

## Human Perception Based Color Image Segmentation

Neeta Gargote<sup>1</sup>, Savitha Devaraj<sup>2</sup>, Shravani Shahapure<sup>3</sup>

<sup>1,2</sup> *Department of Electronics Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India.*

<sup>3</sup> *Department of Electronics and Telecommunication Engineering, PIIT, New Panvel, India.*

*Email: nita\_9000@yahoo.com<sup>1</sup>, saitha82@gmail.com<sup>2</sup>*

### ABSTRAKSI

Segmentasi citra warna mungkin merupakan tugas yang paling penting dalam analisis citra dan pemahaman. Sebuah sistem segmentasi gambar warna berdasarkan persepsi manusia baru disajikan dalam makalah ini. Sistem ini menggunakan arsitektur jaringan syaraf. Syaraf di sini menggunakan fungsi aktivasi sigmoid jamak. Fungsi aktivasi sigmoid jamak adalah kunci untuk segmentasi. Jumlah langkah yaitu ambang batas dalam fungsi sigmoid jamak tergantung pada jumlah cluster dalam gambar. Nilai ambang batas untuk mendeteksi klaster dan label mereka ditemukan secara otomatis dari urutan turunan pertama dari histogram kejenuhan dan intensitas dalam ruang warna HSI. Dalam hal ini penggunaan utama dari jaringan saraf adalah untuk mendeteksi jumlah objek secara otomatis dari gambar. Ini label-label objek dengan rata - rata warna mereka. Algoritma ini yang dihasilkan dapat diandalkan dan bekerja memuaskan pada berbagai jenis gambar berwarna.

**Kata Kunci:** Segmentasi citra, Persepsi Manusia, Jaringan Syaraf, Aktivasi Sigmoid Jamak, Fungsi Histogram

### ABSTRACT

Color image segmentation is probably the most important task in image analysis and understanding. A novel Human Perception Based Color Image Segmentation System is presented in this paper. This system uses a neural network architecture. The neurons here uses a multisigmoid activation function. The multisigmoid activation function is the key for segmentation. The number of steps ie. thresholds in the multisigmoid function are dependent on the number of clusters in the image. The threshold values for detecting the clusters and their labels are found automatically from the first order derivative of histograms of saturation and intensity in the HSI color space. Here the main use of neural network is to detect the number of objects automatically from an image. It labels the objects with their mean colors. The algorithm is found to be reliable and works satisfactorily on different kinds of color images.

**Keywords:** Image Segmentation, Human Perception, Neural Network, Multisigmoid Activation, Histogram Function

## 1. INTRODUCTION

The process of partitioning a digital image into multiple regions (set of pixels) is called image segmentation[1]. The partitions are different objects in image which have the same texture or color. The result of the image segmentation is a set of regions that collectively cover the entire image. All of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics. Some of practical applications of image segmentation are: image processing, computer vision, face recognition, medical imaging, digital libraries, image and video retrieval [11]. Different color spaces like HSI, RGB, YIQ, etc. have been suggested in image processing, each suitable for different domains[2]. HSI color space is used because a color in this space is represented in three dimensions: one which codes the color itself (H) and another two which explain details of the color, saturation (S) and intensity (I). The color perceived by human is a combination of three color stimuli such as red (R), green (G), and blue (B), which forms a color space.

There are two types of Image segmentation

1. Binerisation (constant threshold)
2. Adaptive thresholding (variable threshold)

### 1. Binerisation :

Image is an aggregate of pixels which refers as spatial domain. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression  $g(x,y)=T[f(x,y)]$ ; where,  $f(x,y)$  is the input image,  $g(x,y)$  is the processed image and  $T$  is an operator on  $f$ , defined over some neighborhood of  $(x,y)$ .

In addition,  $T$  can operate on a set of input images, such as performing the pixel by pixel sum of  $K$  images for noise reduction. The operator  $T$  is applied at each location  $(x,y)$  to yield the output  $g$ , at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.

The simplest form of  $T$  is when the neighborhood of size  $1 \times 1$  (i.e. a single pixel). In this case  $g$  depends on only the value of  $f$  at  $(x,y)$ , and  $T$  becomes a gray-level (also called an intensity or mapping) transformation function of the form  $s=T(r)$ ; where  $r$  is the gray level of  $f(x,y)$  at any

point  $(x,y)$ ,  $s$  is the gray level of  $g(x,y)$  at any point  $(x,y)$  and  $T$  is the transformation that maps a pixel value  $r$  into a pixel values.

The values of pixels before processing will be denoted by  $r$  and after processing denoted by  $s$ . The effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below certain point say  $m$  and brightening the levels above  $m$  in the original image. This technique is known as contrast stretching. The values of  $r$  below  $m$  are compressed by the transformation function into a narrow range of  $s$ , toward black. The opposite effect takes place for values of  $r$  above  $m$ . A mapping of this form is called a thresholding function. And  $T(r)$  produces a two level (binary) image. This is also called as binerisation.

Suppose that the image,  $f(x,y)$ , composed of light objects on a dark background, in such a way that object and background pixels have gray levels grouped into two dominant modes. So to extract objects from the background, threshold  $T$  is to be selected that separates these modes. Then any point  $(x,y)$  for which  $f(x,y) > T$  is called an object point or background point. In this way thresholding can be done.

## 2. Adaptive thresholding:

If  $T$  depends on the spatial coordinates  $x$  and  $y$ , the threshold is called dynamic or adaptive.

If an image with two types of light objects on a dark background is there. Then we can select a point  $(x,y)$  as  $T1 < f(x,y) \leq T2$  belonging to one object class and  $f(x,y) > T2$  belonging to the other object class and  $f(x,y) \leq T1$  to the background in multilevel or adaptive thresholding.

Thresholding may be viewed as an operation that involves tests against a function  $T$  of the form  $T = T[x,y,p(x,y),f(x,y)]$ , where  $f(x,y)$  is the gray level of point  $(x,y)$  and  $p(x,y)$  is some local property of this point

When  $T$  depends only on  $f(x,y)$  i.e. only on gray level values the threshold is called global. If  $T$  depends on both  $f(x,y)$  and  $p(x,y)$ , the threshold is called local.

Both the methods are simple and fast. But these methods work only for bimodal histogram and does not give multiple segments.

Despite the large number of image segmentation algorithms available, no general methods have been found to process the wide diversity of images encountered in real world applications. Robust color image segmentation algorithm is still a very challenging research topic.

Here in this paper we have presented an Adaptive Neuro System for color image segmentation. Till the time lot of systems are proposed on gray level or monochrome images but on color image very few technologies are available. Here we have presented a novel Human Perception Based Color Image Segmentation System for color image segmentation. The proposed system uses a neural network[3] with architecture similar to the multilayer perceptron (MLP) network. The main difference is that neurons here use a multisigmoid activation function. The multisigmoid function is the key for segmentation. The number of steps i.e. thresholds in the multisigmoid function are dependant on the number of clusters in the image. The threshold values for detecting the clusters and their labels are found automatically from the first order derivative of histograms of saturation and intensity in the HSI color space. Here, the main use of neural network is to detect the number of objects automatically from an image. The advantage of this method is that no a prioriknowledge is required to segment the color image. It labels the objects with their mean colors.

## 2. HISTOGRAM THRESHOLDING ALGORITHM

This algorithm is used to find out number of clusters and to compute the multi-level sigmoid function for the neurons. It is crucial to determine the number of clusters in an image so as to segment the objects appropriately. The main endeavour here is to find number of clusters without a prioriknowledge of the image. To achieve this, first the histograms of given color image for saturation and intensity planes are found out. from the derivative curve of histogram we find the clusters in saturation and intensity planes independently. A group of saturation/intensity values with histogram derivative transiting from negative to positive (zero crossing) and subsequent positive to negative (zero crossing) is considered into one cluster. Similarly other clusters are found out with the help of zero crossings. The example is as shown in Figure 1. The threshold value is the first zero crossing (shown in Figure 1) and average of two subsequent zero crossings is considered as label (target). The average value as a target helps to segment the object with a color appropriate to its original color. Hence in this system objects are colored with their mean color i.e. system tries to maintain the color property of the object even after segmentation. This can be helpful in image post-processing.

Once threshold and target values are calculated, a neural network activation function is constructed as in Equation 1.

$$f(x) = \sum_k \left( \frac{y_k - y_{k-1}}{1 + e^{-\frac{(x - \theta_k)}{\theta_0}}} + y_{k-1} \right) \times [\mu((x - y_{k-1}) \times d^2) - \mu((x - y_k) \times d^2)] \quad (1)$$

Where  $\mu$       Step function  
 $\theta_k$       Thresholds

- $y_k$  Target level of each sigmoid, will constitute the system labels  
 $\theta_o$  Steepness parameter  
 $d$  Size of neighborhood.

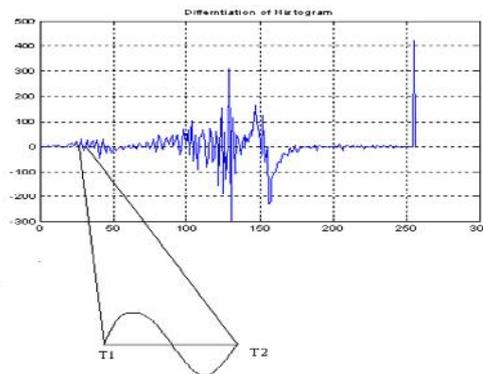


Fig 1 First Order Derivative with Threshold

### 3. NEURALNETWORKALGORITHM

This system consists of two independent neural networks one each for saturation and intensity planes. The neural network consists of three layers namely input layer, hidden layer, and output layer as depicted in Figure 2. The input to a neuron in the input layer is normalized between [0-1]. The output value of each neuron is between [0-1]. Each layer is having a fixed number of neurons equal to the size (I x J) of the image. All neurons are having primary connection weight as 1. Each neuron in one layer is connected to respective neuron in the previous layer with its  $d$ th order neighborhood. There are no connections between neurons in the same layer. Neural network tuning block is used to update the connection weight as in Equation 2 and 3 by taking into consideration the output error in network. At every training epoch, the error is calculated by taking difference between the actual output and the desired output of neuron.

$$\Delta W_{ji} = n \left( \frac{-\partial E}{\partial O_j} \right) \frac{\partial O_j}{\partial I_j} O_i \quad (2) \quad \text{For Output Layer}$$

$$\Delta W_{ji} = n \left( \sum_k k \left( -\frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial I_k} w_{kj} \right) \right) \frac{\partial O_j}{\partial I_j} O_i \quad (3) \quad \text{For Hidden Layer}$$

Where

- $I_j$  Total input to the  $j^{\text{th}}$  neuron  
 $O_j$  Output of the  $j^{\text{th}}$  neuron  
 $W_{ji}$  Weight of link from neuron  $i$  in one layer to neuron  $j$  in the next layer  
 $O_i$  Output of the  $i^{\text{th}}$  neuron  
 $E$  Error in the network's output

$n$  Learning rate.

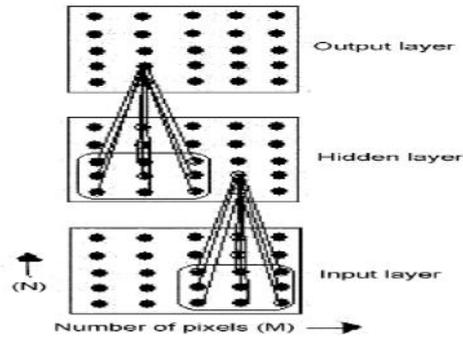


Fig 2 Neural Network Architecture

As the training progresses, a pixel gets the color depending upon its surrounding pixel colors.

#### 4. FLOWCHART

The flow chart of the complete algorithm is as given below.

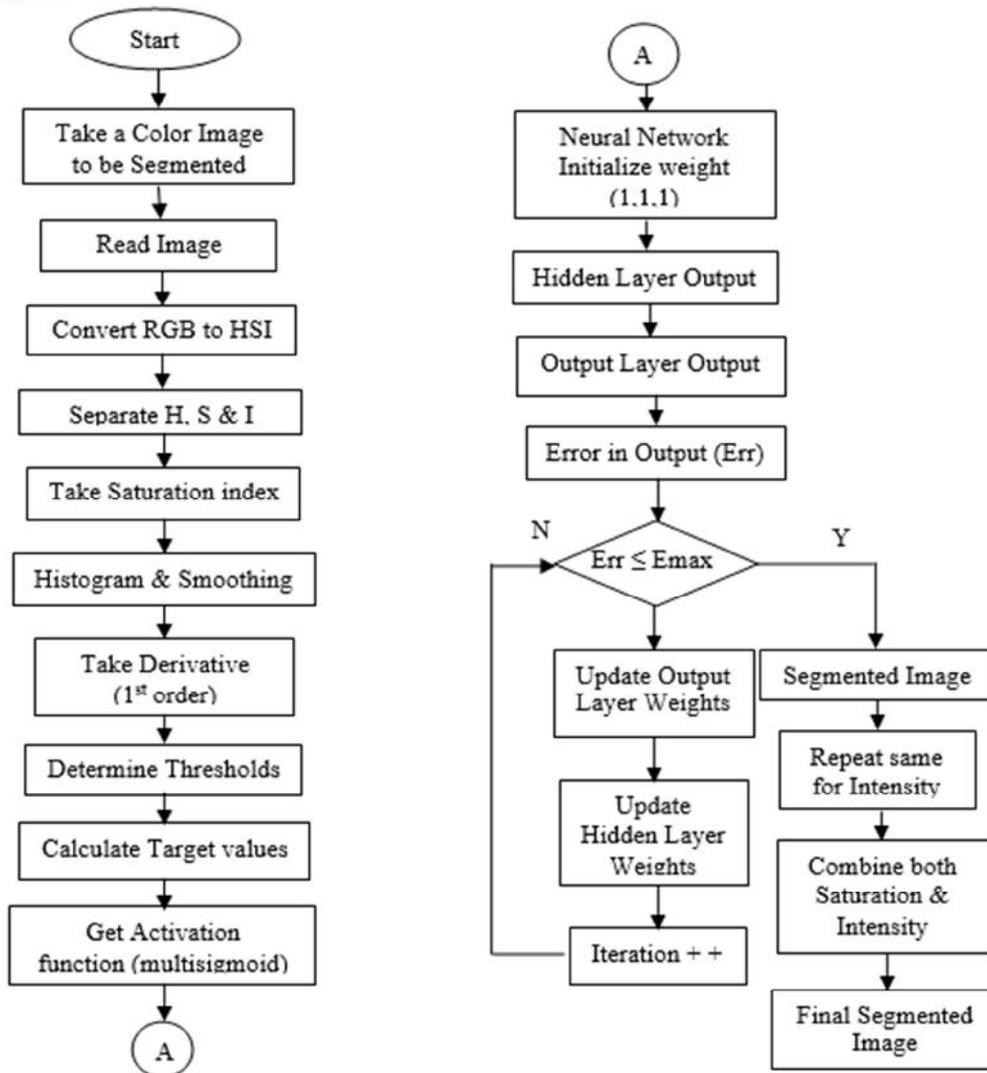


Fig 3 Complete Flow Chart of the Segmentation System

Here we have taken RGB image, converted to HSI and then processed on it. In the algorithm we have separated the hue, saturation and intensity part. From this we have taken the saturation part first.

Above figure shows the detailed flow chart of the segmentation system. In this system we calculated the segmented image of saturation and intensity independently. Lastly we combine the effects of both saturation and intensity to get the Segmented output image. The system has proven to be robust and real time applicable.

### 3. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in matlab environment on a Intel ‘Celeron’ M Processor, 1.73 GHz, 533 MHz RAM. For all the experiments, the proposed method uses a second order (3 x 3) neighborhood scheme for neuron connection scheme. A third order FIR (Finite impulse response) filter with cutoff frequency 0.5 is used to smooth the histogram. Experimentally it is found that 0.5 is a suitable cutoff frequency value for most of the images. The segmentation results for the different figures are depicted. It can be observed from the experimental results that without a priori knowledge system could isolate the objects properly and are labeled with their mean colors.

Here we have taken RGB image of pepper and processed on them. Initially we have converted RGB to HSI image. Then separated hue, saturation and intensity part of it. After that we have taken intensity part and processed on it. We had gone upto 2 iterations. Below is the experimental result of 2 iterations is shown. We can see that the value or error function goes reducing.

RGB image converted to HSI image whose saturation part is as shown in figure 4. Figure 4(a) shows the histogram of an image, 4(b) shows the smoothed histogram, 4(c) shows the differentiation of smoothed histogram, 4(d) shows multisigmoid activation function 4(e) shows the segmentation results of its saturation components. Similarly the results for intensity components of an image are shown in figure 5(a) to 5(e). Final result of segmentation is shown in figure 6(b) which is combination of 4(e) and 5(e) as well as the hue part of the image.

Segmentation results for different images as shown in figure 7(a) to 10(a) are as shown in figure 7(b) to 10(b).

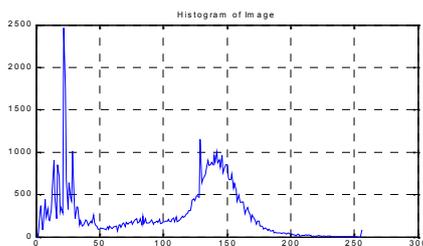


Fig 4 (a) Histogram of Image  
Histogram

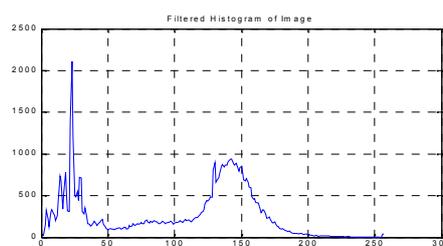


Fig 4(b) Smoothed

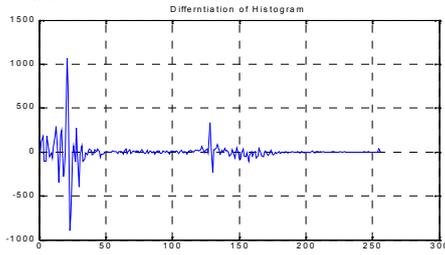


Fig 4 (c) Differentiation of Histogram

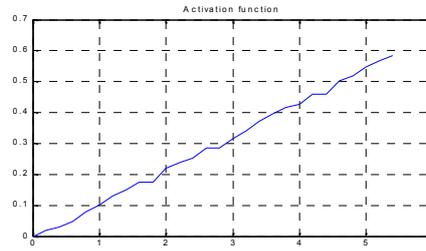


Fig 4(d) Activation Function

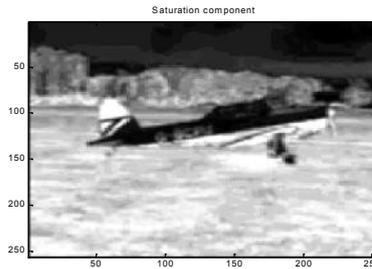


Fig 4 (e) Segmented Image

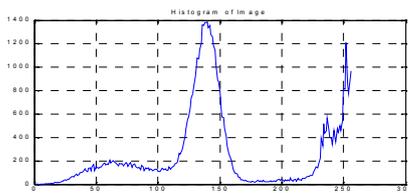


Fig 5 (a) Histogram of Image

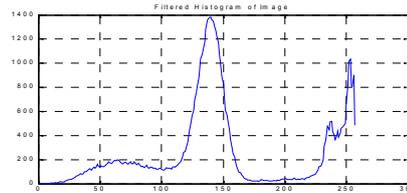


Fig 5(b) Smoothed Histogram

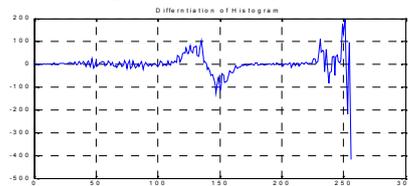


Fig 5 (c) Differentiation of Histogram

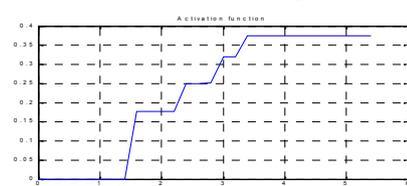


Fig 5(d) Activation Function

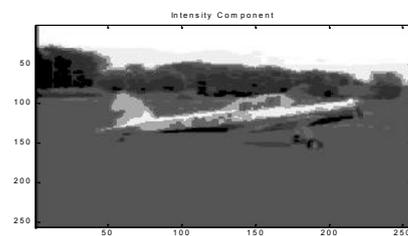


Fig 5 (e) Segmented Image



Fig 6 (a)Original Image (b)Segmented Image

**TABLE 1**

Description of the errors in the segmentation:

Image	Saturation Component		Intensity Component	
	Error1	Error2	Error1	Error2
Aeroplane	6.3591e-004	6.3538e-004	0.0060	0.0060
Bird	2.4201e-005	2.4096e-005	2.2126e-004	2.2126e-004
Multicolor	0.0057	0.0057	0.0293	0.0289
Structure	1.2908e-004	1.2908e-004	0.1177	0.1176
Hand	2.1228e-005	2.1228e-005	0.0029	0.0029



Fig 7 (a)Original Image (b)Segmented Image

Fig 8 (a)Original Image (b)Segmented Image

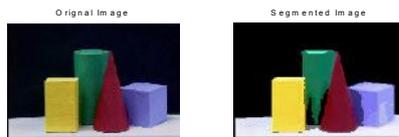


Fig 9 (a)Original Image (b)Segmented Image

Fig10(a)Original Image (b)Segmented Image

#### 4. CONCLUSION

A novel system for color image segmentation based on adaptive thresholds is described. The segments in images are found automatically based on adaptive multilevel threshold approach. One of the advantages of this system is that it does not require a priori information about the number of objects in the image. The use of first order derivative of histogram is found to be a powerful method to find clusters in the image. The neural network with multisigmoid function tries to label the objects with its original color even after segmentation. This Segmentation

system is tested on several images and its performance was found satisfactory. The system can be used as a primary tool to segment unknown color images. Experimental result show that the system can segment the heavily multicolor textured object from multicolored background satisfactorily. Also as the network once trained do not need to train again and again the system found to be simple and with real time.

## REFERENCES

- [1] R. C. Gonzalez, R.E. Woods *Digital Image Processing*, Pearson Education Pet. Ltd. New Delhi.
- [2] A.K. Jain, *Fundamentals of Digital Image Processing*, Upper Saddle River, NJ: Prentic Hall, 1989.
- [3] Simon Haykin, *Neural Networks A Comprehensive Foundation*, Pearson Education, 1999.
- [4] J.S. Weszka, "A Survey of threshold selection techniques", *Computer Graphics Image Process*, 7:259-265, 1978.
- [5] Jacek M. Zurada, *Introduction to Artificial Neural Systems*, Jaico publishing house, Mumbai, 2002.
- [6] K.S.Deshmukh, G.N. Shinde, A.V. Nandedkar, Y.V. Joshi, "Multilevel approach for color image segmentation", *Fourth Indian Conf. on Computer Vision, Graphics and Image Processing*, 338-344, 2004.
- [7] V. Boskovitz, Hugo Guterman, "An adaptive neuro fuzzy system for automatic image segmentation and edge detection", *IEEE Transaction fuzzy systems*, 10(2): 247-262, 2002.
- [8] Xiu-Qi Li, Shu-Ching Chen, Mei-Ling Shyu, Borko Furht, "An efficient content-based visual image retrieval system", *IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, 2002.
- [9]. S.K. Pal, A. Rosenfeld, "Image enhancement and thresholding by optimization of fuzzy compactness", *Pattern Recognition Letter*, 7: 77-86, 1988.
- [10]. A. A. Younes, I. Truck, and H. Akdaj, Color Image Profiling Using Fuzzy Sets, *Turk J Elec Engin*, vol. 13, no. 3, 2005.
- [11] Computer Vision Test Images, <http://www.cs.cmu.edu/~cil/v-images.html>

- [12] Manolis Lourakis, "ECVNET Image Database",  
<http://zeus.ics.forth.gr/forth/ics/cvrl/proj/ecvnet/imagedb>