Colorization of Gray Level Images by Neural Network

[*]Hossein Ghayoumi Zadeh [1], Hojat Jafari[1], Ali Hayati[2], Alireza Malvandi[1], Mohammad Fiuzy[1], Javad Haddadnia[1]

[1.] Sabzevar Tarbiat Moallem University/Department of Electrical Engineering, Sabzevar, Khorasan Razavi Iran
[2]Islamic Azad University Sabzevar Branch, Sabzevar, Khorasan Razavi Iran
Corresponding:[*] h.ghayoumizadeh@gmail.com

Abstract: in this paper, first, we color the gray level pictures but later we want to perform the coloring of human pictures automatically and intelligently by means of the implemented neural network. Because of the technique applied in this paper, this method can be used in colorizing medical images. Color images achieved have good distinction and separation. For this purpose, a combination of artificial neural networks and some image processing algorithms was developed to transfer colors from a user-selected source image to a target grayscale image. The proposed method can be used to separate the objects in gray images. Our method is based on a simple premise: neighboring pixels in space-time that have similar intensities should have similar colors. We formalize this premise using a quadratic cost function and obtain an optimization problem that can be solved efficiently using standard techniques. In our approach an artist only needs to annotate the image with a few color scribbles, and the indicated colors are automatically propagated in both space and time to produce a fully colorized image or sequence.
[Hossein Ghayoumi Zadeh, Hojat Jafari, Ali Hayati, Alireza Malvandi, Mohammad Fiuzy, Javad Haddadnia. **Colorization of Gray Level Images by Neural Network.** *Academ Arena* 2017;9(11):1-7]. ISSN 1553-992X (print); ISSN 2158-771X (online). http://www.sciencepub.net/academia. 1. doi:10.7537/marsaaj091117.01.

Keywords: colorization, Equalization, neural networks, gray level.

1. Introduction

Colorization is the art of adding color to a monochrome image or movie. This is done in order to increase the visual appeal of images such as old black and white photos, classic movies or scientific illustrations. Various semi-automatic colorization approaches have been published previously. They all involve some form of partial human intervention in order to make a mapping between the color and the intensity. Luminance keying also known as pseudo coloring [1] is a basic colorization technique which utilizes a user defined look-up table to transform each level of grayscale intensity into a specified hue, saturation and brightness, i.e. a global color vector is assigned to each grayscale value. Welsh et al. [2] proposed techniques where rather than choosing colors from a palette to color individual components, the color is transferred from a source color image to a target grayscale image by matching luminance and texture information between the images. This approach is inspired by a method of color transfer between images described in Reinhard et al. [3] and image analogies by Hertzmann et al. [4]. Another well known approach to colorization [5] assumes that small changes take place between two consecutive frames; therefore, it is possible to use optical flow to estimate dense pixel to pixel correspondences. Chromatic information can then be transferred directly between the corresponding pixels. There are some approaches [6], [7], [8] which make use of the assumption that the homogeneity in the gray-scale domain indicates homogeneity in the color domain and vice versa. This assumption provides a possibility to propagate color from several user-defined seed pixels to the rest of the image. In [9], colorization is done through luminance-weighted chrominance blending and fast intrinsic distance computations. Shi et al. [10] color the grayscale images by segmentation and color filling method, where an image is first segmented into regions and then the desired colors are Used to fill each region. Since the existing automatic image segmentation algorithms usually cannot segment the image into meaningful regions, only color filling of each segmented region cannot produce natural colorized results. Sykora et al. [11] suggested using unsupervised image segmentation in cartoons colorization. However the method usually cannot get ideal results for other types of images and is restricted to only cartoons. A major difficulty with colorization, however, lies in the fact that it is an expensive and time-consuming process. For example, in order to colorize a still image an artist typically begins by segmenting the image into regions, and then proceeds to assign a color to each region. Unfortunately, automatic segmentation algorithms often fail to correctly identify fuzzy or complex region boundaries, such as the boundary between a subject's hair and her face. Thus, the artist is often left with the task of manually delineating complicated boundaries between regions. Colorization of movies requires, in addition, tracking regions across the frames of a shot. Existing tracking algorithms typically fail to robustly track non-rigid regions, again requiring massive user intervention in the process.

2. Algorithm

The first step in colorizing the gray level images is to remove noise and perform threshold operation on images so that colorization is done accurately. If the primary picture is similar to fig.1, the figure histogram needs to be examined carefully presented in fig. 2.

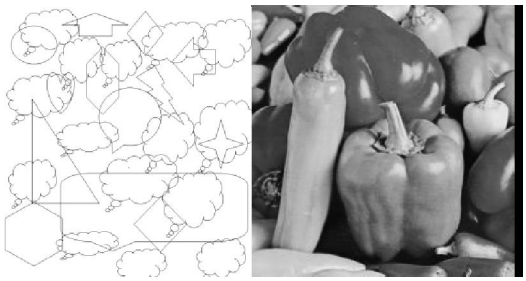


Figure (1): Gray level main image to colorize

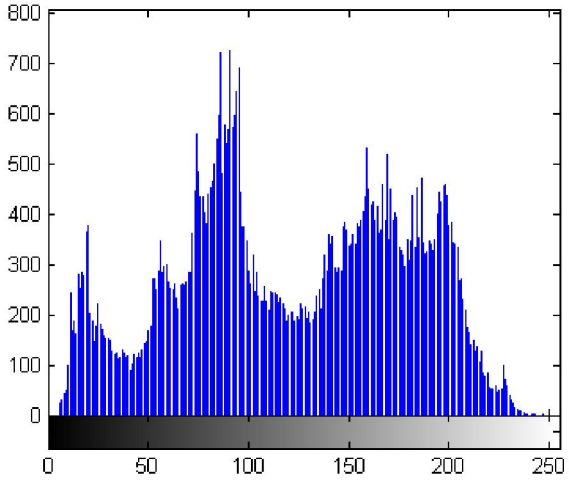


Figure (2): Histogram of the primary figure

As you observe in fig.2, histogram is not even so we use equalization.

3. Histogram Equalization

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal. Histogram equalization often produces unrealistic effects in photographs; however it is very useful for scientific images like thermal, satellite or x-ray images, often the same class of images that user would apply false-color to. Also histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color depth. For example, if applied to 8-bit image displayed with 8-bit gray-scale palette it will further reduce color depth (number of unique shades of gray) of the image. Histogram equalization will work the best when applied to images with much higher color depth than palette size, like continuous data or 16-bit gray-scale images.

To transfer the gray levels so that the histogram of the resulting image is equalized to be a constant:

H [i] =constant for all i

The purposes:

To equally use all available gray levels; for further histogram specification. (Fig.3)

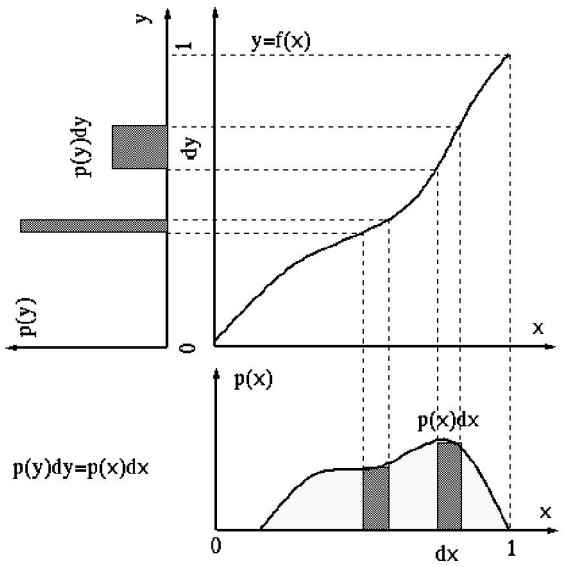


Figure (3): This figure shows that for any given mapping function y=f (x) between the input and output images

The following holds:

$$P(y)dy = p(x)dx \quad (1)$$

i.e., the number of pixels mapped from x to y is unchanged.

To equalize the histogram of the output image, we let p (y) be a constant. In particular, if the gray

levels are assumed to be in the ranges between 0 and 1 ($0<x\leq1$, $0<y\leq1$), then p (y) =1. Then we have:

$$dy=p(x)dx \text{ or } dy/dx=p(x) \quad (2)$$

i.e., the mapping function y=f (x) for histogram equalization is:

$$y = f(X) = \int_0^x p(u)du = p(x) - p(0) = p(x) \quad (3)$$

Where

$$p(x) = \int_0^x p(u)du \ , p(0) = 0 \quad (4)$$

Is the cumulative probability distribution of the input image, which monotonically increases. Intuitively, histogram equalization is realized by the following: If p (x) is high, P (x) has a steep slope, dy will be wide, causing p (y) to be low to keep p (y)dy=p (x)dx; If p (x) is low, P (x) has a shallow slope; dy will be narrow, causing p (y) to be high.

For discrete gray levels, the gray level of the input x takes one of the L discrete values: $x \in \{0,1,2,..,L-1\}$ and the continuous mapping function becomes discrete:

$$\acute{y} = f[x] \triangleq \sum_{i=0}^{x} h[i] = H[x] \quad (5)$$

Where h [i] is the probability for the gray level of any given pixel to be I ($0\leq i\leq L-1$):

$$h[i]= \frac{n_i}{\sum_{i=0}^{L-1} n_i} = \frac{n_i}{N}$$

And

$$\sum_{i=0}^{L-1} h[i] = 1 \quad (6)$$

Of course here h [i] is the histogram of the image and H [i] is the cumulative histogram.

The resulting function $\acute{y}$ is in the range $0\leq \acute{y} \leq 1$ and it needs to be converted to the gray levels $0 \leq y \leq L-1$ by either of the two ways:

$$y = [\acute{y}(L-1) + 0.5] \quad (8)$$

$$y = \left[\frac{y - \acute{y}_{min}}{1 - \acute{y}_{min}}(L-1) + 0.5\right]$$

Where [x] is the floor, or the integer part of a real number x, and adding 0.5 is for proper rounding. Note that while both conversions map $\acute{y}_{max} = 1$ to the highest gray level L-1, the second conversion also maps $\acute{y}_{min}$ to 0 to stretch the gray levels of the output image to occupy the entire dynamic range $0\leq Y<L-1$.

The result is shown in fig.4.

Figure (4): Image equalization

We work in YUV color space, commonly used in video, where Y is the monochromatic luminance channel, which we will refer to simply as intensity, while U and V are the chrominance channels, encoding the color [Jack 2001].

The algorithm is given as input an intensity volume Y (x; y; t) and outputs two color volumes U (x; y; t) and V (x; y; t). To simplify notation we will use boldface letters (e.g. r; s) to denote (x; y; t) triplets. Thus, Y (r) is the intensity of a particular pixel. As mentioned in the introduction, we wish to impose the constraint that two neighboring pixels r; s should have similar colors if their intensities are similar. Thus, we wish to minimize the difference between the color U (r) at pixel r and the weighted average of the colors at neighboring pixels:

$$J(U) = \sum_r (U(r) - \sum_{s\in N(r)} w_{rs} U(s))^2 \quad (9)$$

Where w_{rs} is a weighting function that sums to one, large when Y (r) is similar to Y (s), and small when the two intensities are different. Similar weighting functions are used extensively in image segmentation algorithms (e.g. [Shi and Malik 1997; Weiss 1999]), where they are usually referred to as affinity functions. We have experimented with two weighting functions. The simplest one is commonly used by image segmentation algorithms and is based on the squared difference between the two intensities:

$$w_{rs} \propto e^{-(Y(r)-Y(s))^2/2\sigma_r^2} \quad (10)$$

A second weighting function is based on the normalized correlation between the two intensities:

$$w_{rs} \propto 1 + \frac{1}{\sigma_r^2}(Y(r) - \mu_r)(Y(s) - \mu_r) \quad (11)$$

Where μ_r and σ_r are the mean and variance of the intensities in a window around r.

The correlation affinity can also be derived from assuming alocal linear relation between color and intensity [Zomet and Peleg 2002; Torralba and Freeman 2003]. Formally, it assumes that the color at a pixel U (r) is a linear function of the intensity Y (r):

U (r) = aiY (r) +bi and the linear coefficients ai, bi are the same for all pixels in a small neighborhood around r. This assumption can be justified empirically [Zomet and Peleg 2002] and intuitively it means that when the intensity is constant the color should be constant, and when the intensity is an edge the color should also be an edge (although the values on the two sides of the edge can be any two numbers). While this model adds to the system a pair of variables per each image window, a simple elimination of the ai, bi variables yields an equation equivalent to equation 1 with a correlation based affinity function. The notation r €N (s) denotes the fact that r and s are neighboring pixels. In a single frame, we define two pixels as neighbors if their image locations are nearby. Between two successive frames, we define two pixels as neighbors if their image locations, after accounting for motion, are nearby. More formally, let vx (x, y), vy (x,y) denote the optical flow calculated at time t. Then the pixel (x0, y0, t) is a neighbor of pixel (x1, y1, t +1) if:

$$\left\| \left(x_0 + v_x(x_0), y_0 + v_y(y_0)\right) - (x_1, y_1) \right\| < T \qquad (12)$$

The tow field $v_x(x0), v_y(y0)$ is calculated using a standard motion estimation algorithm [Lucas and Kanade 1981]. Note that the optical flow is only used to define the neighborhood of each pixel, not to propagate colors through time. Now given a set of locations ri where the colors are specified by the user u (ri) = ui, v (ri) = vi we minimize J (U), J (V) subject to these constraints. Since the cost functions are quadratic and the constraints are linear, this optimization problem yields a large, sparse system of linear equations, which may be solved using a number of standard methods. Our algorithm is closely related to algorithms proposed for other tasks in image processing. In image segmentation algorithms based on normalized cuts [Shi and Malik 1997], one attempts to find the second smallest eigenvector of the matrix D -W where W is a n pixels×npixels matrix whose elements are the pair wise affinities between pixels (i.e., the r; s entry of the matrix is wrs) and D is a diagonal matrix whose diagonal elements are the sum of the affinities (in our case this is always 1). The second smallest eigenvector of any symmetric matrix A is a unit norm vector x that minimizes xTAx and is orthogonal to the first eigenvector. By direct inspection, the quadratic form minimized by normalized cuts is exactly our cost function J, that is xT (Dj-W)x = J (x). Thus, our algorithm minimizes the same cost function but under different constraints. In image denoising algorithms based on anisotropic diffusion [Perona and Malik 1989; Tang et al. 2001] one often minimizes a function similar to equation 1, but the function is applied to the image intensity as well.

4. Edge removing

In the provided method the kind of edge removing is very significant. The more edge vector segmentation, the more details must be presented on the image color. Because of this reason, SOBEL algorithm is utilized. With regard to type of the edge vector, the desired colors are put on the image (fig.5).

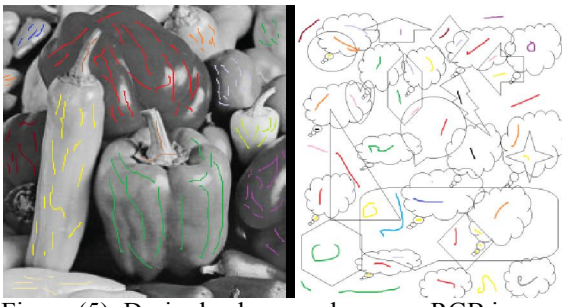


Figure (5): Desired colors are drawn on RGB image.

The result of colorization can be observed in fig.6.

Figure (6): Colorized image on RGB image

Looking accurately at the image, we can notice noises and disturbances on the image that should be reduced and minimized so a middle filter is used for this purpose that is a middle mask presented in figure 7 is applied for each color.

$$\frac{1}{9}\times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Figure (7): Low- pass filter, a quiet plaice

The results obtained from this filter are illustrated in figure 8.

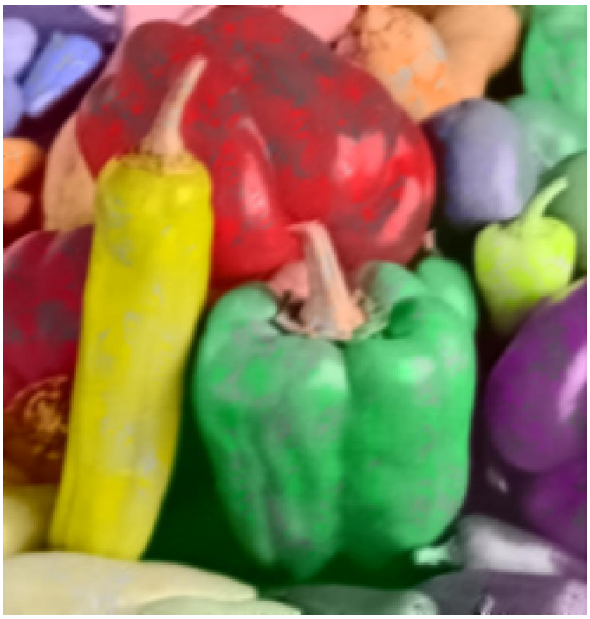
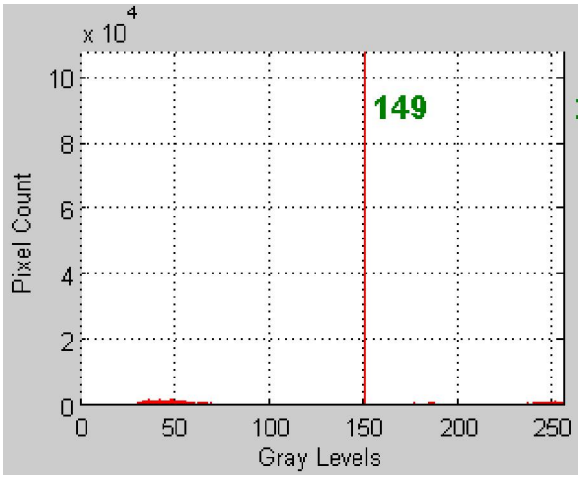


Figure (8): The reduction of disturbance on the colorized image

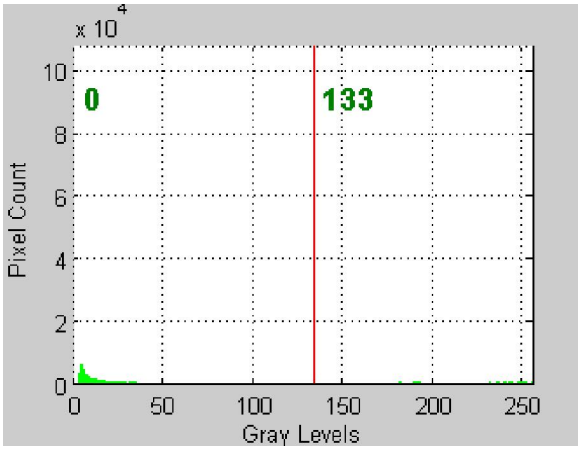
5. Neural network

In the previous section, the gray level pictures were converted into RGB, but here we want the human face pictures to be colored automatically [12, 13]. In this method, the primary picture which is used for training the neural network is broken down [14]. It is broken down into main RGB colors [15]. Figure 9 shows the tested sample and figure 10 shows this break down.

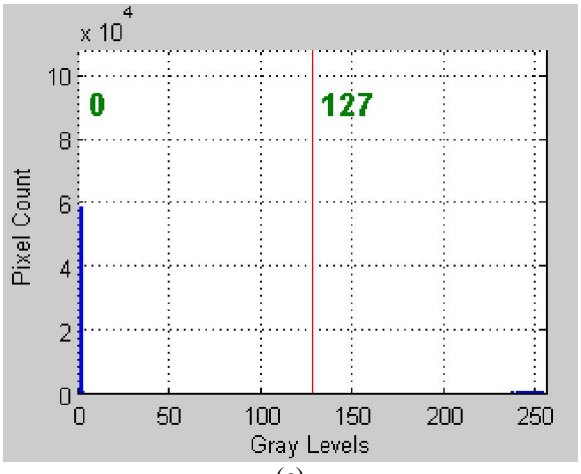
Figure (9): The result from colorization on the image



(a)



(b)



(c)

Figure (10): (a) Histogram of red band. (b) Histogram of green band. (c) Histogram of blue band

The applied algorithm is shown in figure 11. [16]

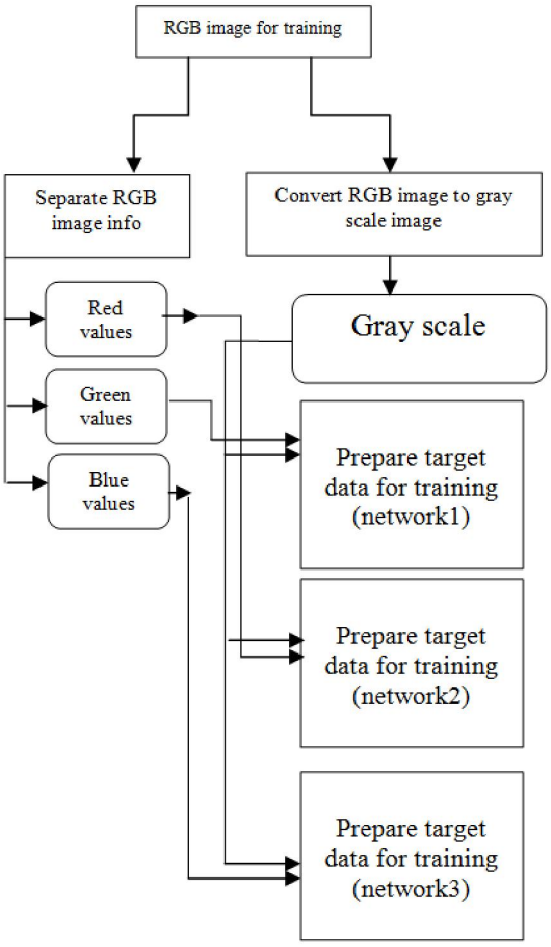


Figure (11): The applied algorithm for neural network in coloring

Now, first, we color a picture which was originally in gray level then insert it into neural network, in order to train the neural network. The picture which is used for coloring is shown in figure 12:

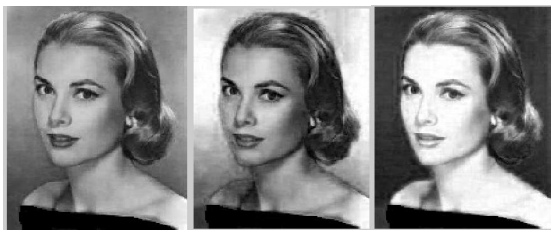
Figure (12): The primary picture of gray level type

Now, by using the figure 13-a pattern, the coloring is carried out and the result is shown in figure 13-b.

(a) (b)

Figure (13): (a) picture coloring pattern; and (b) the result of coloring.

Now according to what stated previously, the figure 13b is then broken down, and the obtained results are shown in figure 14.



(a) (b) (c)

Figure (14): (a) green band. (b) Blue band. (c) Red band

Then this break down is prepared and inserted along with the colored picture to train the network. To test the network, another picture, such as figure 15-a, was inserted into the trained network and the results are shown in figure 15-b.

(b) (a)

Figure (15): (a) a gray level type picture inserted to test the neural network; and (b) the picture obtained from coloring through neural network.

It should be noted that the pictures used for testing the neural network, have to be similar to

trained pictures regarding their physical structure, otherwise there would be improper pictures.

Conclusion

In this paper, the gray level pictures were converted into color pictures, which was first carried out by a pre-defined pattern but later it was performed automatically by a neural network.

Reference

1. R.C.Gonzalez and R.E. Woods, Digital Image Processing (second ed.), AddisonWesley Publishing, Reading, MA (1987).
2. T. Welsh, M. Ashikhmin and K. Mueller, Transferring color to greyscale images, in: ACM SIGGRAPH 2002 Conference Proceedings (2002) pp. 277-280.
3. E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, Color transfer between images, IEEE Transactions on Computer Graphics and Applications 21 (2001) (5), pp. 34-41.
4. A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless and D.H. Salesin, "Image analogies", ACM SIGGRAPH 2001 Conference Proceedings, 327-340(2001).
5. Z. Pan, Z. Dong and M. Zhang, A new algorithm for adding color to video or animation clips, in: Proceedings of WSCG International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (2004) pp. 515-519.
6. T. Horiuchi, Estimation of color for gray-level image by probabilistic relaxation, in: Proceedings of IEEE International Conference on Pattern Recognition (2002) pp. 867-870.
7. T. Horiuchi and S. Hirano, Colorization algorithm for grayscale image by propagating seed pixels, in: Proceedings of IEEE International Conference on Pattern Recognition (2003) pp. 457- 460.
8. A. Levin, D. Lischinski and Y. Weiss, Colorization using optimization, in: ACM SIGGRAPH 2004 Conference Proceedings (2004) pp. 689-694.
9. Liron Yatziv and Guillermo Sapiro, Fast image and video colorization using chrominance blending, in: IEEE Transactions on Image Processing, Vol. 15, No. 5, May 2006, pp. 1120-1129.
10. Shi, J., Malik, J., 1997. Normalized cuts and image segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 731-737.
11. Sy´kora, D., Buria´nek, J., Zara, J., 2003. Segmentation of Black and White Cartoons, In: Proceedings of Spring Conference on Computer Graphics, pp. 245-254.
12. changjiang, Z., X.Wang and H.Zhang,2006. Application of a neural network to automatic gray level adjustment for medical images. Networking, Sensing and Control, ICNSC, pp: 1064 – 1069.
13. Luiz, F., M. Vieira, 2003. Automatically choosing source color images for coloring grayscale images. In Proc. SIBGRAPI'03, pp.1530-1538.
14. karthikeyani.V, K. Duraiswamy and P.Kamalakkannan, 2007. Conversion of Gray-scale image to Color Image with and without Texture Synthesis, In Proc. IJCSNS, v.7, no.4, pp.11-16.
15. Rybak, I.A., A.V. Golovan, V.I. Gusakova, L.N. Podladchikova, N.A.Shevtsova,1992. A neural network system model for active perception and invariant recognition of grey-level images. Neural Networks. IJCNN, PP: 1-6.
16. Karlk, B.; Saroz, M., Coloring gray-scale image using artificial neural networks, Adaptive Science & Technology, 2009. 2nd International Conference on, pp.366–371.

10/13/2017