JEI Journal of Electronic Imaging

RESEARCH PAPER

CQ100: a high-quality image dataset for color quantization research

M. Emre Celebi[a,*] **and María-Luisa Pérez-Delgado**[b]

[a]University of Central Arkansas, Department of Computer Science and Engineering, Conway, Arkansas, United States

[b]University of Salamanca, Escuela Politécnica Superior de Zamora, Zamora, Spain

ABSTRACT. Color quantization (CQ) is a classical image processing operation that reduces the number of distinct colors in a given image. Although the idea of CQ dates back to the early 1970s, the first true CQ algorithm, median-cut, was proposed later in 1980. Since then, hundreds of publications have investigated the topic of CQ, proposing dozens of algorithms. A vast majority of these publications demonstrate their results on small datasets, containing a handful of images of mixed quality. Furthermore, the reproducibility of CQ research is often limited due to the use of private test images or public test images with multiple non-identical copies on the World Wide Web or restrictive licenses. To address these problems, we curated a large, diverse, and high-quality dataset of 24-bit color images called CQ100 and released it under a permissive license. We present an overview of CQ100 and demonstrate its use in comparing CQ algorithms.

 [DOI: 10.1117/1.JEI.32.3.033019]

Keywords: image processing; color quantization; data clustering; dataset; CQ100

Paper 230303V received Mar. 14, 2023; revised May 3, 2023; accepted May 11, 2023; published Jun. 7, 2023.

1 Introduction

24-bit color images have become commonplace since the turn of the millennium.[1,2] These images typically contain hundreds of thousands of distinct colors, which complicate their display, storage, transmission, processing, and analysis. Color quantization (CQ) is a common image processing operation that reduces the number of distinct colors in a given image. Figure 1 shows the pencils image (Ref. 3, CC0 license, 768×512 pixels) and its quantized versions with 4, 16, 64, and 256 colors obtained using the median-cut algorithm.[4] It can be seen that the reproduction is reasonably accurate with only 64 colors and is nearly indistinguishable from its original with 256 colors.

CQ is composed of two phases: color palette design and pixel mapping. In the former phase, a small set of colors, called the color palette, representing the input colors is selected, whereas, in the latter phase, each pixel in the input image is assigned to one of the palette colors. The purpose of CQ is to reduce the number of distinct colors in a given image to a significantly smaller number with minimal distortion. Since natural images often contain a wide range of colors, faithful reproduction of such images with a small color palette is a challenging problem. In fact, CQ can be characterized as a large-scale combinatorial optimization problem.[5]

There are several ways to classify CQ algorithms. Image-independent algorithms design a universal color palette without regard to any particular input image, whereas image-dependent ones design a custom color palette based on the distribution of the colors in a given input image.

*Address all correspondence to M. Emre Celebi, ecelebi@uca.edu

1017-9909/2023/$28.00

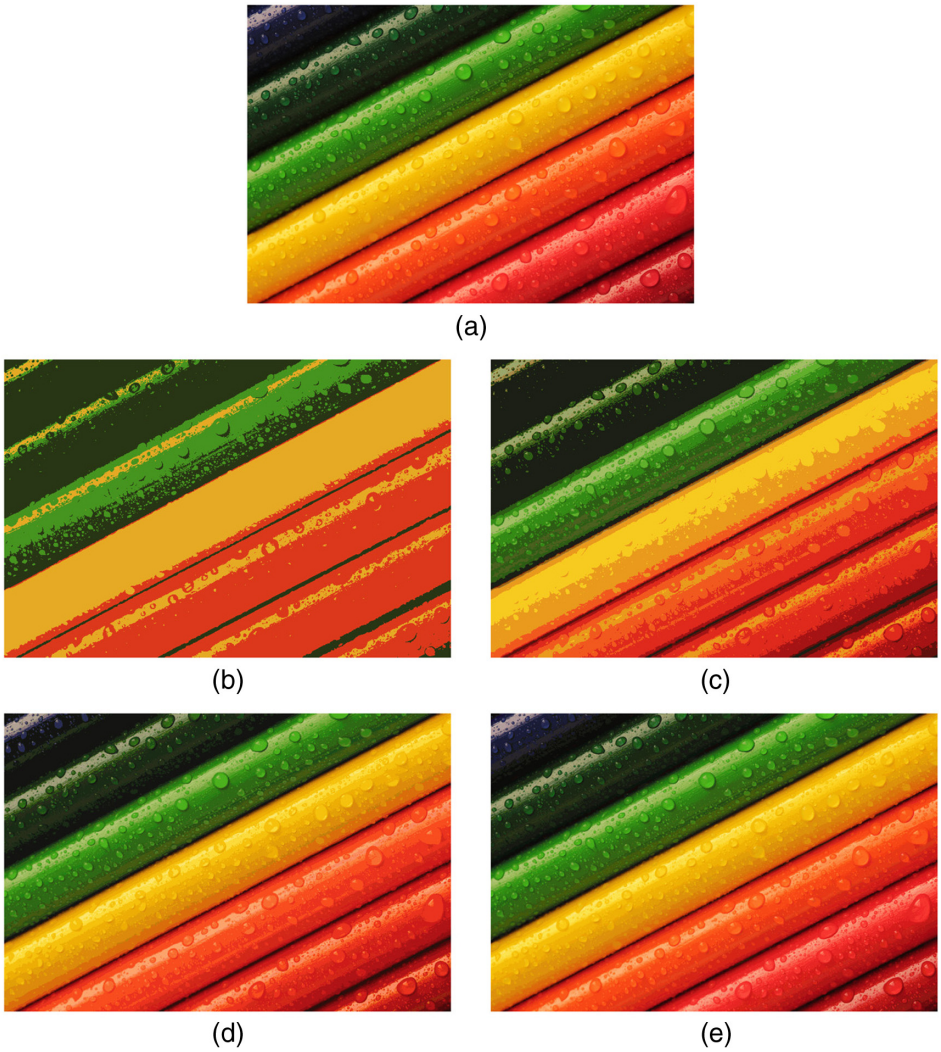

Fig. 1 Pencils and its various quantized versions: (a) original (117,157 colors), (b) 4 colors, (c) 16 colors, (d) 64 colors, and (e) 256 colors.

Unsurprisingly, a vast majority of CQ algorithms are image-dependent. Another classification scheme is based on the nature of the underlying clustering algorithm.[6] Hierarchical algorithms recursively find nested clusters in a top-down (or divisive) or bottom-up (or agglomerative) fashion. In contrast, partitional algorithms find all the clusters simultaneously as a partition of the data without imposing a hierarchical structure on the data.[7] Early CQ algorithms were mostly hierarchical, whereas modern CQ algorithms tend to be partitional.[8] Yet another classification scheme is based on whether or not the palette size is allowed to change during the CQ process. Static algorithms assume that the palette size is a constant to be specified by the user in advance, whereas dynamic algorithms compute the palette size automatically at run-time. A vast majority of CQ algorithms proposed to date are image-dependent and static. Therefore, in this paper, we focus primarily on such algorithms. (The extraction of relevant colors (i.e., colors that stand out to an observer[9]) from a color image, which is an important task in specific application domains such as fine arts[9,10] and medicine,[11,12] is outside the scope of this study.)

Many popular partitional algorithms are based on the local optimization of an objective function, which is typically nonsmooth and nonconvex with numerous local optima. Hence, such algorithms are often highly dependent on initialization[13] and can easily get stuck in a poor local optimum. By contrast, metaheuristic-based algorithms are formulated based on global optimization and, thus, are less sensitive to initialization.

CQ was a necessity in the past due to the limitations of the display hardware, many of which could not handle the number of colors that can be present in a typical 24-bit image. Over the past

decades, 24-bit display hardware have become ubiquitous. However, CQ is still used in many visual computing applications as a preprocessing step. Modern applications of CQ include non-photorealistic rendering, image matting, image dehazing, image compression, color-to-grayscale conversion, image watermarking/steganography, image segmentation, content-based image retrieval, color analysis, saliency detection, and skin detection. For specific references, see Ref. 8.

Despite the over four decades of research on CQ, there is currently no benchmark dataset on which CQ algorithms can be developed, tested, and compared.[8] Many CQ studies employ a subset of the USC-SIPI image database (Available at Ref. 14) or the Kodak lossless true color image suite (Available at Ref. 15). The USC-SIPI dataset contains 14 scanned images of relatively poor quality (in modern standards). Furthermore, the copyright status of many of these images is unknown. The Kodak dataset contains 24 relatively high-quality images in the public domain. However, the dataset is neither sufficiently diverse nor sufficiently large to permit comprehensive evaluation of CQ algorithms. (For example, in the USC-SIPI dataset, three (≈21%) images (4.1.01, 4.1.03, and 4.1.04) are portrait photos, two images (4.1.04 and 4.1.02) show the same person, and two images (4.1.07 and 4.1.08) depict identical objects. On the other hand, one-third of the Kodak images portray water bodies. In addition, the dataset contains pairs of images featuring identical objects [i.e., kodim02 depicts a subregion of kodim01, whereas kodim06 and kodim11 depict the same boat captured in two separate yet similar scenes.]) To address these problems, we curated a large, diverse, and high-quality dataset of 24-bit color images called CQ100 and released it under a permissive license.

The remainder of this paper is organized as follows. Section 2 gives a detailed description of our dataset. Section 3 demonstrates the use of our dataset in comparing CQ algorithms. Finally, Sec. 4 concludes the paper and suggests future research directions.

2 Description of the Dataset

Our objective was to collect a large, diverse, and high-quality dataset of images and release it under a permissive license. We started by collecting over 100 images from three public sources:

- Wikimedia Commons (Available at Ref. 16): A repository of over 88 million freely useable media files.
- PxHere (Available at Ref. 3): a repository of over one million CC0 -licensed images.
- Kodak lossless true color image suite: a collection of 24 Kodak photo CD images placed in the public domain by the Eastman Kodak Company.

During the image collection process, we paid particular attention to the diversity of content and license compatibility issues. Once we obtained a tentative dataset with 100+ images, we eliminated the images with outlying aspect ratios using the following iterative process. We first computed the mean (m) and standard deviation (s) of the aspect ratios (Ordinarily, the aspect ratio of an $H \times W$ image is given by W/H. However, to treat portrait and landscape images uniformly, we computed the aspect ratio as $\max(W/H, H/W)$.) of the current set of images and then eliminated those with outlying aspect ratios, that is, aspect ratios outside the range $[m - 2s, m + 2s]$. We added a few more images to the set (from the aforementioned public sources) and then repeated the above outlier removal process until we were left with 100 images. The characteristics of this dataset were:

- Sources: Wikimedia Commons (73), PxHere (23), and Kodak lossless true color image suite (4).
- Categories: animals (18), food (17), miscellaneous (6), objects (23), people (8), places (11), plants (6), and vehicles (11).
- Licenses: public domain (14), CC0 (25), CC BY (7), and CC BY-SA (54).

The images in our initial dataset had dimensions ranging from 768×512 to 6498×8123. To facilitate comparisons, it is desirable to have all images contain the same number of pixels. A simple way to achieve this goal is to resize all images to the same dimensions (and thus the same aspect ratio). To this end, we performed a grid search between the minimum ($l = 1.25$) and maximum ($u = 1.64$) aspect ratios in the dataset with a step size of $(u - l)/1000$. The optimal

aspect ratio minimizing the average relative deviation (Given an image, let a and $\tilde{a}$ be its aspect ratios before and after resizing, respectively. The relative deviation in the aspect ratio caused by the resizing operation is then $\varepsilon = |\tilde{a} - a|/a$.) turned out to be 1.49973, which is very close to the standard 3:2 aspect ratio commonly used in 35-mm film photography and digital single-lens reflex camera (DSLR) photography. Accordingly, we resized [We used the resize operator (with the default Lanczos resampling filter) of the convert program of ImageMagick 7.1.0-33 (available at Ref. 17)] all portrait and landscape images in our initial dataset respectively to 512×768 and 768×512 to achieve our target aspect ratio of 1.5. These resizing operations caused an average of only 5.35% relative distortion in the aspect ratio, which is negligible. Other than resizing, file renaming and file format conversion were the only modifications made to the original images. Most of the files originally had meaningless or long and overly descriptive names, some reaching 120 characters. Therefore, we renamed each file concisely to accurately reflect its contents. This renaming operation reduced the mean filename length from about 40 to about 12. As for file formats, 95 of the images were originally stored in the JPG format and 5 in the PNG format. We converted all images to the uncompressed binary PPM format. PPM is popular in visual computing because it is uncompressed, and PPM files are easy to read and write, owing to their extremely simple structure. As mentioned earlier, each image in CQ100 has one of four types of licenses (For an overview of these licenses, visit Ref. 18.) (listed in decreasing permissiveness):

- Public domain: no known copyright.
- CC0 (Creative Commons zero 1.0 universal): no rights reserved.
- CC BY (Creative Commons attribution): others may share and adapt the work, even commercially, as long as they credit the copyright holder for the original work.
- CC BY-SA (Creative Commons attribution-share alike): others may share and adapt the work, even commercially, as long as they credit the copyright holder for the original work and license their new works under the same license as the original.

Since these licenses are mutually compatible, we released CQ100 under the least permissive of them, namely the CC BY-SA 4.0 license. Nevertheless, this license allows anyone to share and modify our dataset with the restrictions noted above. The metadata associated with each image include the following 16 attributes: original image filename, modified image filename, image category, source URL, license, license URL, author, author URL, modifications made to the original image, additional notes about the original image, original image width, original image height, original image number of colors, modified image width, modified image height, and modified image number of colors.

To demonstrate the diversity of CQ100, we show thumbnails for images 1 through 25, 26 through 50, 51 through 75, and 76 through 100 in Figs. 2–5, respectively. The numeric id around each thumbnail indicates the alphabetical order of the filename of the corresponding image, e.g., 1: adirondack_chairs; 2: astro_bodies,..., 99: wool_carder_bee; and 100: yasaka_pagoda. A more detailed view of a subset of CQ100 is given in Fig. 6, which displays a sample image from each category. Note that for each image featured in Figs. 2–6, the original license is given in the respective caption.

Figures 7 and 8 show box plots of the distribution of respectively the number of distinct colors and colorfulness[19] per image for CQ100, Kodak, and USC-SIPI datasets. It can be seen that CQ100 has a greater color diversity compared to the other two datasets.

3 Demonstration of the Dataset

In this section, we demonstrate the use of our CQ100 dataset on a CQ task. First, we describe the experimental setup and present a brief qualitative assessment. Then, we conduct a more detailed quantitative assessment and discuss its implications.

3.1 Experimental Setup and Qualitative Assessment

We compare the following 21 CQ algorithms (listed in chronological order) with respect to their effectiveness [An effective CQ algorithm is one that produces minimal distortion, as quantified by a chosen image fidelity metric (see Sec. 3.2).]:

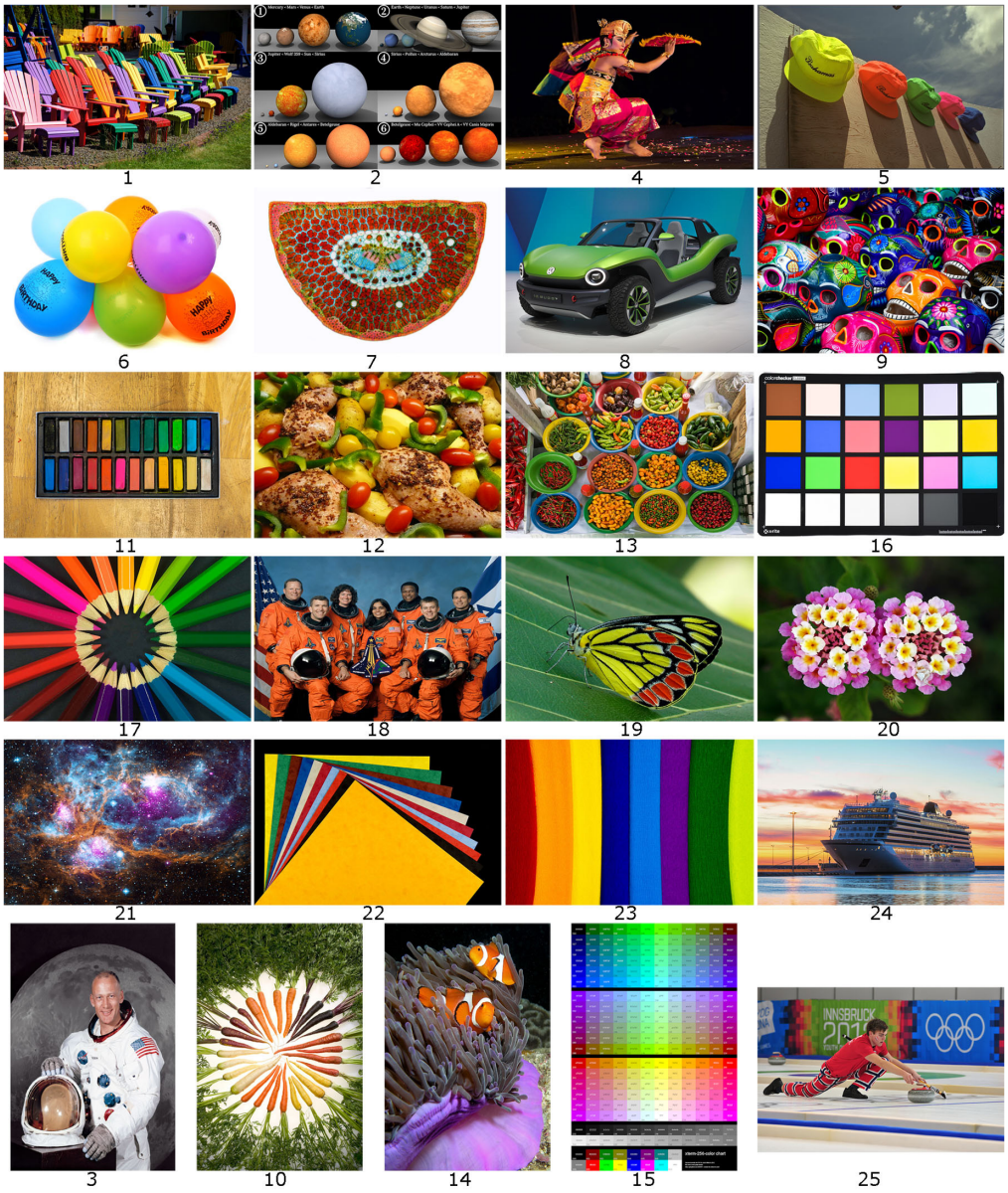

Fig. 2 Thumbnails for images 1 through 25 (3, 5, 10, 18, and 21 are in the public domain; 6, 15, 16, and 23 are licensed under CC0; the others are licensed under CC BY-SA).

- Median-cut algorithm[4] (divisive hierarchical) (MC)
- Popularity algorithm[4] (partitional) (POP)
- Octree algorithm[20] (agglomerative hierarchical) (OCT)
- Wan et al.'s marginal variance minimization algorithm[21] (divisive hierarchical) (WAN)
- Orchard and Bouman's binary splitting algorithm[22] (divisive hierarchical) (BS)
- Wu's variance minimization algorithm[23] (divisive hierarchical) (WU)
- Dekker's self-organizing map algorithm[24] (partitional) (SOM)
- Split-and-merge algorithm[25] (agglomerative hierarchical) (SAM)
- Batch k-means algorithm initialized using VCL[26] (partitional) (WSM)
- Fuzzy c-means algorithm initialized using WU[27] (partitional) (WFCM)
- Adaptive distributing units algorithm[28] (partitional) (ADU)
- Variance-cut algorithm with Lloyd iterations[29] (divisive hierarchical) (VCL)
- Ant-tree algorithm[30] (metaheuristic) (ATCQ)

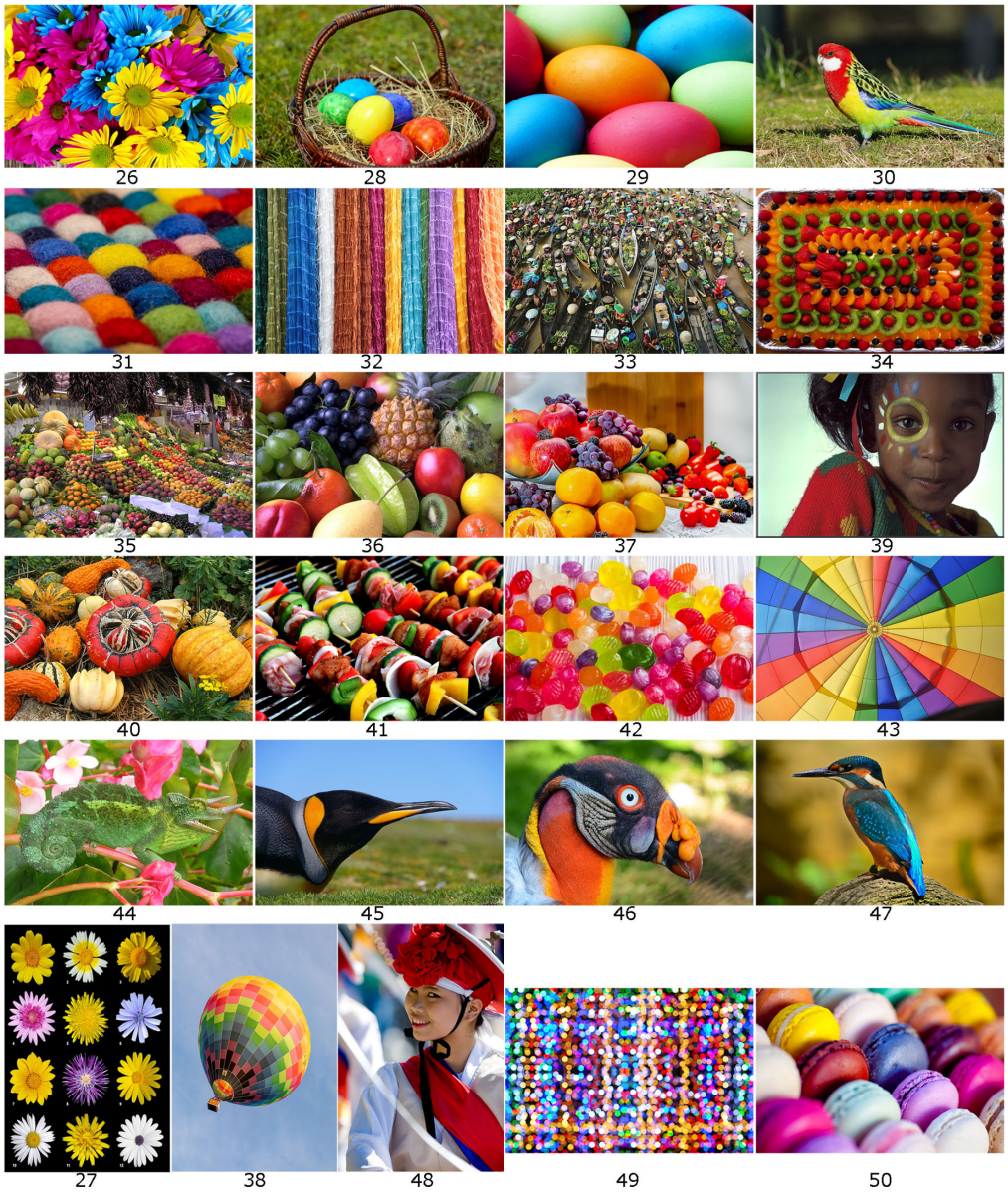

Fig. 3 Thumbnails for images 26 through 50 (35, 39, 44, and 48 are in the public domain; 26, 28, 29, 36, 37, 41, 42, 47, and 49 are licensed under CC0; 34, 45, and 50 are licensed under CC BY; and the others are licensed under CC BY-SA)

- Firefly algorithm combined with ATCQ[31] (metaheuristic) (FFATCQ)
- Artificial bee colony combined with ATCQ[32] (metaheuristic) (ABCATCQ)
- Shuffled-frog leaping algorithm[33] (metaheuristic) (SFLA)
- WU combined with ATCQ[34] (metaheuristic) (WUATCQ)
- Particle swarm optimization combined with ATCQ[35] (metaheuristic) (PSOATCQ)
- BS combined with ITATCQ[36] (metaheuristic) (BSITATCQ)
- Iterative ATCQ[37] (metaheuristic) (ITATCQ)
- Iterative online k-means algorithm[38] (partitional) (IOKM)

These algorithms represent the CQ work published between 1980 and 2022. (For a modern survey of CQ, see Ref. 8.) However, it should be noted that our objective is not to perform an exhaustive comparison of the CQ algorithms published over the past 40 years but to demonstrate the use of the presented dataset in comparing several popular CQ algorithms.

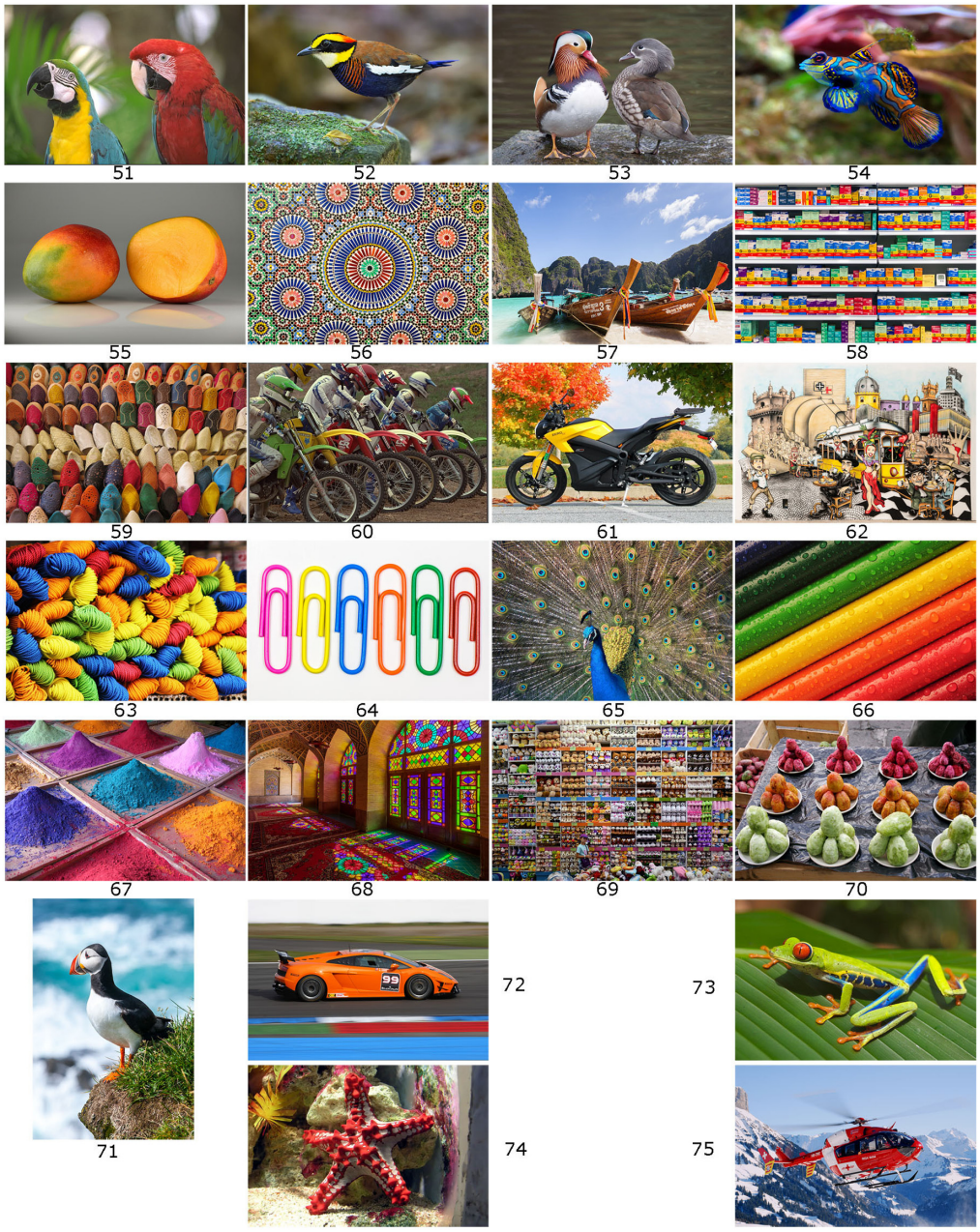

Fig. 4 Thumbnails for images 51 through 75 (51, 60, 73, and 74 are in the public domain; 63, 64, and 66 are licensed under CC0; 67 is licensed under CC BY; and the others are licensed under CC BY-SA).

We execute each algorithm with the default parameter values suggested by its authors (see Table 1) and quantize each image in our dataset to 4, 16, 64, and 256 colors separately. The cases of 4 and 256 colors represent the two extremes. It can be argued that a typical natural image can hardly be quantized to fewer than 4 colors, and most images can be reproduced relatively accurately with no more than 256 colors.

Figures 9–12, respectively show the shopping bags, motorcycle, umbrellas, and common jezebel images quantized using three algorithms (in each figure, going from top to bottom, the subfigures are arranged in progressively decreasing quality). Given an input image, for each algorithm, we display the corresponding reduced-color output image and a grayscale error image that allows us to visualize the differences between the input and output. The error image is obtained by amplifying the pixelwise normalized Euclidean differences between the input and

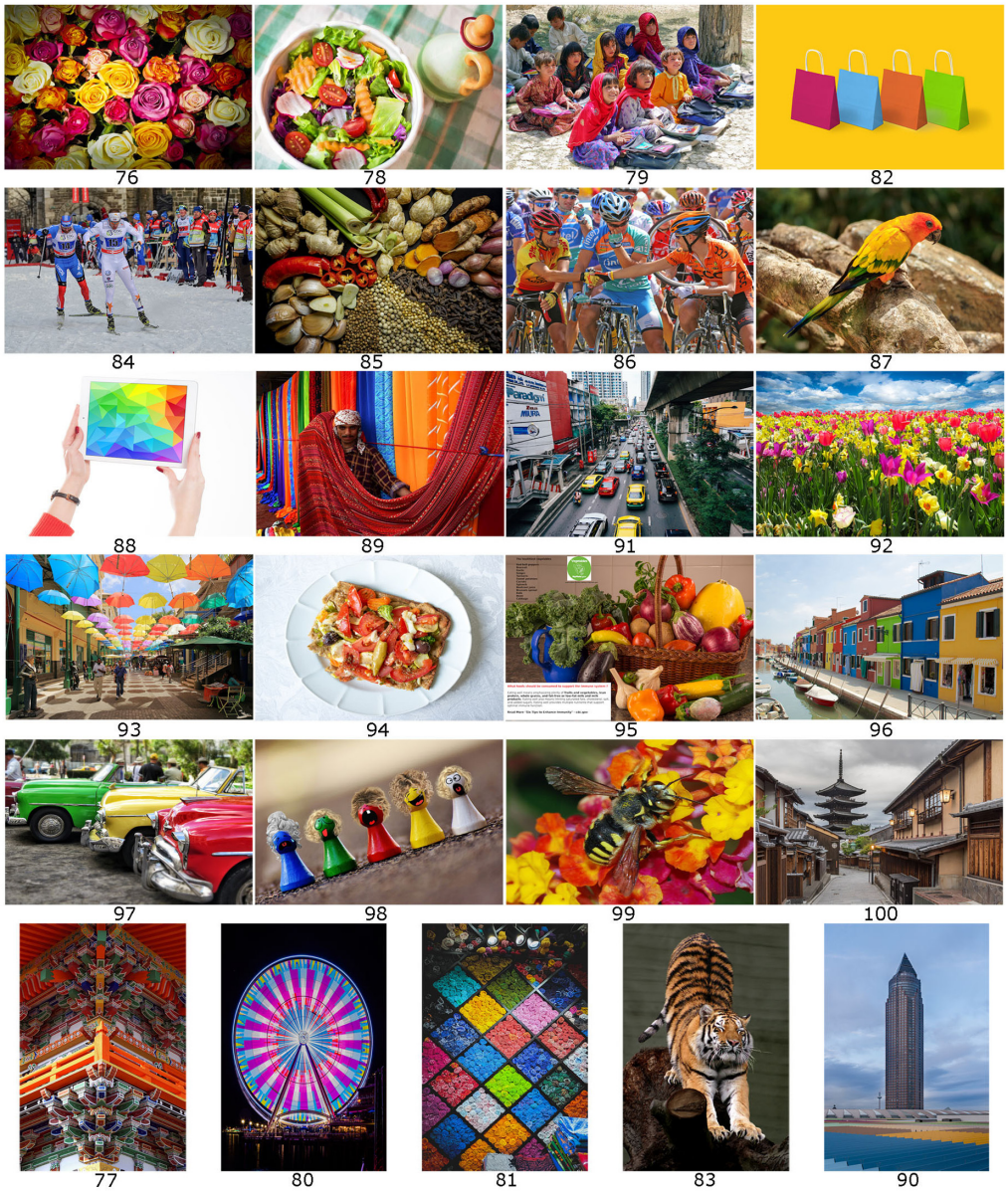

Fig. 5 Thumbnails for images 76 through 100 (79 is in the public domain; 76, 78, 82, 86, 88, 91, 92, 97, and 98 are licensed under CC0; 84, 89, and 96 are licensed under CC BY; and the others are licensed under CC BY-SA).

output by a factor of four and then negating them for better visualization (see below). Hence, the cleaner/lighter the error image, the better the reproduction of the input image.

Let I_{RGB} and $\tilde{I}_{RGB}$ respectively denote the $H \times W$ original input and quantized output images in the standard RGB (sRGB) color space.[39] $I_{\mathrm{RGB}}(r,c)$ and $\tilde{I}_{\mathrm{RGB}}(r,c)$ are then three-dimensional vectors containing the RGB values of the pixel with (row, column) coordinate (r,c) in I_{RGB} and $\tilde{I}_{\mathrm{RGB}}$, respectively ($r \in \{1,\ldots,H\}$ and $c \in \{1,\ldots,W\}$). The corresponding pixel in the 8-bit error image E is then computed as

$$E(r,c) = 255 - \frac{4}{\sqrt{3}} \|I_{\mathrm{RGB}}(r,c) - \tilde{I}_{\mathrm{RGB}}(r,c)\|_2, \tag{1}$$

where $\|\cdot\|_2$ denotes the Euclidean (ℓ_2) norm. Note that the division by a $\sqrt{3}$ is necessary for the RGB -to-grayscale conversion, and pixels in E with negative values are clipped to zero.

Fig. 6 Sample images from CQ100: (a) red-eyed tree frog (animals), (b) fruits (food), (c) color checker (miscellaneous), (d) nylon cords (objects), (e) schoolgirls (people), (f) cosmic vista (places), (g) daisy bouquet (plants), and (h) sports bicycles (vehicles). Images (a), (e), and (f) are in the public domain; the others are licensed under CC0.

3.2 Quantitative Assessment and Discussion

Quantitative assessment remains to be one of the least explored aspects of CQ.[8] Most CQ studies employ pixelwise image fidelity metrics such as the mean squared error (MSE), or its variants such as the peak signal-to-noise ratio (PSNR), computed in the RGB color space. Recently, Ref. 40 investigated 25 fidelity metrics and concluded that MSE computed in the CIELAB color space was among the best. Following their recommendation, we compute the MSE metric as follows:

$$\mathrm{MSE}(I_{\mathrm{CIELAB}}, \tilde{I}_{\mathrm{CIELAB}}) = \frac{1}{\mathrm{HW}} \sum_{r=1}^{H} \sum_{c=1}^{W} \| I_{\mathrm{CIELAB}}(r,c) - \tilde{I}_{\mathrm{CIELAB}}(r,c) \|_2^2, \tag{2}$$

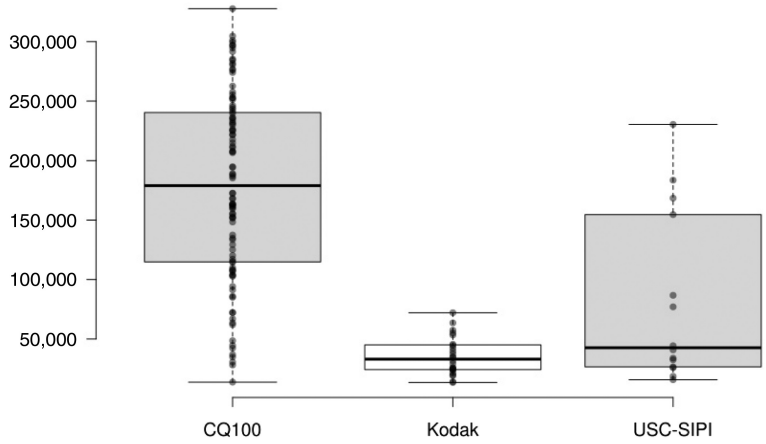

Fig. 7 Box plot of the distribution of the number of distinct colors per image for CQ100, Kodak, and USC-SIPI datasets.

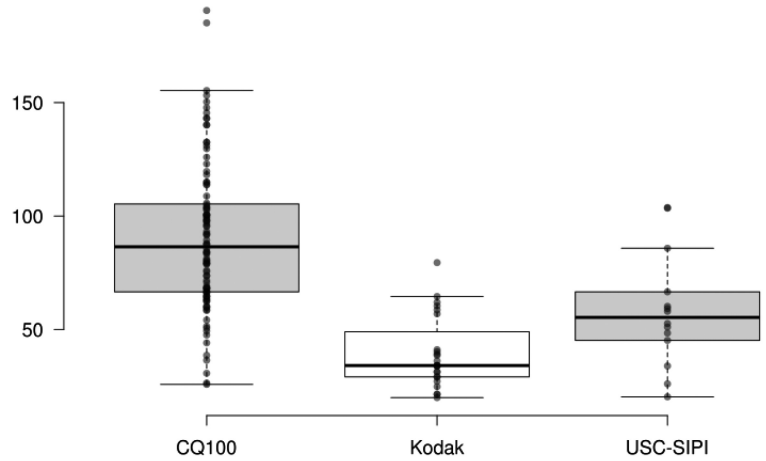

Fig. 8 Box plot of the distribution of the colorfulness per image for CQ100, Kodak, and USC-SIPI datasets

where I_{CIELAB} and $\tilde{I}_{\text{CIELAB}}$ respectively denote the $H \times W$ original input and quantized output images in the CIELAB color space. Thus, MSE represents the average color distortion in the CIELAB color space with respect to the squared Euclidean (ℓ_2^2) norm. Note that in the RGB -to- CIELAB conversion, we assume a working color space of SRGB and a D65 reference white.

The presented CQ100 dataset includes the 100 (true-color) input images and their metadata (as described in Sec. 2), 8400 (reduced-color) output images (21 CQ algorithms $\times$100 input images $\times\{4,16,64,256\}$ colors), and Microsoft Excel worksheets containing the MSE for each input/output image combination. (Other image fidelity metrics can be computed over the provided input/output images.) By contrast, to the best of our knowledge, for the USC-SIPI and Kodak datasets, there is no public repository containing the output images produced by a variety of algorithms and the corresponding MSE values.

To determine if there are any statistically significant differences among the CQ algorithms, we employ two nonparametric statistical tests:[41] the Friedman test[42] and the Iman-Davenport test.[43] These tests are alternatives to the parametric two-way analysis of variance (ANOVA) test. Their advantage over ANOVA is that they do not require normality or homoscedasticity, assumptions that are often violated in machine learning or optimization studies.[44–48]

Given B blocks (subjects) and T treatments (measurements), the null hypothesis (H_0) of the Friedman test is that populations within a block are identical. The alternative hypothesis (H_1) is that at least one treatment tends to yield larger (or smaller) values than at least one other treatment. The test statistic is computed as follows.[49] In the first step, the observations within each block are ranked separately, so each block contains a separate set of T ranks. If ties occur, the tied

Table 1 Parameter setting for each CQ algorithm.

Algorithm	Parameter setting
MC/POP/WAN/WU	Number of bits: 5
OCT	Maximum depth: 6
SOM	Sampling factor: 1
SAM	Initial number of clusters: 20× number of colors
WSM	Decimation factor: 2
	Maximum number of iterations: 100
	Convergence threshold: 0.001
WFCM	Decimation factor: 2
	Maximum number of iterations: 100
	Convergence threshold: 0.001
	Weighting exponent: 2
ADU	Learning rate: 0.015
VCL	Decimation factor: 2
	Number of bits: 5
	Number of iterations: 10
ATCQ	$\alpha = 0.25$, 0.3, 0.35, and 0.4 for 4, 16, 64, and 256 colors, respectively.
FFATCQ	Number of fireflies: 5
	$\alpha = \{0.25, 0.30, 0.35, 0.40, 0.45\}$
	Number of iterations: 20
ABCATCQ	Number of food sources: 5
	$\alpha = \{0.25, 0.30, 0.35, 0.40, 0.45\}$
	Number of iterations: 20
SFLA	Number of frogs: 8
	Number of memeplexes: 2
	Number of iterations: 20
PSOATCQ	Number of particles: 5
	$\alpha = \{0.25, 0.30, 0.35, 0.40, 0.45\}$
	Number of iterations: 20
BSITATCQ	Number of iterations: 20
ITATCQ	$\alpha = 0.35$
	Iterations: 20

observations are given the mean of the rank positions for which they are tied. If H_0 is true, the ranks in each block should be randomly distributed over the columns (treatments). Otherwise, we expect a lack of randomness in this distribution. For example, if a particular treatment is better than the others, we expect small ranks to "favor" that column. In the second step, the ranks in each column are summed. If H_0 is true, we expect the sums to be fairly close—so close that we can attribute differences to chance. Otherwise, we expect to see at least one difference between

Fig. 9 (a) Shopping bags (13,752 colors) and its various quantized versions (4 colors): (b) WFCM, (d) WUATCQ, and (f) BSITATCQ. Subfigures (c), (e), and (g) are the error images corresponding to subfigures (b), (d), and (f), respectively.

pairs of rank sums so large that we cannot reasonably attribute it to sampling variability. The test statistic is given as

$$\chi_r^2 = \frac{12}{\mathrm{BT}(T+1)} \sum_{j=1}^{T} R_j^2 - 3B(T+1), \tag{3}$$

where R_j ($j \in \{1,2,\ldots,T\}$) is the rank sum of the jth column. χ_r^2 is approximately chi-square with $(T-1)$ degrees of freedom. H_0 is rejected at the α level of significance if the value of (3) is greater than or equal to the critical chi-square value for $(T-1)$ degrees of freedom.

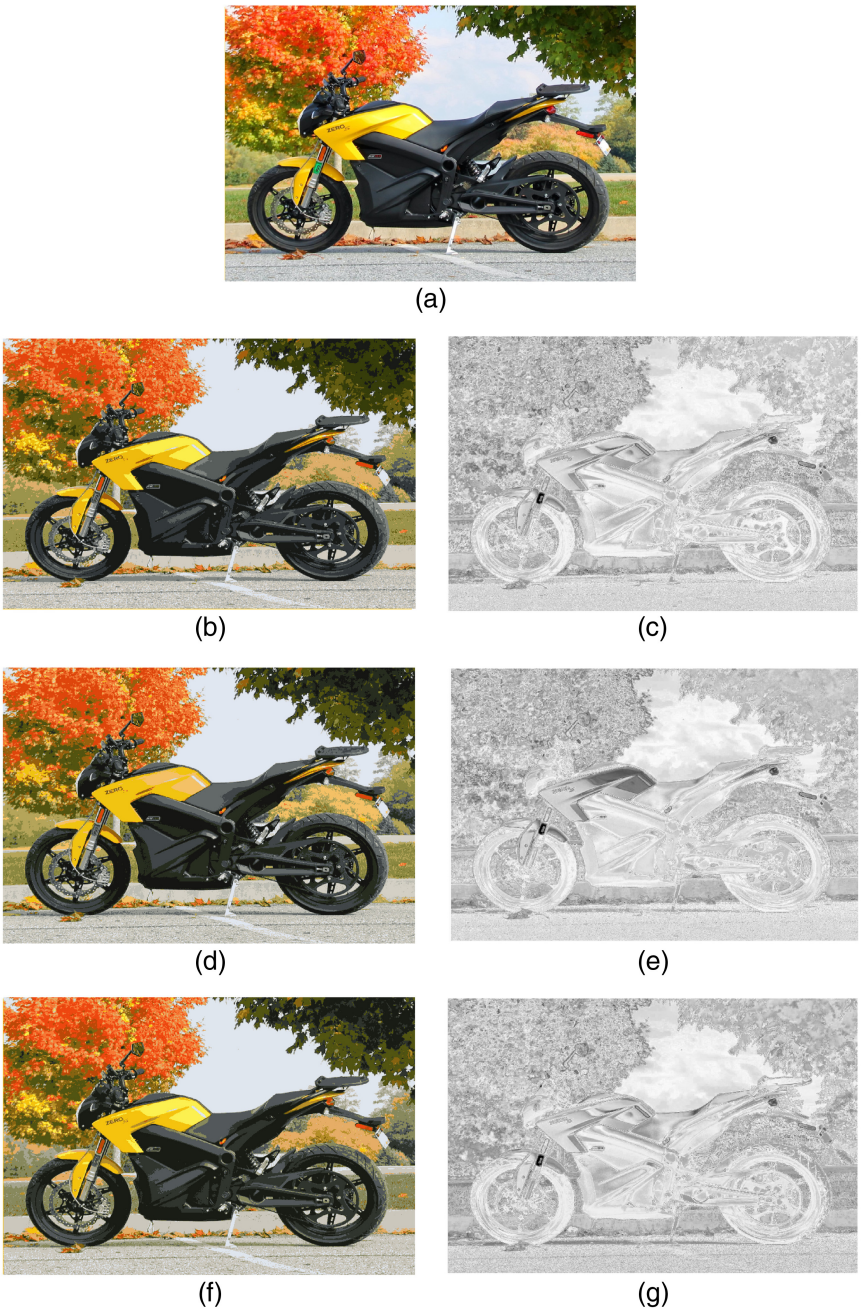

Fig. 10 (a) Motorcycle (212,020 colors) and its various quantized versions (16 colors): (b) PSOATCQ, (d) ADU, and (f) BSITATCQ. Subfigures (c), (e), and (g) are the error images corresponding to subfigures (b), (d), and (f), respectively.

Reference 43 proposed the following alternative statistic

$$F_r = \frac{(B-1)\chi_r^2}{B(T-1)-\chi_r^2}, \tag{4}$$

which is distributed according to the F-distribution with $(T-1)$ and $(T-1)(B-1)$ degrees of freedom. Compared to χ_r^2, this statistic is not only less conservative but also more accurate for small sample sizes.[43]

In this study, blocks and treatments correspond to images and CQ algorithms, respectively. For each $K \in \{4,16,64,256\}$, our goal is to determine if at least one algorithm is significantly better than at least one other algorithm at the $\alpha = 0.05$ level of significance. If this is the case,

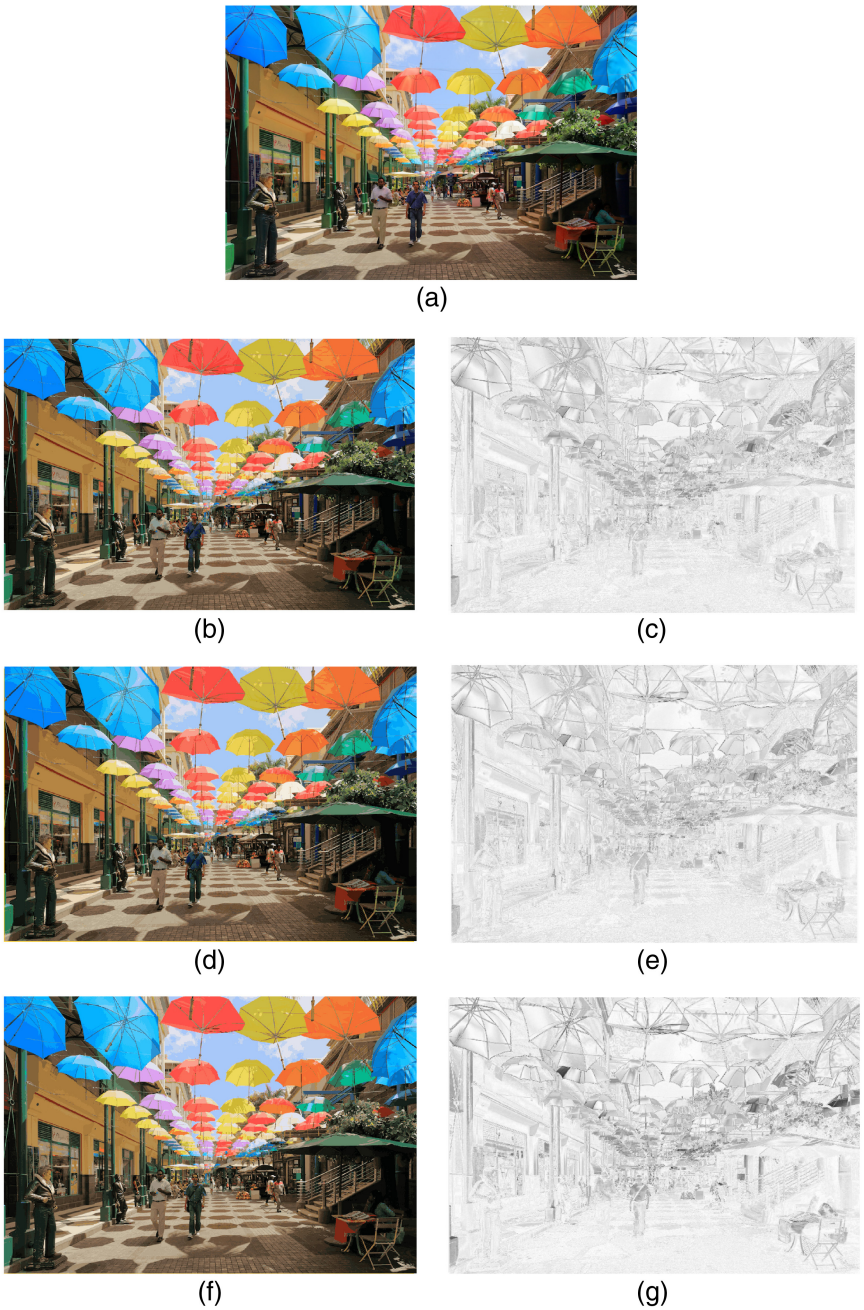

Fig. 11 (a) Umbrellas (217,673 colors) and its various quantized versions (64 colors): (b) SFLA, (d) ABCATCQ, and (f) WFCM. Subfigures (c), (e), and (g) are the error images corresponding to subfigures (b), (d), and (f), respectively.

we perform multiple comparison testing to determine which pairs of algorithms differ significantly. For this purpose, we employ the Bergmann-Hommel test[50] (also at the $\alpha = 0.05$ level), a powerful multiple comparison test that has been used successfully in various machine learning studies.[13,41,51–53] (The power of a binary hypothesis test is the probability that the test correctly rejects the null hypothesis.) Bergmann-Hommel is a dynamic test that considers the logical relations among the hypotheses and is strictly more powerful[54] than various alternative tests that control the familywise error rate (The familywise error rate is the probability of falsely rejecting at least one null hypothesis when performing multiple comparison tests.) such as Nemenyi,[55] Holm,[56] and Shaffer[57] tests.

Table 2 gives the mean rank of each CQ algorithm over the dataset for $K \in \{4,16,64,256\}$ (lower ranks are better). The last column gives the mean of the (mean) ranks over the four K

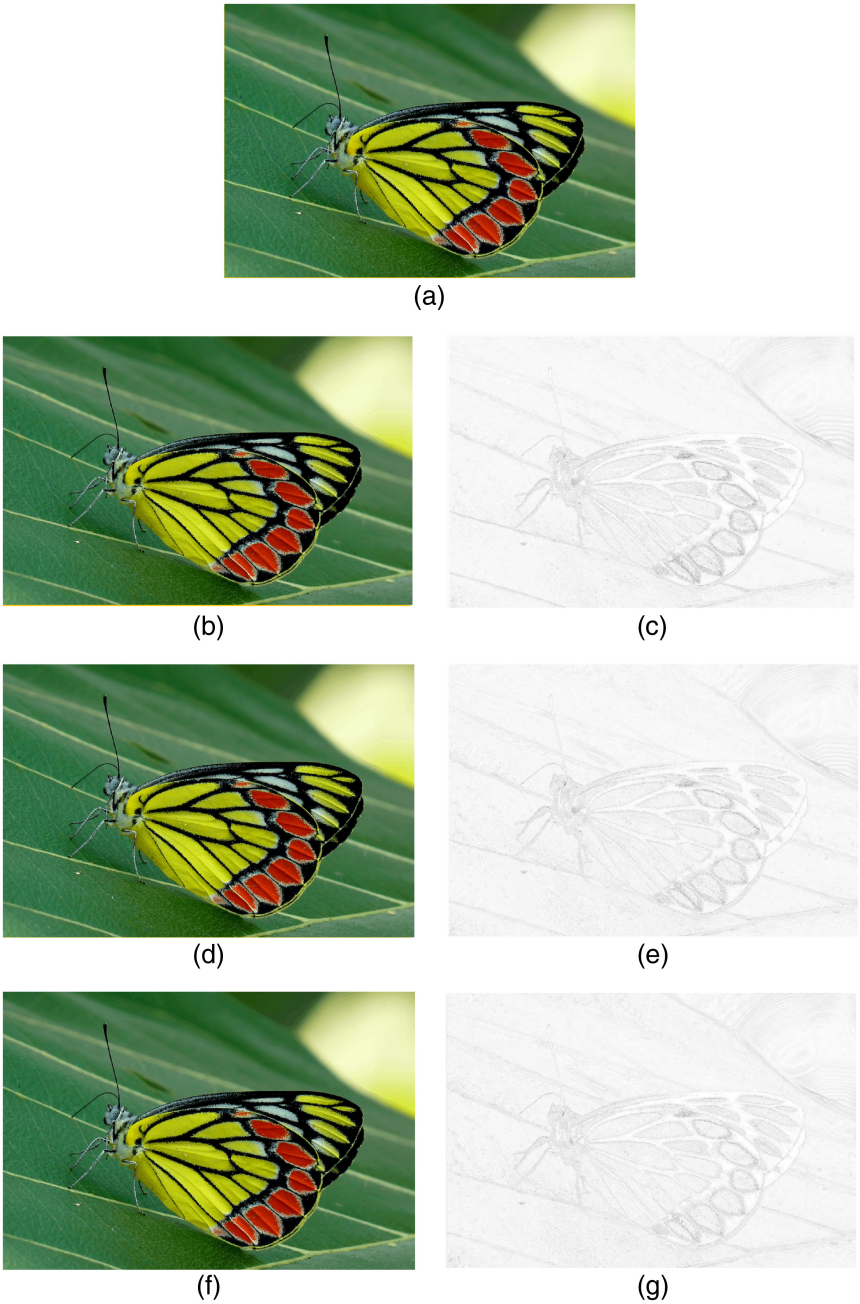

Fig. 12 (a) Common jezebel (137,446 colors) and its various quantized versions (256 colors): (b) ADU, (d) WSM, and (f) IOKM. Subfigures (c), (e), and (g) are the error images corresponding to subfigures (b), (d), and (f), respectively.

values. Unsurprisingly, the top ranks are occupied by partitional and metaheuristic-based algorithms. As mentioned earlier, the Bergmann-Hommel test is a powerful multiple comparison test. However, this power comes at the expense of a very high computational cost. More specifically, the test takes exponential time in the number of hypotheses;[58] thus, it cannot handle more than ten algorithms, even on a high-performance CPU. To address this problem, we eliminate the patently inferior algorithms,[48] namely all seven hierarchical algorithms (i.e., BS, MC, OCT, SAM, VCL, WAN, and WU) and the five partitional or metaheuristic-based algorithms that rank in the bottom half (i.e., ATCQ, FFATCQ, ITATCQ, POP, and SOM). Thus, the nonparametric statistical analyses below are conducted for the remaining nine algorithms (Even if the Bergmann-Hommel test could handle all 21 algorithms, it would have been extremely unwieldy to analyze the test results involving

Table 2 Mean rank of each CQ algorithm for $K \in \{4,16,64,256\}$.

Algorithm	$K = 4$	$K = 16$	$K = 64$	$K = 256$	Mean
ABCATCQ	6.50	5.61	3.75	4.12	5.00
ADU	11.45	8.00	5.06	2.99	6.88
ATCQ	14.71	18.08	18.01	17.39	17.05
BS	13.41	15.05	14.68	13.55	14.17
BSITATCQ	10.20	8.56	8.04	6.81	8.40
FFATCQ	8.77	11.61	12.06	12.48	11.23
IOKM	6.94	4.91	4.45	4.98	5.32
ITATCQ	9.84	11.36	12.42	10.44	11.02
MC	17.61	19.34	19.89	20.06	19.23
OCT	14.08	15.78	15.29	13.21	14.59
POP	20.77	20.89	20.87	20.70	20.81
PSOATCQ	6.19	3.21	2.71	3.58	3.92
SAM	10.48	10.35	12.24	13.98	11.76
SFLA	6.28	5.41	5.93	7.78	6.35
SOM	17.03	13.98	10.02	8.61	12.41
VCL	12.51	13.14	15.53	17.17	14.59
WAN	15.08	16.83	18.19	18.78	17.22
WFCM	4.53	5.18	7.94	10.66	7.08
WSM	7.04	4.67	3.78	3.32	4.70
WU	8.85	10.59	11.61	11.92	10.74
WUATCQ	8.72	8.45	8.53	8.46	8.54

$\binom{21}{2} = 420$ hypotheses, as opposed to $\binom{9}{2} = 36$ hypotheses, and such an analysis would have more than likely yielded complex and uninterpretable rules.), namely ABCATCQ, ADU, BSITATCQ, IOKM, PSOATCQ, SFLA, WFCM, WSM, and WUATCQ.

For $K = 4$ colors, both Friedman and Iman-Davenport tests detect a statistically significant difference in image fidelity (or effectiveness) among the CQ algorithms: $\chi_r^2(8) = 158.990$ with $p = 9.613\text{e} - 11$ and $F_r(8,792) = 24.555$ with $p = 6.675\text{e} - 34$. The results of the Bergmann-Hommel test are given in column 2 of Table 3. For example, the test rejects the null hypothesis "ABCATCQ versus ADU" (i.e., the two algorithms are equally effective). Since we know from Table 2 that ABCATCQ has a lower (or better) rank than ADU for $K = 4$, the rejection of the above hypothesis means ABCATCQ is more effective than ADU for $K = 4$ and the difference between the two algorithms is statistically significant at the $\alpha = 0.05$ level. The results of the Bergmann-Hommel test can be summarized succinctly as follows:

$$\text{WFCM} > \{\text{ABCATCQ, PSOATCQ, SFLA, WSM}\} > \{\text{ADU, BSITATCQ, WUATCQ}\},$$

where a notation such as $\{\text{A, B}\} > \text{C}$ indicates that there is no statistically significant difference between algorithms A and B, and these two algorithms are significantly more effective than (or superior to) algorithm C. The above (summary) rule can be interpreted as follows:

- WFCM is the best algorithm;
- {ADU, BSITATCQ, WUATCQ} is the worst group of algorithms; and
- {ABCATCQ, PSOATCQ, SFLA, WSM} is in between.

Table 3 Results of the Bergmann-Hommel test for $K \in \{4,16,64,256\}$ (✓: rejected and ✗: not rejected).

Null hypothesis	$K = 4$	$K = 16$	$K = 64$	$K = 256$
ABCATCQ versus ADU	✓	✓	✗	✓
ABCATCQ versus BSITATCQ	✓	✓	✓	✓
ABCATCQ versus IOKM	✗	✗	✗	✓
ABCATCQ versus PSOATCQ	✗	✓	✗	✗
ABCATCQ versus SFLA	✗	✗	✓	✓
ABCATCQ versus WFCM	✓	✗	✓	✓
ABCATCQ versus WSM	✗	✗	✗	✗
ABCATCQ versus WUATCQ	✓	✓	✓	✓
ADU versus BSITATCQ	✗	✗	✓	✓
ADU versus IOKM	✓	✓	✗	✓
ADU versus PSOATCQ	✓	✓	✓	✗
ADU versus SFLA	✓	✓	✗	✓
ADU versus WFCM	✓	✓	✓	✓
ADU versus WSM	✓	✓	✗	✗
ADU versus WUATCQ	✗	✗	✓	✓
BSITATCQ versus IOKM	✓	✓	✓	✓
BSITATCQ versus PSOATCQ	✓	✓	✓	✓
BSITATCQ versus SFLA	✓	✓	✓	✗
BSITATCQ versus WFCM	✓	✓	✗	✓
BSITATCQ versus WSM	✓	✓	✓	✓
BSITATCQ versus WUATCQ	✗	✗	✗	✓
IOKM versus PSOATCQ	✗	✓	✓	✓
IOKM versus SFLA	✗	✗	✗	✓
IOKM versus WFCM	✓	✗	✓	✓
IOKM versus WSM	✗	✗	✗	✓
IOKM versus WUATCQ	✗	✓	✓	✓
PSOATCQ versus SFLA	✗	✓	✓	✓
PSOATCQ versus WFCM	✓	✓	✓	✓
PSOATCQ versus WSM	✗	✓	✓	✗
PSOATCQ versus WUATCQ	✓	✓	✓	✓
SFLA versus WFCM	✓	✗	✓	✓
SFLA versus WSM	✗	✗	✓	✓
SFLA versus WUATCQ	✓	✓	✓	✓
WFCM versus WSM	✓	✗	✓	✓
WFCM versus WUA TCQ	✓	✓	✗	✗
WSM versus WUATCQ	✓	✓	✓	✓

Observe that the above summary does not include IOKM. This is because, as Table 3 shows, while IOKM is inferior to WFCM and superior to {ADU, BSITATCQ}, it cannot be included in group {ABCATCQ, PSOATCQ, SFLA, WSM } since it is not superior to WUATCQ. Hence, an alternative rule, including IOKM but excluding WUATCQ, is

WFCM > {ABCATCQ, IOKM, PSOATCQ, SFLA, WSM} > {ADU, BSITATCQ}.

For $K = 16$ colors, both Friedman and Iman-Davenport tests detect a statistically significant difference in image fidelity (or effectiveness) among the CQ algorithms: $\chi_r^2(8) = 203.771$ with $p = 9.240\text{e} - 11$ and $F_r(8792) = 33.835$ with $p = 4.888\text{e} - 46$. The results of the Bergmann-Hommel test are given in column 3 of Table 3. In this case, the results can be summarized by a single rule that covers all nine algorithms:

PSOATCQ > {ABCATCQ, IOKM, SFLA, WFCM, WSM} > {ADU, BSITATCQ, WUATCQ}, which can be interpreted as follows:

- PSOATCQ is the best algorithm;
- {ADU, BSITATCQ, WUATCQ} is the worst group of algorithms; and
- {ABCATCQ, IOKM, SFLA, WFCM, WSM} is in between.

For $K = 64$ colors, both Friedman and Iman-Davenport tests detect a statistically significant difference in image fidelity (or effectiveness) among the CQ algorithms: $\chi_r^2(8) = 330.423$ with $p = 1.416\text{e} - 10$ and $F_r(8,792) = 69.663$ with $p = 1.763\text{e} - 86$. The results of the Bergmann-Hommel test are given in column 4 of Table 3. In this case, the results can be summarized by three alternative rules:

- PSOATCQ > {ADU, IOKM, WSM} > {BSITATCQ, WFCM, WUATCQ};
- {ABCATCQ, ADU, IOKM, WSM} > {BSITATCQ, WFCM, WUATCQ}; and
- PSOATCQ > {ADU, IOKM, SFLA} > {BSITATCQ, WFCM, WUATCQ}.

Observe that {BSITATCQ, WFCM, WUATCQ} is the worst group of algorithms in every case.

For $K = 256$ colors, both Friedman and Iman-Davenport tests detect a statistically significant difference in image fidelity (or effectiveness) among the CQ algorithms: $\chi_r^2(8) = 394.704$ with $p = 1.872\text{e} - 10$ and $F_r(8,792) = 96.413$ with $p = 1.441\text{e} - 111$. The results of the Bergmann-Hommel test are given in column 5 of Table 3. In this case, the results can be summarized by two alternative rules:

- {ADU, PSOATCQ, WSM} > IOKM > {BSITATCQ, SFLA} > {WFCM, WUATCQ} and
- {ABCATCQ, PSOATCQ, WSM} > IOKM > {BSITATCQ, SFLA} > {WFCM, WUATCQ}

Observe that, in either case, {WFCM, WUATCQ} is the worst group of algorithms; {BSITATCQ, SFLA} is the second-worst group; and IOKM is superior to the preceding two groups.

General remarks with respect to effectiveness (recall that lower ranks are better):

- ABCATCQ is always in the best or second-best group;
- ADU ranks progressively lower (This behavior is likely due to the constant learning rate used in the experiments, as suggested by Ref. 28. For better performance, this rate should probably be set as a function of K.) with increasing K (it is in the worst group for $K \in \{4,16\}$ but the best or second-best group for $K \in \{64,256\}$);
- BSITATCQ ranks progressively lower with increasing K (nevertheless, it is in the worst group for $K \in \{4,16,64\}$ and the second-worst group for $K = 256$);
- IOKM is always in the best or second-best group;
- PSOATCQ is in the best group for $K \in \{16,64,256\}$ and the second-best group for $K = 4$;
- SFLA is in the second-best group for $K \in \{4,16,64\}$ and the second-worst group for $K = 256$;
- WFCM ranks progressively higher (The lackluster performance of WFCM, which is a fuzzy generalization of WSM, was demonstrated earlier by Ref. 27 on a limited number of images.) with increasing K (it is the best algorithm for $K = 4$ but in the worst group for $K \in \{64,256\}$);

- WSM ranks progressively lower with increasing K (it is always in the best or second-best group); and
- WUATCQ is always in the worst group.

Finally, for each K value, if we were to recommend a single algorithm based on effectiveness considerations, it would be: WFCM for $K = 4$, PSOATCQ for $K \in \{16,64\}$, and ADU for $K = 256$.

It is important to emphasize that in our experiment, we compared these CQ algorithms along a single dimension (MSE in the CIELAB color space). There are many other criteria on which one can compare CQ algorithms, including computational efficiency, simplicity (conceptual and implementation), and ease of use (e.g., as measured by the number of user-defined parameters). For example, while IOKM is inferior to PSOATCQ based on effectiveness (as quantified by MSE), it is vastly superior based on the other three criteria mentioned above. Therefore, there is no such thing as a universal CQ algorithm, and the best algorithm depends highly on the application requirements.

4 Conclusions and Future Research Directions

In this paper, we presented CQ100, a large, diverse, and high-quality dataset of 24-bit color images released under the CC BY-SA 4.0 license. We also demonstrated how CQ100 can be used to compare CQ algorithms based on the popular MSE metric (computed in the CIELAB color space). Future work includes a multivariate comparison of CQ algorithms over CQ100 based on several image fidelity metrics.

The use of CQ100 is not restricted to CQ. For example, it can also be used to develop, test, and compare filtering[59] or segmentation[60,61] algorithms. However, for some applications, the images in the dataset may need to be annotated by one or more human experts. For example, in a segmentation application, the annotation for a given image could include bounding boxes around objects of interest or a segmentation mask for the entire image. CQ100 is significantly larger and more diverse than its two main competitors (i.e., USC-SIPI and Kodak). Although the current size of the dataset appears to be adequate for its primary target application, which is an unsupervised learning task, it should be expanded if it is to be used for supervised learning tasks.

Data, Materials, and Code Availability

The CQ100 dataset presented in this paper is available at Ref. 62. The dataset includes the 100 (true-color) input images and their metadata, 8400 (reduced-color) output images (21 CQ algorithms $\times 100$ input images $\times \{4,16,64,256\}$ colors), and Microsoft Excel worksheets containing the MSE for each input/output image combination.

Acknowledgments

This material is based upon work supported by the National Science Foundation (Grant No. OIA-1946391). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The authors gratefully acknowledge the following open-source software:

1. median-cut algorithm (GIMP, Spencer Kimball, and Peter Mattis);
2. octree algorithm (ImageMagick, John Cristy);
3. marginal variance minimization algorithm (Utah Raster Toolkit and Craig E. Kolb);
4. binary splitting algorithm (Charles A. Bouman);
5. variance minimization algorithm (Xiaolin Wu);
6. self-organizing map algorithm (Anthony Dekker);
7. split-and-merge algorithm (Luc Brun);
8. multipletest software[41] (Salvador García and Francisco Herrera); and
9. BoxPlotR software[63] (Michaela Spitzer, Jan Wildenhain, Juri Rappsilber, and Mike Tyers).

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. G. Sharma, M. J. Vrhel, and H. J. Trussell, "Color imaging for multimedia," *Proc. IEEE* **86**(6), 1088–1108 (1998).
2. R. Ramanath et al., "Color image processing pipeline," *IEEE Signal Process. Mag.* **22**(1), 34–43 (2005).
3. https://pxhere.com/.
4. P. Heckbert, "Color image quantization for frame buffer display," *ACM SIGGRAPH Comput. Graph.* **16**(3), 297–307 (1982).
5. X. Wu, "Color quantization by dynamic programming and principal analysis," *ACM Trans. Graph.* **11**(4), 348–372 (1992).
6. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.* **31**(3), 264–323 (1999).
7. M. E. Celebi, ed., *Partitional Clustering Algorithms*, Springer (2015).
8. M. E. Celebi, "Forty years of color quantization: a modern, algorithmic survey," *Artif. Intell. Rev.* (2023).
9. J. L. Nieves et al., "Computing the relevant colors that describe the color palette of paintings," *Appl. Opt.* **59**(6), 1732–1740 (2020).
10. J. L. Nieves et al., "Psychophysical determination of the relevant colours that describe the colour palette of paintings," *J. Imaging* **7**(4), 72 (2021).
11. M. E. Celebi and A. Zornberg, "Automated quantification of clinically significant colors in dermoscopy images and its application to skin lesion classification," *IEEE Syst. J.* **8**(3), 980–984 (2014).
12. C. Barata et al., "Clinically inspired analysis of dermoscopy images using a generative model," *Comput. Vis. Image Underst.* **151**, 124–137 (2016).
13. M. E. Celebi, H. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.* **40**(1), 200–210 (2013).
14. https://sipi.usc.edu/database/database.php?volume=misc.
15. http://r0k.us/graphics/kodak/.
16. https://commons.wikimedia.org/wiki/Main_Page.
17. https://imagemagick.org/.
18. https://creativecommons.org/.
19. D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," *Proc. SPIE* **5007**, 87–95 (2003).
20. M. Gervautz and W. Purgathofer, "A simple method for color quantization: octree quantization," in *New Trends in Computer Graphics*, N. Magnenat-Thalmann and D. Thalmann, eds., pp. 219–231, Springer (1988).
21. S. J. Wan, P. Prusinkiewicz, and S. K. M. Wong, "Variance-based color image quantization for frame buffer display," *Color Res. Appl.* **15**, 52–58 (1990).
22. M. Orchard and C. Bouman, "Color quantization of images," *IEEE Trans. Signal Process.* **39**(12), 2677–2690 (1991).
23. X. Wu, "Efficient statistical computations for optimal color quantization," in Graphics Gems II,J. Arvo, ed., pp. 126–133, Academic Press (1991).
24. A. Dekker, "Kohonen neural networks for optimal colour quantization," *Netw. Computat. Neural Syst.* **5**(3), 351–367 (1994).
25. L. Brun and M. Mokhtari, "Two high speed color quantization algorithms," in *Proc. 1st Int. Conf. Color in Graph. and Image Process.*, pp. 116–121 (2000).
26. M. E. Celebi, "Improving the performance of k-means for color quantization," *Image Vis. Comput.* **29**(4), 260–271 (2011).
27. Q. Wen and M. E. Celebi, "Hard vs. fuzzy c-means clustering for color quantization," *EURASIP J. Adv. Signal Process.* **2011**(1), 118–129 (2011).
28. M. E. Celebi, S. Hwang, and Q. Wen, "Colour quantisation using the adaptive distributing units algorithm," *Imaging Sci. J.* **62**(2), 80–91 (2014).
29. 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.* **10**(2), 329–344 (2015).
30. M.-L. Pérez-Delgado, "Colour quantization with ant-tree," *Appl. Soft Comput.* **36**, 656–669 (2015).
31. M.-L. Pérez-Delgado, "Artificial ants and fireflies can perform colour quantisation," *Appl. Soft Comput.* **73**, 153–177 (2018).
32. M.-L. Pérez-Delgado, "The color quantization problem solved by swarm-based operations," *Appl. Intell.* **49**(7), 2482–2514 (2019).
33. M.-L. Pérez-Delgado, "Color image quantization using the shuffled-frog leaping algorithm," *Eng. Appl. Artif. Intell.* **79**, 142–158 (2019).
34. M.-L. Pérez-Delgado and J.-Á. Román-Gallego, "A hybrid color quantization algorithm that combines the greedy orthogonal bi-partitioning method with artificial ants," *IEEE Access* **7**, 128714–128734 (2019).
35. M.-L. Pérez-Delgado, "Color quantization with particle swarm optimization and artificial ants," *Soft Comput.* **24**(6), 4545–4573 (2020).

36. M.-L. Pérez-Delgado, "A mixed method with effective color reduction," *Appl. Sci.* **10**(21), 7819 (2020).
37. M.-L. Pérez-Delgado, "Revisiting the iterative ant-tree for color quantization algorithm," *J. Visual Commun. Image Represent.* **78**, 103180 (2021).
38. A. Abernathy and M. E. Celebi, "The incremental online k-means clustering algorithm and its application to color quantization," *Expert Syst. Appl.* **207**, 117927 (2022).
39. M. Anderson et al., "Proposal for a standard default color space for the internet—sRGB," in *Proc. Color and Imaging Conf.*, pp. 238–245 (1996).
40. B. Ortiz-Jaramillo et al., "Evaluation of color differences in natural scene color images," *Signal Process. Image Commun.* **71**, 128–137 (2019).
41. S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *J. Mach. Learn. Res.* **9**, 2677–2694 (2008).
42. M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Am. Stat. Assoc.* **32**(200), 675–701 (1937).
43. R. L. Iman and J. M. Davenport, "Approximations of the critical region of the friedman statistic," *Commun. Stat. Theory Methods* **9**(6), 571–595 (1980).
44. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.* **7**, 1–30 (2006).
45. J. Luengo, S. García, and F. Herrera, "A study on the use of statistical tests for experimentation with neural networks: analysis of parametric test conditions and non-parametric tests," *Expert Syst. Appl.* **36**(4), 7798–7808 (2009).
46. S. García et al., "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability," *Soft Comput.* **13**, 959–977 (2009).
47. S. García et al., "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics* **15**(6), 617–644 (2009).
48. J. Carrasco et al., "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review," *Swarm Evol. Computat.* **54**, 100665 (2020).
49. W. W. Daniel, *Applied Nonparametric Statistics*, 2nd ed., PWS-KENT Publishing Company (1990).
50. B. Bergmann and G. Hommel, "Improvements of general multiple test procedures for redundant systems of hypotheses," in *Multiple Hypotheses Testing*,P. Bauer and G. Hommel andE. Sonnemann, eds., pp. 100–115, Springer (1988).
51. J. Derrac et al., "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Computat.* **1**(1), 3–18 (2011).
52. R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: recent advances and perspectives," *Inf. Fusion* **41**, 195–216 (2018).
53. U. Johansson et al., "Rule extraction with guarantees from regression models," *Pattern Recognit.* **126**, 108554 (2022).
54. G. Hommel and G. Bernhard, "Bonferroni procedures for logically related hypotheses," *J. Stat. Plann. Inference* **82**(1–2), 119–128 (1999).
55. P. B. Nemenyi, "Distribution-free multiple comparisons," PhD thesis, Princeton University (1963).
56. S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Stat.* **6**(2), 65–70 (1979).
57. J. P. Shaffer, "Modified sequentially rejective multiple test procedures," *J. Am. Stat. Assoc.* **81**(395), 826–831 (1986).
58. G. Hommel and G. Bernhard, "A rapid algorithm and a computer program for multiple test procedures using logical structures of hypotheses," *Comput. Methods Programs Biomed.* **43**(3–4), 213–216 (1994).
59. M. E. Celebi, H. A. Kingravi, and Y. A. Aslandogan, "Nonlinear vector filtering for impulsive noise removal from color images," *J. Electron. Imaging* **16**(3), 033008 (2007).
60. Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(8), 800–810 (2001).
61. M. Mignotte, "Segmentation by fusion of histogram-based k-means clusters in different color spaces," *IEEE Trans. Image Process.* **17**(5), 780–787 (2008).
62. https://data.mendeley.com/datasets/vw5ys9hfxw/2.
63. M. Spitzer et al., "BoxPlotR: a web tool for generation of box plots," *Nat. Methods* **11**, 121–122 (2014).

M. Emre Celebi received his PhD in computer science and engineering from the University of Texas at Arlington, United States. Currently, he is a professor and the chair of the Department of Computer Science and Engineering at the University of Central Arkansas, United States. He has published 170+ articles on image processing/analysis and data mining. According to Google

Scholar, his work has received more than 14,000 citations so far. He is a senior member of the IEEE and a fellow of the SPIE.

María-Luisa Pérez-Delgado received her engineering degree in computer science from the University of Valladolid, Spain and a PhD from the University of Salamanca, Spain, where she is currently a professor. Her research interests include artificial intelligence, optimization, graph theory, and data mining. She has published several research articles and books in these areas.