Image Quantization via k-means Clustering for Analyzing Oil Spill Data

Zoya Khan, Jessie Wang, Ainsli Shah

MATH 2015: Linear Algebra and Probability
Dr. Evan Sorenson
Fall 2024

Abstract

In this report, we will discuss an application of image quantization, via a k-means clustering algorithm, to analyze a dataset of oil spill satellite images and evaluate the effectiveness of this method (for data reduction) in accurately computing the percentage of oil spills. Our findings explore the limitations of k-means clustering in the analysis of dynamic datasets that have feature and environmental variability.

1 Introduction

Oil spills pose significant threats to environmental and societal health, particularly impacting aquatic ecosystems and water consumption. Two of the main ways marine life can be negatively affected by oil are through fouling (or oiling) and oil toxicity. Fouling/oiling describes the physical coating of animals and plants with oil. For example, oil can coat birds' wings, affecting their ability to fly, as well as strip sea otters' fur of insulation, exposing them to hypothermia. Oil toxicity refers to the way toxic compounds in oil can cause severe health problems such as heart damage, stunted growth, immune system defects, and death [**?**]. Hence, a high incidence of oil spills in port areas prompts additional urgency for developing efficient detection mechanisms [2]. Satellite imagery over bodies of water is the principal method used to identify and monitor oil spills, but the detection software relies on accurate differentiation of oil and water from these images [7]. In this project, we used Python to load and standardize a dataset of 8 images and flatten them into pixel vectors. The oil and water in our dataset are differentiated more by contrast and light rather than different hues of red, green, and blue, and RGB color space has drawbacks when representing shading results or quick changing of illumination. For these reasons, we decided to convert to the HSV (hue, saturation, brightness) color space. HSV is better for the distinction of color

and is frequently used in computer vision and image investigation for element recognition or image segmentation. It has applications such as road traffic sign recognition and facial recognition [8]. To detect the percentage of oil in each picture, we implemented k-means clustering to segment each image based on clusters/centroids within the image's HSV space. Through our research, we attempt to determine the optimal number of k-means clusters to divide the pixels into for the most accurate percentage results. Our paper will provide an analysis of our results, a discussion of the challenges encountered while processing variable satellite images with a less dynamic algorithm, and conclude with an analysis of our findings and applications.

2 Methodology

We have compartmentalized our methodology (and problems we encountered/accommodated for) in the following points:

A. *Image Preprocessing & HSV Enhancement:* We loaded a limited dataset of 8 satellite images of oil spills from a larger dataset [9] via Python libraries and preprocessed them to ensure uniform dimensions and quality. For this, we normalized pixel intensity values to a defined range and standardized image dimensions across the 8 images to reduce computational bias.

Figure 1: Eight chosen satellite images. Images 1–8 (from left to right) correspond with the labeled data in the Results/Analysis section below.

(a) We initially wanted to differentiate oil spillage from ocean water through an analysis of the differentiations in RGB values. Because of the overlap in RGB values

within oil spills and ocean water, our k-means algorithm wasn't able to effectively detect a difference. To accommodate for this, we shifted to using an HSV (hue, saturation, and brightness) contrast scale to exaggerate the difference in oil/water color and accommodate for variance from any light sources, reflections, etc.

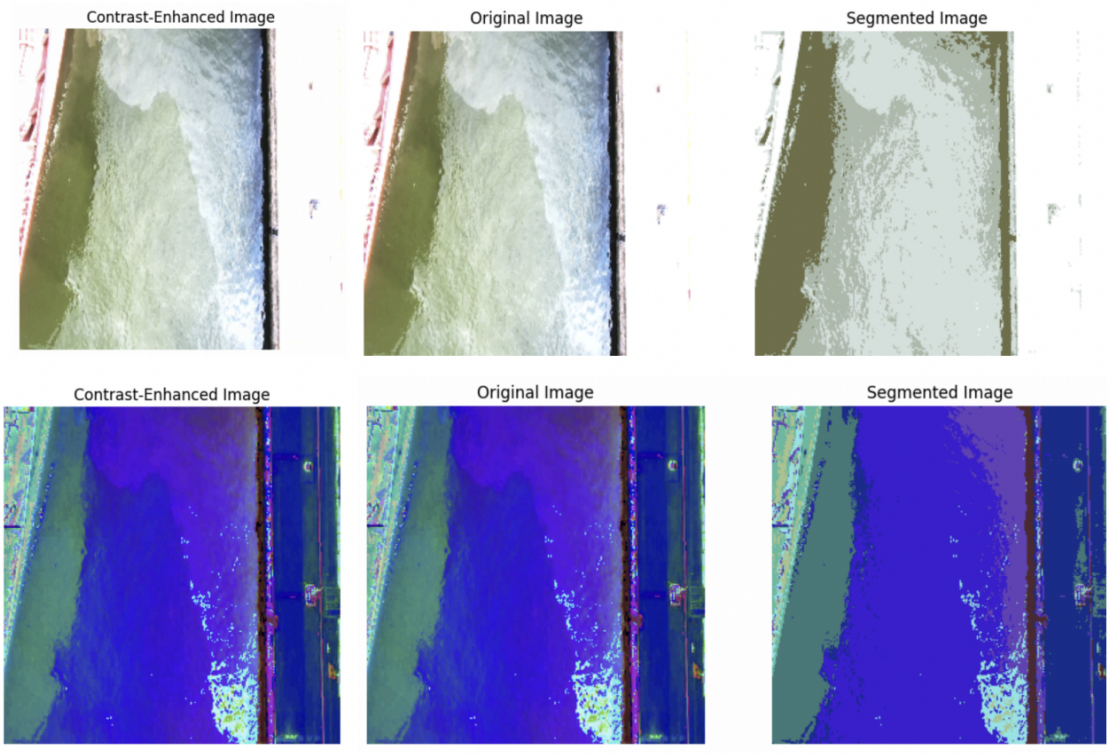

Figure 2: k-means algorithm (with 8 clusters) detected 98.0% oil spill from RGB segmented image vs 16.3% (more accurate to manual/binary masking calculation) for HSV segmented image.

(b) We then flattened the image matrix into a vector for processing: converted each image into a 2D matrix representing its pixels then flattened the matrix into a 1D vector representing each pixel in terms of its RGB/HSV color channels/values

B. *K-means clustering algorithm/mathematical explanation:*

(a) We tested k = 2, 5, 10, 15, 20, and 25 clusters for each image in the dataset.

(b) We also created 2D plots (within Python) to visualize clusters from flattened images.

(c) *K-means algorithm logic:* [3, 1]

i. K-means define clusters of points based on the nearest cluster centroid to a point. These centroids are found by alternating between assigning data points to clusters based on current centroids and choosing centroids based on the current assignment of data points to clusters.

ii. The algorithm steps:

A. Choose K cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$ randomly from the image. These centroids represent the centers of the k clusters.

B. Each data point (in this case, a pixel in the image) is assigned to the cluster with the nearest centroid.

C. Determining distance between each point on the image and the cluster centroids assigns points to each based on distance, and then moves the centroid to the mean of the assigned points:

$$D = \|X - Z\| = \left[\sum_{i=1}^{n}(x_i - z_i)^2\right]^{1/2}$$

where $X = (x_1, x_2, \ldots, x_n)$ and $Z = (z_1, z_2, \ldots, z_n)$.

D. Update centroids as the mean of assigned pixels.

E. Repeat until convergence (no significant change or max iterations).

F. For each pixel, the Euclidean distance (distance metric typically used) is calculated between the current pixel and each mean to assign the pixel to a cluster [**?**]. The mean of the cluster (recomputing centroid of each cluster as mean of all pixel values of that cluster) is then updated

iii. For each cluster C, we compute the new centroid as the mean of the points which have been assigned to the cluster, this process is then repeated (stopping condition is when centroids no longer change significantly or a maximum number of iterations is reached) until completion

iv. The algorithm completes when the clusters stabilze, meaning further iterations don't significantly change the assignments or centroids.

C. *Segment images based on centroids and apply the k-means algorithm.*

D. *Binary masking to find baseline oil spill percentage:* We used binary masking and manual calculations to find a standard oil spill percentage to use as a baseline for each image. This baseline was used to understand what k/cluster amount was most effective in accurately identifying oil spills.

(a) To find an accurate baseline to compare the processed images to, we manually added the mask to the images - coloring the visible oil spill in white and the rest black. This was done by importing the images into the notability app and manually identifying discoloration in the water, which inevitably would produce different results than the k-means clustering algorithm, however, it was necessary to arrive at a baseline for what the oil spill percentage is. This has some consequences which will be discussed.

(b) We also attempted to use the methods within the cv2 library - different thresholding methods, though the results produced with that were visibly inaccurate as shown in the results section and we did not use those percentages as the baseline.

(c) *Calculate the percentage of oil spill pixels:* This percentage was a calculation of the proportion of pixels in oil spill clusters relative to the total number of pixels in the image (standardized in A(preprocessing)).

i. We defined the oil color threshold to be 0.2, 0.2, 0.2 for color values to account for the fact that the color/HSV of ocean water and oil spills aren't very significantly different.

ii. Repeated k-means clustering and oil percentage calculations for each k = 2, 5, 10, 15, 20, 25.

iii. Determined the optimal k by comparing k-means results to the manual/baseline percentages.

3 Results and Analysis

An example of the complete oil spill analysis for this satellite image (with a clear oil/water differentiation) is below:

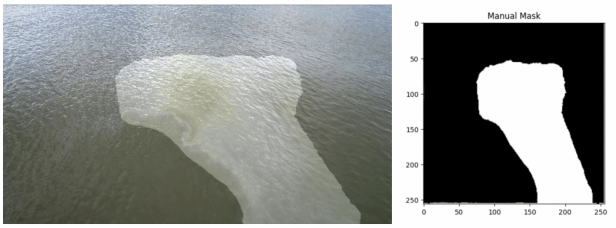

Figure 3: Original and manual mask (30.59%).

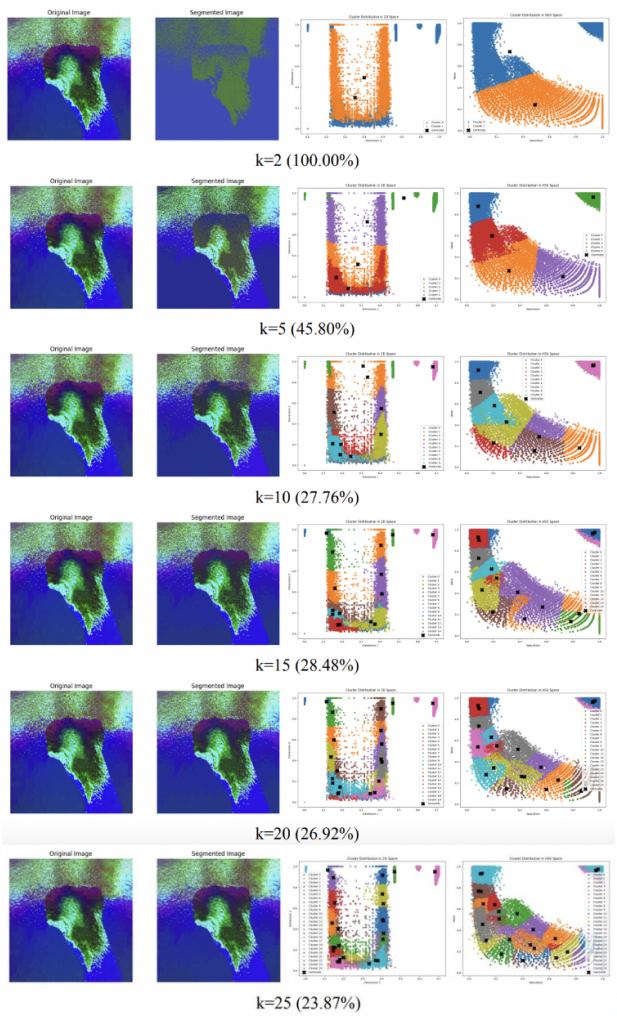

Figure 4: Complete oil spill analysis for Image 7 with a clear oil/water differentiation.

For most images, 5-15 clusters (k) were the range of accurate oil spill percentage predictions. However, as shown in Table 1 (below), the number of clusters still varied for each image because of different segmentations and different HSV values interacting differently with the standard threshold (resulting in different detections).

Image	k=2	k=5	k=10	k=15	k=20	k=25	Baseline
1	11.35	0.72	3.45	3.24	2.57	2.84	18.05
2	10.76	76.29	7.11	10.62	16.43	16.86	18.82
3	56.89	5.48	17.36	13.26	11.17	8.04	17.93
4	34.95	23.07	15.4	10.81	10.06	9.50	18.29
5	24.20	17.82	21.38	6.75	7.88	11.17	19.39
6	100.00e	45.80	27.76	28.48	26.92	23.87	30.59
7	55.21	40.66	15.29	16.56	17.96	16.78	33.63
8	48.47	23.70	28.4	21.89	24.28	24.34	35.03

Figure 5: Oil spill percentage calculations (highlighted number of clusters with the highest accuracy).

Despite these fluctuations, the k-means clusters were consistent in their calculations for $k \geq 10$ clusters, meaning the proportions of detected oil spill regions stabilized across the eight images, and increasing the number of clusters beyond k=10 had little impact on the final computation. Graph 1 shows that, across all 8 images, the oil spill percentages leveled off for $k \geq 10$ clusters (k= 10, 15, 20, and 25). This plateau correlates to an HSV threshold sensitivity and shows that the algorithm was able to consistently capture the defined oil spill region within the feature space with more clusters – after k=10, the algorithm's segmentation only refined the established groupings/regions with more clusters rather than changing the detection.

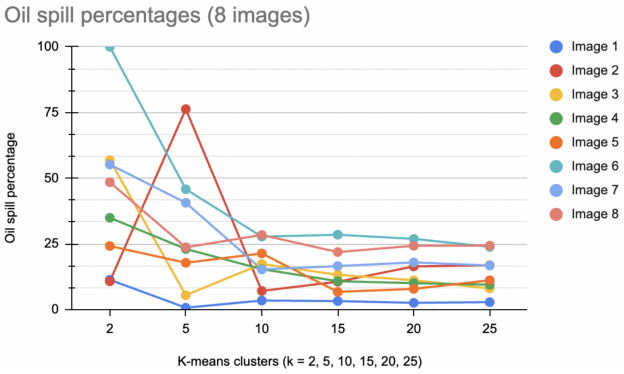

Figure 6: Line plot with oil spill percentages per k-means cluster number

The masking methods within the cv2 library did not work very well with the oil spill data, even for data where the spill was a clearly outlined large area with obvious discoloration. Different thresholding methods were tested including Otsu, binary, and triangle. The Otsu threshold produced a mask that covered the oil spill most accurately as seen below in Figure 4. For this specific image, it produced a percentage of 42.22 percent - close to the value produced by k means clustering with k=5. From the manually applied mask as shown below, calculating the percentage of the oil spill was feasible by simply taking the percentage of 0 pixels, and this percentage is displayed in the table above.

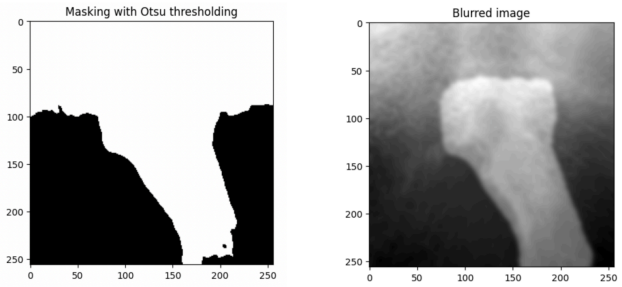

Figure 7: Otsu thresholding most effectively masked the spill/water in the satellite images.

4 Discusssion and Conclusion

K-means clustering is a simple algorithm that partitions data points into k clusters based on their solidarity (essentially minimizing the variance within clusters while maximizing the variance between clusters). Because satellite images of oil spills are in a dynamic environment, the standardization of the k-means algorithm is less effective at detecting because it doesn't accommodate for variation within the environment or images themselves.

Specifically, the k-means algorithm assumes a pre-defined, fixed number of clusters (k) which doesn't interact well with oil spill images that have varying complexities (like the number of regions with spills, distributions of the spills, etc). k-means also uses Euclidean distance to define similarity. For non-linear and high-dimensional data, like oil spill images with high pixel intensities, this assumption doesn't accurately capture meaningful distinctions between oil spills and other environmental features (objects, water, etc).

Because the k-means algorithm treats each pixel independently, it is unable to incorporate any contextual information about the environment. Therefore, lighting variability, different satellite angles, and variations in HSV values aren't accounted for. K-means applies uniform segmentation thresholds across the image, which negates any difference in intensity/color values. Finally, because the k-means algorithm is standardized, it is unable to distinguish between objects (like ships, shadows, etc) and oil spills, resulting in false positives and inaccurate percentage calculations.

4.1 Problems we encountered:

4.1.1 Issues with comparing with the manual mask:

Though by applying the manual mask we were able to determine a baseline for the percentage of the oil spill, the main issue within this method is the inconsistency with the algorithm in how it determines which region displayed discoloration. With the clustering methods, it is able to distinguish the color of the pixel by exact values, yet when manually outlining where the oil spill is, the threshold for discoloration is no longer a strict cutoff through human error.

We attempted to maintain relatively the same threshold across images when applying the mask, however for images such as image 8 listed below, the oil spill is widely spread throughout the water and not a complete area such as the example shown in the methods section. On the one hand, it demonstrates the ability of k-means clustering to identify the different layers within the discoloration of image 8, though it also diminishes the significance of the percentage calculated with the manual mask.

Figure 8: Image 8: original image, manually masked image, and segmented image (k=5)

4.1.2 Variance within color (RGB and HSV) is hard to account for with a standard threshold:

To standardize our methodology, we defined our oil color threshold to be 0.2, 0.2, 0.2. Each image in our limited dataset interacted differently with this predefined threshold because of factors like glare from satellite images, oil emulsion/sheen/patch thickness, sunlight, etc. These external factors introduced variability in pixel intensity/HSV values that a fixed threshold isn't able to dynamically adapt to, leading to inconsistent performance. The k-means algorithm started differentiating too much within the segmented image which meant it wasn't detected by the threshold; subclusters fell below the fixed threshold and oil pixels were excluded from the final calculation. For example, in Figure 6, Image 5 had 21.38% oil spill detected for k=10 clusters, but increasing to 15 clusters dropped this percentage to 6.75%. Similarly, Image 7 had 40.66% for 5 clusters (more accurate to our baseline calculation), but dropped to 15.29% with 10 clusters.

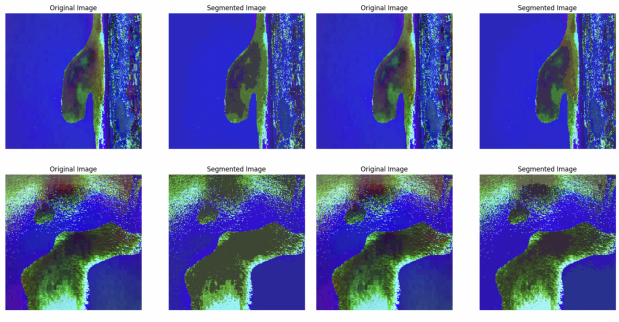

Figure 6. Image 5 (k=10 vs k=15) and Image 7 (k=5 vs k=10)

Figure 9: Image 5 (k=10 vs k=15) and Image 7 (k=5 vs k=10)

4.1.3 Isolating external agents/variables is difficult:

Image 4, a satellite image of an oil spill by a port, had ships, crates, and other objects in the image. The algorithm was unable to differentiate between these external objects and the oil spill, and as a result, the final segmented images (shown below, in Figure 7) included the objects as part of the final calculation. Because isolating environmental differences is very situational, a fixed model won't be accurate in its analysis. Context awareness can be improved if additional features (texture, edge detection, patterns, etc), adaptive clustering (weighted k-means (nearby pixels are more likely to belong to the same cluster)), or object detection models are incorporated to differentiate.

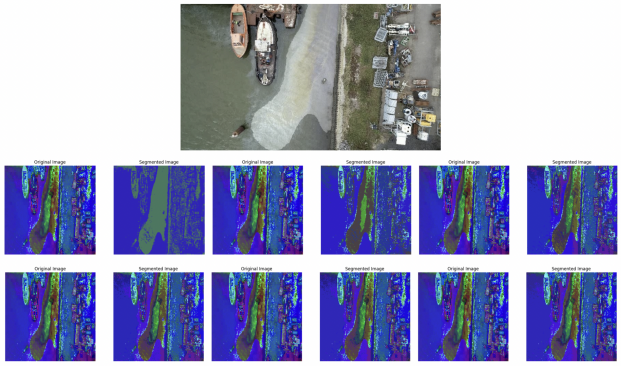

Figure 10: Segmentation and calculated oil spill percentages for Image 4 (with boats/objects): k=2 (34.95%), k=5 (23.07%), k=10 (15.47%), k=15 (10.81%), k=20 (10.06%), and k=25 (9.50) (from left to right)

5 Application

This project focused on k-means clustering as a model for quickly analyzing the presence of spills in ocean water; we specifically tried to understand which level of k-means clustering would be the most effective at accurately calculating the percentage of oil spills in a water body from satellite images. Because of this focus on accuracy, we limited our dataset to eight satellite images so we could see how the k-means algorithm interacted with unique features/environments. As we completed our code/computations, our project evolved to focus on understanding the difficulties in applying a standardized algorithm to a variable dataset. Some differences that the k-means algorithm was unable to account for include varying threshold/HSV values per satellite image, variability in oil appearance (emulsion/oil thickness/glare), and external objects (that were included in the segmented image alongside oil spills). Before being implemented for larger datasets, effective detection models should prioritize being able to dynamically accommodate both varying image features and environmental factors. A potential solution is incorporating other techniques alongside the k-means algorithm to make the model more robust and dynamic.

References

[1] Celebi, M. (2011). Improving the performance of K-means for color quantization. *Image and Vision Computing*, 29(4), 260–271. Retrieved from https://doi.org/10.1016/j.imavis.2010.10.002

[2] De Kerf, T., Sels, S., Samsonova, S., Vanlanduit, S. (2024). A dataset of drone-captured, segmented images for oil spill detection in port environments. *Scientific Data*, 11. Retrieved from https://www.nature.com/articles/s41597-024-03993-8

[3] Li, Y., Wu, H. (2012). A Clustering Method Based on K-means Algorithm. *Physics Procedia*, 25, 1104–1109. Retrieved from https://doi.org/10.1016/j.phpro.2012.03.206

[4] Nallakaruppan, M., Gagadevi, E., Lawanya Shri, M., et al. (2024). Reliable water quality prediction and parametric analysis using explainable AI models. *Scientific Reports*, 14. Retrieved from https://www.nature.com/articles/s41598-024-56775-y

[5] Oil Spills. (2020). Retrieved December 7, 2024, from https://www.noaa.gov/education/resource-collections/ocean-coasts/oil-spills

[6] Peuquet, D. (1992). An algorithm for calculating minimum Euclidean distance between two geographic features. *Computers & Geosciences*, 18(8), 989–1001. Retrieved from https://www.sciencedirect.com/science/article/pii/009830049290016K

[7] Roberts, D. L., & Heldon, D. (2022). Spotting Spills from Space. *NOAA Office of Response and Restoration*. Retrieved from https://blog.response.restoration.noaa.gov/spotting-spills-space

[8] Saravanan, G., Yamuna, G., Nandhini, S. (2016). Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. *2016 International Conference on Communication and Signal Processing (ICCSP)*. Retrieved from https://doi.org/10.1109/ICCSP.2016.7754179

[9] Sels, S., Vanlanduit, S., & De Kerf, T. (2024). Annotated RGB images of Oil Spills in a Port Environment. *Zenodo*. Retrieved from https://doi.org/10.5281/zenodo.10555314