Automated Colorization of a Grayscale Image with Seed Points Propagation

Shaohua Wan, *Senior Member, IEEE,* Yu Xia, Lianyong Qi, *Member, IEEE,* Yee-Hong Yang, *Senior Member, IEEE,* and Mohammed Atiquzzaman, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a fully automatic image colorization method for grayscale images using neural network and optimization. For a determined training set including the gray images and its corresponding color images, our method segments grayscale images into superpixels and then extracts features of particular points of interest in each superpixel. The obtained features and their RGB values are given as input for, the training colorization neural network of each pixel. To achieve a better image colorization effect in shorter running time, our method further propagates the resulting color points to neighboring pixels for improved colorization results. In the propagation of color, we present a cost function to formalize the premise that neighboring pixels should have the maximum positive similarity of intensities and colors; we then propose our solution to solving the optimization problem. At last, a guided image filter is employed to refine the colorized image. Experiments on a wide variety of images show that the proposed algorithms can achieve superior performance over the state-of-the-art algorithms.**

***Index Terms*—Neural network, Segmentation, Colorization, Optimization.**

I. Introduction

FULL automatic colorization is the process of adding color information to a greyscale image without significant user interaction. It is an ill-posed problem to determine the RGB values of a pixel due to lack of any prior information when in the grayscale value. Considering this defect, we need to find additional knowledge of the greyscale pixel to accomplish the colorization process.

In general, existing methods of providing additional color information of grayscale pixels fall into two broad categories: Scribble-based colorization methods [1], [2], [3], [4], [5] and example-based colorization methods [6], [7], [8], [9], [10], [11]. The Scribble-based approach requires the user to scribbles the objective grayscale image and then colorize it through a colorization optimization algorithm. A sufficient good understanding of the scene is required for the user to scribble adequate color graffiti inside the area; it is probably hard, if not impossible, for a typical user to provide enough appropriate color scribbles to achieve a good result.

Instead of obtaining chromatic values from the user, example-based colorization methods take a different approach to accomplish the task. A reference image of the same type serves as input in this approach, where it's the color feature information of the reference image is simultaneously passed to the objective grayscale image [12], [13], [14], [11], [15]. The method requires less effort and skill from the user over the Scribble-based method. The difference between the example-based colorization methods and the scribble based ones is that the scribble based require users to add color scribbles while the example based need transferring the color information from reference images to the gray image. They do not depend on the user's skills or experience to choose appropriate colors for a plausible colorization. The initial example-based colorization methods include image analogies [12] and color transfer algorithm [14]. Welsh *et al.* [11] proposed luminance and texture information matched colorizing method, which matches the features of selected swatches between the gray and color images. This type of work was later enhanced by Irony *et al.* [9] using spatial consistency criteria, that transferred color information to pixels of high confidence just in the same kind of segmented image regions, which then use the scribble based method to colorize. The global optimization framework is proposed by Charpiat *et al.* [6], which is based on colorizing a pixel by minimizing the cost function via the graph cuts methods. The superpixel matching based colorization conducted by Gupta *et al.* [8], which improves spatial coherency by feature correspondence and space weighting. These colorization methods can work without any human intervention. However, the quality of the colorized image is highly dependent on the selection of the reference images, which is a daunting task. To solve this problem, a colorization method that is steady when illumination changes among objective grayscale images and reference images was introduced by Liu *et al.* [10]. In his work, these images are downloaded directly from the Internet based on keywords provided by the user. A major limitation of this approach is the requirement of the reference color image to have a similar viewing angle to the target grayscale image. Therefore, their method can only meet the colorization requirements of special scenes such as places of interest and is not suitable for colorizing images of more general scenes. Chia *et al.* [7] developed a belief propagation-based method which also exploits suitable reference images from the collected Internet images for colorization. However,

S.Wan is with School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China.

Y.Xia is with School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, China.

L.Qi is with School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China.

Y. Yang is with Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

M. Atiquzzaman is with School of Computer Science, University of Oklahoma, Norman, OK,USA.

Corresponding author: Shaohua Wan(shaohua.wan@ieee.org).

their method requires the user to provide manual segmentation cues for all major foreground objects and semantic text labels to search for similar reference images on the Internet.

Several authors [16], [17], [18], [19] have been inspired by the gray-to-color mapping operation when building large-scale image data, successfully fulfilling the colorization through machine learning techniques. But several approaches ask the user to complement the semantics of the picture scene during the training process, which not only improves the computational complexity, but also makes the coloring result greatly affected by the accuracy of semantic description and segmentation. The colorization method using deep learning obtains a model by training a broad range of images to find all kinds of possibilities that exist in the actual situation [16], [18], [19]. Cheng *et al.* [16] have developed a neural network-based colorization method, in which a series of features are used to train a colorization model to estimate the chromaticity value of every pixel. Since this colorization method relies heavily on high-performance segmentation models to segment images, it is limited to predefined segmentation categories and tends to fail when being used to colorize an image containing unknown segmentation classes. A Fully Convolutional Network (FCN) proposed by Iizuka *et al.* [18], extracts different level features as initial information for the training process of the colorization model. Furthermore, generating only one colorization model from the training set not only slows the model convergence time, but also produces artifacts due to low convergence accuracy. Such approaches require high computational cost and storage space. Operations in different scenarios need to converge through a large number of iterations to achieve the accuracy requirements. Moreover, most of the current excellent colorization algorithms using deep learning require semantic segmentation before colorization. The automatic extraction of such advanced feature descriptors is quite hard, which also affects their applicability for colorization, and if not impossible, to generate corresponding semantic descriptors for all actually existing items. To address the above problems, the Gabor filter [20], [21] is used to extract part of the feature as input information for every pixel. The Gabor filters work stably even in the condition that the object rotates, scales, and translates, which makes it attractive. Different from the method of using all the pixels in the database to train the colorization model, the proposed method is to train the model for the set of interest points using the large image dataset.

Altogether, this paper presents an innovative approach for colorization using neural network and color propagating optimization. The neural network is trained using learning and the color propagation is obtained by optimizing an objective function which is defined based on color similarity. In our proposed method, we first use SLIC (Simple Linear Iterative Clustering) [22] to segment grayscale images into superpixels, and extract local features from each interest point that is the geometric center of a corresponding grayscale superpixel, and input them to the neural network. Next, the colorization network is trained based on those centers. We selected the SUN attribute dataset [23] from numerous scene-based data sets to evaluate the colorization algorithm discussed in this paper. Given a trained network, our next goal is to extend the colorization from the centers to their neighbors, also known as propagation in the image colorization community. For propagating the indicated color points, we present a cost function to formalize the premise that neighboring pixels should have maximum positive similarity, in terms of intensities and colors. Then, we propose our solution to the optimization problem. Finally, for acquiring the grayscale interest point from the segmented input image, our method uses local features to get their corresponding color values by the trained model. We introduce the image-guided filter [24] as a refining processor to remove potential artifacts, smooth the images.

We trained our colorization model just for the set of interest points instead of for all pixels, resulting in greatly reduced time cost for convergent iteration operation and improved accuracy. Consequently, we can also propagate seed color pixels to generate the complete colorized image. In this method, without any semantic segmentation, we apply texture descriptors to improve the target scene type recognition ability. Furthermore, since our model is trained based on points at locations determined by superpixels, the memory requirement is significantly reduced.

The major contributions of this paper are as follows:

1) The characteristics of superpixels are used to determine the locations of interest points, and to select their appropriate descriptors.
2) A neural network is designed to train the colorization model.
3) A new cost function for the propagation of color points, and a solution to the corresponding optimization problem has been proposed.

The rest of the paper is organized as follows. Section II presents the framework of our proposed algorithm and gives our proposed efficient implementation. The experimental results are presented in Section III. Finally, Section IV concludes our paper.

II. Colorization

Our goal is to automatically propagate the indicated color of interest points obtained from the trained colorization model to achieve a fully colorized image. A collection of interest points is determined using the SLIC image segmentation method. Given a collection of interest points from all training images, the colorization method proposed in this paper uses a feature extraction method to describe the characteristics of the input target grayscale image at the first step. Secondly, the trained model is used to colorize the interest points of the target grayscale image. We present a cost function that is used to propagate the pixel values from colorized interest points to the whole image region. At last image smoothing process is conducted by the guided image filter for artifacts removal.

Among the steps in our proposed method, except those in the propagation part, including the selection of interest points, the extraction of descriptors and the training of colorization

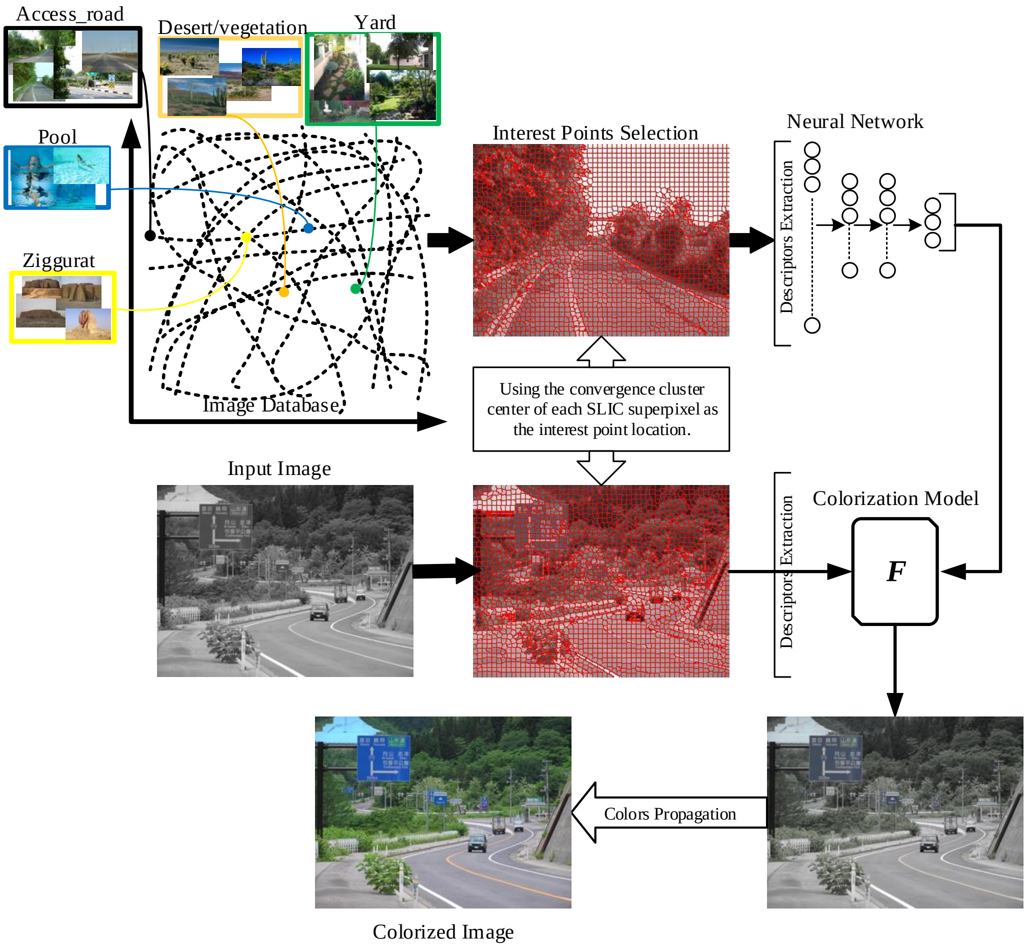


Fig. 1. The proposed colorization method flow chart: we simultaneously begin by selecting interest points from each SLIC superpixel and by extracting features from each interest point location for the whole image dataset to train the proposed colorization model. Then, the target image is segmented by SLIC to find interest points' locations. Finally, we colorize these interest points using the trained colorization network and propagate colors to the whole image using our colors propagation method.

neural network, all the operations are done in the RGB color space. The correlation of RGB values ensures that the shading results are the appropriate result. The colorized result can be proper and right by keeping the correlation of RGB values, which is demonstrated in our experiments. However, during color propagation, the YUV color space is used [2], where Y denote intensity, while U and V encoding the color.

The design and realization for image colorization are illustrated as a structure chart in Figure 1. According to the chart, our approach is made up of 5 important parts: (a) interest points selection, (b) descriptors extraction, (c) model training, (d) interest points colorizing and propagation and (e) refinement of colorized image. The details of each part are covered in the following paragraphs.

A. *Interest Point Selection*

The superpixels algorithm performs the pixel clustering operation to make them in an atomic level region. This procedure not only has perceptual meaning but can also be used to replace the rigid structure of the pixel grid [25]. By eliminating image redundancy, it is possible to filter out better original images for feature acquisition and reduce the difficulty of subsequent work [26]. SLIC superpixels are fast to compute, produce high-quality segmentation, and simple to use. It can adhere well to image boundaries with low computational complexity. In this paper, we select the convergence cluster center of each SLIC superpixel as the interest point location. SLIC converts images from RGB color space to Lab color space, the (L, a, b) color values and (x, y) coordinates corresponding to each pixel form a 5-dimensional vector $V = [L, a, b, x, y]$. The algorithm initializes T superpixel cluster centers $C_t = [l_t, a_t, b_t, x_t, y_t]$ with $t \in \{1, ..., T\}$ at regular grid steps each of size S. For an image with N pixels, the number of superpixels is $T = N/S^2$, the approximate area of each superpixel is $S \times S$. Then, the nearest pixels from a $2S \times 2S$ square neighborhood around each superpixel cluster center are determined based on the distance measure Ds:

$$Ds = d_{lab} + \frac{m}{S} d_{xy}$$

where

$$d_{lab} = \sqrt{(L_t - L_i)^2 + (a_t - a_i)^2 + (b_t - b_i)^2},$$
$$d_{xy} = \sqrt{(X_t - X_i)^2 + (Y_t - Y_i)^2} \quad (1)$$

m is used to adjust the weight of d_{xy}. Then the average vector values of all the pixels are calculated for each superpixel, T cluster centers are updated, and the iteration is repeated until convergence. In the particular SLIC superpixel segmentation scheme of grayscale images, the attribute of each superpixel is defined by two parameters, region size, and regularizer [26]. The region size determines the size of each atomic region, and the regularizer determines the spatial proximity and the pixel similarity. In our case, the region size of superpixel S^2 is set to 6×6 and the regularizer m is set to 10.

B. Descriptor Computation

It is a morbid problem to rely solely on the gray value of one pixel to infer its initial RGB value. Currently, without prior knowledge of the scenario, there is no general solver for such problems. Therefore, more details of the pixel's characteristics are necessary to enhance its information richness and describe its local neighborhoods in a more robust way. In our algorithm, local structures, textures, and context information are involved. The first step in the processing of RGB color images is the conversion to obtain a grayscale image. The formula for converting color images into grayscale images is $Y = 0.299R + 0.587G + 0.114B$. This operation ensures that the input and output images for training and testing steps are at the same luminance. Multiple options for feature descriptors are available, including SIFT, Gabor, etc. A study showed that patch feature and DAISY descriptors [27] make sense on the colorized images by Cheng *et al*. [16]. Therefore, in our method, the length and width of the patch feature for each pixel combination are 7, the DAISY feature [27] and the Gabor feature [28], [20], [21] are combined for each pixel. All the descriptors turn into a 121×1 dimensional feature for every interest point of the original and the reference image.

a) ***Patch features:*** The algorithm in this paper only needs to obtain the gray value of one pixel in the 7×7 neighborhood, and the 49-dimensional feature vector $D_P(\mathbf{r})$ of the pixel r is determined. This is the most intuitive representation of the structure around the pixel.

b) ***DAISY features:*** We characterize the dense features by the DAISY feature vector $D_D(\mathbf{r})$ of the pixel r. The $D_D(\mathbf{r})$ in this paper is 32-dimensional, that is eight orientations in four positions. Considering the dense matching process [27], the DAISY features can significantly improve the coloring effect in complex scenes dense matching. But as in [16], DAISY is ineffective for sparse-texton images.

c) ***Texture features:*** The Gabor feature $D_T(\mathbf{r})$ becomes the ultimate feature involved in this algorithm. The Gabor feature works through a Gabor filter to obtain local features come by any pixel. Gabor filters have important properties that are constant for image deformation. In this study, the Gabor feature vector is a 40-dimensional vector, which is constructed based on 40 Gabor filters [28], [20], [21] in five scales and eight orientations.

The last descriptor $D(\mathbf{r})$ of pixel $\mathbf{r}$ is the integration of the patch, DAISY and Gabor descriptors for the original and the corresponding grayscale image. $D(\mathbf{r})$ is defined by:

$$D(\mathbf{r}) = D_P(\mathbf{r}) \cup D_D(\mathbf{r}) \cup D_T(\mathbf{r}) \tag{2}$$

where $\cup$ denotes the concatenation operator.

The experimental results in [29] are used to illustrate the sensitivity of the selected descriptors to the color diversity and texture complexity of the images. We draw a conclusion that the more colors and the more complexity of texture, the longer convergence time and the lower final convergence accuracy.

C. Network Design

Nowadays, deep learning has achieved great success in the field of computer vision. It can be expressed as a corresponding color image from a monochrome one when a deep learning method is used to solve the colorization problem. The goal of the colorization approach is to estimate RGB values for all the pixels in the monochrome image. The colorization model in this paper is used to learn a continuous function of gray-to-color mapping, which is represented as a correspondence between features obtained in every pixel from the initial monochrome image and color values in the relevant output image.

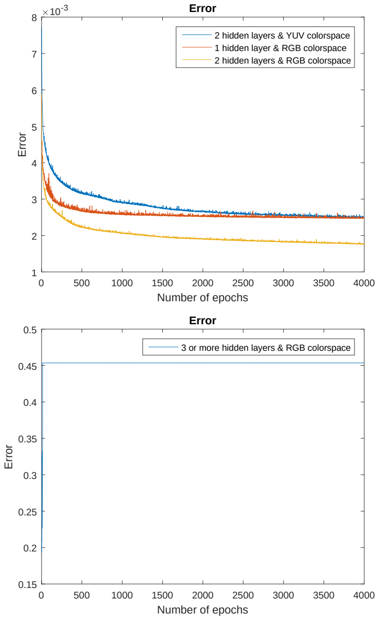


Fig. 2. The convergence curve of the interest points in the training step for the proposed network in a combination of different color spaces and hidden layers.

d) ***Network Structure:*** We built a concise but efficient network, which is 1 input layer and 2 hidden layers and 1 output layer, to serve in the proposed colorization approach. To find the properly hidden layers, Figure 2 indicates the MSE curve of the interest points when training for the proposed network in a combination of different color spaces and hidden layers. Obviously, it can be concluded that a network built by two hidden layers in RGB space is optimal. Meanwhile, it consists of 121 input layer neurons, 60 neurons in each hidden layer and 3 output layer neurons.

e) ***Internal Settings:*** In order for all neurons to be interconnected between layers, meanwhile, each connection is weighted, our colorization network uses a fully connected network. For the calculation of the optimal weight values, the classical error back-propagation algorithm is used to enable errors propagating to the hidden layer. We choose ReLU (Rectified Linear Units) [30] as the activation function to maintain relative intensity information through the layers. Due to the huge training dataset, it becomes infeasible to load all the data at once. Therefore, mini-batches learning is used in this paper to train the colorization model. Mini-batches learning is a variant of the stochastic gradient descent method, which uses a small number of training samples to calculate the gradient and update the model parameters. Batch size refers to the number of training samples in each mini-batch. For the same neural network structure, adjusting the batch size to an appropriate value can achieve the best optimization in time and the final convergence precision. In our case, the batch size is set to 400. Meanwhile, the sigmoid activation function is used in the output layer. Figure 3 shows the architecture of the proposed network.

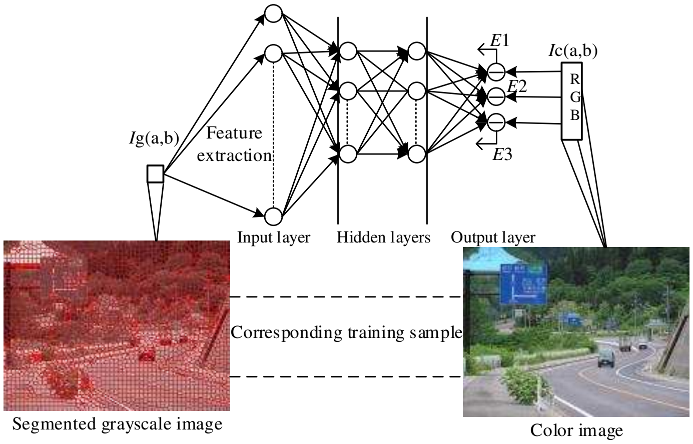


Fig. 3. The architecture of our neural network.

D. *Colors Propagation*

In Section 3.3, we get the colorization model from training on all interest points. Then, we can determine the indicated colors (colorized interest points) of a test grayscale image via the colorization model,

$$G = F(D) \tag{3}$$

where the feature vector is represented by $D = [D_P \cup D_D \cup D_T] \in \mathbb{R}^{N\times 121}$ of each interest point location in target image, and N the number of superpixels. For our colorization method, the Quarter Video Graphics Array (QVGA) resolution is used to resize all images. F denotes the trained colorization function. $G = [g_1, g_2, ..., g_N] \in \mathbb{R}^{N\times 3}$ denotes the corresponding RGB values of all interest points.

From our trained network, we obtain the colorization of all interest points or the centers of superpixels in one test grayscale image. The next goal for this study is to solve the colorization problem of the remaining grayscale pixels. In this section, we come up with an automatic color propagation algorithm to output a fully colorized image. Our algorithm is based on a simple assumption that each adjacent pixel should have the maximum positive similarity of intensities and colors. That is to say, they have the same changing attributes relative to their expectations. The YUV color space is introduced in this step, where Y denotes the monochromatic luminance channel, in short, intensity. U and V are the chrominance channels[31], [2].

We wish to impose the constraint that ensures the maximum positive similarity between the intensities and colors of two near pixels $\mathbf{r}$, $\mathbf{s}$. Thus, we wish to maximize the cross correlation coefficient between the intensities and colors of two neighboring pixels $\mathbf{r}$, $\mathbf{s}$:

$$J(C) = \sum_{\mathbf{r}} \left(\frac{E[(Y(\mathbf{r}) - E[Y(\mathbf{s})])(C(\mathbf{r}) - E[\omega_{\mathbf{rs}} C(\mathbf{s})])]}{\sigma_{Y(\mathbf{r})}\sigma_{C(\mathbf{r})}} \right) \tag{4}$$

where

$$\begin{cases} C = \{U, V\} \\ \mathbf{s} \in N(\mathbf{r}) \end{cases} \tag{5}$$

and C represents the chrominance channel U or V. $\omega_{\mathbf{rs}}$ is a weighting function that sums to one, E calculates the expectation and σ calculates the variance. The weighting function $\omega_{\mathbf{rs}}$ is larger when $Y(\mathbf{r})$ is close to $Y(\mathbf{s})$, and smaller when the two intensities are different. The constraint $\mathbf{s} \in N(\mathbf{r})$ denotes that $\mathbf{r}$ and $\mathbf{s}$ are neighboring pixels. As to the weighting function, we have several possible choices to use. For example, in [2], two weighting functions that are based on the squared difference and normalized correlation between the two intensities are used:

$$\omega_{\mathbf{rs}} \propto e^{-(Y(\mathbf{r})-Y(\mathbf{s}))^2/2\sigma_{\mathbf{r}}^2} \tag{6}$$

$$\omega_{\mathbf{rs}} \propto 1 + \frac{1}{\sigma_{\mathbf{r}}^2}(Y(\mathbf{r}) - \mu_{\mathbf{r}})(Y(\mathbf{s}) - \mu_{\mathbf{r}}) \tag{7}$$

where $\mu_{\mathbf{r}}$ and $\sigma_{\mathbf{r}}$ are the mean and variance of the intensities in a window around $\mathbf{r}$.

In order to use the correlation expression of neighboring regions, we present a new weighting function that is derived by assuming a local linear relation of both luminance and colors [12], [14]:

$$\omega_{\mathbf{rs}} \propto \frac{\sigma_{\mathbf{r}}}{\sigma_{\mathbf{s}}} + \frac{1}{Y(\mathbf{s})}\left(\mu_{\mathbf{r}} - \frac{\sigma_{\mathbf{r}}}{\sigma_{\mathbf{s}}}\mu_{\mathbf{s}}\right) \tag{8}$$

Now given a set of locations $\mathbf{r}_i$ where the colors are specified by the trained colorization model, we maximize $J(C)$ subject to the constraint as specified by Equation (8). Since the cost function is quadratic and the constraint is linear, substituting Equation (8) into Equation (4), then our optimization problem yields a sparse system of linear equations. The proposed algorithm is closely related to space transformation algorithms based on two-dimensional principal component analysis (2DPCA) [32], which is equivalent to finding the optimal projection axis X_{opt} that is the unitary vector of matrix G_t, where G_t is an $n \times n$ nonnegative definite matrix whose elements are the pairwise affinities between pixels (i.e., the $\mathbf{r}$, $\mathbf{s}$ entry of the matrix is $\omega_{\mathbf{rs}}$ from Equation (8)). Indeed, the optimal projection axis is the unitary vector that maximizes $J(X)$, i.e., the eigenvector of G_t corresponding to the largest eigenvalue [33]. By direct inspection, the quadratic form maximized by 2DPCA is exactly our cost function J, that is

$$X^T G_t X = \max_X J(X) \tag{9}$$

where X is the eigenvector that corresponds to the largest eigenvalue of G_t. Thus, the proposed method maximizes the same cost function but under a different constraint:

$$C^T \omega_{\mathbf{rs}} C = \max_C J(C) \tag{10}$$

that is

$$C^T \omega_{\mathbf{rs}} C = \max_C \sum_{\mathbf{r}} \left(\frac{E[(Y(\mathbf{r}) - EY(\mathbf{s}))(C(\mathbf{r}) - E\omega_{\mathbf{rs}} C(\mathbf{s}))]}{\sigma_{Y(\mathbf{r})} \sigma_{C(\mathbf{r})}} \right) \tag{11}$$

where

$$\begin{cases} C = (U, V) \\ \mathbf{s} \in N(\mathbf{r}) \\ \omega_{\mathbf{rs}} \propto \frac{\sigma_{\mathbf{r}}}{\sigma_{\mathbf{s}}} + \frac{1}{Y(\mathbf{s})} \left(\mu_{\mathbf{r}} - \frac{\sigma_{\mathbf{r}}}{\sigma_{\mathbf{s}}} \mu_{\mathbf{s}} \right) \end{cases} \tag{12}$$

E. *Color Refinement*

The color image produced by the colorization method proposed in Section 3.4 has some slight artifacts, and the guided image filter can solve this problem [24]. Like the bilateral filter [34], the guided image filter can also be used as an edge-preserving smoothing operator, but with better performance near the edges. It generates a filtered image by computing the information contained in the guidance image. The selection of guidance images can take into account the input image I itself. In our case, the guidance image is set to the input grayscale image, and two important parameters of guided image filter have been given, which is $nhoodSize = 15 \times 15$, $smoothvalue = 65$, where *nhoodSize* is size of the window and *smoothvalue* is the parameter for the filtering regularization term. Figure 4 shows the effect comparison before and after color refinement.

The color seed points that the proposed color propagation method needs are generated by the trained colorization model and every input of this model in training is a pixel feature. The blue colorizing results without color refinement are mostly caused by the descriptors generated within the objects area containing the sky, ocean, etc. The characteristics of these objects include large connected regions, the monotonous color of blue, low-texture and many samples in the dataset. Despite expanding the search window, this problem is unsolvable. So blue tint is generated around the object when it has regions of low-texture. For this type of regions, if their neighborhood colors are the correct targets, the blue tint can be filtered by the guided filter, which is the reason why the color refinement method is used and the filtering window is more than twice as large as the descriptor search window.

III. Experimental Results

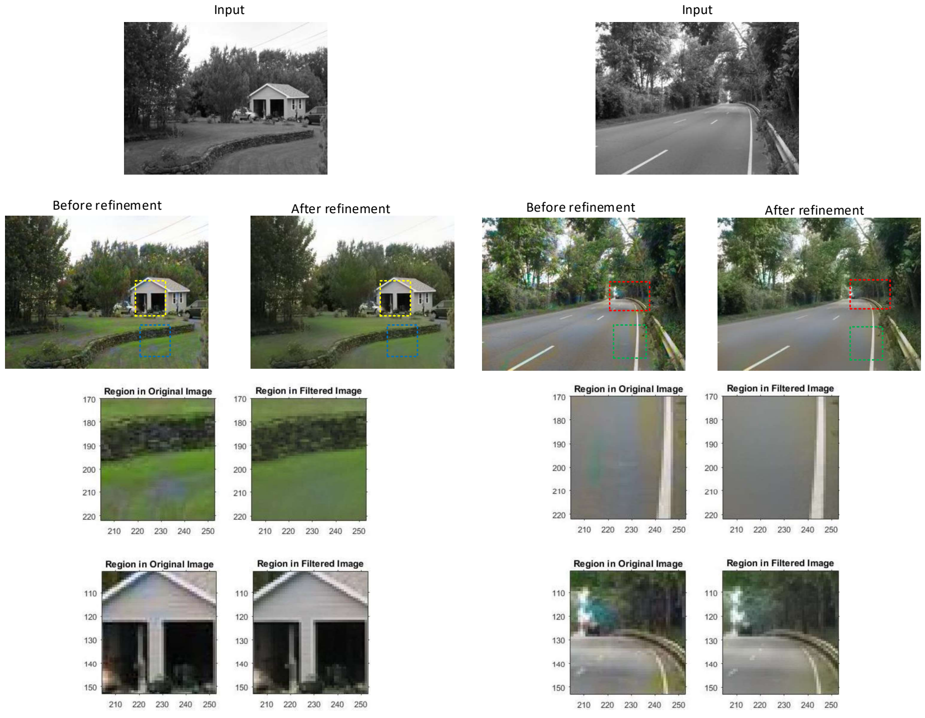
We analysis their experimental results by comparing it with the verification subset of the SUN attribute database [23]. in this part. Figure 5 shows part of them. Overall, the SUN database covers over 700 different scenes and contains over 14,000 images. This makes it a popular database for advanced scene understanding and fine-grained scene recognition. Our algorithm trains 20 images for each scene category and establishes a corresponding colorization network model. All images, whether training or target, are normalized to QVGA resolution. The colorization methods involved are detailed in Sections 3.3 and 3.4, so we won't cover those here again. For the performance verification, we select about 400 random samples per mini-batch in every data set as an input into the algorithm, which costs about 41.5 seconds. The time for the trained model to colorize the QVGA resolution grayscale image is approximately 0.46 seconds. We evaluated colorized image quality using two widely acknowledged objective image quality metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measurement (SSIM)[35].

A. *Colorization Results*

Figure 6 compares the performance of the proposed algorithm with other excellent algorithms by evaluating their experimental results. Algorithms for comparison include the dark coloring method conducted by Cheng *et al.* [16] as well as the CNN based colorization method conducted by Zhang *et al.* [19]. We select the input image from the validation set instead of the training set. Obviously, the method of [16] produces unexpected by-products during the grayscale image colorization process. An analysis found the problem stems from the fact that the output of FCN serves as the semantic segmentation mask in all training images [36]. Therefore, the colorization quality of the method [16] is dominated by the precision of semantic segmentation and target type derived from FCN. The method of [19] generates colorization results with the good visual effect, but the color of their results appears to be too colorful. The methods in this paper are able to achieve colorization results that are closer to the actual color than they are.

In order to verify the sensitivity of the proposed method to colors and textures, we have studied the convergence performance of different categories in the training step. In [29], it is the convergence curve of different scene samples. Indeed each scenario type has different convergence time and accuracy. Dealing with the input image which contains complex color and texture features, current methods cause long convergence time and high mean square error (MSE). Obviously, after data analysis, the curves calculated from the small batch gradient MBGD can be divided into 3 parts based on the time periods. The first part is the period [0, 150], at which point the curve drops steeply. In period 2, [150, 1000], the curve is also significantly reduced, and when the range exceeds 1000 periods, the change slows down and becomes gentle.

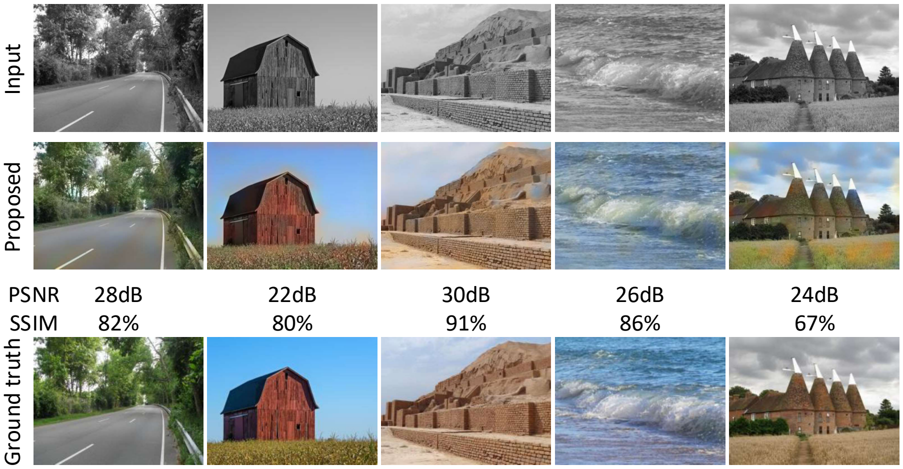
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.2976573, IEEE Transactions on Multimedia



(a) Access road

(b) Yard

Fig. 4. The effect comparison before and after color refinement.



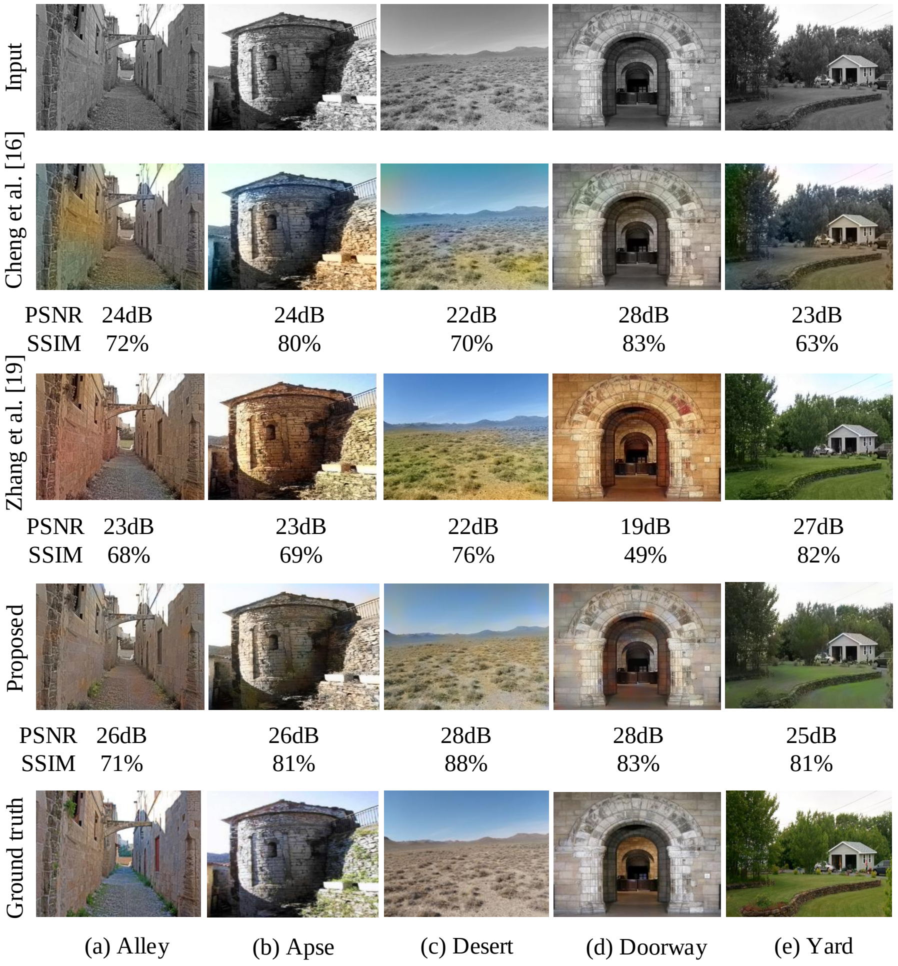
(a) Access road (b) Barn (c) Ziggurat (d) Wave (e) Oast house

Fig. 5. In different scenarios, the colorized results of the proposed colorization method is compared with ground truth. The corresponding PSNR and SSIM values [35] are shown under the second row.

Figure 7 presents, in some extreme cases, a few of the unexpected coloring results produced by our proposed method, as well as comparisons with other methods [16], [19]. It also generates many different color values surrounding similar textons.

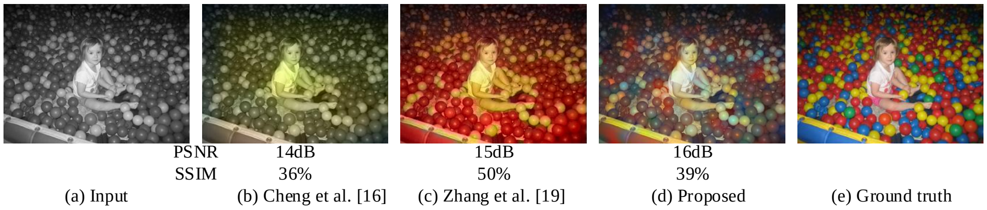
To conclude, it can be seen from the colorization results that even in the case where all the colorization methods underperform, the proposed method also exhibits better performance than other algorithms. The proposed method trained the texture descriptor and built the corresponding colorization model, makes it work reasonably and stable inside the target for extra fine results.

In Figure 8, different rendering results for lighting scenarios with varying illumination are compared. The images of this case include a scene taken at the University of Alberta at different times, see Figure 8. Apparently, the results prove that our method achieves better colorization performance than the existing excellent methods [16], [19]. We can see that the method of [16] creates unnatural results. While the method [19] generates colorization results with good visual effects, it colorizes the road as a lawn, which is incorrect. Our method works well using these images which exhibit varying



(a) Alley (b) Apse (c) Desert (d) Doorway (e) Yard

Fig. 6. We compare our colorized results with the other colorization methods [16], [19] for different scenarios. The corresponding PSNR and SSIM values are shown under results of every method.



(a) Input (b) Cheng et al. [16] (c) Zhang et al. [19] (d) Proposed (e) Ground truth

Fig. 7. We compare ours with the other colorization methods [16], [19] for the extreme case.

illumination of the same scene.

B. Colorizing Performance

Figure 9 shows the mean of PSNR and SSIM in different epochs, that is colored images of different scenes generated by the proposed colorization method in the training step. The test is performed by using the Leave One Out Cross Validation, and 5 test images per scene class from 21 different scenes in the SUN attribute database are selected. In Figure 9, the abscissa is the scene classes, the ordinate is the PSNR and SSIM value. The number of epochs represented by each color curve is listed in the lower right corner of these figures. From the two figures, we can see that the data result of PSNR and SSIM will increase with the number of epochs, which means that the higher number of epochs, the better quality of the color image after colorized.

Figure 10 shows the PSNR and SSIM mean values of colorized images using the proposed and the state-of-the-art colorization method. The scene classes and verification test methods are the same as Figure 9. In Figure 10, the abscissa is scene classes, and the ordinate is the PSNR and SSIM value. The different colorization methods represented by the color curves are listed in the lower right corner of these figures. The contrast algorithm selected is composed of the example-based colorization methods proposed in the literature [6], [8], [9], [11] and deep learning based approaches [16], [19]. Among

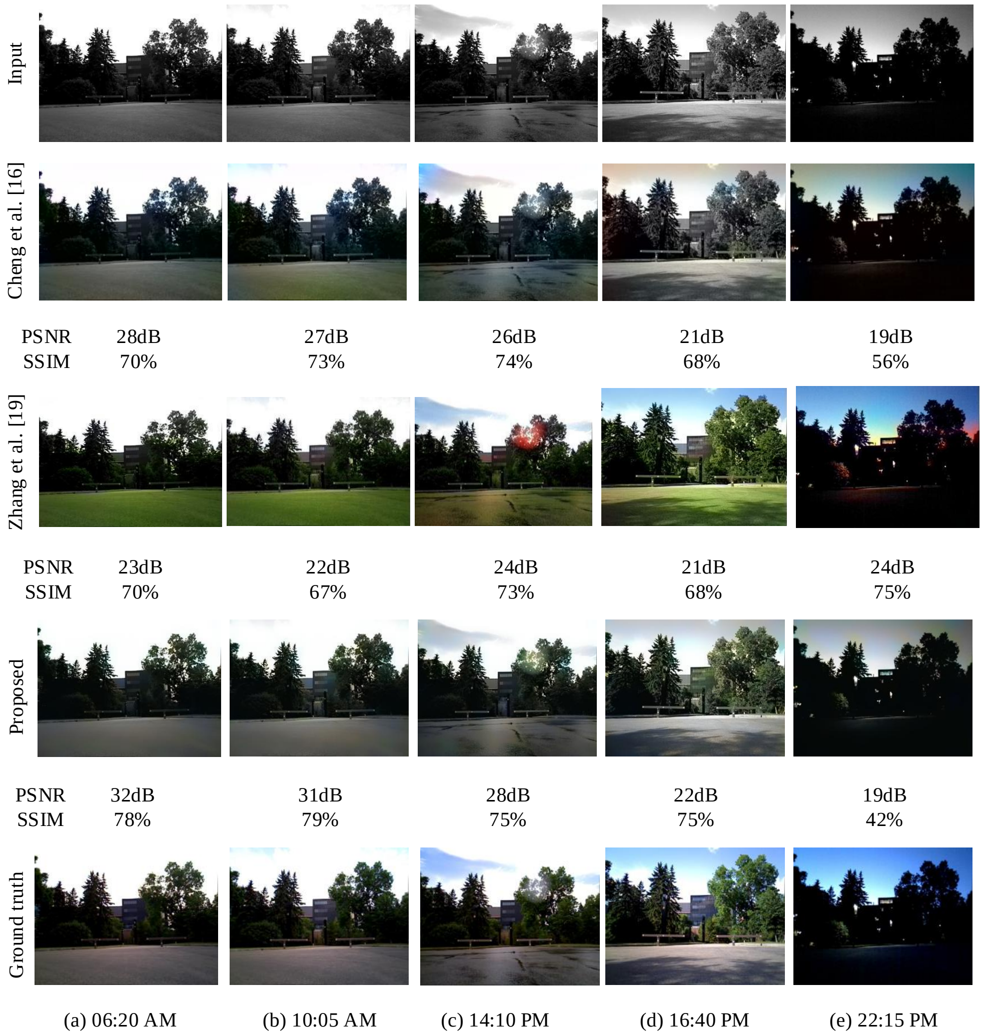


Fig. 8. We compare our colorized results with the other colorization methods [16], [19] for different periods in University of Alberta. The corresponding PSNR and SSIM values are shown under results of every method.

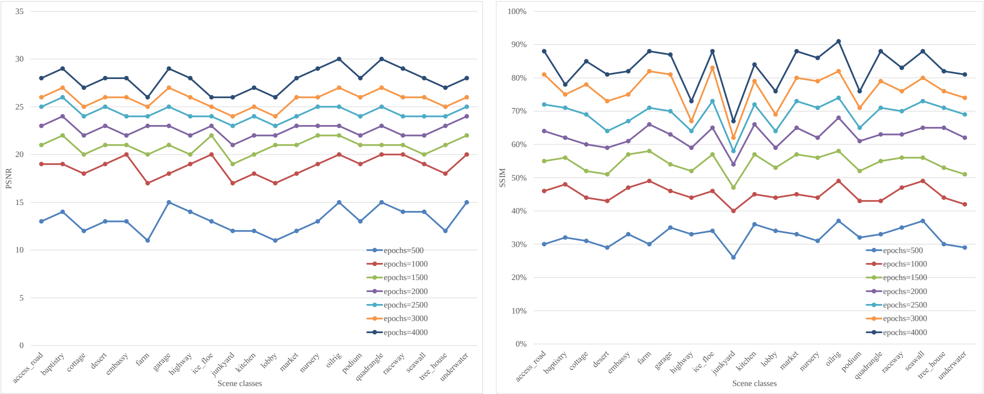


Fig. 9. The PSNR and SSIM curves of the colorized image under different scenes.

them, [16], [19] use the public colorization model provided by them to test. [6], [8], [9], [11] strictly follows the parameter configuration in their literatures. The selection principle of the reference color image is to visually select one or more color images, which has the same overall structural distribution as the target grayscale image. It can be seen from Figure 10 that the proposed algorithm outperforms other comparison algorithms in evaluating the PSNR of the image quality after colorized, and SSIM which measures the similarity of two images.

We calculated the mean of SSIM and PSNR to evaluate

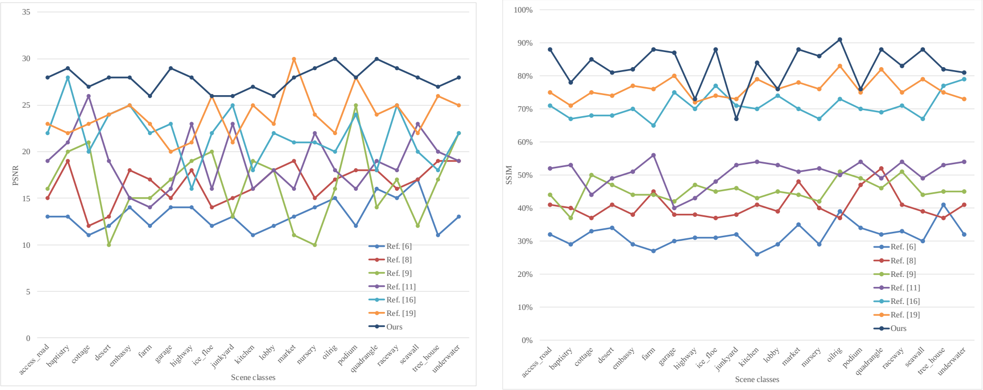


Fig. 10. The PSNR and SSIM curves of the colorized image under different methods.

the colorizing performance of the state-of-the-art colorization methods [16], [19] and the proposed method. 400 images are randomly chosen from the SUN dataset, which includes a total of 131067 images [37]. We show the results in Table 1. We can see that our approach has a median of 26dB PSNR and 83% SSIM, which is close to the ground truth. This strongly indicates that our method is able to generate realistic colorizations.

As seen from Tables II,III, the proposed colorizing algorithm outperforms other control algorithms in terms of both reflecting PSNR of the quality of the colored images and measuring SSIM of the similarity of the two images. The images colorized by the proposed algorithm are similar to the real scene images in brightness, contrast ratio and object structure attribute. Among them, the example-based colorization methods in [8], [6], [9], [11] transfer second-order statistics based on the image or image area block as the basic computing unit in , the example-based colorization methods, it will result in the insufficient processing of the texture edges and other details in the colorizing process. In addition, since the color images used in the aforementioned methods are subjectively selected with a limited amount, the abundance evaluation value of the coloring results in [16] is higher than that of the above methods. This is because the pixels are used for training and testing. It is found that the evaluation indexes of [16] in some scenes, such as ”lobby” and ”underwater”, are closer to those of the proposed algorithm and [19]. For explanation, [19] can accurately segment and distinguish these scenes when FCN is applied to semantic segmentation. Nevertheless, the scenes in most of these images have a simple structure and uncomplicated background texture. When the semantic labels of the objects contained in the images are not clear, the colorizing ability of this method is significantly weakened. According to intuitive observation, [19] can achieve the desired coloring effect; but it excessively seeks for the color richness of the colored images, resulting in over brightness of colored images. Moreover, some large fine-grained areas with a large area, such as highways and squares, are identified as lawns for colorizing. Similar to [19], [38], [39], [18] has developed similar system, which leverages large-scale data and CNNs. The methods differ in their CNN architectures and loss functions. The networks can learn to combine low and high-level cues to perform colorization, and have been shown to produce satisfactory results.

C. *Running Time*

The proposed method is able to process images of several resolutions at a high speed. Table IV shows the mean of 50 running time computations on 400 images of each resolution to get a reliable value. It can be seen that our method is faster than the state-of-the-art methods, and our computation time increases nearly linearly with the image resolution. Therefore, the proposed method is suitable for near real-time applications.

D. *Complexity*

It can be seen from Figure 1 that the proposed method belongs to the cascade operation, and the results generated between the steps can be regarded as independent data. Therefore, we can analyze the complexity of each step included in the algorithm when using it to colorize an image with N pixels, and then use the summation rule to calculate the complexity of the proposed algorithm.

Achanta *et al*. have demonstrated the $O(N)$ complexity of SLIC in their paper (refer to [26] for details): SLIC only calculates the distance from each cluster center to the pixels in the $2S \times 2S$ region. Limiting the size of the search area significantly reduces the number of distance calculations, which can greatly speed up the algorithm and control the algorithm to linear complexity. This method not only reduces the distance calculation but also makes the complexity of the SLIC independent of the number of superpixels.

In the descriptor extraction step, the dimension of each descriptor is deterministic and can be regarded as a constant 121 when computing complexity. So the descriptor extraction executed $121N$ times, in which the complexity of this step is $O(N)$. In this paper, using a trained model to colorize the center point of each superpixel can be regarded as an assignment operation, and the complexity is $O(1)$. So the complexity of the colorizing step is $O(N)$. The color propagation algorithm for each pixel is implemented by the defined affinity function which is a positive semidefinite symmetric matrix,

TABLE I
MEAN OF SSIM AND PSNR OF OUR METHOD AND THE STATE-OF-THE-ART COLORIZATION METHOD.

	PSNR	SSIM
Cheng et al.[16]	24.11dB	72%
Zhang et al.[19]	23.23dB	76%
Ours	**26.18dB**	**83%**

TABLE II
THE PSNR VALUE OF THE COLORIZED IMAGE UNDER DIFFERENT METHODS

Scene category	[8]	[6]	[9]	[11]	[16]	[18]	[19]	Proposed method
access road	13.11	15.01	16.41	19.04	22.43	24.26	23.13	**27.78**
baptistry	13.48	19.29	20.14	21.15	28.09	25.02	22.21	**29.18**
cottage	10.85	12.18	21.43	26.03	20.24	20.34	22.91	**26.70**
desert	12.23	13.31	9.89	19.45	24.36	23.37	24.17	**28.11**
embassy	14.09	18.47	15.25	15.37	25.16	24.41	25.34	**28.08**
farm	12.36	17.61	15.41	14.51	22.80	23.57	23.73	**26.31**
garage	14.09	15.32	17.32	16.23	23.15	25.45	20.18	**29.12**
highway	14.25	18.26	19.43	23.39	16.07	24.61	21.16	**28.42**
icefloe	12.31	14.27	19.86	16.19	22.05	23.71	25.78	**26.37**
junkyard	13.41	15.16	13.11	23.09	25.17	22.58	20.89	**25.81**
kitchen	11.21	16.19	19.03	16.07	18.16	26.08	25.49	**27.02**
lobby	12.05	17.79	18.22	18.31	21.74	25.62	23.35	**26.18**
market	13.07	19.11	11.43	16.24	20.87	26.55	29.85	**28.31**
nursery	14.12	15.16	10.14	22.27	21.28	26.72	24.23	**29.15**
oilrig	15.31	17.22	16.27	18.19	20.25	20.49	22.38	**30.01**
podium	12.24	18.23	25.27	16.28	24.17	27.87	28.01	**28.28**
quadrangle	16.02	17.93	14.13	19.12	18.16	26.91	24.10	**30.17**
raceway	15.21	16.11	17.17	18.18	25.12	25.15	25.27	**29.16**
seawall	17.08	17.17	12.43	23.21	20.19	22.39	22.13	**28.05**
treehouse	11.34	19.25	17.24	20.17	18.18	25.48	26.01	**27.21**
underwater	13.15	18.91	22.12	19.07	22.13	24.75	24.82	**28.11**

TABLE III
THE SSIM VALUE OF THE COLORIZED IMAGE UNDER DIFFERENT METHODS.

Scene category	[8]	[6]	[9]	[11]	[16]	[19]	Proposed method
access road	32%	41%	44%	52%	71%	75%	**88%**
baptistry	29%	40%	37%	53%	67%	71%	**78%**
cottage	33%	37%	50%	44%	68%	75%	**85%**
desert	34%	41%	47%	49%	68%	74%	**81%**
embassy	29%	38%	44%	51%	70%	77%	**82%**
farm	27%	45%	44%	56%	65%	76%	**88%**
garage	30%	38%	42%	40%	75%	80%	**87%**
highway	31%	38%	47%	43%	70%	72%	**73%**
icefloe	31%	37%	45%	48%	77%	74%	**88%**
junkyard	32%	38%	46%	53%	71%	73%	**67%**
kitchen	26%	41%	43%	54%	70%	79%	**84%**
lobby	29%	39%	45%	53%	74%	76%	**76%**
market	35%	48%	44%	51%	70%	78%	**88%**
nursery	29%	40%	42%	52%	67%	76%	**86%**
oilrig	39%	37%	51%	50%	73%	83%	**91%**
podium	34%	47%	49%	54%	70%	75%	**76%**
quadrangle	32%	52%	46%	49%	69%	82%	**88%**
raceway	33%	41%	51%	54%	71%	75%	**83%**
seawall	30%	39%	44%	49%	67%	79%	**88%**
treehouse	41%	37%	45%	53%	77%	75%	**82%**
underwater	32%	41%	45%	54%	79%	73%	**81%**

TABLE IV
MEAN OF 50 RUNNING TIME (SECONDS) COMPUTATIONS ON 400 IMAGES OF DIFFERENT RESOLUTIONS, AND COMPARISON TO THE STATE-OF-THE-ART METHODS.

	QVGA	VGA	SVGA	XGA
Cheng et al.[16]	6.18	19.72	35.24	59.72
Zhang et al.[19]	0.62	2.46	3.42	5.92
Ours	**0.46**	**1.71**	**2.63**	**4.18**

so the computational complexity of the colors propagation algorithm is $O(N)$. We use the guided image filter for color refinement, which is non-iterative, does not exhibit gradient inversion, and the computational complexity is independent of the filter radius. So, the complexity of the guided image filter is $O(N)$. To sum up, the time complexity and spatial complexity of grayscale image colorization using the proposed method are both $O(N)$.

IV. Conclusion

In this article, we present an innovative approach to colorize grayscale photographs free from any manual intervention. Our approach first selects interest points based on SLIC superpixels and then trains a colorization model for training set using a neural network. When colorizing input grayscale images, we first colorize the interest points by the trained model and propagate to fully image using our proposed optimization method. Grayscale descriptors contribute indispensable information to help build a suitable neural network for superior colorization performance. Furthermore, the chrominance noise is reduced by a guided image filter to ensure a consistently high-quality colorization result. A large number of experimental results prove that the method in our paper is superior to other current excellent algorithms.

References

[1] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 351–354.

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 689–694.

[3] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 2007, pp. 309–320.

[4] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1214–1220.

[5] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE transactions on image processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[6] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," in *European conference on computer vision*. Springer, 2008, pp. 126–139.

[7] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 156.

[8] 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*. ACM, 2012, pp. 369–378.

[9] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example." in *Rendering Techniques*. Citeseer, 2005, pp. 201–210.

[10] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 152.

[11] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *ACM transactions on graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 277–280.

[12] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 327–340.

[13] S. Liu and X. Zhang, "Automatic grayscale image colorization using histogram regression," *Pattern Recognition Letters*, vol. 33, no. 13, pp. 1673–1681, 2012.

[14] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.

[15] Y. Xiang, B. Zou, and H. Li, "Selective color transfer with multi-source images," *Pattern Recognition Letters*, vol. 30, no. 7, pp. 682–689, 2009.

[16] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 415–423.

[17] A. Deshpande, J. Rock, and D. Forsyth, "Learning large-scale automatic image colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 567–575.

[18] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 110, 2016.

[19] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[20] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7905–7916, 2015.

[21] J.-K. Kamarainen, V. Kyrki, and H. Kalviainen, "Invariance properties of Gabor filter-based features-overview and applications," *IEEE Transactions on image processing*, vol. 15, no. 5, pp. 1088–1099, 2006.

[22] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua, "A fully automated approach to segmentation of irregularly shaped cellular structures in EM images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2010, pp. 463–471.

[23] G. Patterson, C. Xu, H. Su, and J. Hays, "The SUN attribute database: Beyond categories for deeper scene understanding," *International Journal of Computer Vision*, vol. 108, no. 1-2, pp. 59–81, 2014.

[24] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[25] 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. 2274–2282, 2012.

[26] ——, "SLIC superpixels," Tech. Rep., 2010.

[27] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[28] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *JOSA A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[29] Y. Xia, S. Qu, and S. Wan, "Scene guided colorization using neural networks," *Neural Computing and Applications*, pp. 1–14, 2018.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[31] J. Keith, *Video demystified*. Newnes, 2005.

[32] J. Yang, D. D. Zhang, A. F. Frangi, and J.-y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2004.

[33] J. Yang and J. Y. Yang, "From image vector to matrix: a straightforward image projection technique-IMPCA vs. PCA," *Pattern Recognition*, vol. 35, no. 9, pp. 1997–1999, 2002.

[34] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 664–672, 2004.

[35] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369.

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[37] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3485–3492.

[38] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 577–593.

[39] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.