FEATURE EXTRACTION FROM IMAGES

Mary June A. Ricaña

SN: 2019-01551

Section: WFU-FX

Table of Contents

Introduction

Feature extraction is a dimensionality reduction process where the areas of interest of a photo are taken and analyzed. This reduces the amount of processes an algorithm or a machine will go through in analyzing an image since only those that are essential will be extracted.

For example, in the picture on the right, we want to extract the features defining a cat. Thus, the code or software will detect, say, the geometric features of its body shape, and we say that we have extracted this feature. If we have larger training datasets, we can train a model, as in machine learning, to identify if a photo has a cat in it and where it is located.

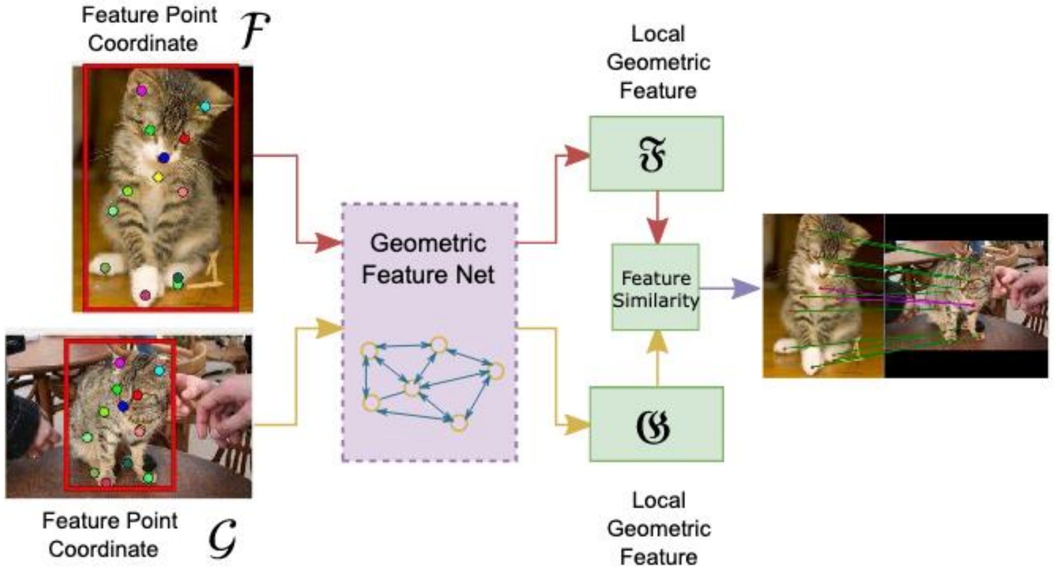


Figure 1. The graph neural network transforms the coordinates into features of the points, so that a simple inference algorithm can successfully do feature matching. [3]

Objectives

In this activity, we will be using feature extraction to identify the individual sand particles in the photo on the right. We will be using the imaging software, ImageJ, to do this. We will also display the features of each sand particle such as their size, particle number, and shape.

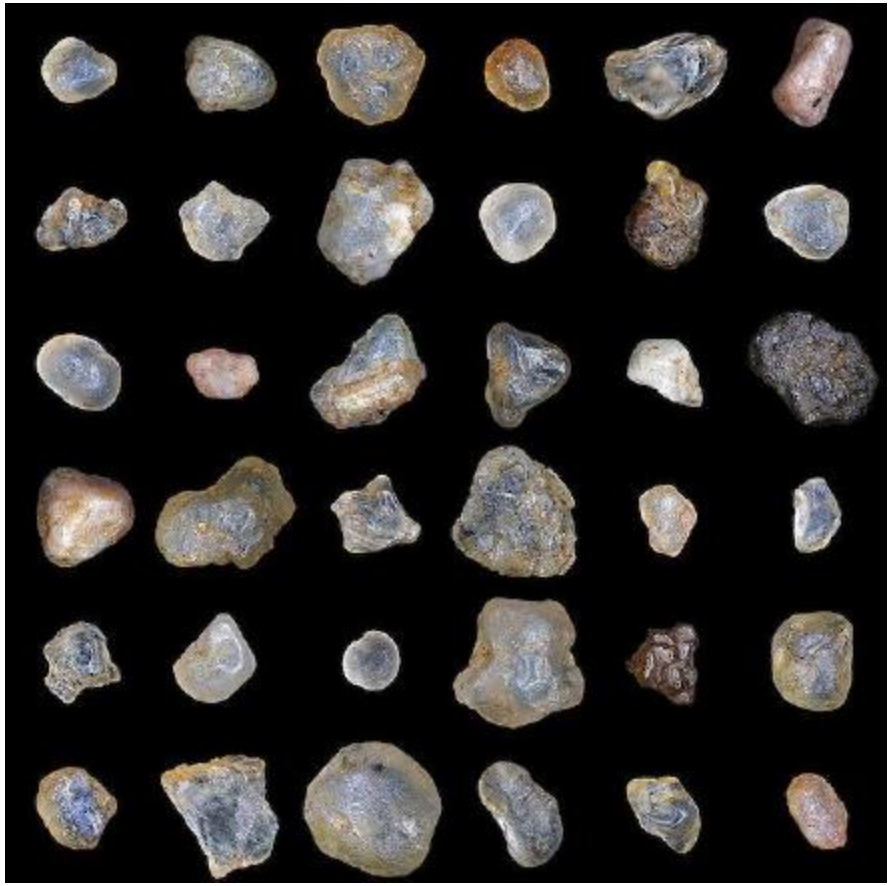


Figure 1. Photo of a collection of sand particles taken from the Feature Extraction Module of AP 157

Setting up a scale for the image

Since there is no scale included in the image, we set our own scale using known knowledge from the internet on the sizes of sand particles.

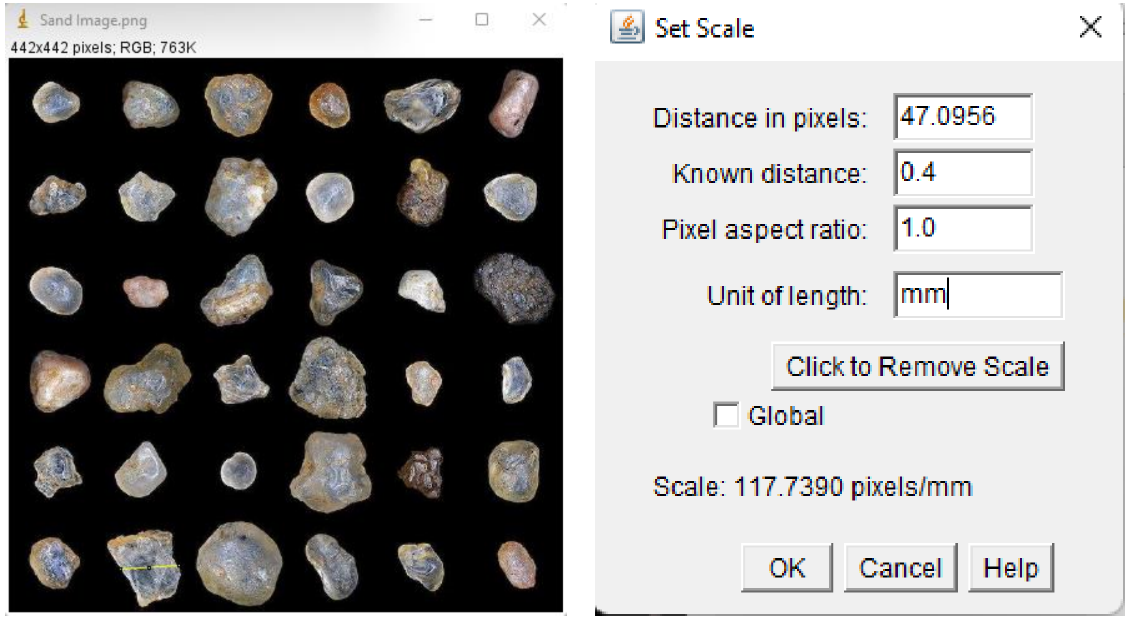


Figure 3. Sand particle image opened through ImageJ and a screenshot of the set scale tab.

Millimeters (mm)		Micrometers (µm)	Phi (ϕ)	Wentworth size class	
	4096		-12.0	Boulder	Gravel
	256		-8.0	Cobble	Gravel
	64		-6.0	Pebble	Gravel
	4		-2.0	Granule	Gravel
	2.00		-1.0	Very coarse sand	Sand
	1.00		0.0	Coarse sand	Sand
1/2	0.50	500	1.0	Medium sand	Sand
1/4	0.25	250	2.0	Fine sand	Sand
1/8	0.125	125	3.0	Very fine sand	Sand
1/16	0.0625	63	4.0	Coarse silt	Silt
1/32	0.031	31	5.0	Medium silt	Silt
1/64	0.0156	15.6	6.0	Fine silt	Silt
1/128	0.0078	7.8	7.0	Very fine silt	Silt
1/256	0.0039	3.9	8.0	Clay	Mud
	0.00006	0.06	14.0		

Figure 4. Wentworth (1922) grain size classification [2]

I made use of the size of a medium to coarse sand grain from the Wentworth classification shown above. I set my distance to be 0.4 mm for the particle on the

Process for feature extraction via ImageJ

After setting up the scale, we adjusted our image to be 8-bit so that it would become grayscale for when we apply thresholding to separate the particles from the background. In the second image, we see that the sand particles are extracted from the background. Next, we convert the thresholded image to binary and then invert it so that we can analyze the black particles.

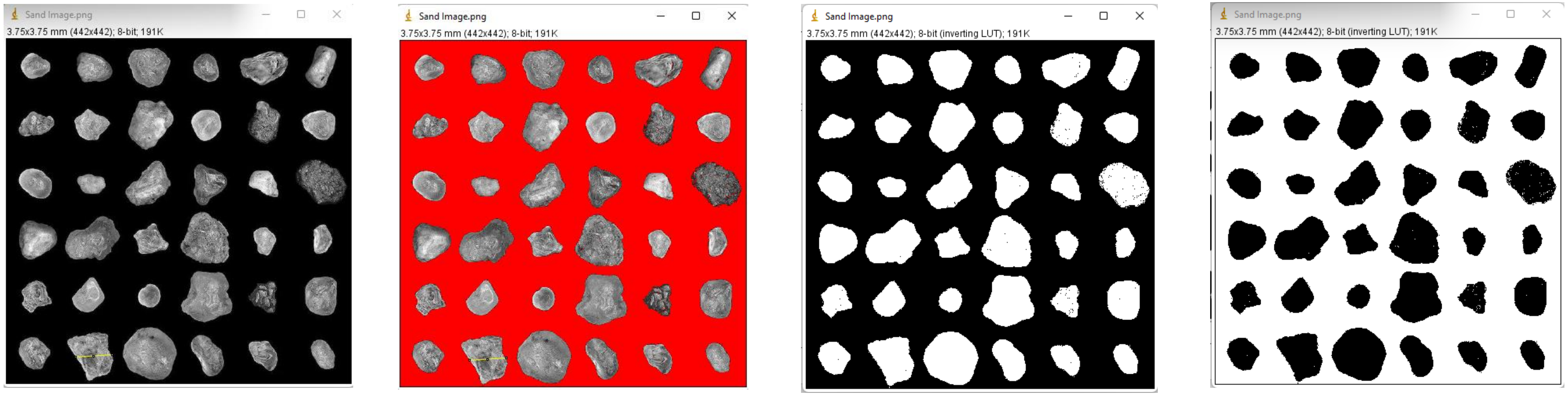
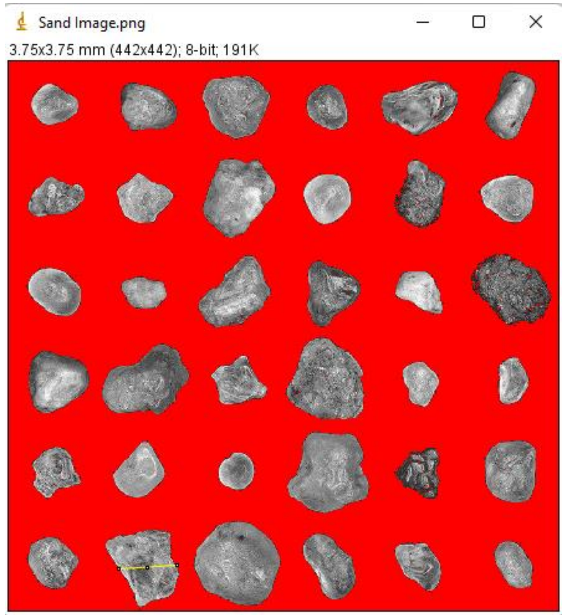


Figure 5. Screenshots of the sand image taken from ImageJ. Process: Grayscale > Thresholded > Binary > Invert

Thresholding

We separate the sand particles from the background by choosing the lower bound to be 0 and the upper bound to be 29. This allows us to choose only the pixels of interest so that it is easier to analyze.

I made sure that the outline of each particle was separated from the background, but there were still areas within the perimeter that were not fully separated.



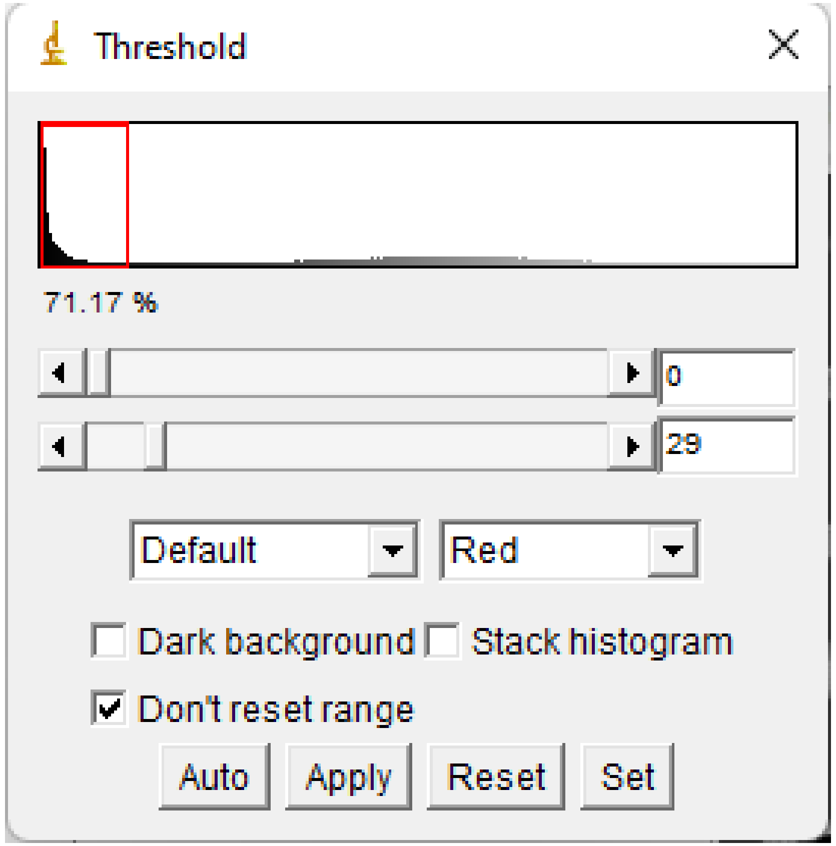


Figure 6. Thresholded photo of the sand particles (left) and a screenshot of its threshold adjustment (above)

Particle analysis

We want ImageJ to find the area, shape descriptors, and centroid of each particle in its analysis. We set the analysis size to be from 4 mm^2 to infinity and its circularity to be from 0.00 to 1.00.

In the figure on the right, we see that the software was able to extract the sand particles from the image and number them from one to 36 in red. We also see the outlines of each particle in black.

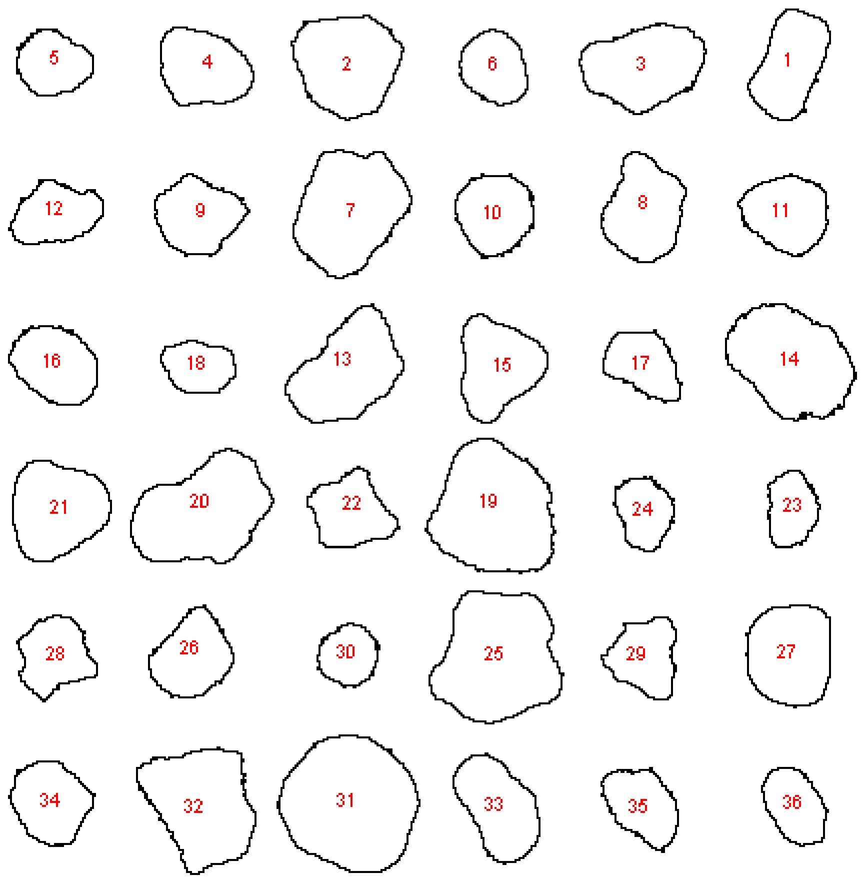


Figure 7. Photo of the outlines of the sand particles analyzed and numbered through ImageJ with threshold from 0-29 and circularity from 0-1.

Shape descriptors

ImageJ was also able to display a table of the necessary features of each sand particle such as their area, circularity, roundness, and solidity.

In Imagej, the circularity given by $4\pi(Area)/(Perimeter^2)$, while roundness is $4\,(Area)/\pi(Major\ Axis^2)$. Their difference is visualized in the figure below.

Results

File Edit Font Results

	Label	Circ.	AR	Round	Solidity
1	Untitled-1	0.78	1.98	0.51	0.98
2	Untitled-1	0.91	1.00	1.00	0.98

Figure 8. Circularity VS Roundness in ImageJ [1]

	Label	Area	X	Y	Circ.	AR	Round	Solidity
7	Sand Image.png	0.186	1.563	0.919	0.772	1.291	0.774	0.941
8	Sand Image.png	0.112	2.814	0.918	0.698	1.393	0.718	0.915
9	Sand Image.png	0.089	0.920	0.942	0.821	1.153	0.867	0.932
10	Sand Image.png	0.087	2.179	0.933	0.762	1.052	0.951	0.933
11	Sand Image.png	0.091	3.417	0.932	0.826	1.171	0.854	0.948
12	Sand Image.png	0.074	0.318	0.936	0.712	1.590	0.629	0.907
13	Sand Image.png	0.182	3.435	1.557	0.651	1.412	0.708	0.934
14	Sand Image.png	0.149	1.555	1.597	0.758	1.579	0.633	0.929
15	Sand Image.png	0.112	2.194	1.580	0.751	1.175	0.851	0.923
16	Sand Image.png	0.091	0.318	1.573	0.848	1.353	0.739	0.954
17	Sand Image.png	0.066	2.810	1.571	0.748	1.527	0.655	0.920
18	Sand Image.png	0.053	0.924	1.581	0.851	1.432	0.698	0.944
19	Sand Image.png	0.224	2.176	2.192	0.779	1.137	0.880	0.945
20	Sand Image.png	0.194	0.941	2.197	0.734	1.570	0.637	0.917
21	Sand Image.png	0.133	0.324	2.186	0.861	1.073	0.932	0.960
22	Sand Image.png	0.082	1.568	2.197	0.725	1.132	0.883	0.882
23	Sand Image.png	0.053	3.439	2.186	0.735	1.553	0.644	0.926
24	Sand Image.png	0.055	2.813	2.199	0.785	1.346	0.743	0.913
25	Sand Image.png	0.230	2.193	2.813	0.777	1.120	0.893	0.927
26	Sand Image.png	0.089	0.896	2.808	0.752	1.263	0.792	0.929
27	Sand Image.png	0.129	3.434	2.800	0.880	1.198	0.835	0.970
28	Sand Image.png	0.078	0.313	2.805	0.726	1.080	0.926	0.871
29	Sand Image.png	0.073	2.817	2.811	0.681	1.147	0.872	0.883
30	Sand Image.png	0.051	1.559	2.804	0.807	1.067	0.937	0.928
31	Sand Image.png	0.257	1.566	3.433	0.858	1.069	0.935	0.968
32	Sand Image.png	0.176	0.936	3.437	0.645	1.298	0.770	0.903
33	Sand Image.png	0.107	2.196	3.448	0.773	1.781	0.562	0.932
34	Sand Image.png	0.089	0.296	3.425	0.812	1.152	0.868	0.937
35	Sand Image.png	0.063	3.453	3.440	0.804	1.598	0.626	0.946
36	Sand Image.png	0.073	2.806	3.439	0.727	1.571	0.637	0.933

Figure 9. Screenshot of the measurement results from ImageJ

Roundness and circularity

From the table shown in the previous slide, all of the roundness and circularity values are in the range of $0.7 \leq a, b < 1.00$, where a and b are the roundness and circularity values, respectively.

To obtain the image in Figure #, we set the circularity to be from 0 to 1, which accounts for any shape up to a perfect circle at b = 1. If we lower the upper bound of the circularity, say to 0.7, we would not be able to get the rounder particles, as seen in Figure #.

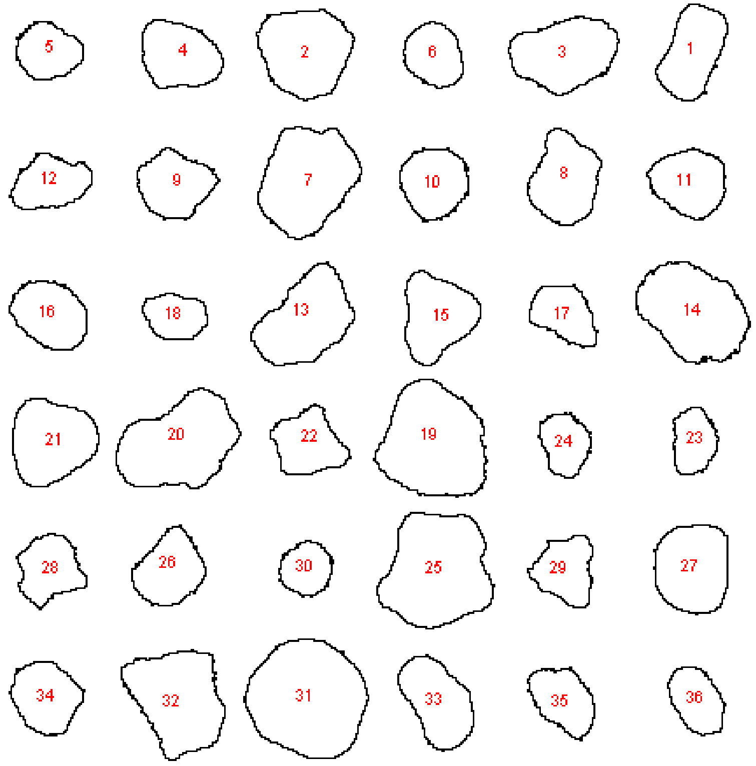


Figure 10. Photo of the outlines of the sand particles analyzed and numbered through ImageJ with threshold from 0-29 and circularity from 0-1.

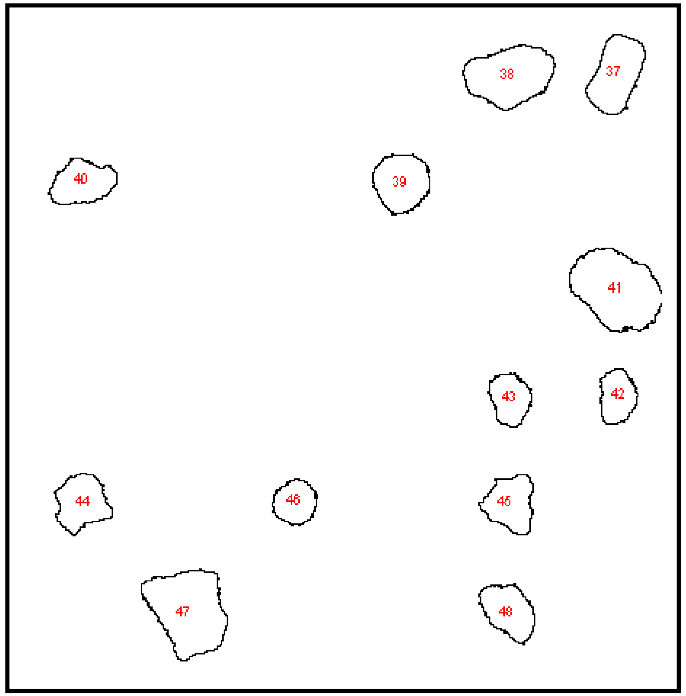


Figure 11. Photo of the outlines of the sand particles analyzed ImageJ with threshold from 0-40.

Particle porosity

The particle porosity is defined as the percentage of void space in a rock. I have no quantitative data here, but I observed that the porosity of the particles was affected by the thresholding.

In my case, I only made sure to separate the outline of each particle completely, not the parts inside the perimeter. I did this because if the outlines deviated a bit (threshold: 0 to 5), the software counted it as another particle and it resulted to having particle count to 37, and not all of the particles were detected. Thus, the there were certain parts within the rock that got blended along the background.

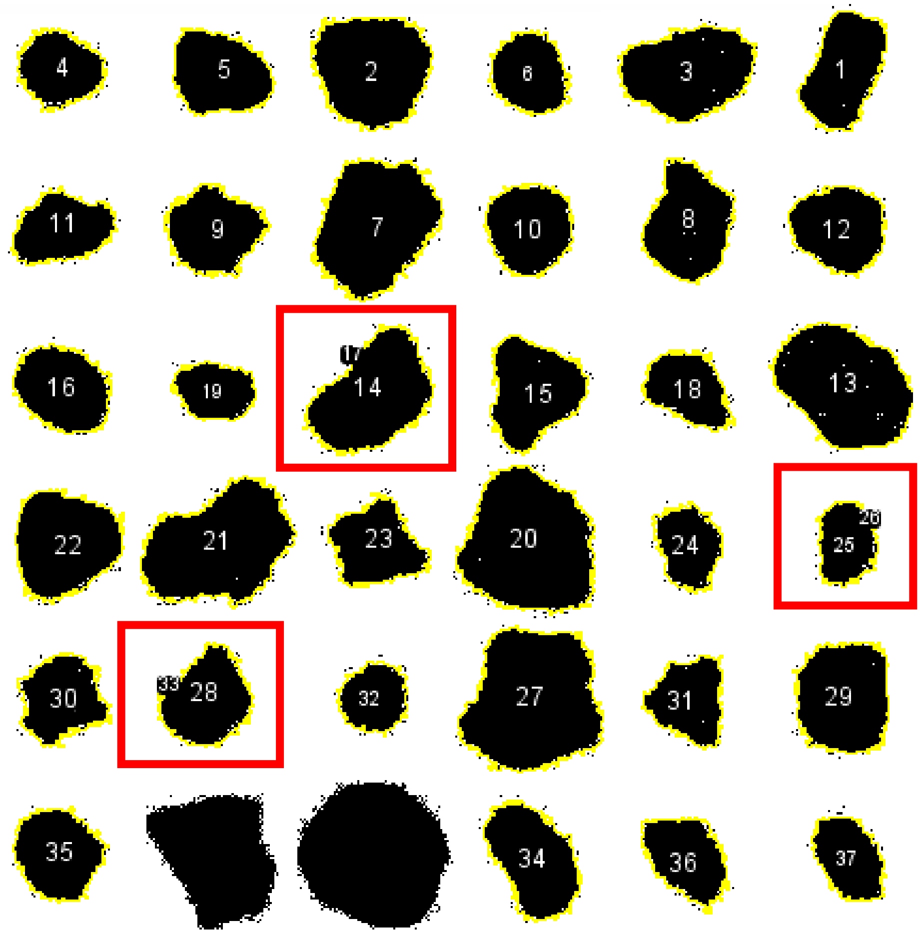


Figure 12. Photo of the outline and mask of the sand particles analyzed ImageJ with threshold from 0 to 5.

Particle porosity

As can be seen in the figure, when the threshold was increased a little bit, particles 14, 28, and 25 have a smaller particle detected while two of the biggest particles in the last row was not detected.

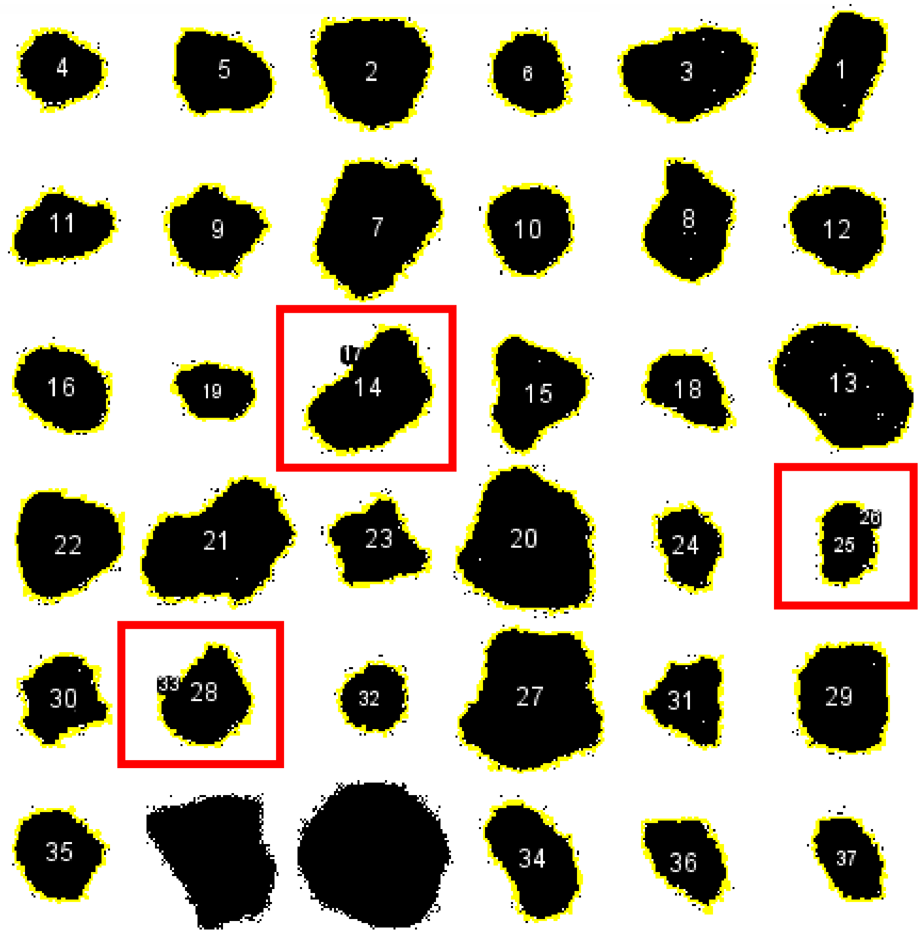


Figure 12. Photo of the outline and mask of the sand particles analyzed ImageJ with threshold from 0 to 5.

Overall analysis

In this activity, we tried to extract the features of a sand particle from an image. Using ImageJ, we were able to find the roundness and circularity, and number the particles from one to 36.

In thresholding the image, the best range taken was from 0 to 29, otherwise, if the threshold was lower, then the outline of the mask would include the background of the particles, and would result to a higher particle count. If we increase the threshold, then the outline would not be defined anymore and only a few of the particles would actually be available for analysis. The threshold of 0 to 29 allowed me to analyze exactly 36 particles, which is exactly the number of particles that we have.

Overall analysis

Meanwhile, for the circularity of the particles, we chose to analyze from 0 to 1. This allows us to account for all of the shapes of the particles. Otherwise, if we lessened the circularity, then the analysis would only be available for those still within that range. The maximum circularity for analysis is one.

For the porosity, this is affected by the thresholding that we did in the earlier steps. Because we needed to cover the outlines of the particles, some of the areas within it can be affected, such that they get blended into the background. This results in a decrease in the total area covered by the outline, which was seen in the screenshot of the values attached in the previous slides.

Reflection

Technical

When I did this activity, it was very easy to follow since the module has laid out the steps completely. It was still challenging to manipulate the threshold so that the analysis would be directed towards the actual particles.

At first, I tried to increase my threshold and it resulted to having up to 50 particle count since the software counted the sides of the particles as separate ones. Meanwhile, when I tried to overfit my threshold, some of the particles were not detected because their shapes were not recognized by the threshold I set.

I was able to figure these out through trial and error. I was also able to understand the concept more by understanding the definitions of ImageJ on how it calculates the features such as circularity and roundness.

Reflection

Personal Experience

This activity is especially interesting to me because I started studying feature extraction via machine learning as one of my readings for my thesis. I was able to incorporate some of my readings in my analysis here, like when we extract features from an image, we may use these data points to real world data if we want to identify objects.

For example, if we want to create a program that identifies a cat in an image, then we feed the algorithm a lot of datasets and allow it to learn or supervise it in making out the features of a cat. After this, it would be able to identify cats in a real life dataset already.

The same principles are also in place here in the activity, such that we should not overfit or underfit the data (i.e. in thresholding) so that the software would be able to take all of the shapes and sizes of the particles into account by itself.

References

[1] O'Connell, C. (2016). Difference between roundness and circularity measurements in Image J. Research Gate. Obtained from https://www.researchgate.net/post/Difference_between_roundness_and_circularity_measurements_in_Image_J_shape_descriptors_function

[2] (2008) Wentworth scale. In: Chesworth W. (eds) Encyclopedia of Soil Science. Encyclopedia of Earth Sciences Series. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-3995-9_634

[3] Z. Zhang and W. S. Lee, "Deep Graphical Feature Learning for the Feature Matching Problem," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5086-5095, doi: 10.1109/ICCV.2019.00519.