Vector Based approach for Color Image Quantization

Navneet Melarkode

School of Computer Science Engineering

Vellore Institute of Technology, Vellore, India

navneetmelarkode@gmail.com

Varun Agarwal

School of Computer Science Engineering

Vellore Institute of Technology, Vellore, India

varunswaika@gmail.com

Abstract

This paper proposes a novel algorithm for quantizing colors from an image and consequently forming a color palette associating with the said image. The produced quality of the color palette is better than the state-of-the-art algorithms while improving on the time complexity of the algorithm runtime as well as a novel approach to using lesser computational power when compared to its counterpart algorithms. The proposed algorithm employs graph algorithms by arranging all the pixels in the picture into an undirected graph. Each graph traversal selects one representative color. This process is repeated until a desired number of colors is reached. The algorithm can be used as a general vector quantization algorithm. This paper extends it to 3 dimensions, one catering to each component of the RGB values of the pixel.

Keywords

Color Image Quantization, Color Palette Generation, Vector Quantization, Image Processing

1. Introduction

In recent times, images play a vital role in shaping all the fields they are involved in. Color Image Quantization involves two significant procedures. The first being generating a color palette and the other being mapping every single pixel of the image to the closest color on the palette. This research concerns the former by generating a user generated number of colors for the palette of any image.

Every image consists of pixels, and each pixels has three components, red, blue and green components. Each pixel denotes a color and is represented as a tuple (red, green, blue). Each of these individual components can take a value from 0 to 255. Recent studies indicate common problem surrounding colors. The ambiguity around distance between two colors is still prevalent and there is no coherence around this topic. Some researchers find that Euclidean distance may not be the most apt color distance metric as it does not quantify for luminosity, while others say that the Hue distance metric does not take the base color into account.

Palette designing has gained lot of traction over the years. Some of the reasons of color quantization are bought into play only because of how integral the palette is as a preprocessing step as mentioned by G. Ramella [1]. Shu-Chien Huang [2] proposed a fast K-means color quantization method that builds upon the strengths of the traditional octree method and works towards improving the computational weight aspect. Another approach that uses a slight variant of K-Means algorithm for Color quantization uses the formulation of MacQueen as studied by S. Thompson et al. [3]. They incorporate an adaptive cluster finding method which increases the speed of algorithm runtime. Pérez-Delgado [4] use a Swarm intelligence-based color quantization method to generate the required color palette. In this particular study, the Artificial Bee Colony implementation was employed to facilitate the Quantization process. M. Frackiewicz et al. [5] employ a fast K-means clustering algorithm with functional changes that involve but are not limited to downsampling the original image to preserve the quality but reduce the iterations. Along similar lines, Z. Hu et al. [6] took inspiration from the multipopulation idea. Their research uses a self-adaptive hybrid differential evolution to generate different buckets indicative of color quantization. H. Park et al. [7] use self-organizing maps coupled with Octree as a means for color quantization. It improves upon the drawback of the traditional Self-organizing maps approach by successfully generating accurate results even when provided with images that have less colors. G. Schaefer and L. Nolle [8] mimic the human visual system by applying a different metric for evaluation. Instead of minimizing the mean squared error, the article tries to maximize a custom metric that works well for a variety of images. M. Celebi et al. [9] use hierarchical clustering to strike an optimal balance between efficiency and effectiveness. It builds upon the Lloyd-Max algorithm. H. Park et al. [10] work with self-organizing maps but showed that if the learning rates and the radius of the neighborhood can be dynamically adjusted during the runtime, the point of convergence can be achieved much faster than counterpart algorithms. C. Ozturk et al. [11] aim to limit the otherwise significant information loss in any image when it undergoes color quantization. To tackle this, they employ an Artificial Bee colony approach for the same. N. An and C. Pun [12] present a hybrid color image segmentation algorithm to produce more robust and reasonable results as compared to its predecessors. M. Celebi [13] propose a fast K-Means algorithm to achieve state of the art results while keeping the computational demands to a minimum. G. Tanaka et al. [14] cited the importance of color to grayscale conversion in modern time.

It is backed by mathematical property and evaluation with de facto algorithms. Y. Hu and B. Su [15] propose an accelerated K-means algorithm to significantly cut down on the original clustering algorithm K-means. X. Zhang et al. [16] implemented a two-stage color quantization method. The main steps involved the use of an LBG algorithm which helps iteratively filter the palette selection. M. Omran et al. [17] worked on employing Swarm intelligence for Color Image Quantization. Particle Swarm Optimization was used as their base algorithm and merged its features with the K-means clustering algorithm, to produce significant improvement as compared to other benchmark algorithms at that time. Z. Bing et al. [18] propose a novel algorithm for color quantization that takes into the account both, representative colors as well as the effective layer and color of each pixel. The algorithm is proven to have an adjustable characteristic to cater to quality aspect. E. Gu et al. [19] incorporate a color image quantization method to achieve minimum visible distortion. T. Taşdizen et al. [20] propose a novel approach for the color image quantization problem using a genetic approach. They are then evaluated with heuristic algorithms and the traditional K-means algorithm. P. Scheunders [21] compared and contrasted the various algorithms of the time used in Color Image Quantization. The main algorithm analysed was the C-means algorithm and it was understood that the said clustering algorithm fizzes out in superiority when it comes to computing time. The other algorithms compared involved hierarchical competitive learning algorithms. P. Scheunders [22] propose a novel clustering algorithm that involves the use of genetic algorithms combined with the classical C-means clustering algorithm. The genetic approach combination results in an enhanced quality of reconstructed color quantized image. G. Joy and Z. Xiang [23] make simple modifications to the existing median cut algorithm to generate a new revamped Color Image Quantization tool. The first step selects the longest longest-dimension instead of the traditional approach of highest pixel count. Instead of bisecting the box's pixel count, the algorithm bisects the center of the color box thus formed from the first step. Ultimately, a 3-2-4-bit cutting color reduction is employed instead of the conventional 3-3-3-bit cutting method. M. Gervautz and W. Purgathofer [24] proposed the famous octree algorithm that set the tone for all further studies in color image quantization. K. Shelley and D. Greenberg [25] describe the use of Color Image Quantization in the field of Frame Buffer Display. The main aim of the study is to identify high quality image displays using Color Image Quantization on frame buffers.

All the aforementioned methods produce images that are made with quantized colors, but they all come with certain trade-offs. While some algorithms are computationally intensive, the others overcome the same by compromising on the generated image's quality. The aim of this paper is to achieve parity between various interests to approximate the similarity between two colors and accurately generate a color palette. The proposed algorithm creates a weighted graph of all the pixel coordinates with the weight being the number of times it occurred in the given picture. Instead of employing a Euclidean approach, the algorithm emulates a cubic boundary region around the heaviest nodes and removes them from the graph. This step eliminates redundant colors and thus by doing so, maximizes the variety and richness of colors extracted from the image for the palette generation. The process is repeated until the desired number of nodes are left on the graph. The algorithm achieves state-of-the-art results while maintaining the time and space complexity. The model has been compared with well-known Image processing algorithms such as Median cut, K-Means and Octree color quantization. The proposed algorithm involves features from both splitting and clustering algorithms to achieve results.

The remainder of this study is as follows. Section 2 involves a literature survey on most of the color palette as well as color quantization studies performed previously. Section 3 discusses the proposed algorithm and the workflow for the same while Section 4 deals with the experiment.

2. Related work and methods

In this section, we describe the major algorithms often used as the base algorithm for various color image quantization studies as well as color palette generation works.

2.1 Octree

Research studies [7] and [20] use the Octree color image quantization as their base algorithm. The Octree algorithm has been around for many years and inspired many future studies in the field of Color Image Quantization. The Octree algorithm can be group under a hierarchical clustering method. The major drawback concerning the Octree algorithm is the performance throughput of the algorithm.

The algorithm employs the tree data structure. Each node in the tree has up to eight nodes as children. The Octree color quantization method generate color palettes without considering the frequency of each color in the said image. This gives rise to a simple edge case which may occur given the vast variety of images present in the world. For an image with many colors but a highly uneven frequency distribution for each of the colors, the Octree color Image Quantization Algorithm will return a very poor result because of not taking the weighted sum which discards the ambiguity caused by the frequency of each color.

2.2 K-Means

[2-5],[14],[15] employ a modified K-Means clustering algorithm in order to generate the color palette and quantize the color image provided. Despite the modifications, all of the variants are heavily influenced and built upon the base algorithm K-Means clustering algorithm. The K-means algorithm gathers all the pixels and performs a pixel-wise vector quantization over the entire image. To initialize the palette, the algorithm randomly assigns k cluster heads. In each iteration, the pixels are grouped under each cluster depending on the Euclidean distance between itself and the cluster head. The cluster heads are updated with every iteration of the algorithm ensuring the changes in the color dimension are reflected. K-Means algorithm gives accurate results in the field of Color Image Quantization but it comes at the cost of high computational necessities. This is chiefly the reason for all the further researched for K-Means clustering in Color Image Quantization aim to improve the computation time while maintaining a similar quality of results generated.

2.3 Distance metric

The standard method to measure distance between any two given colors is the Euclidean distance.

$$Distance = \sqrt{\sum_{i=1}^{3}(x[i] - y[i])^2} \qquad (1)$$

The Euclidean distance is the absolute distance between any two points in the vector plane. For the 3D color space, the Euclidean distance is modified accordingly to take into account the sum of squares of individual components of both the pixels. In equation (1), x and y represent two points on the 3D color plane. Every point on the 3D color plane constitutes of 3 components namely, Red, Green and Blue.

2.4 Evaluation standard

2.4.1 Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{3}(y[i][j] - \tilde{y}[i][j])^2 \qquad (2)$$

The Mean Squared Error (MSE) is the sum of squared differences of each individual components in the 3D color plane. In terms of evaluation, the MSE metric indicates how the quantized reconstructed image compares to the original image in terms of clarity.

2.4.2 Mean Absolute Error

The Mean Absolute Error (MAE) is the sum of absolute differences of each individual components in the 3D color plane. In terms of evaluation, the MAE metric indicates how the quantized reconstructed image compares to the original image in terms of vibrance.

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{3}|y[i][j] - \tilde{y}[i][j]| \qquad (3)$$

In equation (2) and (3), y and $\tilde{y}$ are the reconstructed and original images respectively. n is the number of pixels in the image. A key distinction between MSE and MAE to note is that the MSE penalizes the reconstructed image more for a highly deviated color palette.

3. Proposed method

We present an algorithm that produces state of the art color quantization results without being excessively computationally intensive. It builds on the advantages of traditional algorithms such as the K-means algorithm and the octree algorithm for Color Image Quantization. The proposed method is divided into 2 phases.

The first phase of the algorithm involves reduction of dimensions. The input image is passed through a function to generate a frequency map. For a given threshold distance, the reduction algorithm iterates over all the colors in the image and clusters all the colors within the threshold distance together using a weighted mean. Distance used for this purpose is the standard Euclidean distance where individual components of the pixel are squared and then the root of its sum is returned.

The second phase of the algorithm involves the color palette generation aspect of the proposed method. The output of the first phase is given as an input for this phase. The input is then converted to a graph with each color being represented as a node. The node is connected to other nodes if the distance between the two colors is less than the respective threshold distance. Once the graph is created, the graph iterates over the entire graph to check for the node with the maximum indegree. By doing so, we find out the color that is the most representative at any given point of time. This is done k times where k is the size of the palette to be generated. In this case, the distance metric used is the absolute distance where the sum of absolute differences of each of the color's individual components is returned. In every iteration, the most representative color along with all its neighbors are removed from the graph. This ensures that at any particular time, the color palette can never have the same representative colors.

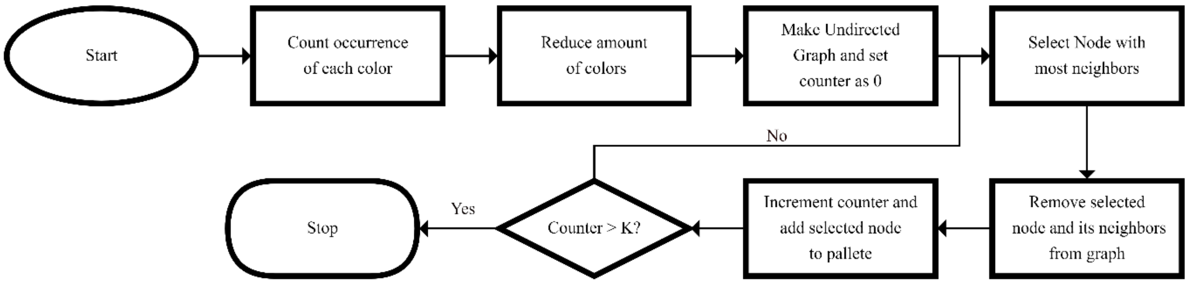

Figure 1: Flowchart for proposed algorithm

Figure 1 encapsulates the entire algorithm in a series of sequential steps. A reduction algorithm is applied to reduce the dimensions of the provided image and then it is subjected to a Undirected Graph Algorithm for the required size of color palette. With every iteration, a representative color is chosen and its neighbors are deleted. The algorithm ends once we obtain the required number of colors.

3.1 Graph creation

```
Initialize two-dimensional adjacency matrix
Initialize threshold distance d_max
for i ← 0 to length(color array)
```

```
    for j ← i+1 to length(color array)
        dis ← distance (color_i, color_j)
        if d_max greater than dis
            Add respective edge in adjacency matrix
    end for
end for
```

Algorithm 1: Creating undirected graph for 3D color space

3.2 Reduction algorithm

```
Initialize new color array [To store output]
Initialize frequency map for new color array
Initialize threshold distance d_max
for i ← 0 to length(color array)
    counter ← false
    for j ← 0 to length(new color array)
        dis ← distance (color_i, newcolor_j)
        if d_max greater than dis
            frequencymap_j ← frequencymap_j + 1
            counter ← true
    if counter is false
        Add color_i to new color array
Return new color array
```

Algorithm 2: Reducing dimensions of image before creating a graph

3.3 Vector Based - Color Image Quantization (VB-CIQ)

```
Generate frequency map of all colors in image
Initialize palette of k size
Use reduction algorithm on colors
```

```
Create a graph with the reduced number of colors
for i ← 0 to k
    maxdegree ← 0
    maxnode ← initnode
    for node in graph:
        if degree(node) > maxdegree
            maxdegree ← degree(node)
            maxnode ← node
    palette_i = color of maxnode
    for neighbor in maxnode.neighbors
        remove neighbor from graph
    remove maxnode from graph
return palette
```

Algorithm 3: VB-CIQ algorithm

4. Experimental setup and results

Our proposed method was evaluated on six standard image processing color images. The images are shown in figure 2. Additional information about the aforementioned images is given in the table 1 as follows.

Image name	Resolution	Number of colors
Yacht	512x512	150053
Sailboat	512x512	168459
Pepper	512x512	183525
Mandrill	512x512	15699
Lenna	512x512	148279
Airplane	512x512	77041

Table 1: Image used for Evaluation and Experimentation

Figure 2: Yachts, Sailboats, Pepper, Mandrill, Lenna, Airplane

All the images have been taken from http://www.hlevkin.com/hlevkin/06testimages.htm and http://www.imageprocessingplace.com/root_files_V3/image_databases.htm . The reason for choosing images from these sources was to test algorithms regarding Color Image Quantization on images that were uncompressed. Each of these images have a file format of either .tiff/.tif/.bmp. For high quality images, a Tagged Image File Format (TIFF) /Bitmap (BMP) file format serves as an excellent representative.

Methods	Features of Methods
Octree	(i) Underlying tree structure. Node capacity 8 children at max. (ii) Nodes pruned until desired number of colors obtained
K-Means	(i) Random initialization of required number of colors in palette (ii) Clustering algorithm on the basis of minimum distance to cluster head (iii) Centroid of all nodes in cluster is cluster head
Maximum Coverage	(i) Uniformly visits all pixels (ii) Neural network-based pixel selection
Median Cut	(i) Recursive algorithm that quantizes colors into buckets (ii) Ordering on the basis of color component with maximum range

Table 2: Color Quantization methods

Table 2 surveys and compares the benchmark algorithms that we used to evaluate the performance of our proposed method. The table also explains and elucidates each of the aforementioned algorithms. The algorithms include K-Means, Fast Octree, Median cut and Maximum coverage in no particular order.

IMAGE	METHOD	Size of Palette					
		8	16	32	64	128	256
LENNA	VB-CIQ	**503.57**	369.28	**207.75**	**130.69**	**67.29**	**42.31**
	FASTOCTREE	987.65	507.32	450.46	265.21	122.62	55.95
	K-MEANS	911.16	1022.71	663.17	515.05	455.07	292.21
	MAXIMUM	2213.19	1173.66	507.05	292.80	150.82	80.74
	MEDIAN CUT	610.96	**362.09**	234.57	140.24	86.13	55.72
AIRPLANE	VB-CIQ	798.20	**326.32**	**154.43**	**66.13**	**48.53**	**29.76**
	FASTOCTREE	1927.09	575.31	289.58	140.95	63.25	41.80
	K-MEANS	798.60	600.43	399.43	285.63	209.16	160.30
	MAXIMUM	2574.37	1316.23	548.39	218.78	154.85	78.22
	MEDIAN CUT	**755.16**	531.59	344.63	181.62	117.84	72.50
MANDRILL	VB-CIQ	2144.22	865.10	**465.02**	**228.15**	**87.91**	**11.14**
	FASTOCTREE	4301.89	1794.17	576.35	414.42	214.95	30.59
	K-MEANS	1877.52	1514.89	1413.24	963.01	686.76	472.76
	MAXIMUM	2654.93	1128.41	588.52	249.65	90.50	24.49
	MEDIAN CUT	**1341.51**	**769.34**	480.04	285.08	166.67	75.67
PEPPER	VB-CIQ	1515.35	**604.70**	**408.59**	**241.28**	**182.73**	147.48
	FASTOCTREE	2414.58	918.65	526.52	421.40	261.57	**117.32**
	K-MEANS	1460.67	1337.52	914.32	704.59	477.39	392.46
	MAXIMUM	5518.01	2363.77	1021.70	533.61	282.80	159.28
	MEDIAN CUT	**1181.08**	640.87	459.76	326.85	225.61	144.02
SAILBOAT	VB-CIQ	**1058.43**	**509.99**	**338.09**	**201.43**	**135.67**	**111.10**
	FASTOCTREE	3596.12	1383.01	487.09	396.79	217.35	116.95
	K-MEANS	1221.18	1177.95	643.19	577.15	486.30	356.20
	MAXIMUM	5763.44	1739.15	1134.58	654.31	294.97	174.13
	MEDIAN CUT	1141.98	660.84	359.15	243.69	165.43	113.92
YACHT	VB-CIQ	2052.36	**620.67**	**394.34**	**207.86**	**119.77**	**80.51**
	FASTOCTREE	3928.63	983.25	503.91	375.00	225.65	116.62
	K-MEANS	1664.10	1273.82	948.01	773.44	555.27	371.46
	MAXIMUM	3993.17	1789.83	816.95	530.81	272.50	146.50
	MEDIAN CUT	**1071.80**	641.85	411.96	239.71	156.03	102.07

Table 3: MSE values for all test images using the benchmark algorithms and the proposed method.

IMAGE	METHOD	Size of Palette					
		8	16	32	64	128	256
LENNA	VB-CIQ	**29.06**	**25.18**	**19.05**	15.49	**10.81**	**8.48**
	FASTOCTREE	36.76	30.53	28.63	19.77	13.31	10.29
	K-MEANS	41.28	42.81	35.05	29.98	28.79	23.01
	MAXIMUM	70.16	51.14	32.64	24.94	17.74	12.87
	MEDIAN CUT	32.34	24.69	19.13	**14.65**	11.36	9.04
AIRPLANE	VB-CIQ	**24.14**	**16.96**	**13.41**	**10.39**	8.91	**6.55**
	FASTOCTREE	34.68	23.88	21.90	12.99	9.58	8.57
	K-MEANS	35.72	29.10	26.49	20.64	17.05	15.89
	MAXIMUM	75.61	55.18	35.19	21.39	18.26	12.73
	MEDIAN CUT	24.90	19.37	15.18	10.67	**8.45**	6.56
MANDRILL	VB-CIQ	57.49	37.41	**28.19**	**18.51**	**9.45**	**3.88**
	FASTOCTREE	73.75	47.24	32.74	25.29	15.30	5.95
	K-MEANS	59.18	52.31	51.48	41.06	35.31	29.65
	MAXIMUM	72.64	47.24	33.65	20.89	12.38	6.86
	MEDIAN CUT	**50.01**	**37.34**	28.97	21.71	16.16	10.26
PEPPER	VB-CIQ	43.03	31.46	25.59	**19.38**	**14.67**	**11.79**
	FASTOCTREE	54.97	34.86	30.00	26.35	19.19	13.10
	K-MEANS	47.52	47.01	39.15	33.45	28.54	26.09
	MAXIMUM	105.71	66.88	46.14	32.68	23.90	17.89
	MEDIAN CUT	**42.57**	**30.82**	**25.28**	20.49	16.35	12.68
SAILBOAT	VB-CIQ	**37.98**	**29.04**	24.19	18.46	**13.43**	**11.27**
	FASTOCTREE	54.76	36.36	29.40	25.23	17.23	13.05
	K-MEANS	45.52	43.51	33.83	31.61	30.06	25.04
	MAXIMUM	109.31	61.13	49.62	37.58	24.75	19.25
	MEDIAN CUT	39.71	29.46	**21.87**	**17.72**	14.38	11.72
YACHT	VB-CIQ	53.79	33.99	26.42	18.96	**13.35**	**10.56**
	FASTOCTREE	69.60	36.57	29.70	23.58	17.13	12.38
	K-MEANS	54.72	44.49	39.76	37.63	30.12	25.44
	MAXIMUM	92.41	60.64	40.95	33.00	23.85	17.33
	MEDIAN CUT	**43.20**	**31.47**	**24.60**	**18.53**	14.64	11.57

Table 4: MAE values for all test images using the benchmark algorithms and the proposed method

Table 3 and 4 depict the performance of our algorithm when pitted against the said algorithms using the MSE and MAE metric. It is to be noted that the reconstructed image using the palette generated from the algorithms will be compared pixel to pixel with the original image. This implies that lower the MSE and MAE values, higher the accuracy of color quantization for the said image.

From Table 3 and 4, it is evident that the VB-CIQ algorithm consistently outperforms state of the art techniques used for color quantization. The Fast Octree Color Image Quantization algorithm compromises on the quality of quantized colors to gain an edge in computational speed. The K-Means algorithm has a very flat gradient descent as the value of k increases as compared to the other algorithms. For a small value of k, the K-Means clustering algorithm produces exceptional results but for larger values of k, it outputs a lackluster color palette. The Maximum coverage algorithm has trouble adjusting to the smaller k values as it tries to accommodate for every color in the image. Less frequently occurring unique colors are given equal priority which skews the MSE and MAE values against its favor. For higher values of k, the maximum coverage can aptly represent all the colors as they are which consequently allows the algorithm to produce a reasonable color palette. The median cut has consistently strived to be the benchmark algorithm for all color quantization-based researches. The findings from our experiment validate the same as it constantly achieves low errors for all values of k invariably. More often than not, our proposed method, the VB-CIQ outperforms its counterpart algorithms. At times, the median cut records a lower MSE and MAE value, but it is evened out for multiple values of k.

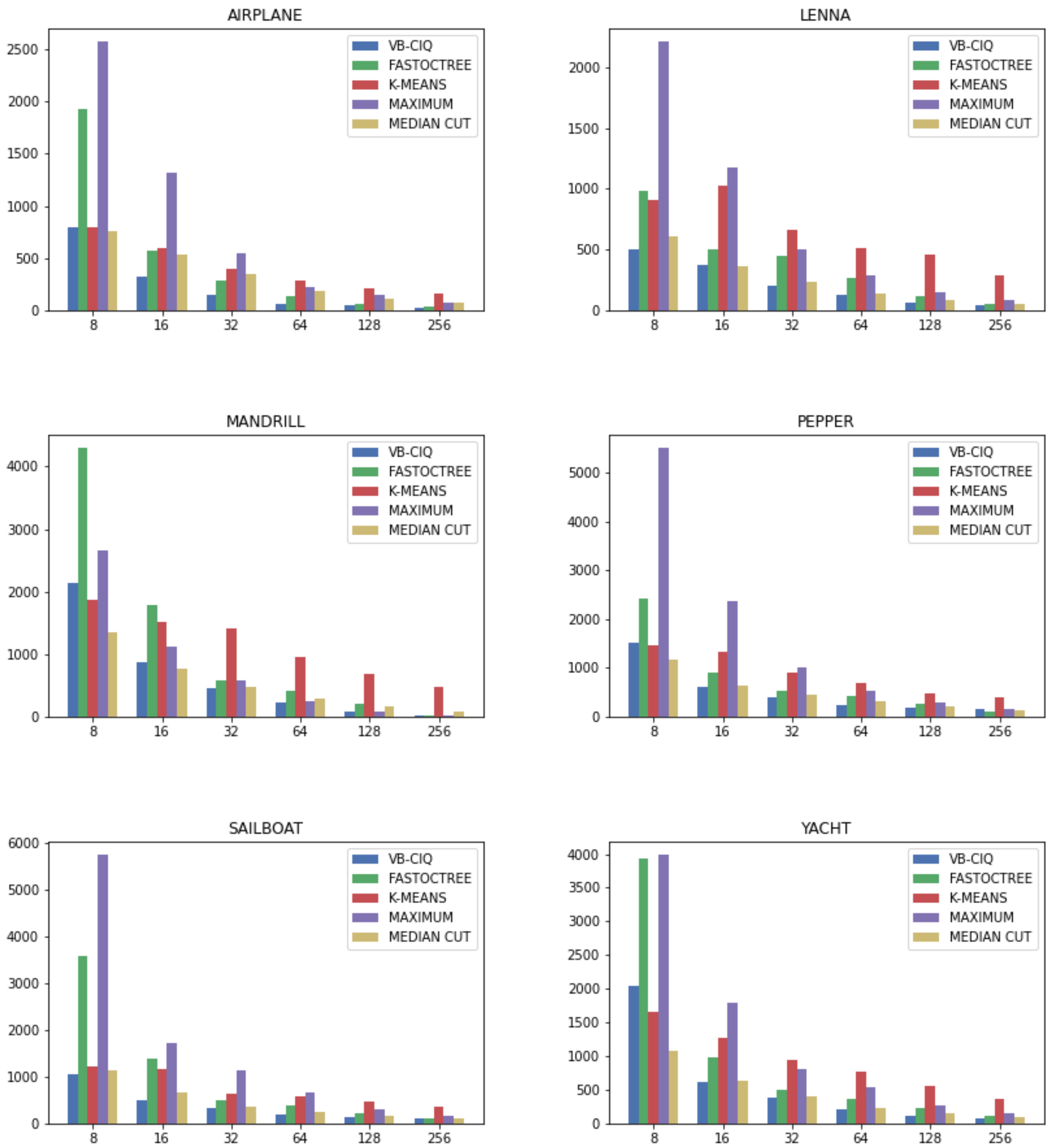

Figure 3: MSE values of each method

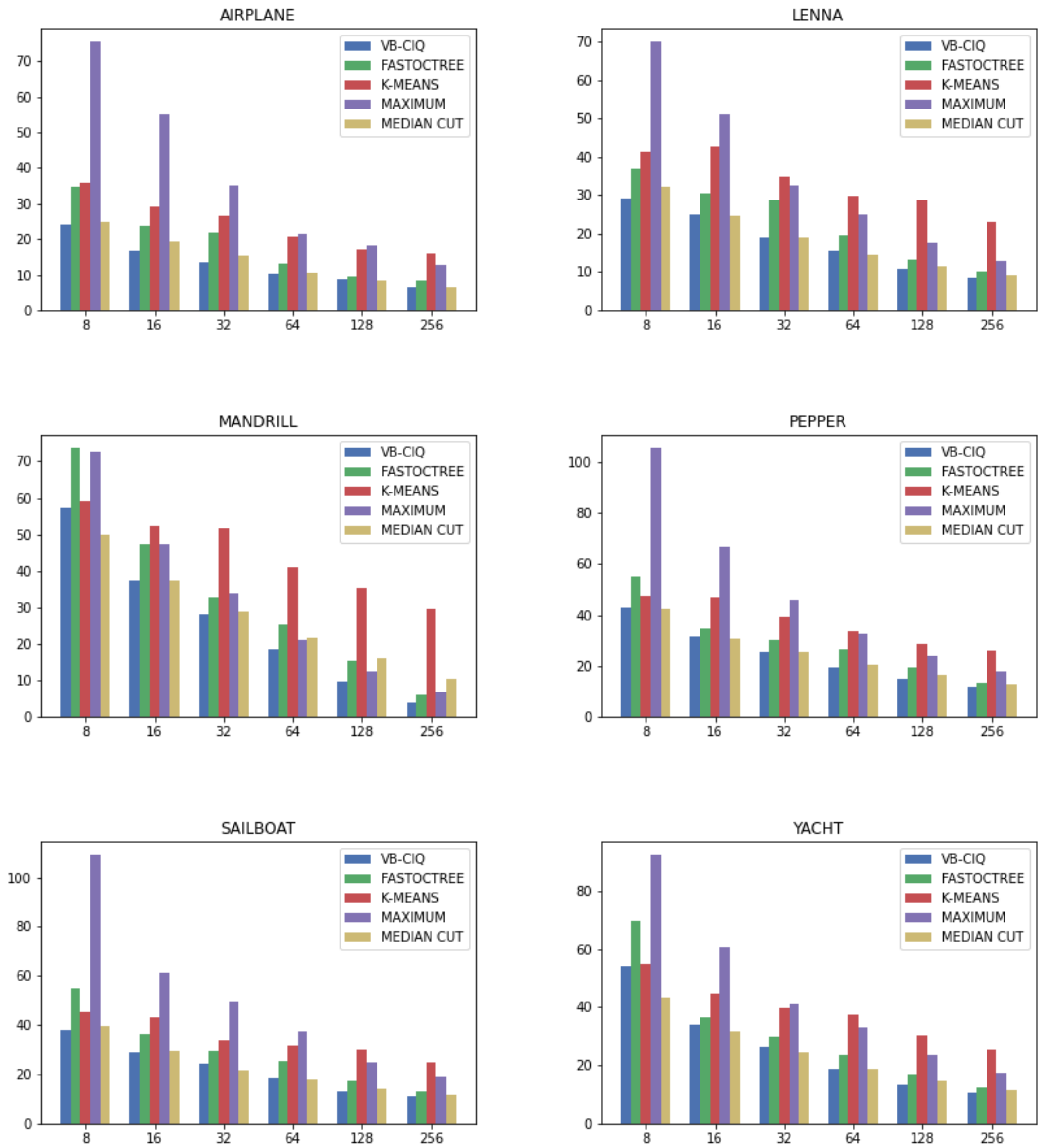

Figure 4: MAE values of each method

From Figures 3 and 4, the flat gradient descent of K-means clustering algorithm is starkly evident. Furthermore, the Maximum Coverage algorithm showcases its adaptability to higher values of k despite starting off poorly for lower values. The compromise in quality made by the Fast Octree algorithm is highlighted in flashes as seen in Mandrill and Yacht graphs. The median cut and VB-CIQ produce equally good results consistently in all the tests as seen from the graphs.

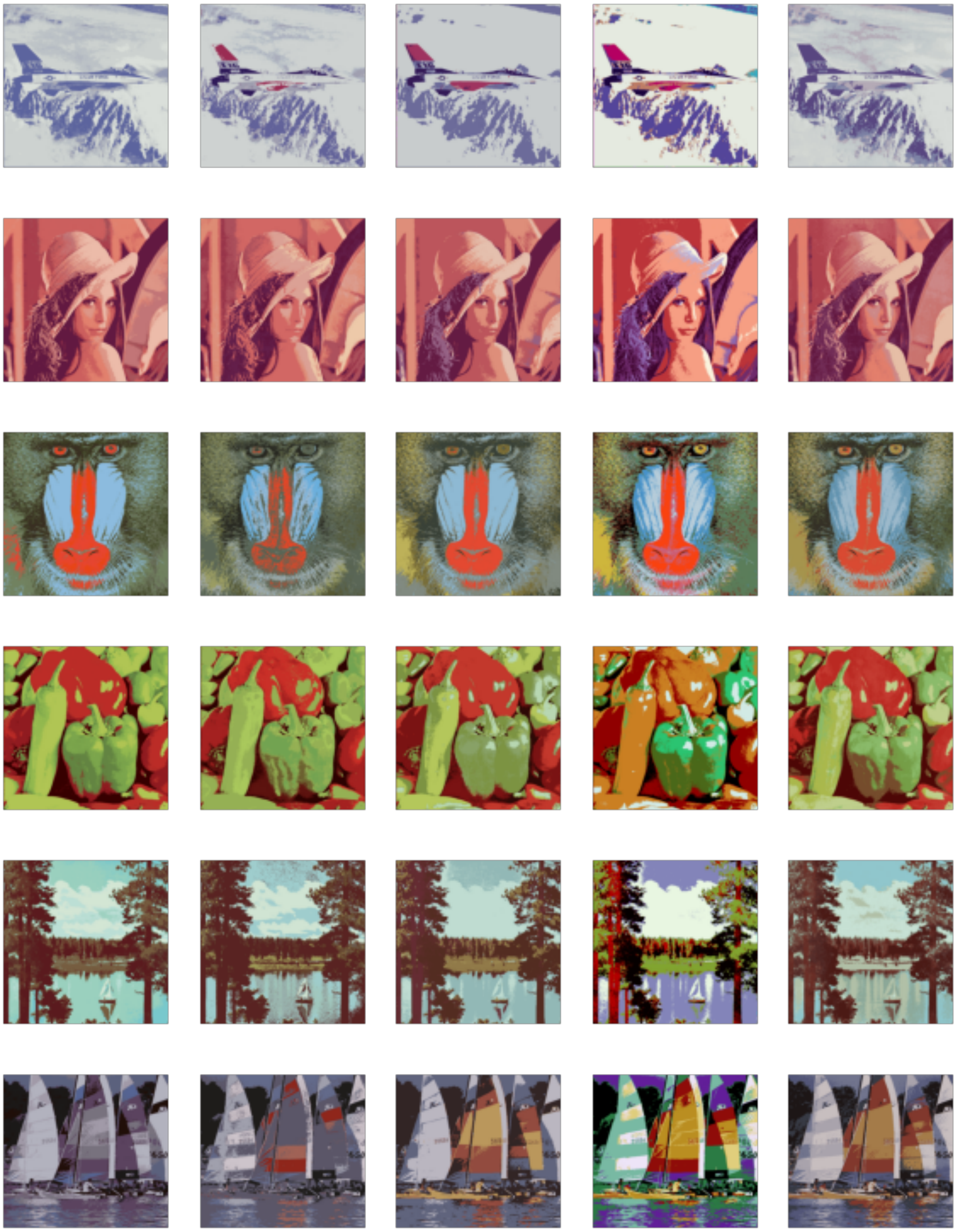

Figure 5: k value of 8 color quantization of VB-CIQ, Fast Octree, K-Means, Maximum Coverage, Median Cut respectively for all test images

Figure 5 shows the reconstructed quantized color image by using the colors present in the palette generated by the algorithms. The VB-CIQ prefers colors that are similar which in turn leads to capturing maximum detail from the image. In Figure 5 (a) Airplane, the text on the chassis of the airplane is not convoluted as the other algorithms. The general shadows and the depth of various objects are encapsulated to a high degree. Consequently, this results in a low vibrant and duller color palette. For a diverse range of colors, the K-Means tries to maximize depth while keeping a vibrant color palette. This results in a slightly more active image but loses the unique characteristics of the original image. The Fast Octree neither captures shadow nor does it highlight the detailing, but the Fast Octree ensures that no color is left out irrespective of the k value. Maximum coverage only targets the vibrance of the color palette, thus losing the originality of the image. All the colors feel manufactured as it tries to cover every color in a small palette of size 8. The median cut maintains a balance between the detailing and preserving the original colors. From Figure 5 (f) Yachts, the Median cut algorithm manages to showcase the representative colors and the details but, from Figure 5 (a) Airplane, it struggles to achieve the same balance. From this figure, we can conclude that our proposed model, VB-CIQ matches or outperforms the Median cut algorithm in most images visually, and not only in the theoretical aspect.

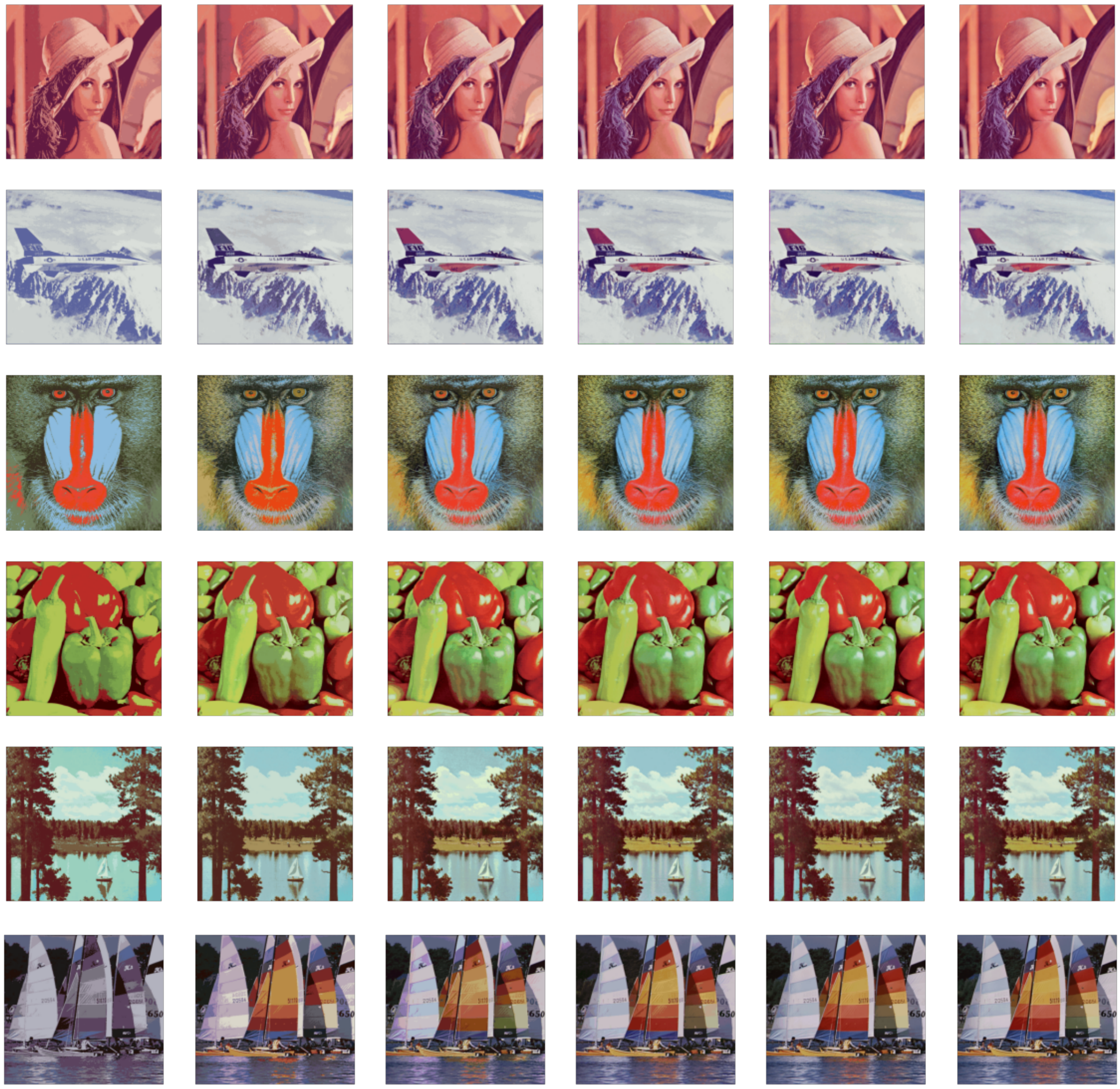

Figure 6: VB-CIQ results for k value 8, 16, 32, 64, 128 and 256 respectively for all test images.

From Figure 6 we can evidently follow the transition of the VB-CIQ color quantization as the K values increase. For a smaller value of K, the algorithm focuses on highlighting the depth, but as the values of K increase, VB-CIQ manages to introduce new colors while maintaining the same level of depth. This can be considered as the reason for its remarkable results when compared to state-of-the-art algorithms. From Figure 6 (f) Yachts, the algorithm slowly introduces the colors but maintains its superior edge over other algorithms by showcasing the depth of the reconstructed image from the very beginning. Ultimately, for a higher value of K, like 256, VB-CIQ can reproduce original images to a very high degree of accuracy while using limited colors.

5. Conclusion

This paper successfully proposes and evaluates a Color Image Quantization algorithm. The proposed algorithm, Vector Based Color Image Quantization (VB-CIQ) algorithm uses an undirected graph to theoretically select the most representative color from a given image. The VB-CIQ uses a novel color reduction method before creating the graph to save computational power while maintaining the same quality of results.

The VB-CIQ is evaluated with state-of-the-art algorithms on standard test images using Mean Squared Error and Mean Absolute Error as the comparison metric. The proposed method consistently yields exceptional results both, theoretically and visually. The algorithm focuses on highlighting the depth for lower values of k, while it generates near accurate images for higher values of k. Future studies can aim to improve the luminosity of color palette for lower values of k.

It is expected that the VB-CIQ serves as the benchmark for future studies in Color Image Quantization. The proposed algorithm can be used in various applications concerning the same.

6. Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] G. Ramella, "Evaluation of quality measures for color quantization," *Multimed. Tools Appl.*, vol. 80, no. 21–23, pp. 32975–33009, 2021, doi: 10.1007/s11042-021-11385-y.

[2] S. C. Huang, "An efficient palette generation method for color image quantization," *Appl. Sci.*, vol. 11, no. 3, pp. 1–17, 2021, doi: 10.3390/app11031043.

[3] S. Thompson, M. E. Celebi, and K. H. Buck, "Fast color quantization using MacQueen's k-means algorithm," *J. Real-Time Image Process.*, vol. 17, no. 5, pp. 1609–1624, 2020, doi: 10.1007/s11554-019-00914-6.

[4] M. L. Pérez-Delgado, "The color quantization problem solved by swarm-based operations," *Appl. Intell.*, vol. 49, no. 7, pp. 2482–2514, 2019, doi: 10.1007/s10489-018-1389-6.

[5] M. Frackiewicz, A. Mandrella, and H. Palus, "Fast color quantization by K-Means clustering combined with image sampling," *Symmetry (Basel).*, vol. 11, no. 8, 2019, doi: 10.3390/sym11080963.

[6] Z. Hu, Q. Su, and X. Xia, "Multiobjective Image Color Quantization Algorithm Based on Self-Adaptive Hybrid Differential Evolution," *Comput. Intell. Neurosci.*, vol. 2016, no. 1, 2016, doi: 10.1155/2016/2450431.

[7] H. J. Park, K. B. Kim, and E. Y. Cha, "An Effective Color Quantization Method Using Octree-Based Self-Organizing Maps," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/5302957.

[8] G. Schaefer and L. Nolle, "A Hybrid Color Quantization Algorithm Incorporating a Human Visual Perception Model," *Comput. Intell.*, vol. 31, no. 4, pp. 684–698, 2015, doi: 10.1111/coin.12043.

[9] M. E. Celebi, Q. Wen, and S. Hwang, "An effective real-time color quantization method based on divisive hierarchical clustering," *J. Real-Time Image Process.*, vol. 10, no. 2, pp. 329–344, 2015, doi: 10.1007/s11554-012-0291-4.

[10] H. J. Park, K. B. Kim, and E. Y. Cha, "An effective color quantization method using color importance-based self-organizing maps," *Neural Netw. World*, vol. 25, no. 2, pp. 121–137, 2015, doi: 10.14311/NNW.2015.25.006.

[11] C. Ozturk, E. Hancer, and D. Karaboga, "Color image quantization: A short review and an application with artificial bee colony algorithm," *Inform.*, vol. 25, no. 3, pp. 485–503, 2014, doi: 10.15388/Informatica.2014.25.

[12] N. Y. An and C. M. Pun, "Color image segmentation using adaptive color quantization and multiresolution texture characterization," *Signal, Image Video Process.*, vol. 8, no. 5, pp. 943–954, 2014, doi: 10.1007/s11760-012-0340-2.

[13] M. E. Celebi, "Improving the performance of k-means for color quantization," *Image Vis. Comput.*, vol. 29, no. 4, pp. 260–271, 2011, doi: 10.1016/j.imavis.2010.10.002.

[14] G. Tanaka, N. Suetake, and E. Uchino, "Derivation of the analytical solution of Color2Gray algorithm and its application to fast color removal based on color quantization," *Opt. Rev.*, vol. 16, no. 6, pp. 601–612, 2009, doi: 10.1007/s10043-009-0118-0.

[15] Y. C. Hu and B. H. Su, "Accelerated k-means clustering algorithm for colour image quantization," *Imaging Sci. J.*, vol. 56, no. 1, pp. 29–40, 2008, doi: 10.1179/174313107X176298.

[16] X. Zhang, Z. Song, Y. Wang, and H. Wang, "Color quantization of digital images," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3768 LNCS, no. 12, pp. 653–664, 2005, doi: 10.1007/11582267_57.

[17] M. G. Omran, A. P. Engelbrecht, and A. Salman, "A color image quantization algorithm based on Particle Swarm Optimization," *Inform.*, vol. 29, no. 3, pp. 261–269, 2005.

[18] Z. Bing, S. Junyi, and P. Qinke, "An adjustable algorithm for color quantization," *Pattern Recognit. Lett.*, vol. 25, no. 16, pp. 1787–1797, 2004, doi: 10.1016/j.patrec.2004.07.005.

[19] E. Gu, D. Xu, J. Wang, and C. Chen, "<title>Perceptually based approach to color quantization</title>," *Image Matching Anal.*, vol. 4552, pp. 292–297, 2001, doi: 10.1117/12.441540.

[20] T. Taşdizen, L. Akarun, and C. Ersoy, "Color quantization with genetic algorithms," *Signal Process. Image Commun.*, vol. 12, no. 1, pp. 49–57, 1998, doi: 10.1016/s0923-5965(97)00035-0.

[21] P. Scheunders, "A comparison of clustering algorithms applied to color image quantization," *Pattern Recognit. Lett.*, vol. 18, no. 11–13, pp. 1379–1384, 1997, doi: 10.1016/S0167-8655(97)00116-5.

[22] P. Scheunders, "A genetic c-means clustering algorithm applied to color image quantization," *Pattern Recognit.*, vol. 30, no. 6, pp. 859–866, 1997, doi: 10.1016/S0031-3203(96)00131-8.

[23] G. Joy and Z. Xiang, "Center-cut for color-image quantization" *The visual computer.,* no. Heckbert 1982, pp. 62–66, 1993.

[24] M. Gervautz and W. Purgathofer, "A Simple Method For Color Quantization: Octree Quantization," *Graph. Gems*, pp. 287–293, 1990, doi: 10.1016/B978-0-08-050753-8.50061-9.

[25] K. L. Shelley and D. P. Greenberg, "Computer Graphics Volume 16, Number 3 July 1982," *Computer (Long. Beach. Calif).*, vol. 16, no. 3, pp. 157–166, 1982.