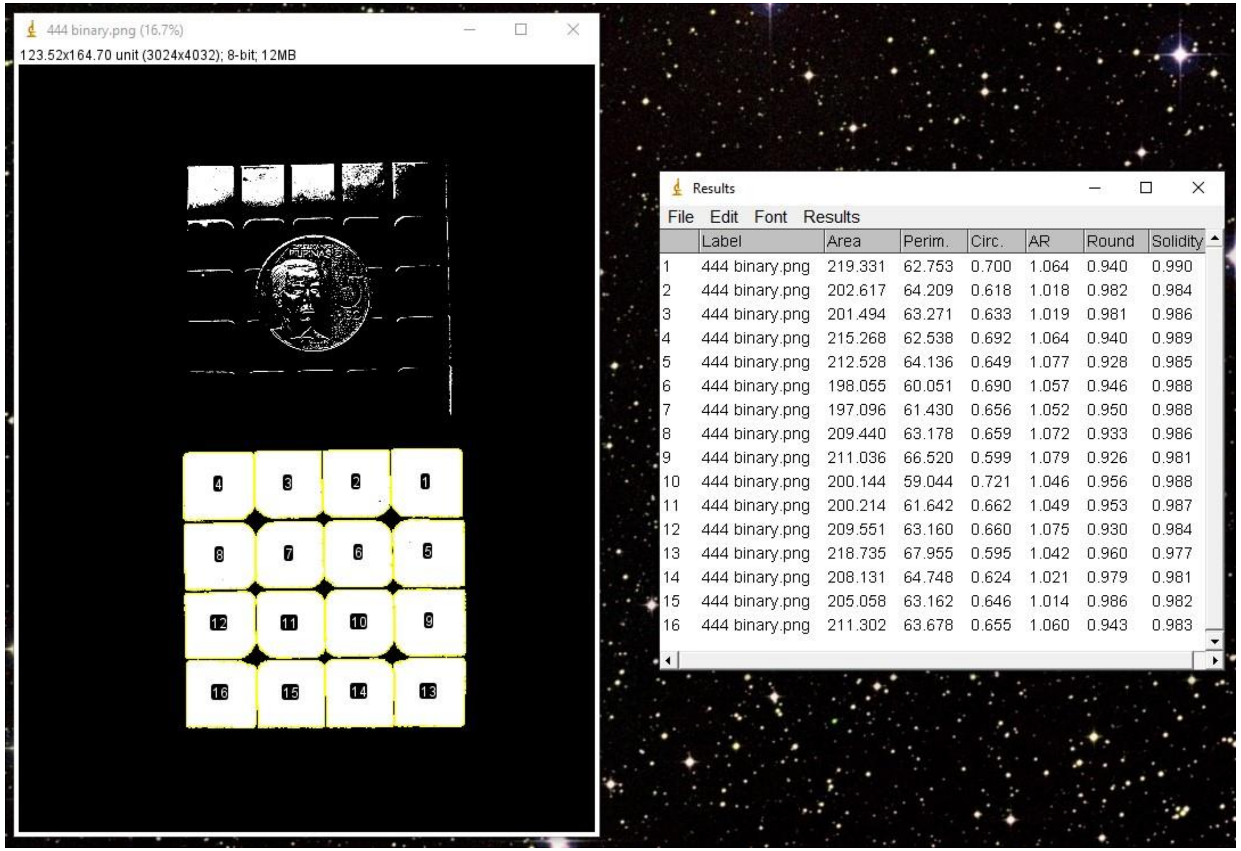


- Since I want the 4x4x4 with white face, it is easy to pick it out from the background and other objects which are significantly darker.
- The image is then binarized to make sure that there are only two values for easier extraction of geometric features.

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Analyze particles results



	Label	Area	Perim.	Circ.	AR	Round	Solidity
1	444 binary.png	219.331	62.753	0.700	1.064	0.940	0.990
2	444 binary.png	202.617	64.209	0.618	1.018	0.982	0.984
3	444 binary.png	201.494	63.271	0.633	1.019	0.981	0.986
4	444 binary.png	215.268	62.538	0.692	1.064	0.940	0.989
5	444 binary.png	212.528	64.136	0.649	1.077	0.928	0.985
6	444 binary.png	198.055	60.051	0.690	1.057	0.946	0.988
7	444 binary.png	197.096	61.430	0.656	1.052	0.950	0.988
8	444 binary.png	209.440	63.178	0.659	1.072	0.933	0.986
9	444 binary.png	211.036	66.520	0.599	1.079	0.926	0.981
10	444 binary.png	200.144	59.044	0.721	1.046	0.956	0.988
11	444 binary.png	200.214	61.642	0.662	1.049	0.953	0.987
12	444 binary.png	209.551	63.160	0.660	1.075	0.930	0.984
13	444 binary.png	218.735	67.955	0.595	1.042	0.960	0.977
14	444 binary.png	208.131	64.748	0.624	1.021	0.979	0.981
15	444 binary.png	205.058	63.162	0.646	1.014	0.986	0.982
16	444 binary.png	211.302	63.678	0.655	1.060	0.943	0.983

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

RESULTS & ANALYSIS

- Actually, I know that my 4x4x4 has a side of about 60 mm, which sets each ‘cubie’ as we call it, to have about 15 mm. This can be confirmed since the extracted area of each ‘cubie’ ranges from about 200-220 mm^2, whose square root is about 14-15 mm.
- Moreover, we see that the perimeter of each ‘cubie’ is about 60 mm, which, when divided by 4 (approximating each ‘cubie’ as a square), is about 15 mm as well.
- Amazing!

Files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%204

REFLECTION

- This activity was straightforward so I was able to follow it easily.
- It is fun to recognize and understand the functions of different options in ImageJ based on our fundamental understandings of image processing. It is cool to know that you can do these manually with coding without the app.
- I am grateful for this no-coding exercise. This is a breather exercise because I have been overwhelmed by coding lately.

SELF-GRADE

- Technical correctness: 35/35
  - I am confident that I understood how to use ImageJ to extract feature from images, and have applied it.
- Quality of presentation: 35/35
  - I have explained each step, and images are clear and concise.
- Self-reflection: 30/30
  - Even though I was slacking in this subject for a while, I managed to get back on track. The topics are really fun and interesting. It helped me understand the ideas behind image tools such as smartphone cameras, Photoshop, etc.
- Initiative: 10/10
  - I used an extra image for extraction.