

Research Article

Global Distribution Adjustment and Nonlinear Feature Transformation for Automatic Colorization

Terumasa Aoki¹ and Van Nguyen²

¹New Industry Hatchery Center (NICHe), Tohoku University, Sendai 9808579, Japan

²Graduate School of Information Science (GSIS), Tohoku University, Sendai 9808579, Japan

Correspondence should be addressed to Terumasa Aoki; aoki@iec.tohoku.ac.jp

Received 28 July 2017; Revised 18 November 2017; Accepted 3 December 2017; Published 10 January 2018

Academic Editor: Jianping Fan

Copyright © 2018 Terumasa Aoki and Van Nguyen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automatic colorization is generally classified into two groups: propagation-based methods and reference-based methods. In reference-based automatic colorization methods, color image(s) are used as reference(s) to reconstruct original color of a gray target image. The most important task here is to find the best matching pairs for all pixels between reference and target images in order to transfer color information from reference to target pixels. A lot of attractive local feature-based image matching methods have already been developed for the last two decades. Unfortunately, as far as we know, there are no optimal matching methods for automatic colorization because the requirements for pixel matching in automatic colorization are wholly different from those for traditional image matching. To design an efficient matching algorithm for automatic colorization, clustering pixel with low computational cost and generating descriptive feature vector are the most important challenges to be solved. In this paper, we present a novel method to address these two problems. In particular, our work concentrates on solving the second problem (designing a descriptive feature vector); namely, we will discuss how to learn a descriptive texture feature using scaled sparse texture feature combining with a nonlinear transformation to construct an optimal feature descriptor. Our experimental results show our proposed method outperforms the state-of-the-art methods in terms of robustness for color reconstruction for automatic colorization applications.

1. Introduction

Automatic colorization is a technology to give correct color information for each pixel of a colorless (gray-scale) image automatically or semiautomatically. This technology is generally classified into two groups: propagation-based methods and reference-based methods. Propagation-based methods are also called interactive methods or semiautomatic methods. These methods require dozens of color feeding given by user and generate color of gray pixels by propagating the provided clues (color information) [1]. In reference-based automatic colorization methods, color image(s) are used as reference(s) to reconstruct original colors of a gray target image [2]. Also, some state-of-the-art systems combine these two approaches. The method that Irony et al. introduced in [3] uses Discrete Cosine Transform (DCT) coefficients of a k -by- k neighborhood around each pixel as

the underlying feature vector. After clustering pixels by using PCA (Principal Component Analysis), only the reliable pixels receive colors from reference(s), and these colors are spread to the whole segment using optimization color interpolation introduced by Levin et al. [1]. On the different attempt, Chia et al. [4] introduced a semiautomatic system for colorizing monochrome images with less user efforts. User is required to provide a label for each target image to use as searching keyword and a segmented image as a filter to obtain the most appropriate reference image from the Internet.

In reference-based methods, the most important task is to find matching pairs of all pixels between two images in order to transfer color information from reference to target pixels. A lot of good local feature-based image matching methods such as SIFT and SURF have already been developed for the last two decades. When we introduce these mature technologies for our purpose, it may seem easily to implement

an excellent automatic colorization system. Unfortunately, as far as we know, there are no optimal matching methods for automatic colorization because the requirements for pixel matching in automatic colorization are wholly different from those for traditional image matching. Although traditional image matching methods usually compute only the correspondences between interest points in two images, automatic colorization methods must find the correspondences between all pixels in reference and target images to obtain a good output image. This causes the following two serious problems.

The first problem is huge computational cost. Since one image usually contains a few hundred interest points whereas the number of pixels in an image is often a few millions, computational cost of pixel matching for automatic colorization is much higher than that of traditional image matching. To solve this problem, pixel clustering is widely used before matching process because it can reduce computation cost by matching each cluster instead of matching each pixel. The methods by Gupta et al. [2] uses superpixel as a fundamental component for matching process. Although there are many proposed methods such as [5, 6] to extract superpixels from input images with high accuracy, the use of superpixel still does not solve this matching problem perfectly since it is difficult to extract a large size of region (superpixel) due to the squared-shape constraint. In fact, superpixel extraction always suffers from the tradeoff between high accuracy and redundancy (the size of each superpixel). For this demand, we apply the recently introduced pixel classification method [7], which enhances the Color Lines concept [8]. This method is a promising solution for arbitrary pixel classification without the strict constraint on the pixel distribution.

The second problem is matching performance. In traditional image matching (local feature-based matching) methods, the systems define the pixels which are more discriminative than surrounding pixels as interest points. Matching process is successfully done since each interesting point is discriminative. Unfortunately, in automatic colorization, all pixels (or all clusters) in reference and target images must be matched to each other regardless of their discriminative powers. This will degrade the matching performance. How to enhance the description ability for nondiscriminative points (or clusters) is a vital problem in automatic colorization.

There exist many researches evolving sparse coding for image reconstruction; however, as far as we know, there are no methods to reconstruct color information from gray-scale images. In this paper, we present a scaled sparse texture feature representation algorithm which extracts the feature descriptor for pixels under the condition that color information will be preserved.

The rest of this paper is organized as follows. The study on the previous works is presented in Section 2. Section 3 discusses in detail our contribution on learning descriptive feature using Expectation Maximization model. Section 4 provides the introduction about the hemisphere-based pixel classification and how it is applied into our work. Section 5 is dedicated for talking about our achievements using the proposed method and in Section 6 we describe the conclusions of this paper and the future plans for our work.

2. Related Works

As described in Section 1, there are two main tasks in matching process for automatic colorization: classifying pixel and finding descriptive feature vector. For general purpose of data classification, there exist many researches which can be applied to many different applications of data analysis. Depending on the flexibilities of classifying data patterns, existing methods can be categorized into two main techniques: fuzzy clustering and crisp clustering. Fuzzy clustering, which is originally introduced in [9], attracts various researches that assess the membership degrees for each data pattern and corresponding prospective clusters. The methods proposed in [10–14] fall on this category. On the different approach, the goal of crisp clustering is nonoverlapping classification; namely, a data point belongs to single specific cluster. Crisp clustering covers most of clustering algorithms and achieves many outstanding results. Thus, wealthy researches have been revealed in this category including k -means [15], its derivatives [16–18], and many other methods [19–24].

Fuzzy c -means clustering algorithm, which was first introduced in [9], attempts to minimize mean square errors between the number of clusters and the distances to cluster centroids simultaneously by updating the set of membership degrees for each data pattern with respect to prospective clusters. The output is a matrix, in which each row is the membership degrees of data points corresponding to the set of available clusters. Although original fuzzy method requires desired number of clusters, the work in [10] tries to address this obstacle by automatically estimating the optimal number using Xie-Beni formula [14]. With the similar purpose, Yu et al. [12] proposed a fuzzy method which uses ant colonial optimization (ACO) to find the optimal clusters. This work also improves the choice of the proper initial centroids instead of randomly initial points. The authors quantize each RGB channel into the cells and use them as initial clusters. Cluster kernels are updated using ACO while pixels are categorized based on fuzzy c -means scheme. Although ant algorithm is a good method, it always suffers from high computational complexity. It requires a lot of efforts to reduce computing cost by using quantized pixel colors and the settlements for proper quantizing levels.

k -means can effectively cluster data patterns to the most proper partitions if a reasonable number of expected clusters is provided. Challenging with the drawback that k -means tends to detect the hyper-spherical-shaped clusters although clusters should take arbitrary shapes and sizes in the real case, Li et al. [25] explored the symmetrical distance instead of using Euclidean distance to measure the dissimilarity between data patterns. The obtained clusters from Su and Chou [16] are more robust for data set containing non-spherical-shaped clusters. Another derivative of k -means is global k -means clustering presented by Likas et al. [18]. k -means heavily suffers from the initial conditions in which cluster kernels are randomly decided and proceeds by iteratively relocating the kernels to minimize the clustering errors. In order to eliminate this situation, Likas et al. propose a global search. It incrementally adds up one optimal kernel

for every searching process iteratively until reaching the expected number k . This method guarantees that an optimal result can be obtained without the random initialization of centroid locations for every search process of a new kernel. One of the other attractive crisp methods was presented as the derivation of k -medoid [25] where Guha et al. initially treat every single point as the representatives of the clusters they belong to. In each successive step, the closest pair of the representatives is merged to one cluster until obtaining k clusters using the distance between them as the measurement. Although the current fuzzy and crisp approaches have achieved the wide range of improvements in clustering algorithms, there still exist major drawbacks when they are used for image processing applications. The pixel classifier model introduced in [7] holds the advantages that can satisfy these demands. It does not need user effort in providing any input parameters for the clustering process, and it can also automatically adapt to the changing of input data. Therefore, this classifier can be a promising method for our current implementation.

On the other hand, with the second task (construction of a descriptive feature vector), there are some introduced works dealing with learning an efficient feature set. In [26], Hazırbaş et al. proposed a method to learn appropriate feature vector from a predefined dictionary. The purpose of this method is to learn the optimal feature set for semantic segmentation while reducing redundancy in constructing a descriptive feature vector. In this work, they use color information of each pixel as parts of the feature dictionary, but it is inapplicable for automatic colorization since color information of a target image is missing. Besides, to reduce the redundancy of feature descriptor, they measure Mutual Information (MI) between each pair of features and try to reduce this number in the learning process such that the larger value of MI denotes higher redundancy of features set. However, MI is not a good measurement for classification and redundancy deduction since it measures the exact number of equal feature values at a particular pixel position. For the purpose of reference-based automatic colorization, reference images are quantized as fewer colors than original images to deduct the redundancy. In other words, the pixels with similar color are usually classified into each group as the matching components. In a different work [27], Omer and Werman exploited the sparse representation method to find descriptive representation of color images. By achieving outstanding results, this research shows its robustness and applicability for image denoising and color image restoration, while, for our target of automatic colorization, color information cannot be used under any condition. Therefore, it strongly desires an efficient feature descriptor, which can help classify pixels into proper clusters and preserve their similarity without requiring color information. In an intuitive way, texture information and gray scale of pixels can be the promising resources. In Section 3, we will introduce a learning algorithm to obtain the optimal feature descriptor for image pixels. Our algorithm is conducted using Expectation Maximization technique to learn a descriptive feature that maximizes the color similarity between the reconstructed color image and the original one.

3. Expectation Maximization (EM) Model for Color Recovery

3.1. Design Concept. As for reference-based automatic colorization, color reference images and a gray target image are available. To reduce the computational cost of matching process between them, we attempt to classify the pixels with similar color into the same class. Since we cannot use color information, we exploit the texture feature as the representative information for pixels to help them keep their original color identity. Although there exist various types of texture features which are used for particular purposes, some of them are also strongly correlated. So it is not necessary to use many texture features simultaneously as pixel's representation. Besides, it is difficult to choose a best descriptive feature for different images since their content and texture are not stable. For these reasons, we present a feature learning method to learn a descriptive feature vector for individual image. For the purpose of application in reference-based image colorization, this feature vector must satisfy three requirements. First, it should be extracted from gray-scale pixels. Second, it should be a good representative for missing color information so that we can use them to classify into "similar color" clusters without knowing original color of each pixel. Third, the distribution of feature vector should be similar to color vector distribution. To evaluate our proposed algorithm, we need to know whether the learned feature vector is representable for missing color information. Fortunately, references are color images, so that we can use them as the training samples in our feature learning algorithm. In automatic image colorization, a reference image is usually selected from the images whose composition or texture characteristics are similar to a target image. As a result, the learned features set from the reference image(s) are useful to handle the characteristics of the pixels extracted from a target image. In our learning strategy, a texture feature dictionary is extracted from gray-scale reference images first. Then, a descriptive feature vector is learned from this dictionary, and original color of reference pixels is used to evaluate the learning result. Finally, the learned feature set is applied to the target gray-scale image.

Now, we assume that the distributions of feature vectors and color channels are similar because feature vectors are representatives for color information of each pixel. Besides, we want to learn the appropriate feature set which can represent the color information of pixels most effectively. This turns into the desire to regress the color distribution model with a given feature set $P(C_{Re} | \theta)$, where θ is a parameter set containing feature dictionary and C_{Re} is the color of recovered image. To simplify the computational process, instead of direct estimation of this model, we try to minimize the difference between original color (C_{Or}) distribution $P(C_{Or})$ and $P(C_{Re} | \theta)$; namely, $MinDiff = abs(P(C_{Or}) - P(C_{Re} | \theta))$. This process is composed of two main steps: estimating the most appropriate feature that satisfies $MinDiff$ and direct optimization of the estimated feature by modifying the distribution traits of feature pixels. These two steps can be generalized by applying Expectation Maximization algorithm.

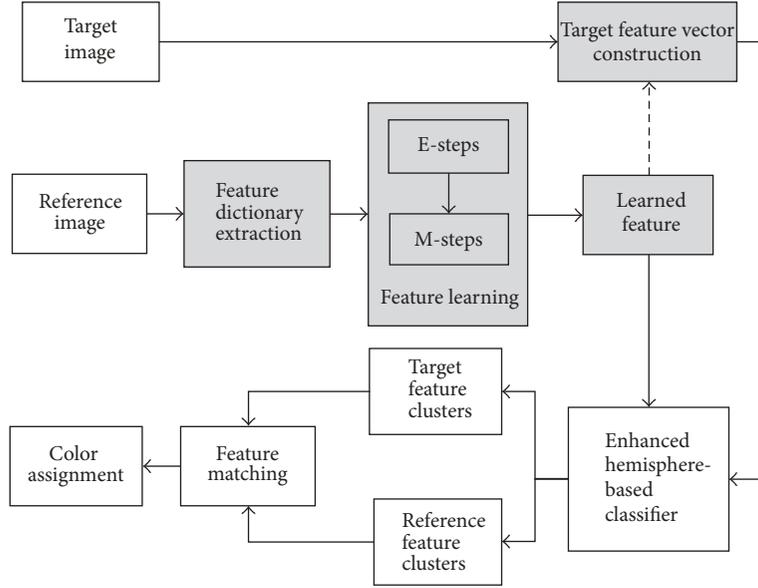


FIGURE 1: Complete system diagram for colorization using learned feature vectors.

3.2. System Overview. Figure 1 shows the system diagram of our proposed colorization system where the shaded blocks are the main contributions of this paper. The rest of this paper will discuss the further details for each step. After learning a proper feature vector for reference images, the corresponding feature types are extracted from target image to create the feature set. The white blocks are the extra steps when applying learned feature vectors to colorization. In these steps, we use enhanced hemisphere-based classifier [8] for classifying pixels into classes.

EM algorithm has been used for fitting Gaussian mixture models in many visual applications such as face detection application [28], and it has been proved to achieve the good results. In our research, we handle the idea of EM algorithm to fit the color distribution model for the feature vectors so that the expected feature set is the most similar to the distribution of RGB color. By this similarity, the correlation between features will be close to that of color channels, and we can describe this relationship by the following formula:

$$\text{MinDiff} \propto \text{abs}(10^{\text{corr}_{F_i}} - 10^{\text{corr}_{C_j}}), \quad (1)$$

where corr_{F_i} is the correlation between feature elements of current learning feature set F_i and corr_{C_j} is the correlation between RGB color elements. We take the exponent of these values to normalize it within the range of PSNR since the correlation range is in between 0 and 1.

When using this learned feature, we can infer the original color for image pixels by simply propagating color from clusters' kernels. The similarity between inferred and the original color implies the lower value of MinDiff, namely, $\text{MinDiff} \propto -\text{PSNR}$ between inferred image and the original image. We can capture this relationship by

$$\text{MinDiff} \propto -10 \log_{10} \frac{m \times n \times \max(I_{ij})^2}{\sum_i \sum_j (I_{ij} - K_{ij})^2}, \quad (2)$$

where m and n are the width and height of an image and I_{ij} and K_{ij} are color values of the original and inferred images.

The PSNR level assesses the quality of our algorithm. We use the method proposed in [8] to cluster feature vectors. PSNR level is calculated between the original color image and the recovering color image by propagating color of cluster kernels to their belonging pixels. The higher value of PSNR means higher similarity between training and recovering color images. Our goal is to learn utmost four-dimensional feature vectors from an n -dimensional dictionary. To do this task, we combine two constraints for learning process. The first constraint is to maximize the similarity between the recovered color image and the original image, which is presented as the PSNR level. The second constraint is to formulate a feature vector which has similar distribution to RGB color. We measure this similarity by using the correlation value. The subtraction between these two values needs to be as maximum as possible. To sum up, we add these two constraints in the following equation:

$$\theta = \arg \max (\text{PSNRF}(I, RI) - \text{avg}_F (\text{abs}(10^{\text{corr}_F} - 10^{\text{corr}_C}))) \quad (3)$$

or

$$\theta = \arg \max \left(10 \log_{10} \left(\frac{m \times n \times \max(I_{ij})^2}{\sum_i \sum_j (I_{ij} - K_{ij})^2} \right) - \text{avg}_F (\text{abs}(10^{\text{corr}_F} - 10^{\text{corr}_C})) \right), \quad (4)$$

where $\theta = \{f_1, f_2, f_3, [f_4], \alpha, \beta\}$ with $\{f_1, f_2, f_3, [f_4]\}$ is the feature set which needs to be estimated in EM algorithm. α, β are the parameters in nonlinear transformation for

optimizing in M-step. i is the color channel and j is pixel index of the original image. PSNR F is the corresponding PSNR of the learning feature set F , and I and RI are the original and recovered color images from classified clusters.

As discussed above, because color information cannot be used in our implementation, we instead exploit pixel intensity, standard deviation of pixels and its neighbors, and the GLCM (Gray-Scale Level Correlation Matrix) [7] features. To be more specific, pixel intensity keeps its original value, and standard deviation feature contains two elements: pixel standard deviation and the average value of its neighborhood. GLCM texture features include five feature values; those are means, dissimilarity, homogeneity, entropy, and correlation feature. Besides, since texture properties vary widely in an image. To catch this variation, we explore the texture feature in three different scales to maximize the ability of handling the appropriate distinct characteristics of pixels. As a result, our learning feature dictionary is the composition of 18-dimensional feature vectors.

Our algorithm is conducted by applying EM (Expectation Maximization) algorithm [28]. In the expectation step (E-step), we pay attention to optimizing the initial feature set which maximizes PSNR and the similarity of the correlation with RGB components. Maximization step (M-step) is the optimization step which applies nonlinear transformation to maximize the similarity between the correlation of feature and color elements. The detail of our algorithm is presented in Sections 3.3 and 3.4.

3.3. Expectation Step (E-Step). E-step initially learns the appropriate three-dimensional elements of a representative feature vector. In this step, we try to find the feature F and set the parameter $(\alpha, \beta) = (0, 0)$ to represent that they are not considered in the E-step. No transformation will be investigated in this early attempt. Since our dictionary is an 18-dimensional vector in each learning iteration, one combination of three different elements is denoted as the feature $F = \{f_1, f_2, f_3, [f_4]\}$. We use this feature as the input for pixel classification process, in which enhanced hemisphere-based clustering method is used. Then the color is propagated to the classified pixels from their cluster kernels. The PSNR value is calculated accordingly by using original and inferred color image following right hand side of formula (4).

Besides, the correlations between considering feature elements are calculated as the measurement for their distribution characteristics. The correlations between RGB color channels are also calculated accordingly. The subtraction between these two values measures the similarity of distribution characteristic of feature and RGB vectors. We define this value as the global parameter and try to simultaneously minimize this parameter as the likelihood between feature and color component since the purpose of learning feature set is to recover color information of pixels. It is intuitively to expect that the larger similarity between the distribution of feature vectors and RGB components means the higher probability that pixels are classified in the respective group in the feature domain.

By using (4), we can handle these two conditions simultaneously. The final obtained features produce highest value of learning cost. And the target of E-step is to define just the most appropriate descriptive feature set for input image.

3.4. Maximization Step (M-Step). In the M-step, we take three-dimensional feature vectors learned from E-step into account as input and continue using (4) as our learning formula. With the purpose to strengthen learned feature vector, in the M-step, we will learn extra feature dimension with the expectation that higher dimensional vectors are more descriptive for gray pixels. We also consider the tradeoff between computational cost and the robust of feature vectors; therefore only one more dimension will be learned to concatenate to the obtained three-dimensional feature vector after E-step.

Since the purpose of feature learning is to cluster the pixels into clusters while preserving their color information after recovering, the distribution of feature vector is expected to be similar with the distribution of RGB vectors. Besides, the correlations have been investigated as the measurement for this scattering information in the E-step. In this optimization, we apply the nonlinear transformation algorithm for individual feature value which can lead to the modification in global distribution of feature vector. For the purpose of our learning process, we intent to maximize the similarity of spatial distribution and color vectors as well as increase the accuracy of classifier. Besides, the clustering algorithm introduced in [8] that we use for pixels classification achieves good results when the vectors have Gaussian or Mixture of Gaussian distributions. To make our learned feature distribute "more Gaussian," in the M-step, we conduct a nonlinear transformation over the learned feature set from the E-step.

As plotted in Figure 2, the logarithm transformation is expected to be able to manipulate the pixels distribution to bell form. We take control of this transformation by using two additional parameters α as the variance and β as the bias component for our operation. We use the log function as the nonlinear transforming operator presented in the following expression:

$$f_i^t = \alpha \log(f_i + \beta), \quad (5)$$

where f_i, f_i^t are the feature value before and after transformation and (α, β) are the variance and bias of transforming operator.

The M-step is performed to find the optimal pair of values (α, β) , which satisfies (4). To make our algorithm more reasonable, we consider the parameter α as the normalizing variable which keeps controlling the value of transformed feature range in $(0, 255)$ while β is adjusted. In this nonlinear transformation, the value of β controls the position of Gaussian distribution on horizontal axis or plane, as the value of means μ of Gaussian model if fitting a normal distribution model for modified feature elements. On the other hand, the value of α affects the magnitude and the width of the Gaussian or moves the Gaussian peak upward or downward vertically. Therefore, by adjusting the value of (α, β) , we can control

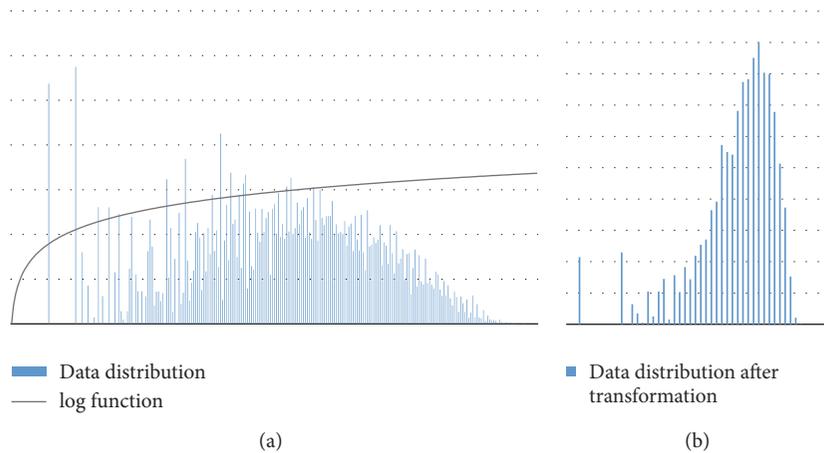


FIGURE 2: Data distribution example with nonlinear logarithm transformations (a). By applying this transformation, data distribution is expected to be more bell-shaped (“more Gaussian”) (b) that successfully modifies the distribution of pixels and supports the accuracy of the classifier proposed in [8].

the correlations between feature elements, and consequently the result of the classification algorithm applied on feature vectors can be controlled.

The evaluation method keeps similar as in the E-step by computing PSNR of the recovered image and the correlation value between new transformed feature elements. A new pair of values (α, β) is handled if formula (4) achieves larger value than the value obtained from E-step. The result of this step is expected to be the couple of (α, β) that change the distribution of feature vector cloud as most similar to that of color points as possible.

4. Enhanced Hemisphere-Based Classification

In [8], the authors introduced a method using enhanced hemisphere concept for color pixel classification where the grid of hemisphere is used to geometrically slice the coordinate systems into grids. Each grid contains a cluster of color pixels. With this approach, the distribution of color pixels is assumed to show mixture of Gaussian bell shapes. The experimental results prove the robustness of the introduced method for different set of color images. These achievements also show that the enhanced hemisphere-based clustering method can be used for various RGB color pixels distribution.

In this paper, we investigate the advantages of the classifier using enhanced hemisphere concept for feature vectors of pixels. As the purpose of our research, we want to present a fully automatic colorization method, which can respond for the different types and changes of image context. Among major steps of automatic colorization, classification is one of the most important processes to group the similar color pixels into the same cluster so that it can significantly reduce the matching cost. We found this capability from the enhanced hemisphere clustering concept fits to our purpose. Beside, in our approach, we use feature vector to carry on the characteristics of pixels, especially keeping the relation of

colors across pixels as much as possible. Since we transform the learned feature elements into Gaussian-like distribution as described in Section 3, the distribution of feature vectors is expected to be similar to the color pixels distribution. For this reason, we try to investigate the enhanced hemisphere-based clustering for feature vectors instead of RGB vectors in the original use in [8].

The process to conduct the enhanced hemisphere-based classifier for feature vector starts with considering each feature value as a color channel. The first step is to estimate the number of hemisphere for feature clustering. We need to calculate the l_1 -norm and define the minimal norm of feature vectors and use it as the initial radius to estimate the next hemisphere boundary till reaching the maximum feature vector norm. The trial radius r_t is iteratively increased; in every attempt, define the corresponding number of feature points at the centroids (n_c) and the boundaries (n_b) of hemispheres, and calculate the decision cost $d_c = n_c/n_b$. The smaller value of d_c stands for lower value of $\sum I_{ij} P(I_{ij} | r_t)$ with I_{ij} being the pixel at position (i, j) in image coordinate system and r_t being the radius of hemisphere at current iteration. As a result, low value of d_c implies the regions where feature vectors gather sparsely (boundary) and vice versa. Based on this value, we define a new hemisphere slice when d_c goes lower than the previous value.

Once getting the number of hemisphere, we go to the next step to define the kernel point and cluster size of each feature cluster. As presented in [8], we define the kernels list by considering the feature points at the centroid region of each hemisphere slice. The radius and size of each cluster depend on the size of hemisphere slice so that we can dynamically estimate these values without requiring any predefined parameters.

Since the hemisphere-based clustering algorithm is a fully automating process, we can use it as a classifier for feature vector in our iterative process for estimating the required parameters in EM model with the outstanding achievements.



FIGURE 3: Original image (a) and the recovered color image (b) using learned three-dimensional feature vector after E-step.

5. Experimental Results and Discussion

In our experiments, we use three types of aforementioned feature vectors including pixel intensity, standard deviations, and GLCM texture feature. For GLCM texture features, we manipulate size of cooccurrence matrices with three scales 5×5 , 9×9 , and 13×13 . Five texture feature types are mean, dissimilarity, homogeneity, entropy, and correlation features that create an 18-dimensional feature dictionary. We perform our algorithm to learn m -dimensional feature vectors. Since higher number of m tends to keep more information about image pixels but requires higher time complexity, we only learn to get m to be equal to utmost 4 at the final result. We use the MIT Urban and Natural Scene dataset [29] as input images to evaluate our proposed algorithm.

We perform the first experiment to demonstrate the robustness and accuracy of E-step. For this purpose, we use the PSNR of reconstructed color image to measure how similar the recovered image and original image are. The number of feature elements is equal to three. The ability in keeping the original color by using only texture feature of pixels indicates the robustness of our algorithm in learning the appropriate feature set. Our algorithm alternatively tries every possible combination between triple of dictionary elements. This guarantees that the most suitable vector is found with the largest level of PSNR.

As demonstrated in Figure 3, the recovered image has the strongly similar color to the original image. This proves that our algorithm can learn the most appropriate feature set from the dictionary. Besides, the enhanced hemisphere-based method can effectively classify pixels into appropriate clusters while preserving their color information. It also shows that the scaled dictionary can effectively handle the characteristic of pixels for different textures.

M-steps are based on the idea that modifies the global distribution of feature vector to optimize the pixel description and classifier simultaneously. On the one hand, we learned an extra feature that makes the feature vectors more

descriptive for different texture types. As demonstrated in Figure 4, the missing color information when using only three-dimensional feature vectors has been compensated by learning fourth feature dimension.

On the other hand, we try to make the distribution of learned features become “more Gaussian.” Figure 5(a) plots the histogram of a feature element before transformation corresponding with the recovered image using this feature vector. The distribution of the feature value is not a “good Gaussian shape.”

Figure 5(b) shows the distribution of feature value after applying nonlinear transformation. It contains two bell shapes that can be inferred as the mixture of two Gaussian distributions. By adjusting the value of two parameters (α , β), we have successfully modified feature vector distribution that makes them more descriptive and discriminative. This proves that the use of nonlinear transform function to modify feature values can manipulate global distribution of feature vectors. After this transformation, classification method introduced in [8] achieves more accurate results that lead to more precision in the color recovery.

Table 1 summarizes our experiments with different images and structures. For each image, we collect the correlation between feature tuples and RGB color as well as the PSNR of reconstructed images before and after M-step to demonstrate the advantage of this optimization method. After the M-step, the correlation gets closer to the value of correlation between color channels; these prove that we have successfully modified the global distribution of pixels in feature domain that preserve the color information. Besides, the higher values of PSNR define the transformation and learning extra feature dimension produces the more descriptive feature vector than original three-dimensional one after E-step.

Figure 6 demonstrates the output of images using learned features as the criteria to classify pixels and cluster kernels as the seed color to reconstruct original pixel colors. It can be seen that, after E-step, there still exist some defects in color recovering results. M-step transforms feature values to

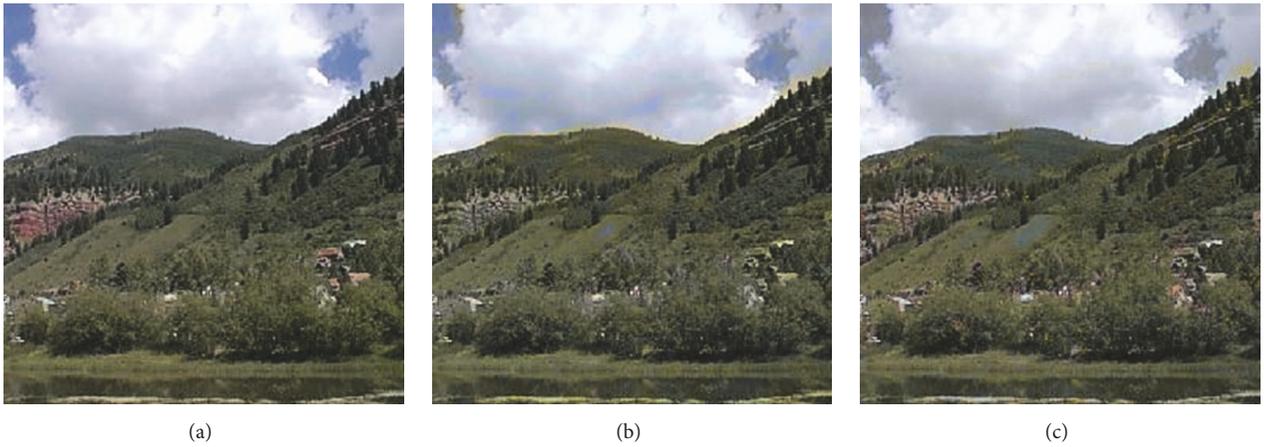


FIGURE 4: An extra feature tuple added after M-step to make learned feature more descriptive. As depicted in this image, the recovered image using three-dimensional feature (b) misses some colors from original image (a), and these defects have been eliminated after adding extra feature in M-step (c).

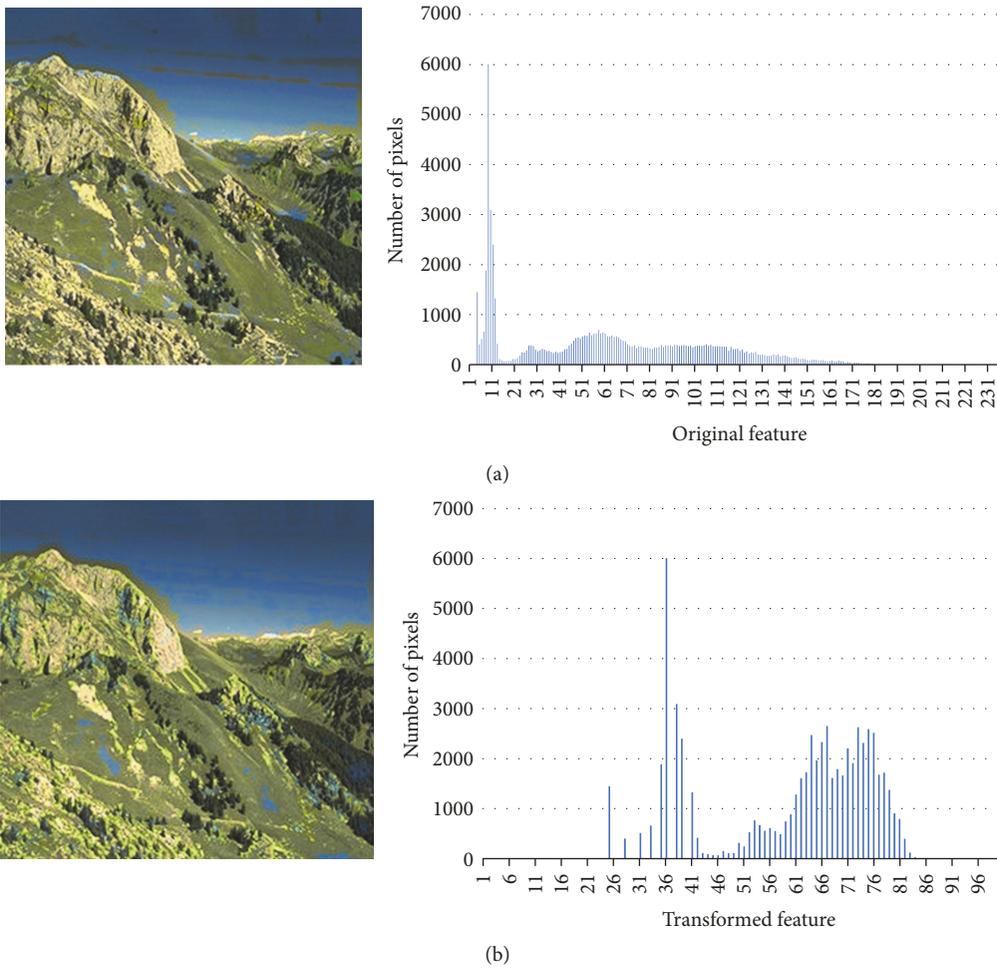


FIGURE 5: The distribution of learned feature values (after E-step) is not Gaussian (a). After nonlinear transformation in M-step, this distribution becomes more Gaussian as plotted in (b) that can help increase the accuracy of classifying feature vectors leading to the more precision in color reconstruction. As shown in (b), the defects are removed in recovered image.

TABLE 1: The results of learning process before and after applying M-step.

| Input | Color correlation | E-step | | M-step | |
|-------|-------------------|---------------------|---------|---------------------|----------------|
| | | Feature correlation | PSNR | Feature correlation | PSNR |
| (1) | 4.65218 | 1.70593 | 26.6034 | 2.98258 | <u>26.8436</u> |
| (2) | 6.17421 | 1.85361 | 27.4339 | 3.54563 | <u>27.6762</u> |
| (3) | 6.52125 | 1.97957 | 25.3198 | 5.1082 | <u>25.5997</u> |
| (4) | 5.45406 | 1.80969 | 26.3328 | 3.33898 | <u>27.0535</u> |
| (5) | 2.63201 | 1.59904 | 27.2102 | 2.61883 | <u>27.5409</u> |
| (6) | 4.21849 | 1.71946 | 27.0152 | 4.97776 | <u>27.6046</u> |
| (7) | 8.3623 | 1.49676 | 26.1107 | 2.59245 | <u>26.3373</u> |
| (8) | 3.81443 | 1.9547 | 23.039 | 3.78879 | <u>23.6149</u> |
| (9) | 6.04486 | 2.06075 | 23.9355 | 5.5012 | <u>25.3075</u> |
| (10) | 8.65296 | 2.36828 | 27.2904 | 6.48897 | <u>27.6309</u> |
| (11) | 6.20452 | 2.36094 | 23.3158 | 5.9459 | <u>23.5803</u> |
| (12) | 5.69085 | 2.15709 | 27.2325 | 5.66991 | <u>27.804</u> |

TABLE 2: The comparisons of PSNR and matching time when using different methods for matching process in colorization.

| Input | Feature learning | | Feature Line | | Superpixel | |
|-------|------------------|--------------------|--------------|--------------------|------------|--------------------|
| | PSNR | Matching time (ms) | PSNR | Matching time (ms) | PSNR | Matching time (ms) |
| (1) | <u>26.0869</u> | 160 | 25.5904 | 142 | 23.2869 | 134069 |
| (2) | <u>25.5709</u> | 109 | 24.8474 | 114 | 24.4649 | 109128 |
| (3) | <u>28.2651</u> | 130 | 24.9636 | 115 | 25.966 | 131581 |
| (4) | <u>29.0486</u> | 140 | 28.9886 | 122 | 27.2218 | 178977 |
| (5) | <u>26.0473</u> | 234 | 16.8502 | 201 | 20.6465 | 291383 |
| (6) | <u>24.5262</u> | 179 | 21.167 | 133 | 22.3369 | 108276 |
| (7) | <u>22.7022</u> | 172 | 19.3676 | 133 | 20.7267 | 70715 |
| (8) | <u>23.4184</u> | 281 | 22.0257 | 123 | 22.391 | 40560 |
| (9) | <u>23.1787</u> | 235 | 22.6112 | 177 | 21.5942 | 122036 |
| (10) | <u>26.5051</u> | 191 | 24.89 | 81 | 24.8915 | 40617 |
| (11) | <u>28.1289</u> | 110 | 26.9708 | 98 | 25.5595 | 52302 |
| (12) | <u>26.3759</u> | 130 | 24.6655 | 108 | 25.6511 | 139367 |

make distribution of feature vectors become more Gaussian that helps to increase the precision of enhanced hemisphere-based classifier. Besides, M-step also learns extra feature elements so that the final feature vectors are more concrete in handling characteristic of color pixels. When recovered images in Figure 6(C) are compared with the results after E-step (Figure 6(B)), we can see the excellence of our learning and optimizing algorithm.

The above experimental data prove the robustness and descriptive feature learning of our proposed algorithm. We conduct further experiment on colorization as the application of learned feature set. We set up three experiments that use our learned feature, Feature Line [30], and superpixel [2] as the material for matching process. We compare two aspects including matching time and PSNR level of colorizing images to assess the performance of our proposed method. The measurement of matching time represents how much redundancy is eliminated in the matching process; the smaller value, the more optimal process. PSNR levels measure how much similarity between colorized images and reference images;

the higher number, the better results. We summarize the obtained results in Table 2.

As demonstrated in Table 2, PSNR levels of our method are distinctly higher than the other two methods indicating that our method can produce more descriptive feature vectors. Besides, comparing matching time of our method and superpixel method we see that we have effectively reduce the computational time or redundancy in matching process while we still can keep more accurate matching results. The matching time of our method is just slightly larger than that of Feature Line approach; however our PSNR levels are much higher and more stable than those obtained of Feature Line method so that it is affordable to sacrifice these computational costs.

The PSNRs levels of colorized images using our method are higher than using Feature Line method indicating more accuracy not only in learned feature vector but also in output of feature classifier.

Figure 7 shows the output colorized images using our proposed method for learning feature vectors in comparison

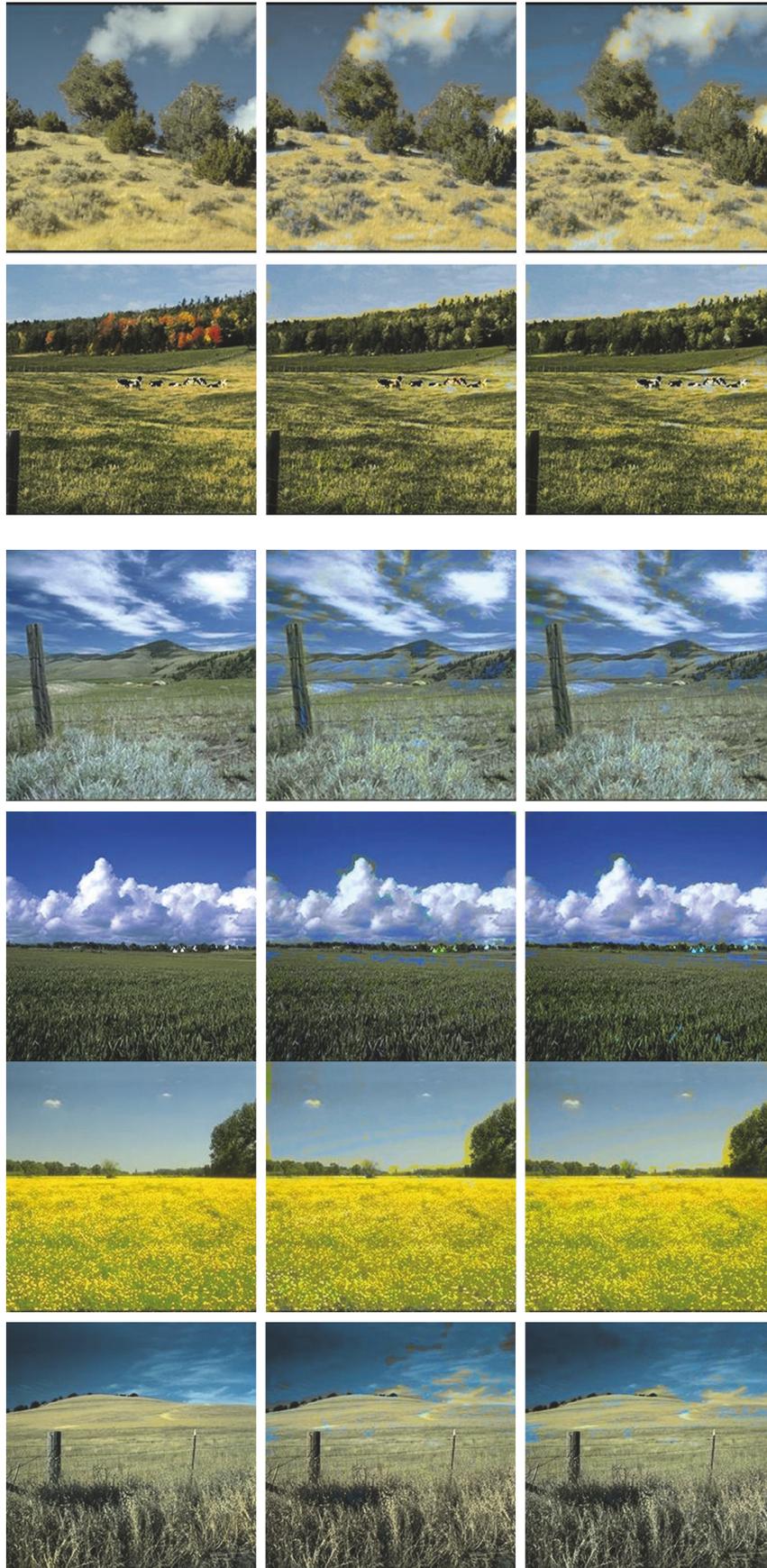
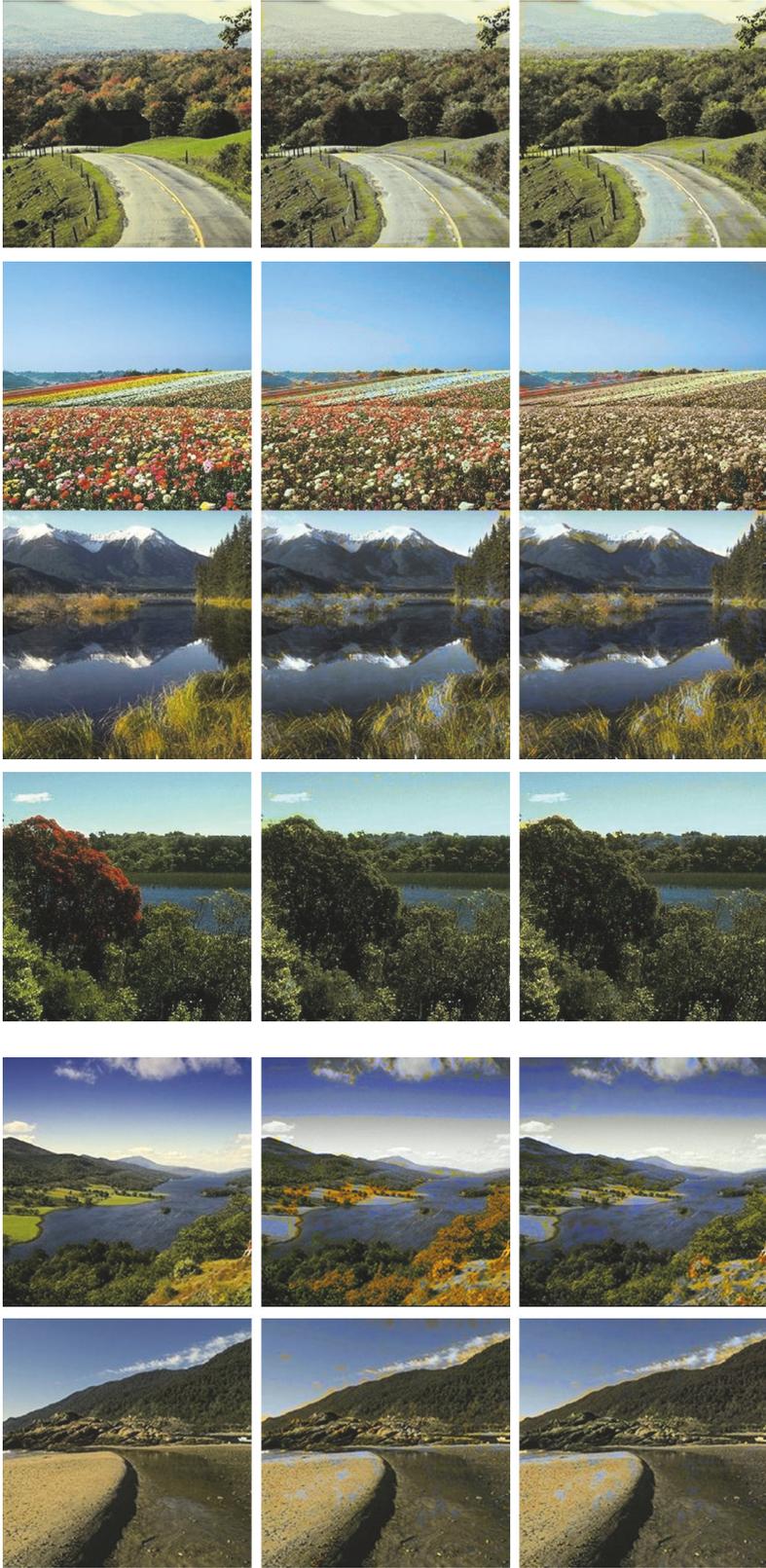


FIGURE 6: Continued.



(A)

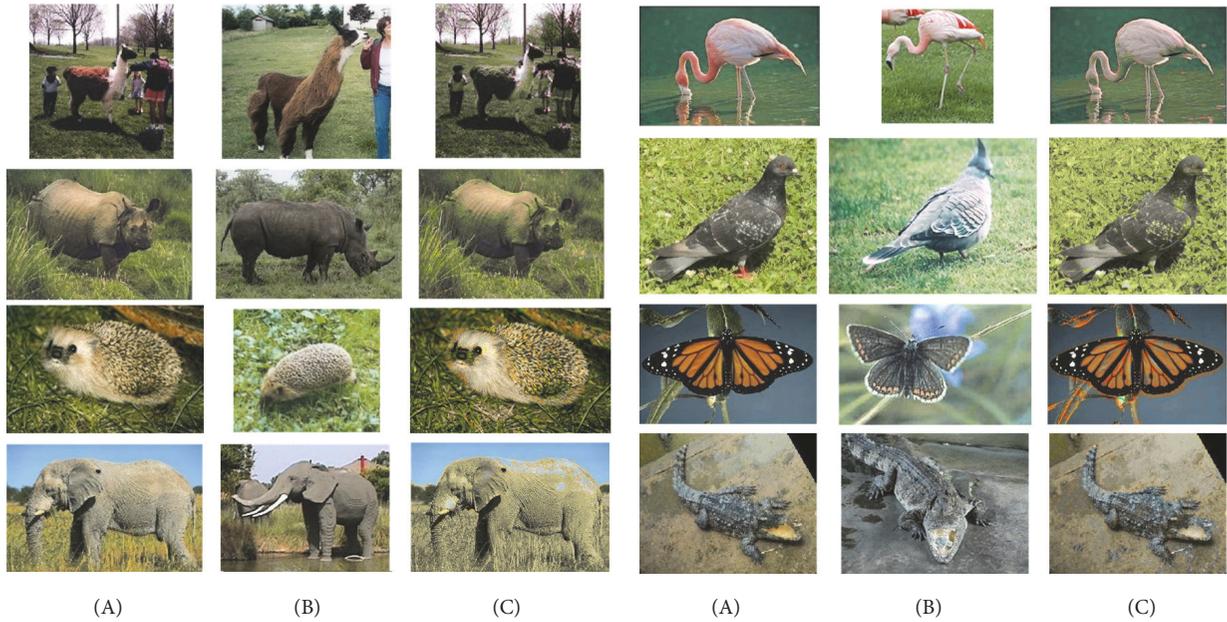
(B)

(C)

FIGURE 6: Color images (A) and recovered images using learned feature vectors after E-step (B) and M-step (C).

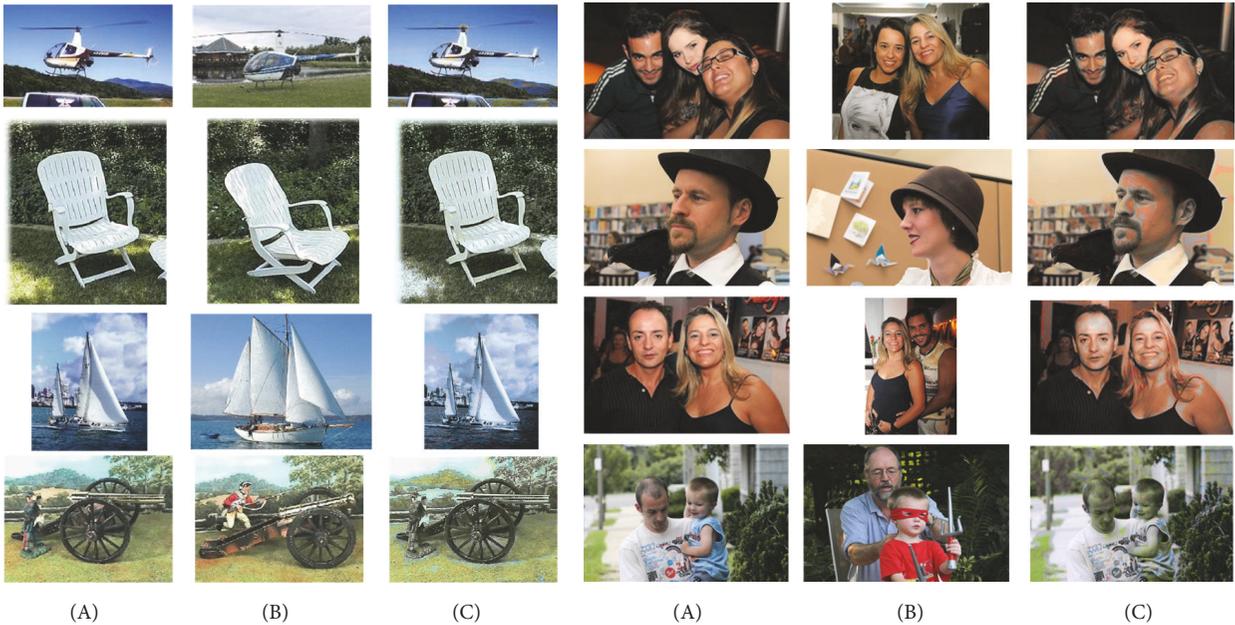


FIGURE 7: The outputs colorized images using our learned feature (C), Feature Line (D), and superpixel (E) method. Output images from our method contain less defects and more even than other two methods.



(a) The outputs colorized images using our proposed method for animals (C), Ground Truth (A), the gray image of which is an input image, and reference image (B)

(b) The outputs colorized images using our proposed method for other creations (bird, insect, and crocodile) (C), Ground Truth (A), the gray image of which is an input image, and reference image (B)



(c) The outputs colorized images using our proposed method for artificial objects (helicopter, chair, sailboat, and folk art object) (C), Ground Truth (A), the gray image of which is an input image, and reference image (B)

(d) The outputs colorized images using our proposed method for humans (C), Ground Truth (A), the gray image of which is an input image, and reference image (B). We find that human's clothes sometimes fail to get correct colors

FIGURE 8

with Feature Line and superpixel method. As can be seen in these images, our method can produce smoother and more even colorization results. It removes the jerky in color assignment if using superpixel method. It also archives more precise matching results than Feature Line method.

Traditionally, outdoor natural-scene images are often used in the experiments in this topic. In order to widen the

applications of image colorization, we applied our proposed method to other types of images such as animals, other creations (bird, insect, and crocodile), artificial objects, and humans. Some results of this experiment are shown in Figures 8(a)–8(d). These figures show our method has enough ability for these types of images. On the other hand, we found that colorization of artificial objects (especially, human's

clothes) is sometimes very difficult (for example, 4th row in Figure 8(d)) because a reference image does not contain the correct color of the clothes at all. This is not the problem for our proposed method but the problem for all reference-based image colorization methods. We think it is an important future work in this topic.

6. Conclusions and Future Work

In this paper, we presented a novel feature learning method using EM technique and nonlinear transformation to build a descriptive feature vector and to optimize the distribution of feature points in n -dimensional space for the purpose of achieving outstanding colorization results. In our method, E-step tries to find the most meaningful combination of feature tuples from dictionary. The obtained results after E-step show flexibility and robustness of our algorithm in learning set of feature types for each input image.

M-step shows its advantage since it can learn appropriate parameters (α, β) and additional feature, the distribution of which is closer to Gaussian. These steps help to satisfy the important assumption to obtain high precision in the classification step using enhanced hemisphere-based method [8].

Besides, we also performed the experiments that use learned feature set for colorization. As demonstrated in the Experimental Results sections, our method produces the more descriptive feature vectors which enhance the accuracy of matching process leading to the higher level of PSNR in color transferring from reference to target pixels. We use feature clusters as the representatives for their belonging pixels to match up similar regions in reference and target images, which can keep the number of clusters stable when the size of an image becomes bigger. So, the computational cost did not increase exponentially when image sizes increase as in superpixel approach.

In conclusion, our method has two advantages which support for the purpose of applying in image colorization work. It can automatically learn the most proper feature vector with the transformation step to correct the remaining minor errors in reconstructing a color image. Also, it can help to increase the precision in matching process while keeping the matching time did not soar too high in large size images.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proceedings of the Proceeding of the ACM Transactions on Graphics (ACM SIGGRAPH '04)*, pp. 689–694, Los Angeles, Calif, USA, August 2004.
- [2] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," in *Proceedings of the 20th ACM International Conference on Multimedia, MM 2012*, pp. 369–378, Japan, November 2012.
- [3] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," *Symposium A Quarterly Journal In Modern Foreign Literatures*, pp. 201–210, 2005.
- [4] A. Y.-S. Chia, S. Zhuo, R. K. Gupta et al., "Semantic colorization with internet images," *ACM Transactions on Graphics*, vol. 30, no. 6, 2011.
- [5] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.
- [6] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.
- [7] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
- [8] V. Ng and T. Aoki, "Enhanced Hemisphere Concept for Color Pixel Classification," in *Proceedings of the International Conference on Multimedia Systems and Signal Processing (ICMSSP)*, pp. 31–35, New Taipei, Taiwan, September 2016.
- [9] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [10] A. W.-C. Liew and H. Yan, "An Adaptive Spatial Fuzzy Clustering Algorithm for 3-D MR Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 22, no. 9, pp. 1063–1075, 2003.
- [11] N. A. M. Isa, S. A. Salamah, and U. K. Ngah, "Adaptive fuzzy moving K-means clustering algorithm for image segmentation," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2145–2153, 2009.
- [12] Z. Yu, O. C. Au, R. Zou, W. Yu, and J. Tian, "An adaptive unsupervised approach toward pixel clustering and color image segmentation," *Pattern Recognition*, vol. 43, no. 5, pp. 1889–1906, 2010.
- [13] S. Das and A. Konar, "Automatic image pixel clustering with an improved differential evolution," *Applied Soft Computing*, vol. 9, no. 1, pp. 226–236, 2009.
- [14] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.
- [15] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [16] M. C. Su and C. H. Chou, "A modified version of the GCA algorithm with a distance based on cluster symmetry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 674–680, 2001.
- [17] F. U. Siddiqui and N. A. M. Isa, "Enhanced moving K-means (EMKM) algorithm for image segmentation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 833–841, 2011.
- [18] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.
- [19] L. Kaufman and P. J. Rousseeuw, *Clustering by means of medoids*, 1987.
- [20] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

- [21] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 144–155, San Francisco, CA, USA, 1994.
- [22] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern Recognition*, vol. 35, no. 6, pp. 1197–1208, 2002.
- [23] A. Oliver, X. Munoz, J. Batlle, L. Pacheco, and J. Freixenet, "Improving clustering algorithms for image segmentation using contour and region information," in *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics*, vol. 2, pp. 315–320, May 2006.
- [24] G. W. Milligan, S. C. Soon, and L. M. Sokol, "The Effect of Cluster Size, Dimensionality, and the Number of Clusters on Recovery of True Cluster Structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 40–47, 1983.
- [25] N. Li, H. Huo, Y.-M. Zhao, X. Chen, and T. Fang, "A spatial clustering method with edge weighting for image segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1124–1128, 2013.
- [26] C. Hazırbaş, J. Diebold, and D. Cremers, "Optimizing the relevance-redundancy tradeoff for efficient semantic segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9087, pp. 243–255, 2015.
- [27] I. Omer and M. Werman, "Color lines: image specific color representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '04*, pp. II946–II953, USA, July 2004.
- [28] S. J. Prince, *Computer Vision (Models, Learning, and Inference) Algorithms*, Cambridge University Press, UK, 2013.
- [29] MIT Urban and Natural Scene Categories Dataset, 2016, <http://cvcl.mit.edu/database.htm>.
- [30] V. Nguyen, V. Sintunata, and T. Aoki, "Automatic image colorization based on feature lines," in *Proceedings of the 11th Joint Conference on Computer Vision*, pp. 126–133, 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

