ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2022.0400

K-MEANS WITH SAMPLING FOR DETERMINING PROMINENT COLORS IN IMAGES

Angelina Cheng[1], Eric Rosenberg[2] and Alina Gorbunova[3]
[1,3]Department of Computer Science, Rutgers University, United States
[2]Department of Applied Computing, Georgian Court University, United States

Abstract

A tool that quickly calculates the dominant colors of an image can be very useful in image processing. The k-means clustering algorithm has this potential since it partitions a set of data into n clusters and returns a representative data point from each cluster. We discuss k-means with sampling for images, which applies k-means clustering to a random sample of image pixels. We found that even with a small random sample of pixels from the image, k-means with sampling exhibits no significant loss of correctness. We examine the usefulness and limitations of k-means clustering in determining the prominent colors of an image and identifying trends in large sets of image data.

Keywords:

K-Means, Clustering, Color, Image

1. INTRODUCTION

The analysis of image data drives world industries of security, science, entertainment, marketing, and historical studies. One way to evaluate image data is through the lens of color theory. A human can look at several images and perhaps identify some prominent color trends. But when analyzing millions of images for color trends, it is necessary to use a program that will process the images efficiently.

A major topic that needs to be addressed when approaching such a problem is this: how can a computer recognize color themes? And how does a quantitative measure of color compare to human opinion? A simple method would just average all the colors that make up an image. However, this will not yield the image's defining color; it usually results in nondescript grey or brown. To achieve our goal, it is necessary to use a different method that can identify a few specific color values that best represent all the colors in the image.

If a data set consists of a very specific type of image with clearly identifiable rules and labels, one might choose to use a supervised learning algorithm for the purposes defined above. For example, if all considered images are photos of flowers, then a programmer can train an algorithm to recognize the different parts of the flower and assign representative colors to them. However, what if the image set has no specifications? For example, a fashion company may find it useful to process millions of photos on social media and determine the most popular colors of this month. Without specific features to expect, it is difficult to tailor a supervised algorithm to be effective and efficient in all possible scenarios.

Thus we consider the *k*-means clustering algorithm. It is an unsupervised method of image processing that will classify data without human training or complex pre-defined labels [1]. This makes *k*-means clustering a valuable candidate for determining color trends across large sets of disparate images. The *k*-means clustering algorithm is used by data science professionals in many industries, such as customer analysis, sales, and health monitoring. It an efficient tool and can identify groups that were not previously defined.

In this paper we will discuss the usefulness of the *k*-means clustering algorithm specifically for determining the most prominent colors of an image. We will show that an efficient color clustering program does not need to process every pixel of an image to give an accurate result. Applying *k*-means clustering to a small fraction of the original image, even as small as one-tenth, will yield results nearly equivalent to applying *k*-means clustering to the complete set of pixels. Yet running *k*-means clustering on a dataset with one-tenth of the pixels will execute in one-tenth of the runtime. We refer to this sampling adjustment to the *k*-means algorithm as "*k*-means with sampling".

2. *K*-MEANS CLUSTERING ALGORITHM

We have introduced the *k*-means clustering algorithm as a tool to sort data into groups. In this section we address further details.

When using this algorithm for the purpose of determining the dominant colors of an image, our data points will be the colors of each pixel. We will refer to an image's set of most prominent colors as its "color theme". We represent color in the RGB color space, with the expectation that the user's purpose is to analyze photos and online images, so the color of each pixel in an image will be represented by three integers, each between 0 and 255 (inclusive). The magnitude of each integer corresponds to the pixel's intensity of red, green, or blue, respectively [2]. When calculating Euclidean distances in *k*-means clustering, a pixel's *R*, *G*, *B* integers will be evaluated as a three-dimensional vector. Given pixel *A* with RGB value $[R1,G1,B1]$ and pixel *B* with RGB value $[R2,G2,B2]$, the Euclidean distance between *A* and *B* is $\sqrt{(R_2-R_1)^2+(G_2-G_1)^2+(B_2-B_1)^2}$.

We now define the generic *k*-means clustering algorithm (that we will later specialize for color selection). Let the integer k be the number of groups into which the data will be partitioned. The algorithm's goal is to cluster the data into k groups, with each group having one centroid. A centroid is the data point in a group for which the Euclidean distance between the centroid and all other data points in the group is minimized. An optimal solution has been found when the algorithm has minimized the sum, over all k clusters, of the sum of the Euclidean distances between the k-th centroid and each data point in the k-th cluster. In practice, this optimization problem is NP-Hard [3], so k-means clustering is generally implemented to settle for an approximate solution after some number of maximum iterations. Informally, we refer to how close centroids are to their respective data points as "compactness".

Heuristic *k*-means clustering algorithm:

(i) Randomly select *k* centroids.

(ii) Assign each data point to the nearest centroid, forming *k* clusters.

(iii) Recalculate the centroid of each cluster.

(iv) Repeat steps 2 and 3 until each centroid does not change location significantly, or until the maximum number of iterations has been reached.

At each iteration of the algorithm, the *k* centroids are the current best representative data point of that cluster. At the termination of the algorithm, it is expected that the centroids represent the *k* most distinct groupings of the data [4].

When *k*-means clustering is applied in image processing, the data points are the RGB color values of each pixel. As previously mentioned, we use minimum Euclidean distances between [*R*, *G*, *B*] vectors to determine centroids and data point reassignments during each iteration. This heuristic *k*-means clustering algorithm has a runtime of $O(n{\cdot}k{\cdot}t{\cdot}d)$, where *n* is the number of data points, *k* is the number of clusters, *t* is the maximum number of iterations, and *d* is the dimension of each data point ($d = 3$ in the case of RGB colors as data points) [5]. With the maximum iteration termination condition, as in step 4 above, the program may terminate before the optimal solution has been found. However, it has been experimentally shown that in most practical cases, the centroids will change only very slightly after 20 iterations [6].

A common way to adjust *k*-means clustering for efficiency is to implement a centroid movement threshold ε. The parameter ε is an upper bound on the Euclidean distance between a centroid's current position and its previous position. When for each centroid this distance changes by less than ε, the algorithm terminates. It has been experimentally shown that the first few iterations of *k*-means clustering produce the largest changes in centroid distances; later iterations mainly consist of small oscillations around their optimal values [6].

3. RELATED WORK

The goal of our research is to analyze the usefulness of the *k*-means clustering algorithm for determining prominent colors, and to present our findings on *k*-means with sampling, an adjustment which improves the runtime of the bare *k*-means clustering algorithm for color analysis. There are several published enhancements to the general *k*-means clustering algorithm to improve runtimes using methods different than *k*-means with sampling and not specifically related to color analysis. Some methods discuss ways to choose better initial centroids (using refinement algorithms to replace random selection), since a poor choice of initial centroids can necessitate more iterations [7]. Methods of avoiding this include choosing centroids from strategic increments in a sorted list or mathematical projection of data points [8].

The repeated *k*-means strategy applies *k*-means clustering multiple times from different initial centroids to increase the probability of a compact result. In our study, we use the repeated *k*-means strategy, with multiple *attempts,* where each *attempt* uses a new choice of random initial centroids. Other common ways to increase *k*-means efficiency include refining centroid assignment criteria [1] and storing data points in a *kd*-tree as the *k*-means clustering algorithm is executed [9]. These runtime-improving techniques are all applicable and may be used in addition to the methods we will discuss in our paper. However, we will not include those techniques in our upcoming analysis and instead we will focus on methods to improve program efficiency specific to the area of digital image analysis.

A factor that may affect the usefulness of *k*-means clustering results for an image is the value of *k*. A *k*-value that is too low may not reveal the true dominant colors of an image, while a high *k*-value may result in redundant colors (as well as a long runtime). There are established methods that select an appropriate *k*-value and solve this problem [10], so we will not discuss the issue in detail in this paper. Additionally, there are methods to addresses another weakness of *k*-means clustering: it does not consider distinct shapes and objects in its determination of prominent colors. The spatial constrained *k*-means approach is proposed to correct this weakness by iteratively separating an image into object regions, while maintaining the generic and efficient properties of basic *k*-means clustering [11].

We chose the *k*-means clustering algorithm because it is a popular, fast, and simple technique. There are other image segmentation algorithms that are also commonly used. The nearest centroid or "Rocchio" method performs non-parametric classification that has centroid-point distance minimization goals similar to *k*-means clustering. There are also graph-based image segmentation techniques, which will analyze images as networks, linking adjacent pixels based on certain properties. Particle swarm optimization (PSO), which uses "swarms" of candidate solutions and a fitness function, has also demonstrated good performance and results when applied to image clustering [12]. Additionally, the partial least squares discriminant analysis (an extension of PLS regression) is a good choice for image analysis; however, it is a supervised learning method that requires some training data to calibrate appropriate probabilities [13]. The *k*-means clustering algorithm we utilize can certainly be used jointly with supervised learning algorithms for certain purposes, such as in the case of color correction based on a set of training images [14].

The method of *k*-means clustering with a selection of random data has been addressed, though not with our specific analyses in prominent color determination. Convergence performance has been analyzed for an optimized *k*-means program that uses random sampling and parallelization on general (non-image) data [15]. In our study, we specifically analyze *k*-means with sampling with small samples to determine color themes.

4. *K*-MEANS WITH SAMPLING

Since the runtime of *k*-means clustering is $O(n{\cdot}k{\cdot}t{\cdot}d)$, the runtime of an image clustering program can be decreased by decreasing *n* or *k* or *t* or *d*. For color analysis we have $d = 3$, the dimension of the RGB color value of a pixel. This cannot be changed. The extent to which it is practical to decrease *k* and *t* depends on the needs of the user. The user may want to determine few or many prominent colors in an image, and that influences what *k* value is chosen. In determining the appropriate *t* value, users should consider that increasing *t* typically yields centroid values closer to their final converged values. Additionally, if the user decides to implement a threshold value ε, lower values of ε correspond to centroids closer to their final converged values.

Lower stopping thresholds require smaller changes to centroid values between iterations before the algorithm can terminate.

The expected runtime O($n \cdot k \cdot t \cdot d$) is proportional to n. In many data processing applications, it is standard to process all or most data points to expect accurate results. However, in this paper we will show this is not necessarily true for the problem of determining the most prominent colors of an image. Tens of thousands of pixels are not needed to identify the most prominent color clusters in an image. It is adequate to study a sample of pixels, especially if many of the pixels are the same color

We investigated the consequences of applying *k*-means to a random sample of the pixels of an image. We reasonably expect hat a random selection of the pixels is a good representation of the entire image, but there is always the risk of an unlucky selection (one that includes mostly outlier colors or does not reflect the proportions of colors in the original image). Hence, using the 30 images in Fig.1, we experimentally measured the disparity between the results of *k*-means clustering on a small random sample of the pixels versus the complete set of all pixels. These images in this data set were chosen from articles from popular online news sites appearing on April 19, 2021. We compared *k*-means results with $k = 4$ and $\varepsilon = 0.5$ (thus computing the four most dominant colors of each image, and terminating *k*-means when each centroid position moved less than .5 between two consecutive iterations).

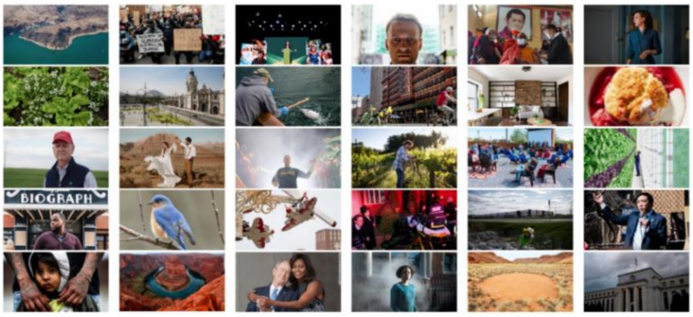


Fig.1. 30 photos from various news sites: New York Times, CNN, MSNBC, NBC. Refs. [16] - [42]

For this analysis, we implemented *k*-means clustering in Python, using the *cv.kmeans*() function on the *cv2* Python module [43]. This module has an added parameter, *attempts*, which stores an integer representing the number of times the program will run *k*-means clustering with different random starting centroids. (The final centroids will be the ones from the attempt with the highest compactness.) Additionally, the *cv.kmeans*() function contains “extra” code outside *k*-means clustering where data is initially ingested and managed. We will ignore the runtime of these “extra” operations in our analysis.

First, we experimentally confirmed the proportional relationship between runtime and input data size. In Fig.2, the vertical axis is the ratio of *k*-means with sampling runtime to normal *k*-means clustering, and the horizontal axis is the fraction of pixels used for *k*-means with sampling. Using the 30 images in Fig.1, we collected average runtimes for *k*-means clustering on samples of 15, 20, 25, ..., 50 percent of the original pixels, and compared these values to the runtime of *k*-means on the entire image. Data is displayed with various values for the number of maximum iterations (represented by different point sizes) and number of attempts (represented by different point colors).

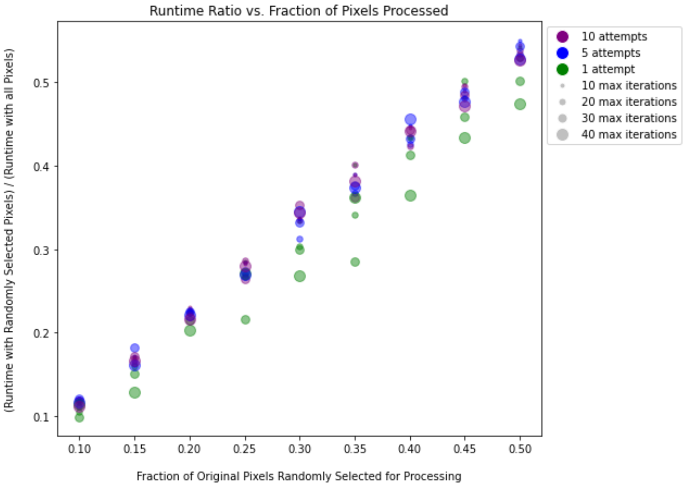


Fig.2. Runtime Ratio

As expected, the runtime ratios are generally proportional to data size. *K*-means clustering on 10% of the original pixels takes approximately 10% of the runtime for *k*-means clustering on all original pixels; similar results follow for samples of 15, 20, 25, ..., 50 percent of the original pixels.

We also observe from Fig.2 that the lower ratios of each sample size are represented by larger green points. This implies that a low number of attempts and a high number of max iterations create the smallest ratios. This occurs because with fewer attempts, there are fewer results to process to choose the most compact one. A high number of maximum iterations broadens the runtime difference between processing small and large data sets because centroids for small data sets are likely to converge sooner. For example, in an image with only 100 pixels, the algorithm is more likely to choose four starting centroids at random whose values are close to the four final centroids, versus an image with 10,000 pixels.

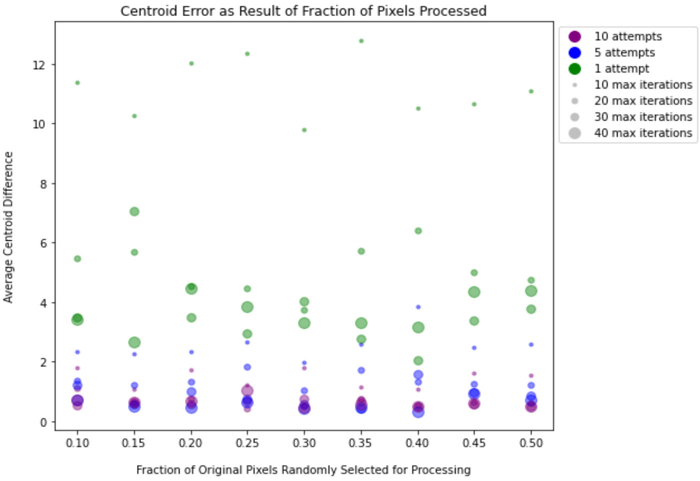


Fig.3. Centroid Error

With these insights into how data size, maximum iteration limit, and number of attempts affect the runtime of *k*-means clustering, we now quantify the error introduced by *k*-means with sampling. To explain how the error is computed, suppose (for illustrative purposes) that k=3, and we process a given image to obtain two sets, each containing three centroids: one set (call it set X) obtained by processing all the pixels, and one set (call it set Y) obtained by processing a random sample of the pixels. Suppose we pair the centroid $X1$ = ($Rx1$, $Gx1$, $Bx1$) in X with the centroid $Y1$ = ($Ry1$, $Gy1$, $By1$) in Y. We define the error $err(X1, Y1)$ by $err(X1, Y1)$ = [$abs(Rx1\text{-}Ry1)$ + $abs(Gx1\text{-}Gy1)$ + $abs(Bx1\text{-}By1)$] / 3, where $abs(\ )$ is the absolute value function. Similarly, if centroid $X2$ in X is paired with centroid $Y2$ in Y we define the error $err(X2, Y2)$ by $err(X2, Y2)$ = [$abs(Rx2\text{-}Ry2)$ + $abs(Gx2\text{-}\ Gy2)$ + $abs(Bx2\text{-}By2)$] / 3, and we can similarly define the error $err(X3, Y3)$ if centroid $X3$ in X is paired with centroid $Y3$ in Y. The total error $err(X, Y)$ for this set of pairings between the three centroids in X and the three centroids in Y is $err(X, Y)$ = $err(X1, Y1)$ + $err(X2, Y2)$ + $err(X3, Y3)$. Since $err(X, Y)$ depends on the pairings between centroids in X and centroids in Y, we find the set of pairings that minimizes this error, and this minimized error is taken to be the error introduced by k-means with random sampling for this image. The Fig.3 displays this error on the vertical axis (labelled "average centroid difference"). This Fig.shows that a low number of attempts and a low number of maximum iterations leads to the highest inaccuracies for the small samples. This is to be expected, since fewer attempts allow for fewer chances to achieve compact results, and fewer iterations of the algorithm reduces the opportunities to get centroids close to optimal centroids.

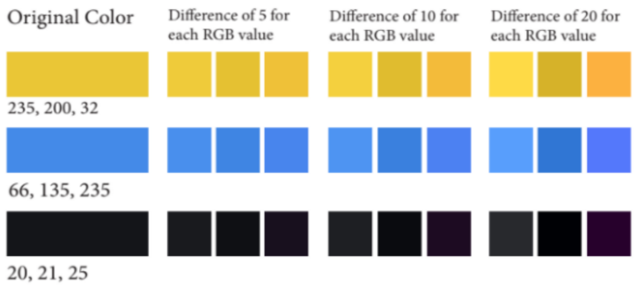


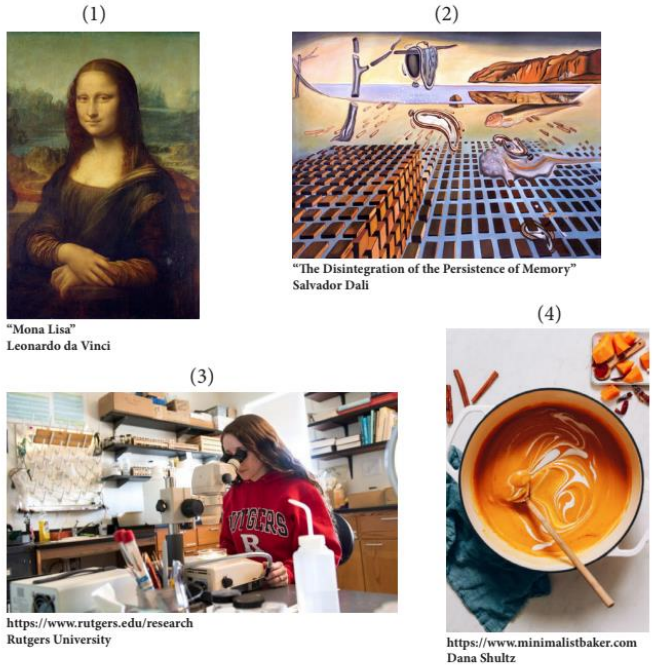
Fig.4. Each entry in the "Difference of [x] for each RGB value" column has 3 color blocks. Each of these color blocks deviates from the original color RGB values by x. The first block shows each RGB value increased by x. The second block shows each RGB value decreased by x. The third block shows the R and B values increased by x, with the G value decreased by x. For example, the "Difference of 5 for each RGB value" column for the first color [235, 200, 32] will have color blocks of these RGB values: [240, 204, 37], [230, 195, 27], [240, 195, 37]

Surprisingly, we observe low error even with few attempts and small samples. The maximum average error between any R, G, or B values is less than 14. In the RGB color space, this value is relatively low since the possible average error can range from 0 to 255. The Fig.4 displays a color and its RGB values in the leftmost column. In the same row of Fig.4, we compare each smaller, individual block of color to the original. At a difference of 5 for each R, G, and B value, the visual disparity between an individual color block and the original is hardly discernible. At a difference of 10, the original color still looks very similar when compared to each individual color block. When each R, G, and B value deviates from the original by 20, the human eye can begin to detect obvious differences. Therefore, a maximal RGB error of at most 14 implies that the centroids from the sampled image are visually extremely close to the centroids from the original image.

The permissible error depends on context. For the user's project, how vital is it that the algorithm outputs the mathematically perfect centroid values for an image, considering that an error of 10 for each R, G, B value is visually trivial? If the user prioritizes program runtime, they can often expect visually accurate results even when using values as low as 5 attempts and 20 maximum iterations on one-tenth of the original pixels. This implies that k-means clustering can achieve near-identical results when applied to a set of all pixels of an image and when applied to one-tenth of the original pixels.

5. LIMITATIONS AND USEFULNESS IN DETERMINING PROMINENT COLORS

The k-means clustering algorithm partitions the pixels of an image into k clusters with k centroids, but do those centroids represent the k most prominent colors of an image? Just how useful is k-means clustering is for determining the color theme of an image? We consider the results of the algorithm to be useful if the algorithm outputs the "correct" colors, but it can be argued that the color theme of an image is a subjective, qualitative trait.



"Mona Lisa"
Leonardo da Vinci

"The Disintegration of the Persistence of Memory"
Salvador Dali

https://www.rutgers.edu/research
Rutgers University

https://www.minimalistbaker.com
Dana Shultz

Fig.5. To examine this, we surveyed twenty people (11 women and 9 men, all 17 years or older). We asked them to consider four images (see Fig.5) and choose four colors that they thought were the most prominent in the image. We compared the participants' responses to the results of our Python-implemented k-means clustering with k = 4 (with t = 25 and ε = 0.5). These results are represented in Fig.6. The first set of 4 columns reflects data for image 1, the next set of 4 columns reflects data for image 2, etc. The four resulting centroids determined by k-means clustering (topmost blocks of colors) are separated from participant choices by a red line. Each row represents the participant's 16 choices (4 choices per each of the 4 images)

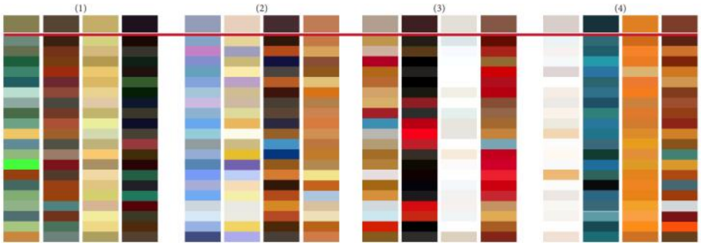


Fig.6. For images 1, 2, and 4, we visually observe that many participant's color choices have some acceptable correlation that supports the usefulness of *k*-means clustering for determining an image's dominant color theme. For example, for image 2, both the algorithm and most participants agreed that the color theme should consist of blue-purple and various shades of brown. Similar claims can be made for images 1 and 4

However, for image 3, the *k*-means clustering algorithm chooses 4 colors that resemble a monochromatic brown theme. Yet the majority of the surveyed people chose a bright red or orange hue that is not represented in the *k*-means result. One could argue that for this image, the *k*-means clustering result for the four most prominent colors is not a useful result; the *k*-means clustering result does not accurately represent what a human perceives to be the important colors of this image. From this observation, it is prudent to discuss the limitations of the *k*-means algorithm. Why might it fail to produce a result that significantly corresponds to human opinion?

One reason may be that the *k*-means algorithm does not make any association between object recognition and identifying color, the way a human might. The algorithm also considers background colors to be equally as important as foreground colors. The *k*-means algorithm is not tailored for the purpose of image analysis, but generally for partitioning data into clusters. It operates without regard for the order or positioning of the original data points.

This can make a difference in image analysis because the positioning of the pixels in an image is important to human color perception. The Fig.7 is image 3 from the survey. The Fig.8 is image 3 with all pixels randomly re-arranged. The Fig.9 is also composed of all the pixels from image 3 re-arranged, but this time arranged into a position that supports the algorithm's result. If a human were asked to identify the prominent color theme of those three images, they might have three very different responses. However, these images each contain the exact same data set of colors. Thus, the *k*-means clustering algorithm would output the exact same color themes for all three images (if the same 4 initial centroids are used for each image).

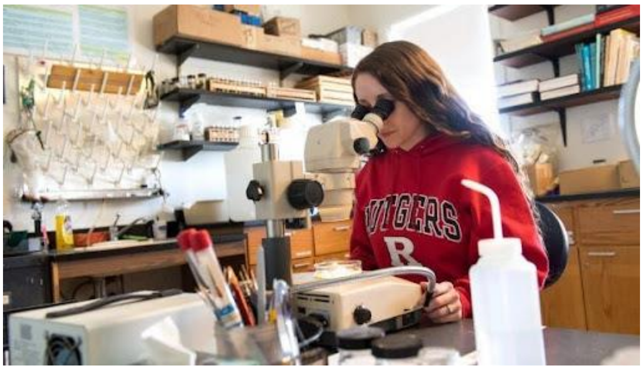


Fig.7. Image from Survey

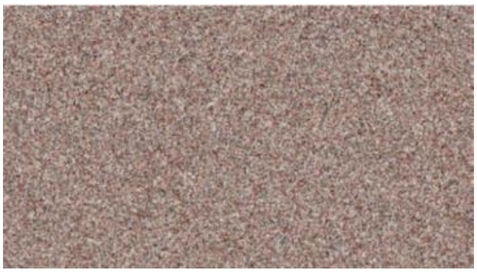


Fig.8. Randomly rearranged pixel

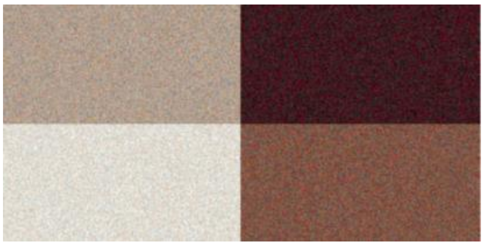


Fig.9. Rearranged Pixel

There are ways to incorporate object detection into the *k*-means clustering algorithm's criteria for determining prominent colors. There are established studies in artificial intelligence and supervised learning related this purpose [44]. However, there is a cost in runtime and complexity. The *k*-means algorithm is a very valuable image processing tool because of its simplicity and linear runtime. Adding object recognition criteria may be costly and especially complex if your data set lacks defined and recognizable features.

Another limitation of *k*-means clustering is that results can vary in usefulness depending on the value of *k*. For some images, the colors that a human perceives to be important may not appear in the *k*-means clustering results if *k* has a low value. For example, we noticed that a red color was consistently chosen by survey participants responding to image 3, yet the algorithm did not output a similar color. However, when executing *k*-means clustering with $k = 6$ instead of $k = 4$ on the same image, a bright red hue was included in the final centroids. The algorithm was only able to isolate this color with a higher *k* value. Furthermore, it can be difficult to determine what *k* value is too low for an image. We found that *k*-means clustering can produce excellent results with $k = 1$ for an image whose most prominent color dominates over half of the pixels, but we have also observed instances that indicate the need for *k* values greater than 10. In addition, it is possible that the *k*-means clustering algorithm may output outlier colors as dominant, especially if an outlier color is selected as an initial centroid. Users may need to account for this by implementing the repeated *k*-means strategy to choose the most compact result, or by implementing a refinement algorithm to discard centroids with small clusters.

Even though *k*-means clustering has its limitations, it is still highly applicable and relevant for quickly determining prominent colors in large diverse data sets.

6. CONCLUSION

The *k*-means clustering algorithm is an efficient tool for determining color themes of large, diverse sets of images. The simplicity of *k*-means clustering as an unsupervised learning algorithm makes it applicable to many types of images. The runtime of *k*-means clustering is proportional to the number of pixels being processed, the maximal number of iterations, and the value of *k*.

The *k*-means clustering algorithm minimizes the distance between centroid values and the data points in their respective clusters, and the produced centroid colors are highly successful in accurately representing an image's most prominent colors. For most images, *k*-means clustering results can be expected to have a strong correlation to survey participant responses. For the minority of images where *k*-means clustering results do not strongly correlate to human opinion, users can address these issues depending on the use case. For example, to prevent the program from choosing outlier colors, users can implement code that checks the sizes of clusters and rejects centroids from very small clusters. If algorithm simplicity is not a priority, *k*-means clustering can be supplemented by object recognition technology to achieve more "human" results.

This study's primary finding is that the application of *k*-means clustering for determining prominent colors can be further optimized (with visually indistinguishable error) by using *k*-means with sampling. Our research shows that for images with high resolutions (containing 5000 pixels or more), *k*-means with sampling with as little as 10% of the pixels will yield results nearly identical to applying *k*-means clustering to all pixels. In this scenario, *k*-means with sampling will execute in 10% of the runtime, which has promising implications for optimizing industry applications.

K-means with sampling is a valuable tool for the many services that analyze large and diverse data sets for color representation. *K*-means with sampling is a simple, fast, and customizable to users' runtime and convergence requirements. The ability to identify near-optimal centroids in images, and only process a small fraction of pixels, significantly increases efficiency in image processing applications. *K*-means with sampling to determine prominent colors in images has applications across all systems and projects that rely on image analysis. Our findings in runtime improvement and limitation analysis may be useful in the industries of entertainment and art, as well as in science and security.

REFERENCES

[1] S. Na, L. Xumin and G. Yong, "Research on K-Means Clustering Algorithm: An Improved K-Means Clustering Algorithm", *Proceedings of Symposium on Intelligent Information Technology and Security Informatics*, pp. 63-67, 2010.

[2] M.W. Celebi, "*Color Medical Image Analysis*", Springer, 2013.

[3] M. Mahajan, P. Nimbhorkar and K. Varadarajan, "The Planar K-Means Problem is NP-Hard", *Theoretical Computer Science*, Vol. 442, pp. 13-21, 2012.

[4] I.P. Kumar, V.P.H. Gopal, S. Ramasubbareddy, S. Nalluri, and K. Govinda, "*Dominant Color Palette Extraction by K-Means Clustering Algorithm and Reconstruction of Image*", Springer, pp. 921-929, 2020.

[5] I.C. Mogotsi, C.D. Manning, P. Raghavan, and H. Schütze, "*Introduction to Information Retrieval*", Cambridge University Press, Cambridge, 2008.

[6] A. Broder, L. Garcia-Pueyo, V. Josifovski, S. Vassilvitskii, and S. Venkatesan, "Scalable K-Means by Ranked Retrieval", *Proceedings of ACM International Conference on Web Search and Data Mining*, pp. 233-242, 2014.

[7] P.S. Bradley and U.M. Fayyad, "Refining Initial Points for K-Means Clustering", *Proceedings of International Conference on Machine Learning*, pp. 91-99, 1998.

[8] P. Franti and S. Sieranoja, "How Much Can k-Means be Improved by using Better Initialization and Repeats?", *Pattern Recognition*, Vol. 93, pp. 95-112, 2019.

[9] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman and A.Y. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 881-892, 2002.

[10] C. Yuan and H. Yang, "Research on K-Value Selection Method of K-Means Clustering Algorithm", *Multidisciplinary Science Journal*, Vol. 2, pp. 226-235, 2019.

[11] M. Luo, Y.F. Ma and H.J. Zhang, "A Spatial Constrained K-Means Approach to Image Segmentation", *Proceedings of International Conference on Multimedia*, Vol. 2, pp. 738-742, 2003.

[12] M. Omran, A.P. Engelbrecht and A. Salman, "Particle Swarm Optimization Method for Image Clustering", *International Journal on Pattern Recognition and Artificial Intelligence*, Vol. 19, pp. 297-321, 2005.

[13] R. Vitale, J.M. Prats-Montalban, F. Lopez-Garcia, J. Blasco and A. Ferrer, "Segmentation Techniques in Image Analysis: A Comparative Study", *Chemometrics*, Vol. 30, pp. 749-758, 2016.

[14] A. Molada Tebar and S. Westland, "Dominant Color Extraction with K-Means for Camera Characterization in Cultural Heritage Documentation", *Remote Sensing*, Vol. 12, pp. 520-534, 2020.

[15] C. Wu, B. Yan and R. Yu, "K-Means Clustering Algorithm and its Simulation Based on Distributed Computing Platform", *Complexity*, Vol. 2021, pp. 1-10, 2021.

[16] D. Kann, "As a Megadrought Persists, New Projections Show a Key Colorado River Reservoir could Sink to a Record Low Later this Year", Available at https://edition.cnn.com/2021/04/19/weather/western-drought- colorado-river-cutbacks-study/index.html, Accessed at 2021.

[17] K.N. Blain, "Adam Toledo's Killing is part of a Brutal Pattern of Child Killings in America", Available at: https://www.msnbc.com/opinion/adam-toledo-s-killing-part-brutal-pattern-child-killings-america-n1264432, Accessed at 2021.

[18] G. Collins and B. Stephens, "Tell Me the One About the Presidential Candidate Who Ran for Mayor", Available at: https://www.nytimes.com/2021/04/19/opinion/indianapolis-yang-giuliani-cuomo.html, Accessed at 2021.

[19] E. Goldberg and Y. Paskova, "They Told Her Women Couldn't Join the Ambulance Corps. So She Started Her Own", Available at:

https://www.nytimes.com/2021/04/19/us/ezras-nashim-womens-EMT.html, Accessed at 2021.

[20] A. Krueger, “Why Reopening Ceremonies Are So Important in New York Right Now”, Available at: https://www.nytimes.com/2021/04/16/nyregion/coronavirus-nyc-reopening.html, Accessed at 2021.

[21] S. Deb, “Regular People Keep Challenging N.B.A. and W.N.B.A. Players”, Available at: https://www.nytimes.com/2021/04/19/sports/basketball/why-the-worst-nba-player-is-probably-still-better-than-you.html, Accessed at 2021.

[22] C. Dema and M. Ives, “How Bhutan Out-Vaccinated Most of the World”, Available at: https://www.nytimes.com/2021/04/18/world/asia/bhutan-vaccines-covid.html, Accessed at 2021.

[23] A. Troianovski, “Navalny’s Network is Disbanding, Citing Pressure From Putin”, Available at: https://www.nytimes.com/2021/04/29/world/europe/navalny-group-putin-russia.html, Accessed at 2021.

[24] M. Eddy, “German Greens and Conservatives Choose Chancellor Candidates”, Available at: https://www.nytimes.com/2021/04/19/world/europe/germany-greens-chancellor-annalena-baerbock.html, Accessed at 2021.

[25] N. Cumming Bruce, “U.N. Panel Is Scathing in its Criticism of a British Report on Race”, Available at: https://www.nytimes.com/2021/04/19/world/europe/britain-race-united-nations-boris-johnson.html, Accessed at 2021.

[26] M. Caro, “Taking Over Victory Gardens to Make a ‘Theater for All’”, Available at: https://www.nytimes.com/2021/04/18/theater/ken-matt-martin-victory-gardens.html, Accessed at 2021.

[27] B. Brantley, “How Helen McCrory Shone, Even in a Haze of Mystery”, Available at: https://www.nytimes.com/2021/04/17/theater/helen-mccrory-appraisal.html, Accessed at 2021.

[28] A. Wall, “Finding Love and All Its Quirks, Even If 2,654 Miles Away”, Available at: https://www.nytimes.com/2021/04/30/fashion/weddings/sarah-lenz-stephen-paskey-wedding.html, Accessed at 2021.

[29] S. Sifton, “New Week! New Recipes!”, Available at: https://www.nytimes.com/2021/04/19/dining/new-week-new-recipes.html, Accessed at 2021.

[30] E. Asimov, “One Year Later: How U.S. Winemakers Averted Disaster”, Available at: https://www.nytimes.com/2021/04/15/dining/drinks/wine-pandemic.html, Accessed at 2021.

[31] New York Times, “Birds by the Billions: A Guide to Spring’s Avian Parade”, Available at: https://www.nytimes.com/2021/04/15/travel/birding-america.html, Accessed at 2021.

[32] P. McClanahan and D. Kamin, “52 Places, Virtually”, Available at: https://www.nytimes.com/2020/04/14/travel/52-places-to-go-virtual-travel.html, Accessed at 2021.

[33] C. Arisman, “On the Water in Alaska, Where Salmon Fishing Dreams Live On”, Available at: https://www.nytimes.com/2021/04/19/travel/alaska-salmon-fishing.html, Accessed at 2021.

[34] R. Kaysen, “The Chelsea Hotel Becomes a New York Battleground”, Available at: https://www.nytimes.com/2021/04/16/nyregion/chelsea-hotel-nyc.html, Accessed at 2021.

[35] S. Franklin, “Homes for Sale in Manhattan, Brooklyn and Queens”, Available at: https://www.nytimes.com/2021/04/15/realestate/housing-market-nyc.html, accessed August19,2021.

[36] M. Roach, “Why Diversity Is an Advantage in a Vegetable Plot”, Available at: https://www.nytimes.com/2021/04/14/realestate/why-diversity-is-an-advantage-in-a-vegetable-plot.html, Accessed at 2021.

[37] CNN Business, “Stock market news today: Dow and S&P 500 updates”, Available at: https://edition.cnn.com/business/live-news/stock-market-news-040721/index.html, Accessed at 2021.

[38] V. Greenwood, “Fairy Circles in Australia May Be Due to Microbes, Study Says”, Available at: https://www.nytimes.com/2021/04/12/science/fairy-circles-australia.html, Accessed at 2021.

[39] A. Mandavilli, “Could the Pandemic Prompt an ‘Epidemic of Loss’ of Women in the Sciences?”, Available at: 2021, https://www.nytimes.com/2021/04/13/health/women-stem-pandemic.html, accessed August 19, 2021.

[40] C. Siemaszko, “Michelle Obama Embraces George W. Bush: Why That Photo Was So Moving”, Available at: https://www.nbcnews.com/news/us-news/michelle-obama-embraces-george-w-bush-why-photo-was-so-n654451, Accessed at 2021.

[41] A.C. Elassar, “A giant, indoor vertical farm aims to bring jobs and fresh produce to Compton”, Available at: https://edition.cnn.com/2021/04/19/business/compton-vertical-farm-plenty-trnd/index.html, Accessed at 2021.

[42] K.C. Rogers, “Summer concerts are almost here, but will they be safe? What you should know.”, Available at: https://edition.cnn.com/travel/article/concert-music-festival-safety-pandemic-wellness/index.html, Accessed at 2021.

[43] “K-means clustering in OpenCV”, OpenCV, Available at: docs.opencv.org/master/d1/d5c/tutorial_py_kmeans_opencv.html, Accessed at 2021.

[44] H.T. Nguyen, E.H. Lee, C.H. Bae and S. Lee, “Multiple Object Detection Based on Clustering and Deep Learning Methods”, *Sensors*, Vol 20, pp. 4424-4434, 2020.