

# Texture Feature Extraction to Colorize Gray Images

Smriti Kumar  
Deptt. of E&TC Engg.  
CSIT Durg  
Chhattisgarh, India.

Deepak Singh  
Deptt. of E&TC Engg.  
CSIT Durg  
Chhattisgarh, India.

## ABSTRACT

This work presents a simple process for gray image colorization, using a colored image which is similar to this gray scale image but not the colorized version of the gray image. This colored image is retrieved from the data base of colored images that has been created for this purpose. Here, the texture properties of the colored images are extracted and stored. For the purpose of colorization these features are compared with those of the gray image to be colorized and the best matching image is found out from the database. For colorization of this gray scale image a decorrelated color space YCbCr is utilized. This technique is completely automatic and no human intervention is required in the process of colorization. Apart from this the technique presented here is very fast and produces good quality results as compared to the conventional colorization methods. Texture features used here to calculate a texture similarity measure are energy, entropy, contrast, homogeneity, autocorrelation based on correlation matrix as well as coarseness and directionality.

## General Terms

Image Retrieval, Gray Scale Image Colorization, Texture Analysis.

## Keywords

Texture Features, Entropy, Energy, Contrast, Autocorrelation, Homogeneity, Coarseness, Directionality.

## 1. INTRODUCTION

Adding colors to gray scale images not only increases the visual appeal of images but also enhances the information content of images. This is especially true in case of scientific images like Computerized Tomography (CT), X-ray images and Magnetic Resonance Imaging (MRI) can be enhanced when colorized for demonstrations, presentations, scientific data and old movies too [1-2]. Colorization techniques can be broadly classified into Hand colorization, Semi Automatic and Automatic colorization techniques.

Hand Colorization techniques are the modes through which artists have been expressing their talents since ages. Now a days many softwares like Adobe photoshop, Paint pro, Black Magic etc. are available for this purpose. Semi automatic colorization is the mapping of luminance values to color values, done by converting the gray image to an indexed image [3]. The choice of the color-map is commonly determined by human decision. Pseudocoloring is a common example of the semi automatic coloring technique for adding color to grayscale images. Automatic Colorization is the technique to fully automate the gray image coloring process. Automatic coloring can be classified into three categories, as per the source of chromatic properties to be transferred to the gray image pixels [4], which are: Transformational Coloring, User Selection and Image Matching.

In the transformational coloring [5], the coloring is done by applying a function which is called a transformation function  $T_k$  upon the intensity value of each pixel  $I_g(x,y)$  resulting in the chromatic value  $I_{ck}(x,y)$  for channel k. Hence it can be said that:

$$I_{ck}(x,y) = T_k[I_g(x,y)] \quad (1)$$

Where  $I_g$  is the intensity value of pixel the  $x, y$ , and  $I_{ck}$  is the  $k$  channel chromatic value for the same pixel).

In user selection coloring [6], the user marks some samples points in the gray image and selects the colors of each, using a tool similar to a brush. There after the computer is responsible of spreading the color on the bounded region containing the colored samples.

In Image Matching technique, a colored image is selected manually to be a reference for the color palette used in coloring process. The pixels of the gray image  $I_g$  are matched with a source colored image  $I_s$  pixels.

Technique of interest here is Image Matching/ Coloring by Reference colored image. This technique is based on the fact that for each gray pixel  $I_g(x,y)$  in target gray scale image there exists a colored pixel  $I_s(x_2, y_2)$  in reference or source image such that the distance  $E$  (which is some similarity measure) between them exhibits the minimum value. The chromatic properties in ' $I_s$ ' are transferred to ' $I_g$ ' and the achromatic value of ' $I_g$ ' is retained [7-10].

When one represents any typical three channel image in any of the well-known color spaces, the different channels' values will be correlated. For instance, in RGB color space, most pixels for the red and green channel will have large values if the blue channel is large. This implies that if it is desired to alter the chromatic properties of a pixel in a coherent way, then one must modify all color channels in tandem. Thus any type of color modification process gets complicated.

So the prime requirement is an orthogonal color space without correlations between axes. Thus color space with decorrelated axes is a useful tool for manipulating color images in this work. Ruderman et al [11] developed such a color space, called  $\alpha\beta$ , this color space minimizes correlation between channels for many natural scenes, since it is based on data-driven human perception research that assumes that the human visual system is ideal for processing natural scenes. There's little correlation between the axes in  $\alpha\beta$  space, which allows the application of different operations in different color channels without troubles, due to undesirable cross-channel artifacts. Apart from this,  $\alpha\beta$  color space also exhibits logarithmic properties, which means that to a first approximation the uniform changes in channel intensity are equally detectable [11]. This makes  $\alpha\beta$  space, a very popular choice where colorization is concerned.

In the proposed work, another decorrelated color space YCbCr color space has been used for colorization [12-13]. YCbCr color space provides three decorrelated channels Y, Cb and Cr. Channel Y is achromatic luminance channel, and the chromatic channels Cb and Cr correspond to the difference between blue component with a reference value and difference between red components with a reference value, respectively. This space also allows selective transfer of chromatic Cb and Cr channels from color image to grayscale image without introducing cross- channel artifacts.

Apart from decorrelated color space for pixel matching in image colorization by reference technique, other parameters like texture features are also required. Texture is a spatial property, a simple one-dimensional histogram is not useful in characterizing texture. In order to capture the spatial dependence of gray-level values which contribute to the perception of texture, a two-dimensional dependence matrix known as a gray-level co-occurrence matrix is extensively used in texture analysis. Other measures that have been used extensively are the autocorrelation function as well as coarseness and directionality [14].

To colorize a gray scale image automatically the foremost requirement is a database of color images from where the most similar in composition image can be chosen to serve the purpose of source image whose chromatic properties will be transferred to this gray image. Thus a database of various color images from Berkeley database has been created [15] (see Figure 1). For retrieval of the images from this database, the texture features of all these images are extracted and stored, to be compared with the features of target gray scale image to be colorized. All the experiments concerned with this work are performed using MATLAB version 7.0.10.499 (R2010a) on Intel(R) Core(TM) i5-3210 CPU@ 2.50GHz.

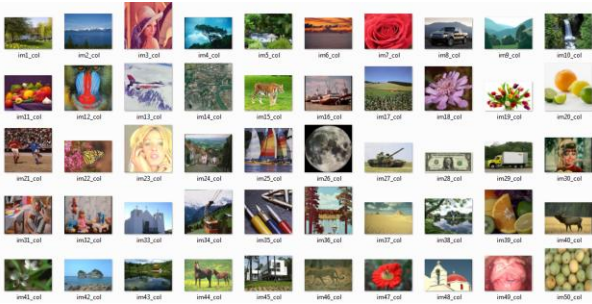


Fig 1: Database

## 2. METHODOLOGY

First step of this method is to extract the texture features of the image database created. Here texture features utilized are energy, entropy, contrast, homogeneity, autocorrelation, coarseness and directionality. These are explained subsequently.

The gray-level co-occurrence matrix [14][16][17]  $P[i, j]$  is defined by first specifying a displacement vector  $d = (dx, dy)$  and counting all pairs of pixels separated by  $d$  having gray levels  $i$  and  $j$ . A gray-level co-occurrence matrix captures spatial distribution of gray levels. Entropy a feature measuring randomness of gray-level distribution is, defined as:

$$Entropy = - \sum_i \sum_j P[i, j] \log P[i, j] \quad (2)$$

Note that the entropy is highest when in an image there are no preferred gray-level pairs for the specified distance vector  $d$ . The features of energy, contrast, and homogeneity are also defined using the gray-level co-occurrence matrix as given below:

$$Energy = - \sum_i \sum_j P^2[i, j] \quad (3)$$

$$Contrast = - \sum_i \sum_j (i - j)^2 P[i, j] \quad (4)$$

$$Homogeneity = - \sum_i \sum_j P[i, j] / (1 + |i - j|) \quad (5)$$

The choice of the displacement vector  $d$  is an important parameter in the definition of the gray-level co-occurrence matrix. Occasionally, the co-occurrence matrix is computed for several values of  $d$  and the one which maximizes a statistical measure computed from  $P[i, j]$  is used. The gray-level co-occurrence matrix approach is particularly suitable for describing microtextures. It is not suitable for textures comprising large area primitives since it does not capture shape properties. Gray-level co-occurrence matrices have been used extensively in remote sensing applications for land-use classification. The autocorrelation function [14][16][17]  $p[k, l]$  for an  $N \times N$  image is defined as follows:

$$p[k, l] = \frac{\frac{1}{(N-k)(N-l)} \sum_{i=1}^{N-k} \sum_{j=1}^{N-l} f(i, j) f(i+k, j+l)}{\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N f^2(i, j)} \quad (6)$$

for  $0 \leq k, l \leq N - 1$

Where,  $f(i, j)$  is the gray level intensity of pixel in the subject image whose coordinates are  $(i, j)$ . For images comprising repetitive texture patterns the autocorrelation function exhibits periodic behavior with a period equal to the spacing between adjacent texture primitives. When the texture is coarse, the autocorrelation function drops off slowly, whereas for fine textures it drops off rapidly. The autocorrelation function is used as a measure of periodicity of texture as well as a measure of the scale of the texture primitives.

Coarseness [14][16][17] scales repetition rates. An image will contain textures at several scales; coarseness aims to identify the largest size at which a texture exists, even where a smaller micro texture exists. Computationally one first takes averages at each point over neighborhood the linear size of which are powers of 2. Average over the neighborhood of size  $2^k \times 2^k$  at the point  $(x, y)$  is:

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i, j) / 2^{2k} \quad (7)$$

Where,  $f(i, j)$  is the gray level intensity of pixel in the subject image whose coordinates are  $(i, j)$ . Then at each point one takes differences between pairs of averages corresponding to non-overlapping neighborhoods on opposite sides of the point in both horizontal and vertical orientations. In the horizontal case this is

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)| \quad (8)$$

At each point, one then picks the best size which gives the highest output value, where  $k$  maximizes  $E$  in either direction. The coarseness measure is then the average of  $S_{opt}(x, y) = 2^{k_{opt}}$  over the picture.

Directionality [14][16][17] is a global property over a region it does not aim to differentiate between different orientations or patterns, but measures the total degree of directionality. Two simple masks are used to detect edges in the image. At each pixel the angle and magnitude are calculated. A histogram,  $H_d$ , of edge probabilities is then built up by counting all points with magnitude greater than a threshold and quantizing by the edge angle. The histogram will reflect the degree of directionality. To extract a measure from  $H_d$  the sharpness of the peaks are computed from their second moments. In other words:

1) The edge strength  $e_d(x,y)$  and the directional angle  $a_d(x,y)$  are computed using approximate pixel-wise derivatives computed by the Sobel edge detector in the 3X3 moving window:

$$e_d(x,y) = 0.5(|\Delta_x(x,y)| + |\Delta_y(x,y)|) \quad (9)$$

$$a_d(x,y) = \frac{\tan^{-1}(\Delta_x(x,y))}{\Delta_y(x,y)} \quad (10)$$

2) Histogram  $H_{dir}(a_d)$  of quantized directional values  $a_d$  - the numbers of the edge pixels  $(x,y)$  with the directional angle  $a_d(x,y)$  and the edge strength  $e_d(x,y)$  greater than a predefined threshold:

– The histogram  $H_{dir}(a_d)$  is relatively uniform for images without strong orientation.

– The histogram  $H_{dir}(a_d)$  exhibits peaks for highly directional images.

3) The degree of directionality relates to the sharpness of peaks:

$$F_{dir} = 1 - r \cdot n_p \sum_{p=1}^{n_p} \sum_{a_d \in} (a_d - a_{dp})^2 H_{dir}(a_d) \quad (11)$$

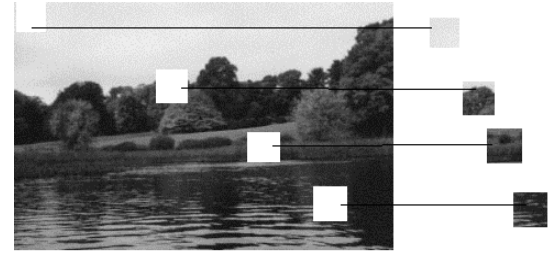
Where,  $n_p$  - the number of peaks,  $a_{dp}$  - the position of the pth peak,  $w_p$  - the range of the angles around the pth peak,  $r$  - normalizing factor related to quantizing levels of  $a_d$  - the quantized directional angle (modulo 180°). Using the above formulae equations (2-11) we extract the texture features of the images in database and store them as texture database.

In the next step a gray scale image (referred to as target image) to be colorized is analyzed for the texture features viz energy, entropy, contrast, homogeneity, autocorrelation, coarseness and directionality. These features are extracted and compared with the texture database and the best matching image from the database is extracted based on the texture feature matching. Now the target gray scale image is divided into small windows of equal size, (see Figure 2). Various window sizes have been tried and it was observed that a window size of 3X3 produces best results although the time consumed for colorization in this case is quiet long. So other window sizes ranging from 5X5 to 15X15 were tried for several images and it has been observed that a window size of 7X7 produces good quality output images and the time consumed is also reduced to a great extent the source image is also divided into windows of equal size and the size of these windows is same as that of the window size chosen for the target image. Each of these windows is now average filtered. Average filtering is done here because a moving average filter smoothens the data by replacing each data point with the average of the neighboring data points defined within the span. This process is equivalent to low pass filtering with the response of the smoothing given by the difference equation [18]:

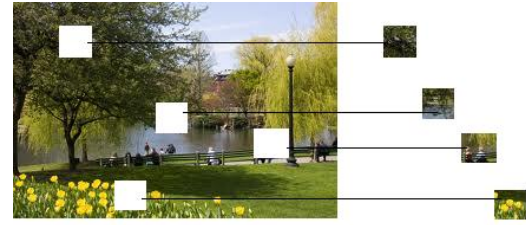
$$y_s(i) = \frac{y(i+N) + y(i+N-1) + \dots + y(i-N)}{2N+1} \quad (12)$$

Where  $y_s(i)$  is the smoothed value for the  $i^{th}$  data point,  $N$  is the number of neighboring data points on either side of  $y_s(i)$ , and  $2N+1$  is the span.

In the next step, the windows of both the target gray scale image and the colored source image are analyzed for texture features and the texture features are extracted (equations(2-11)) for comparison and finding the best matching windows for color transfer. Once the texture features are extracted from each of these windows in target and source images in the form of numerical values and stored, for use in segmentation stage



(a)



(b)

**Fig2: Windowing of Image (a) Target Image (MATLAB Image 'Autumn') (b) Source Image (Berkeley Database).**

This technique uses K means clustering method for image segmentation [4] [19]. The K means algorithm is an iterative process utilized to partition an image into K clusters. The basic algorithm is:

- Pick  $K$  cluster centers, either randomly or based on some parameter.
- The cluster that exhibits minimum distance between pixel and cluster center, is assigned with that pixel.
- By averaging all of the pixels in the cluster, the cluster centers are recomputed.
- Until convergence is attained, above two steps are repeated.

In this case, distance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors.

In this methodology  $K$  is chosen to be 5 and so five cluster centers are chosen. To calculate the cluster center the average of texture features extracted in the form of numerical values is calculated for each of the windows in the target as well as in the source image and this texture feature metric is stored in a matrix. The texture feature metric TFM for a window of size  $(n \times n)$  is calculated as:



$$TFM_m = \sum_{i=1}^x \frac{TF_i}{x} \quad (13)$$

Where,  $m$  is the window number,  $TF$  is the target feature,  $x$  is the total number of features extracted. In the next step five cluster centers are chosen in the range from the minimum to maximum values of elements of the matrix for target image and as discussed above the pixels of the target image are segmented or grouped into five clusters, (see Figure3). A cluster center is calculated here as:

$$CC_i = \frac{i(\max(TFM) - \min(TFM))}{k} \quad \forall i = 1, 2, \dots, k \quad (14)$$

Where,  $CC$  is the cluster center,  $i$  is the center number,  $k$  is the maximum number of cluster. After this the source image windows' texture features are matched with the cluster groups in segmented target image one by one and the best matching source window to a particular cluster in target image is found out (as shown in Figure 4).



Fig 3: Target image Segmented into Five clusters

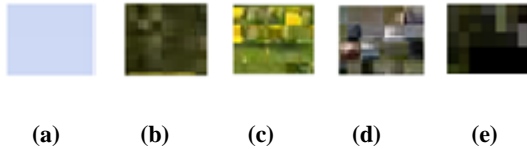


Fig 4: (a) Best Matching Window from Source Image to Cluster 1 in target Image. (b) Best Matching Window from Source Image to Cluster 2 in target Image. (c) Best Matching Window from Source Image to Cluster 3 in target Image. (d) Best Matching Window from Source Image to Cluster 4 in target Image. (e) Best Matching Window from Source Image to Cluster 5 in target Image.

This matching window in the source image is now used to colorize the corresponding cluster's pixel in target image, so the next stage of this methodology is colorization. In the colorization stage each of the clusters in target image is colorized using the source image windows that match best to that target image cluster. The first step in colorization is converting the source image window to a decorrelated color space. In the proposed work, decorrelated color space YCbCr color space has been used for colorization [12].

YCbCr is one of the popular color space in computing. It represents colors in terms of one luminance component/ luma ( $Y$ ), and two chrominance components/ chroma ( $Cb$  and  $Cr$ ).  $Y$  is achromatic luminance channel, and the chromatic channels  $Cb$  and  $Cr$  correspond to the difference between blue component with a reference value and difference between red components with a reference value, respectively. This space

also allows selective transfer of chromatic  $Cb$  and  $Cr$  channels from color image to grayscale image without introducing cross-channel artifacts.

The following equations illustrate conversion of RGB into YCbCr color space and vice versa:

$$Y = 0.2989R + 0.5866G + 0.1145B \quad (15)$$

$$Cb = -0.1688R - 0.3312G - 0.5000B \quad (16)$$

$$Cr = 0.5000R - 0.4184G - 0.0816B \quad (17)$$

And back

$$R = Y + 1.402(Cr - 128) \quad (18)$$

$$G = Y - 0.34414(Cb - 128) - 0.71414(Cr - 128) \quad (19)$$

$$B = Y - 1.772(Cb - 128) \quad (20)$$

The main motive of this work is to make an image take on another image's look and feel. More formally this means that it is desirable here that some aspects of the distribution of data points in YCbCr color space to transfer between images. For this purposes, the mean and standard deviations along each of the three axes suffice. Thus, one computes these measures for both the source and target image windows. Note that one computes the mean and standard deviation for each axis separately in YCbCr Color space. First, is subtracted the mean from the data points:

$$Y^* = Y - \langle Y \rangle \quad (21)$$

$$Cb^* = Cb - \langle Cb \rangle \quad (22)$$

$$Cr^* = Cr - \langle Cr \rangle \quad (23)$$

Then, are scaled the data points comprising the target image by factors determined by the respective standard deviations:

$$Y' = \frac{\sigma_t^Y}{\sigma_s^Y} Y^* \quad (24)$$

$$Cb' = \frac{\sigma_t^{Cb}}{\sigma_s^{Cb}} Cb^* \quad (25)$$

$$Cr' = \frac{\sigma_t^{Cr}}{\sigma_s^{Cr}} Cr^* \quad (26)$$

Where subscript  $t$  stands for target gray scale image and subscript  $s$  stands for colored source image and  $\sigma$  stands for standard deviation. After this transformation, the resulting data points have standard deviations that conform to the source image. Next, instead of adding the averages that were previously subtracted in equations (21-23), one adds the averages computed for the source image, in  $Cb$  and  $Cr$  channels of target image. This step transfers the chromatic properties of matching pixels from source image to the target image. As the above mentioned step is not being implemented for the  $Y$  channel of target image so its luminosity is retained. As a result YCbCr version of the target image is obtained (see Figure 5). Finally, the result is converted back to RGB color space using equations (18-20), and the colorized version of the target gray scale image is obtained (see Figure 6).

Because it is assumed that one wants to transfer one image's appearance to another, it's possible to select source and target images that don't work well together. The result's quality depends on the images' similarity in composition. The methodology is summarized here in the block diagram form (see Figure 7). Here the methodology is divided in two stages. Stage 1 is the retrieval phase and Stage 2 is the colorization phase.

The quality of results obtained by the proposed methodology is compared with the results of some other techniques in literature [8-10] based on various parameters like Mean Square Error (*MSE*), Peak Signal to Noise Ratio (*PSNR*), Normalized Cross correlation (*NK*), Structural Content (*SC*), Maximum Difference (*MD*), Normalized Absolute Error (*NAE*), Structural Similarity between Images (*SSIM*), on the basis of number of colors in the output image and on the basis of Time taken for colorization[20-21].



Figure 5: YCbCr Version of Target Gray Scale Image.



Figure 6: Target Gray Scale Image Colorized

Mean Squared Error [20] is the average squared difference between a reference image and a distorted image. It is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count. Mean Square Error between two images of size  $M \times N$  is thus computed as:

$$MSE = \frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N (x_{j,k} - x'_{j,k})^2 \quad (27)$$

Where,  $x_{j,k}$  and  $x'_{j,k}$  is the pixel intensity value at the coordinates the  $(j,k)$  of the two images being compared respectively.

Peak Signal to Noise ratio [20] is the ratio between the reference signal and the distortion signal in an image, given in decibels. The higher the PSNR, the closer the subject image is to the original. In general, a higher PSNR value should correlate to a higher quality image. *PSNR* for  $n$  bit images is calculated as:

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (28)$$

In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. This is also known as a sliding dot product or sliding inner-product. For image-processing applications in which the brightness of the image and template can vary due to lighting and exposure conditions, the images can be first normalized. The closeness between two digital images can also be quantified in terms of correlation function

[20]. Normalized Cross Correlation between two images can be computed as:

$$NK = \frac{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k} x'_{j,k})}{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k})^2} \quad (29)$$

Structural Content [20] is also correlation based measure and measures the similarity between two images. Structural Content (SC) is given by the equation

$$SC = \frac{\sum_{j=1}^M \sum_{k=1}^N (x_{j,k})^2}{\sum_{j=1}^M \sum_{k=1}^N (x'_{j,k})^2} \quad (30)$$

Maximum Difference [20] is the maximum of the error signal (difference between the reference signal and test image). It is computed as:

$$MD = \text{Max}(|x_{j,k} - x'_{j,k}|) \quad (31)$$

Normalized Absolute Error [20] is average of absolute difference between the reference signal and test image. It is given by the equation:

$$NAE = \frac{\sum_{j=1}^M \sum_{k=1}^N |x_{j,k} - x'_{j,k}|}{\sum_{j=1}^M \sum_{k=1}^N |x_{j,k}|} \quad (32)$$

Structural Similarity between Images (*SSIM*) [21] was proposed, based on the assumption that the human visual system is highly adapted to extract structural information from the viewing field. It follows that a measure of structural information change can provide a good approximation to perceived image distortion. The structural information in an image is defined as those attributes that represent the structure of objects in the scene, independent of the average luminance and contrast. The SSIM system separates the task of similarity measurement into three comparisons: luminance, contrast and structure and the three components are combined to yield an overall similarity measure. This results in a specific form of the *SSIM* index:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (33)$$

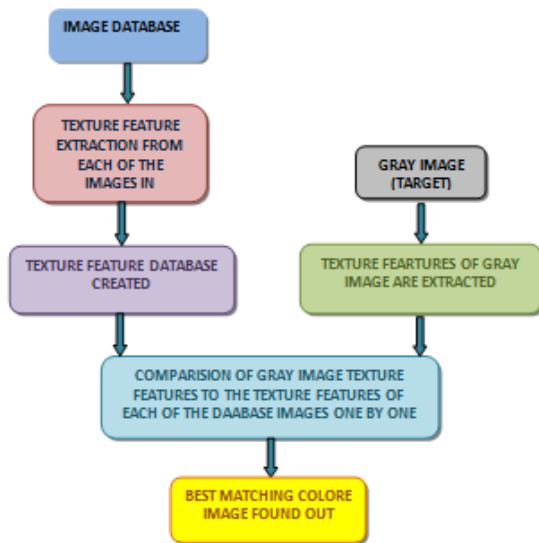
Where,  $x$  and  $y$  are two non negative image signals extracted from two images that are being compared and  $\mu$  is the mean intensity of the two images.  $\sigma$  is the standard deviation used here to estimate the signal contrast.  $C_1$  and  $C_2$  are constants included to avoid instability when  $\mu_x^2 + \mu_y^2$  is very close to zero. Specifically:

$$C_i = (K_i L)^2 \quad (34)$$

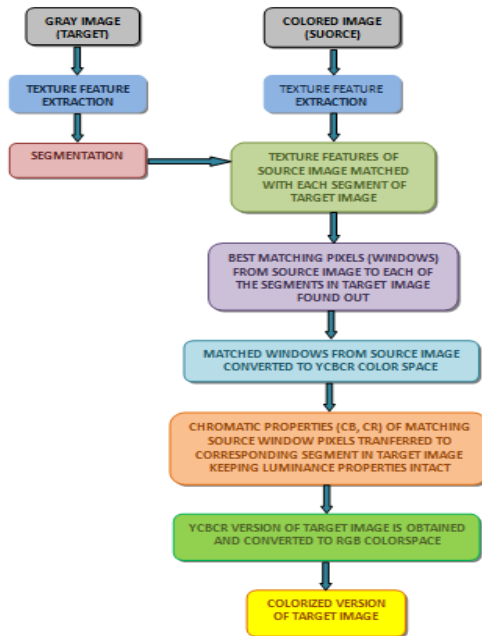
Where  $L$  is dynamic range of pixel values (255 for 8-bit grayscale images), and  $K_i \ll 1$  is a small constant.

Apart from these parameters, the results have also been assessed on the basis of number of unique colors present in the result image. For this simply the image is analyzed pixel by pixel for unique combinations of R, G and B values and counting a combination only once, in case of repeat. This process simply allows the comparison of results obtained through various techniques and through the proposed technique on the basis of number of colors that have been transferred from the colored source image to the target gray scale image in the process of colorization. The results of these comparisons to assess the quality of results obtained have been discussed in subsequent section

earlier texture analysis technique to colorize gray scale images [9] (here it is referred to as jitter colorization) and with the



Stage1: Retrieval Stage



Stage2: Colorization Stage

Fig 7: Methodology in Block Diagram

### 3. RESULTS

The proposed method was tested on various gray scale images for colorization and believable results were obtained. Some of these results are presented here (see Figure 8). These results display the target gray image, the corresponding colored image retrieved from the database and the colorized version of the target gray image obtained from the proposed methodology. The results obtained from the proposed methodology are compared with the results of a very popular gray image colorization technique “Global Image Colorization Technique”[8], comparisons have also been made with the

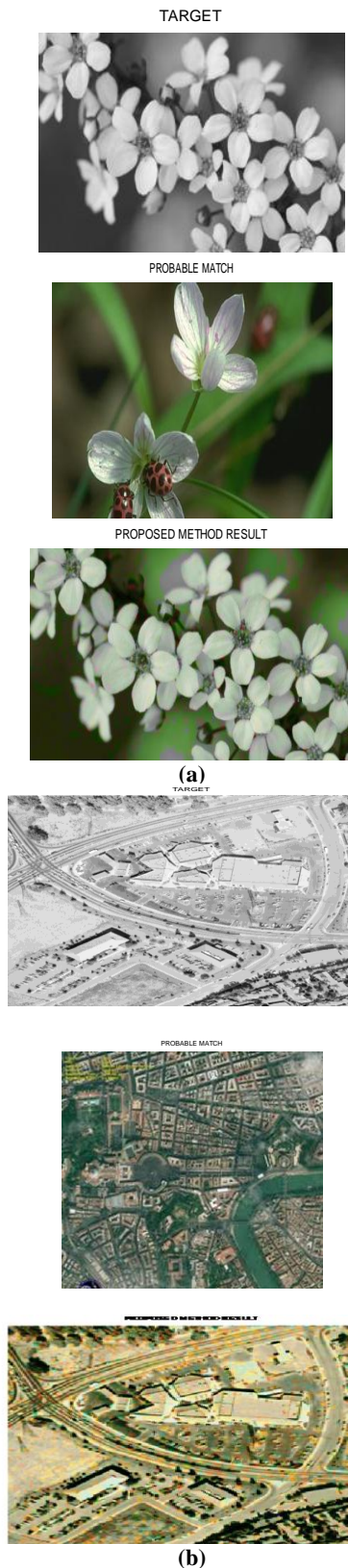
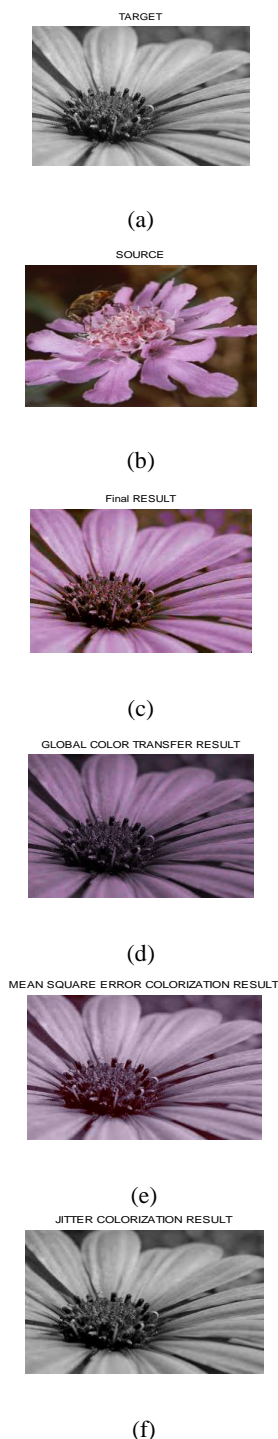


Fig 8: Colorization of Gray Scale Images by Proposed Technique

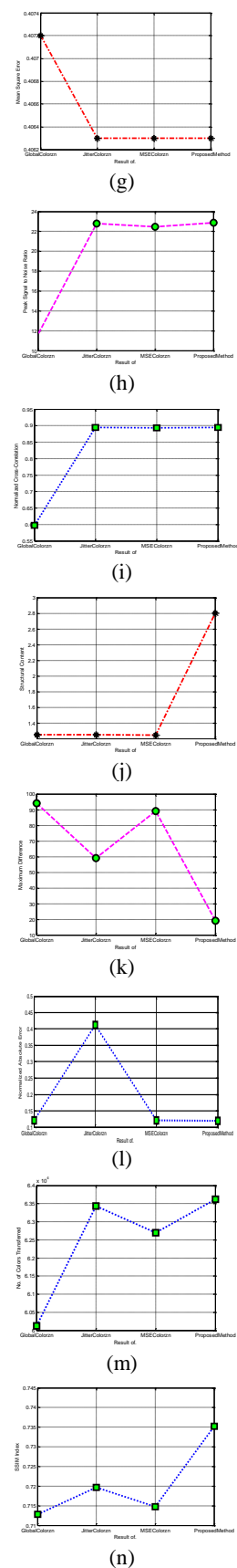
techniques that use Mean Square Error as a metric to compute difference metric for pixel matching in the process of colorization [10] (here it is referred to as MSE colorization). It

is observed that the result of proposed method performs better in terms of various parameters used to assess the quality of images obtained and even the number of colors transferred is maximum in case of proposed methodology, (see Figure 9).



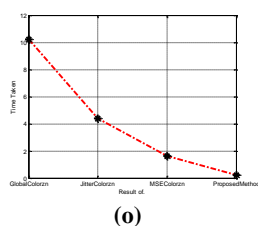
**Fig 9:(a) Target Gray Scale Image (b) Source Colored Image (c) Proposed Method (d) Global Colorization Result (e) MSE Colorization Result (f) Jitter Colorization Result**

As it is obvious from the results shown in Figure 9 the proposed method gives colorized version of target gray scale image which is not only better visually but also performs better when compared with the results of other methodologies [8-10], in terms of Mean Square Error, Peak Signal to Noise Ratio, Normalized Cross correlation, Structural Content,



**Fig 9: All results compared in terms of (g) Mean Square Error (h) Peak Signal to Noise Ratio (i) Normalized Cross Correlation (j) Structural Content (k)Maximum Difference(l) Normalized Absolute Error (m) Number of Colors transferred (n) SSIM Index**





**Fig 9: All results compared in terms of (o) Time taken for Colorization**

Maximum Difference, Normalized Absolute Error, Structural Similarity between Images, on the basis of number of colors in the output image and also in terms of time taken for colorization. The target gray scale images shown in Figure 8 and Figure 9 are of the dimension 256x256 and while global colorization method takes around 8 to 10 seconds to colorize such images, proposed method takes around 0.5 seconds for colorization. It is also obvious that Mean Square Error with respect to the target image is least in the results of proposed method. The Peak Signal to noise ratio is also maximum and usually more than 23dB in case of the results of proposed methodology. Similarly the performance in terms of other quality metrics is also good. Even in terms of number of colors in the final output the results of proposed methodology perform very well and above all, Structural Similarity Index between output images and the source image is also much larger in the case of output of proposed methodology, which is a very important image quality metric

#### 4. CONCLUSION

It can be concluded that the proposed methodology to automatically colorize gray scale images using texture features viz energy, entropy, contrast, homogeneity, autocorrelation, coarseness and directionality, successfully retrieves the appropriate colored image from the database and transfers its chromatic properties to the gray scale image. The results obtained through this methodology also perform well in terms image quality measures as well as in terms of number of colors present in the image.

#### 5. REFERENCES

- [1] Ambika Kalia, Balwinder Singh, "Colorization of grayscale images: an overview" Journal of Global Research in Computer Science, Volume 2, No. 8, Pages 34-38, August 2011.
- [2] Przemyslaw Lagodzinski and Bogdan Smolka, "Colorization of Medical Images" Proceedings of 2009 , APSIPA Annual Summit and Conference, Pages 769-772, Sapporo, Japan, Oct 4-7, 2009.
- [3] J. H. Jang and J. B. Ra, "Pseudo-Color Image Fusion Based on Intensity-Hue-Saturation Color Space" In Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Pages 284- 289, Seoul, Korea, August 20 - 22, 2008.
- [4] Noura A.Semary, Mohiy M. Hadhoud, W. S. El-Kilani, and Nabil A. Ismail, "Texture Recognition Based Gray Image Coloring", In the Proceedings of The 24th IEEE, National Radio Science Conference (NRSC2007), Pages. C22, Cairo, Egypt, March 13-15, 2007.
- [5] J. Yoo and S.Oh , "A Coloring Method of Gray-Level Image using Neural Network," Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems, Volume 2, Pages 1203-1206, Singapore, 1997.
- [6] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," ACM Transactions on Graphics, Volume 23, Issue 3, Pages.689–694, 2004.
- [7] E.Reinhard, M.Ashikhmin, B.Gooch And P.Shirley , "Color Transfer between Images," IEEE Computer Graphics and Applications, Sep/Oct, Volume 21, Issue. 5, Pages 34-41, 2001.
- [8] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to grayscale images" In Proceedings of the 29th Annual ACM SIGGRAPH Conference on Computer Graphics and interactive Techniques, Pages 277-280, San Antonio, Texas, July 23-26, 2002.
- [9] Alexei A. Efros, William T. Freeman, "Image Quilting for Texture Synthesis and Transfer," In Proceedings of SIGGRAPH- 01, International Computer Graphics, Annual Conference Series, Pages 465–470,Florida, 2001.
- [10] Dr. H.B. Kekre, Sudeep Thapade & Nikita Bhandari, "Colorization of Greyscale Images Using Kekre's Biorthogonal Color Spaces and Kekre's Fast Codebook Generation", Advances in Multimedia - An International Journal (AMIJ), Volume 1, Issue 3, Pages- 49-58, Sept. 2011.
- [11] Daniel L. Ruderman, Thomas W. Cronin and Chuan-Chin Chiao, "Statistics of cone responses to natural images: implications for visual coding", J. Opt. Soc. Am. A/ Volume 15, No. 8, Pages 2036-2044, August, 1998.
- [12] Noor A. Ibraheem, Mokhtar M. Hasan, Rafiqul Z. Khan, Pramod K. Mishra, "Understanding Color Models: A Review" ARPJ Journal of Science and Technology, Volume. 2, No. 3, Pages 265-275, ISSN 2225-7217, April 2012.
- [13] Anil k. Jain, "Fundamentals of Digital Image Processing" Prentice-Hall, Publication, USA, ISBN -0-13-336165-9.
- [14] Harlick, R.M.Shanmugam, Dinstein I. "Texture features for image classification", IEEE Transaction on System, Man and Cybernetics, SMC-3:6 10, Pages 621, 1973.
- [15] <http://www.eecs.berkeley.edu/Research/Projects/CS/visi on/grouping/segbench/>
- [16] Abbadeni, N. "Computational Perceptual Features for Texture Representation and Retrieval" IEEE Transactions on Image Processing, Volume 20, Issue1 Pages 236 – 246, January 2011.
- [17] Mark S. Nixon, Alberto S. Aguado, Newnes, "Feature Extraction and Image processing", Oxford Publication, a member of the Reed Elsevier group, Edition No. 1, 2002.
- [18] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine vision" Published by McGraw-Hill, Inc., USA, Edition no.1, ISBN 0-07-032018-7, 1995.
- [19] Gonzalez R. C., Woods R. E., and Eddins, S. L., "Digital Image Processing Using MATLAB." Pearson Education New Delhi (India), Edition No. 1 (Reprint), 2004.
- [20] C.Sasivarnan, A.Jagan, Jaspreet Kaur, Divya Jyoti, Dr.D.S.Rao, "Image Quality Assessment Techniques pn Spatial Domain" IJCST Vol. 2, Issue 3, Pages 177-184, ISSN- 2229-4333 ,September 2011.
- [21] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity" IEEE Transactions On Image Processing, Vol. 13, No. 4, April 2004.