Image Colorization using feature matching technique

S.Bhavani(20172104)[1],K.V.Sushena(20172140) Sree[2],Raghu Ram T.V.S(20172070)[3]

***Abstract*— We present a new example-based method to colorize a gray image. As input, the user needs only to supply a reference color image which is semantically similar to the target image. We extract features from these images at the resolution of superpixels, and exploit these features to guide the colorization process. Our use of a superpixel representation speeds up the colorization process. More importantly, it also empowers the colorizations to exhibit a much higher extent of spatial consistency in the colorization as compared to that using independent pixels. We adopt a fast cascade feature matching scheme to automatically find correspondences between superpixels of the reference and target images. Each correspondence is assigned a confidence based on the feature matching costs computed at different steps in the cascade, and high confidence correspondences are used to assign an initial set of chromatic values to the target superpixels. To further enforce the spatial coherence of these initial color assignments, we develop an image space voting framework which draws evidence from neighboring superpixels to identify and to correct invalid color assignments. Experimental results and user study on a broad range of images demonstrate that our method with a fixed set of parameters yields better colorization results as compared to existing methods**

I. INTRODUCTION

The goal of image colorization is to add colors to a gray image such that the colorized image is perceptually meaningful and visually appealing. A key challenge of this problem is that it is under constrained since there are potentially many colors that can be assigned to the gray pixels of an input image (e.g. leaves may be colored in green, yellow and brown). Hence, there is no one correct solution to the colorization problem and human intervention often plays an important role in the colorization process.In this project we tried to use a reference color image to colorize a gray scale image. We have attempted to see the working of several features to see their impact on colorization. Object labels were assigned to images and better results were obtained in colorization.

II. FEATURE EXTRACTION

A. Superpixel Extraction

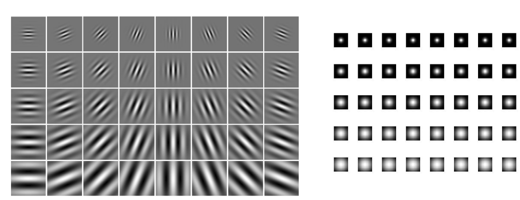
The grayscale and refernce colour image have been segmented into superpixels using the in-built command in MATLAB. The number of pixels within a superpixel has been set to around 40.For each superpixel in input gray image and reference color image, we compute 172dimensional (2+2+40+128) feature vector based on their intensity, standard deviation, gabor features and SURF descriptors. To compute this feature vector for a superpixel, we compute a 172dimensional feature vector at each image pixel and then compute the mean value of all feature vectors that belong to the pixels within a superpixel to represent that superpixel.An advantage of using a superpixel based representation is that it speeds up the colorization method. More importantly, it also affords our method with an ability to maintain stronger spatial coherency in the colorization as compared to that using individual pixels.

B. Gabor feature

We apply Gabor filters to an imagewith eight orientations varying in increments of $\pi/8$ from $0to7\pi/8$, and with five exponential scales $exp(i\pi), i = 0, 1, 2, 3, 4$ to compute a 40-dimensional feature at each pixel.The Gabor feature for the superpixel is then computed as the average Gabor feature of all pixels within the superpixel.

$$
\begin{aligned}
F(u_1, u_2) &= \exp\left(-\frac{(\hat{u_1}^2 + \gamma^2 \hat{u_2}^2)}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda}\hat{u_1}\right), \\
\hat{u_1} &= u_1 \cos\theta + u_2 \sin\theta \quad \text{and} \\
\hat{u_2} &= -u_1 \sin\theta + u_2 \cos\theta,
\end{aligned}
$$

Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.The eight orientations(a) and five scales(b) considered for the image are as follows: Gabor



(a) (b)

Fig. 1. (a) The real part of the Gabor kernels at five scales and eight orientations with the following parameters: = 2, k = =2, and f = p2. (b) The magnitude of the Gabor kernels at five different scales. The kernels exhibit desirable characteristics of spatial frequency, spatial locality, and orientation selectivity.

filters are directly related to Gabor wavelets, since they can be designed for a number of dilations and rotations. However, in general, expansion is not applied for Gabor wavelets, since this requires computation of bi-orthogonal wavelets, which may be very time-consuming. Therefore, usually, a filter bank consisting of Gabor filters with various scales and rotations is created.

C. SURF features

The scale space is created using the gaussian kernels. The scale space is divided into levels and octaves. The interest points are the extrema among 8 neighbors in the current level and its 2 $\times$ 9 neighbors in the level above and below.

Hessian Interest points

A hessian based blob detector is used to find the interest points. The determinant of the hessian matrix expresses the extent of the response and is an expression of the local change around the area.

$$\mathcal{H}(x, z\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

$$L_{xx}(x,\sigma) = I(x) * \frac{\partial^2}{\partial x^2} g(x) \tag{2}$$

$$L_{xy}(x,\sigma) = I(x) * \frac{\partial^2}{\partial xy} g(x) \tag{3}$$

$L_{xx}(x,\sigma)$ is the convolution of the image with second derivative of the gaussian. The second order gaussian kernels must be discretized and cropped before we can apply them. The kernel is approximated by a box filter. The grey areas correspond to 0, white as positive and black as negative. The determinant is calculated approximately as:

$$Det(\mathcal{H}_{approx}) = D_{xx}D_{xy} - (wD_{xy})^2 \tag{4}$$

The extrema are found by examining the points across the octaves.

Description

The SURF descriptor is computed for an area with size 20s. It is based on the Haar wavelet responses. The interest area is divided into 4 $\times$ 4 subareas which is described by the values of the wavelet response in the x and y directions. The interest areas are weighted with a gaussian centered at the interest point to give some robustness for deformations and translations. The components are calculated as:

$$v = \sum dx, \sum |dx|, \sum dy, \sum |dy| \tag{5}$$

For each subarea a vector v is calculated based on 5 $\times$ 5 samples. The descriptor for a interest point is the 16 vectors for the subareas concatenated. Finally the descriptor is normalized to achieve invariance to contrast variations that will represent themselves as a linear scaling of the descriptor.

D. Intensity feature

A two-dimensional feature vector is computed for each superpixel based on the intensity values.The first dimension is the average intensity values of all pixels within the superpixel S, where I(x,y) is the intensity of pixel (x, y) and n is the total number of pixels within the superpixel Si. The second dimension is computed as the average intensity values of the neighboring superpixels of Si, where η represents the neighboring superpixels of Si and N is the number of neighboring superpixels.

$$f_1(i) = \frac{1}{n} \sum_{(x,y)\in i} I_{(x,y)}$$

Fig. 2. The average intensity

$$f_2(i) = \frac{1}{N} \sum_{j \in \eta} f_1(j),$$

E. Standard Deviation feature

Similar to intensity, we also compute a two-dimensional feature based on the standard deviation values in small pixels neighborhoods around each image pixel.The standard deviation feature for the superpixel is then computed in the same way as that computed for the intensity feature.

F. Local Binary Pattern

The local binary pattern is used as a feature vector.A sliding window (3 x 3) was passed over the image and for each pixel in the image, the intensity difference between it and it's 8 neighbouring pixels were computed. If the centre pixel intensity is greater than the neighbouring pixel, then value is taken as 1 otherwise it is taken as 0.This gives a 8-bit binary number. This binary number is then converted into decimal to get the intensity at the centre pixel. For this Local Binary Pattern Image, the histograms were computed for each superpixel.This gives a 256 dimensional vector for each superpixel for both the gray and colour image. The euclidean distance between the histogram of both the target and reference superpixel are compared and the set of reference superpixels that match the target superpixel atmost are found.

G. Gray Level Co-occurence Matrix

The gray level co-occurence matrix (C) is created by considering the value of a pixel at a particular location and the pixel to its immediate right. Initially an empty matrix of size 255 $\times$ 255 is generated. When a pixel of value i is encountered in the image with the immediate neighbor as j, the corresponding location C_{ij} is incremented by 1. The matrix C is then divided by the sum of all the individual values in C. Various features can be computed using the matrix C such as contrast, entropy, homogeneity and energy.

$$Contrast = \sum_i \sum_j (i-j)^2 c_{ij} \tag{6}$$

$$Entropy = -\sum_i \sum_j c_{ij} log c_{ij} \tag{7}$$

$$Energy = \sum_i \sum_j c_{ij}^2 \tag{8}$$

$$Homogeneity = \sum_i \sum_j \frac{c_{ij}}{1+|i-j|} \tag{9}$$

III. Feature Matching

We exploit the features extracted in the previous sections to find the correspondences between the target and reference superpixels and use these to assign a initial set of colour values to the target superpixels.For each target superpixel, one can search among all reference superpixels across all feature types to find the reference superpixel which is most similar to the target. This however demands large processing time. For greater efficiency, we instead employ a fast cascade feature matching scheme which continually prunes the search space at each step of the cascade and concentrates the search only on reference superpixels which are sufficiently similar to the target. To ensure that the search space are pruned

reliably, we exploit the more discriminative Gabor and SURF features at the initial cascade steps to sieve out a set of matching reference superpixels for a target superpixel, before relying on the intensity and standard deviation features to find its final matching reference superpixel. In our work, we found feature matches to be largely unaffected by using SURF before after Gabor and intensity before after standard deviation. Let r_i denote the set of reference superpixels

which are extracted from the reference image I_r. Consider a target superpixel t_i. Starting at the first cascade step, we find a set of α reference superpixels from r_i which are most similar to t_i based on the Gabor features. Let this set of α reference superpixels be denoted as ϕ_i. We compute distance between two features of the same type by the Euclidean distance measure. Following that, at the second level, we

find α_2 reference superpixels from ϕ_i which is most similar to the currently considered target superpixel t_i based on the SURF features. Intensity and standard deviation features are then used in the third and fourth levels respectively to find the set of reference superpixels which are most similar to ti. Let Υ_i denote the set of reference superpixels found by the cascade filtering process to be most similar to ti at the final step of the cascade. The reference superpixel r_a within Υ_i which correspond to t_i is then identified as one with the least matching cost across different feature types to t_i.

$$a = \arg\min_b \ F(r_b, t_i), \quad r_b \in \Upsilon_i,$$

$$\begin{aligned} F(r_b, t_i) &= w_1 C_1(r_b, t_i) + w_2 C_2(r_b, t_i) \\ &+ w_3 C_3(r_b, t_i) + w_4 C_4(r_b, t_i). \end{aligned}$$

We denote C_1, C_2, C_3 and C_4 as the Euclidean distance between the Gabor, SURF, intensity and standard deviations features, and w as their accompanying weights.We fixed w_1, w_2, w_3 and w_4 to be 0.2, 0.5, 0.2 and 0.1 respectively.We then transfer the mean intensity(r,g,b) value of corresponding reference superpixel to the target superpixel.

The initially colored image is then processed further using an optimization based colour interpolation algorithm[9].The algorithm is based on the principle that neighboring pixels with similar luminance should also have similar colors. The algorithm attempts to minimize the difference J(C) between the color assigned to a pixel p and the weighted average of the colors assigned to its neighbors, where the weights are determined by the similarity of their luminance.The code from [9] has been used to implement the color interpolation algorithm.

IV. Image space voting

The correspondences found by the above matching step assign colors to superpixels based solely on image features. While our use of multiple feature types improves color assignments significantly as compared to that using a single feature type, there could be some visually invalid assignments due to incorrect correspondences found. To improve the color assignments, we enforce spatial consistency in the colorization by explicitly voting for the color assignments in the image space. Here,our basic intuition is that color assignment for a superpixel is likely to be correct if its neighboring superpixels which have similar image properties are also assigned similar colors. Consequently, we can exploit neighboring superpixels to identify and to correct invalid color assignments.

Here in the it is clustered based on its corresponding superpixels based on their initial color values with k-means clustering. Densely populated clusters provide strong evidence for the correct color assignments of its member superpixels, while superpixels from sparsely populated clusters indicate that such superpixels have little support for its color assignments. In this regard, the clustering procedure identifies invalid color assignments by pooling evidence from its neighboring superpixels together, where the confidence of a color assignment for a target superpixel is computed as the number of member superpixels belonging to the same cluster as the target superpixel. Here, we identify sparsely populated clusters as those which have less than $\frac{1}{2k}$ the number of superpixels in segment si. We reassign colors to superpixels from sparsely populated clusters by the average color values of the superpixels from the most populated cluster.

V. Object Label based Image colorization

The common objects in the grayscale and the color images were considered and the bounding boxes were obtained manually. The matching of superpixels was then done across the two images only by considering the common objects in the two images. The background superpixels in the color image were matched only to the background pixels in the grayscale image. This was done by considering the previously obtained features.

We performed another experiment by creating mask images for the objects by considering the pixel locations of the objects in both the images. The matching of the superpixels was done for the objects by using the superpixels overlapping with the mask image. The same operation was performed for the background superpixels in both the images.

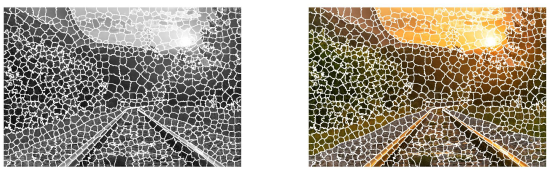
VI. Results

The following are the Reference and the target images used to colorize the target image.Initially the procedure is followed with same image followed by applying the procedure to similar images.

(a) (b)

Fig. 3. (a)Target image (b) Reference image

The superpixels over the image are shown in figure 4.The feature matching is done with the aid of superpixels.



(a) (b)

Fig. 4. (a)Superpixels of Target image (b) Superpixels of Reference image

The micro scribbles after the cascade feature matching are sent to the colorize procedure which fine tunes the color in the super pixels.It is shown in the figure 5.

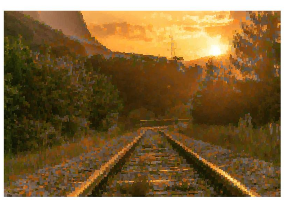
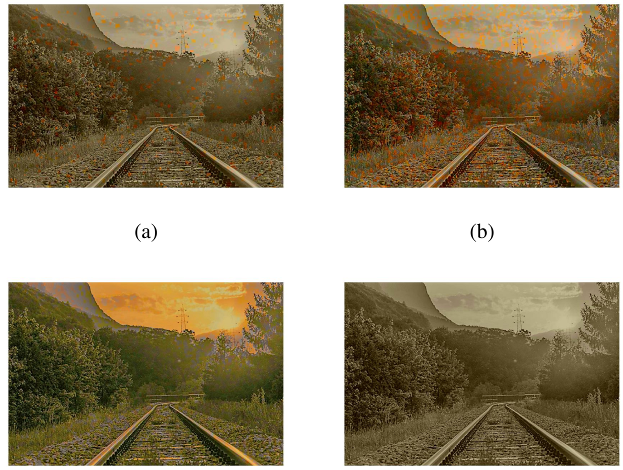


Fig. 5. The micro-scribbles obtained at the end of cascade feature matching.

The image colorization result only if SURF features are applied is shown in figure6(a).The image colorization only if gabor features are used is shown in figure6(b).The image colorization using intensity matching between reference and target is shown in figure6(c).The image colorization using standard deviation matching is shown in figure6(d).

The final colorized image is obtained as shown in the figure 7.



(a) (b)

(c) (d)

Fig. 6. (a)Colorization using only SURF features (b) Colorization using only Gabor features (c)Colorization using only intensity matching (d)Colorzation using only surrounding pixel characteristics

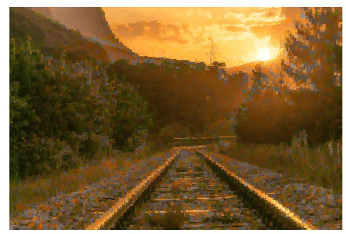
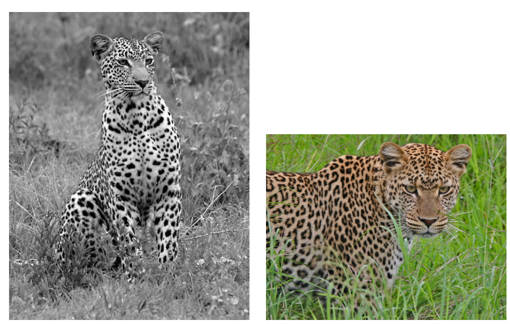


Fig. 7. Final colorized image.

The following are the Reference and the target images used to colorize the target image.Now the algorithm is applied to the similar images and colorization results are shown.



(a) (b)

Fig. 8. (a)Target image (b) Reference image

Initially when the colorization is applied the following result is obtained.It is shown in figure 9(a).And then the same algorithm is applied by putting a rectangular bounding box is applied to the object and colorization is done from object to object and background to background.It is shown in figure 9(b)

(a) (b)

Fig. 9. (a)Without bounding box (b) With rectangular bounding box

Better results are observed after applying the mask in the shape of the object.It is obtained as follows

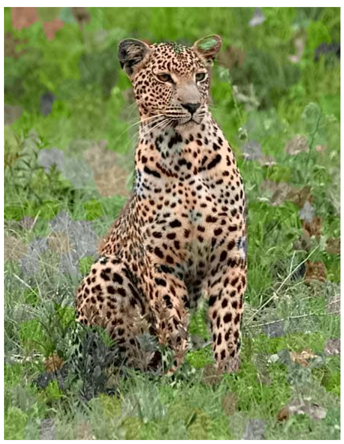


Fig. 10. Colorization With mask applied

VII. Conclusion

Many other features were tried but majority of proper colorization is obtained with SURF,Gabor,Intensity and standard deviation.Other features used like LocalBinaryPatterns,Graylevel co-occurence matrix are tried but were not observed to increase the output efficiency greatly. Better colorization of the images can be applied if more features are applied and colorization will be more specific if labeled images are used and corresponding superpixels are colorized.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf:Speeded up robust features. Computer Vision and Image Understanding (CVIU), 110(3):346359, 2008.

[2] G. Charpiat, M. Hofmann, and B. Scholkopf. Automatic image colorization via multimodal predictions. In Proc. European Conference on Computer Vision, 2008.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(05):603619, 2002.

[5] D. Hasler and S. Sezustrunk. Measuring colourfulness in natural images. SPIE Electronic Imaging, 5007:8795, 2003..

[6] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless,and D. H. Salesin. Image analogies. Proc.SIGGRAPH, pages 327340, 2001.

[7] 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 ACM Multimedia, pages 351354, 2005.

[8] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In Proc. Eurographics Symposium on Rendering, 2005.

[9] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. ACM Transactions on Graphics, 23(3):689694, 2004.