

Colour Based Segmentation Using Image Processing

Chaitra K G

M.Tech II SEM (CSE)

Department of Computer Science Engineering

University B.D.T. College of Engineering, Davanagere, Karnataka, India

(A Constituent College of Visvesvaraya Technological University, Belagavi)

Mohammed Rafi

Department of Computer Science Engineering

University B.D.T. College of Engineering, Davanagere, Karnataka, India

(A Constituent College of Visvesvaraya Technological University, Belagavi)

Abstract

This paper gives an approach to recognize colours in a three dimensional image using colour thresh-holding technique using python with the help of RGB colour model to detect a selected colour by a user in an image. The methods involved for the detection of colour in images are conversion of three dimensional RGB image format and here we are using python so, it should converted into BGR image format.using connected components labelling to detect connected regions in binary digital images and use of bounding box and its properties for calculating the metrics of each labelled region. Further the colour of the pixels is recognized by analysing the RGB values for each pixel present in the image. The algorithm k-means is implemented using machine learning and image processing. The results of this implementation can be used in applications like find the colour of leaf or fruits or vegetablesto know the maturity level of the respected object, segregation of objects based on their colours, intrusion detection.

Keywords: Image processing ,Machine Learning, color detection, RGB image, k-means algorithm.

INTRODUCTION

Colour is one of the most important characteristics of an image, if colour in a live video or in a digital image can be detected, then the results of this detection can be used in various industrial and scientific applications. Colour detection is the fundamental step in many computer vision systems. In this paper Image Processing and machine learning is used for detection of a particular colour in a given image. Image Processing Toolbox provides wide variety of referenced algorithms, methods and applications for image processing, visualization and segmentation. An image is a matrix of pixel values. An image can be represented using many colour models like gray-scale, RGB, HSV etc. Here RGB model is used to detect the colours in an image [4][5]. RGB model is a colour model in which red, green and blue lights are added together in various ways to produce wide range of colours. In RGB model image is composed of a matrix of $M \times N \times 3$ pixels with M rows and N columns of pixels for each red, green and blue colour components of an image. The RGB model uses 8 bits for representation of red, green and blue colour components of an image. Since in an RGB model image is represented in the form of a

matrix, mathematical operations on the image can be applied to detect a particular colour in an image.

LITERATURE SURVEY

In Colour Managing the Third Edition of Billmeyer and Saltzman's Principles of Color Technology, Roy S. Berns, D. Mark Reiman says [1]

This paper aims to Commercial ICC-compliant color-management software was used to produce color-managed CMYK-encoded images for the third edition of Principles of Color Technology. Custom profiles were created for a Scitex Eversmart Pro flatbed scanner and a Kodak Approval proofing device. This enabled objects such as color-order systems and colorimetric-encoded, computer-generated graphical images to be reproduced with reasonable colorimetric accuracy. The GretagMachbeth ColorChecker Color Rendition Chart was used as an independent verification target. Its printed reproduction had an average error of 4.2×10^{-94} (6.4×10^{-ab}). Colorimetric-rendering device profiles enabled the visualization of the book's color gamut and of a calibrated visible.

In Specific Color Detection in Images using RGB Modelling in MATLAB, Visheh Goel, Tarun Jain, Sahil Singhal, Silica Kole says [2]

This paper gives an approach to recognize colors in a two-dimensional image using colour thresholding technique in MATLAB with the help of RGB colour model to detect a selected colour by a user in an image. The methods involved for the detection of colour in images are conversion of three dimensional RGB image into gray scale image and then subtracting the two images to get two dimensional black and white image, using median filter to filter out noisy pixels, using connected

components labelling to detect connected regions in binary digital images and use of bounding box and its properties for calculating the metrics of each labelled region. Further the colour of the pixels is recognized by analysing the RGB values for each pixel present in the image. The algorithm is implemented using image processing toolbox in MATLAB. The results of this implementation can be used in security applications like spy robots, object tracking, segregation of objects based on their colours, intrusion detection.

In A Threshold Selection Method from Gray-Level Histograms, N. Ostusays [3]

This paper aims, A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multithreshold problems. Several experimental results are also presented to support the validity of the method.

METHODOLOGY

To understand the basics of camera analysing, one must know how a computer perceives an image. An image consists of a lot of pixels. Each pixel corresponds to a code. The summation of these codes gives an entire image. After the analysis of these codes, these codes can be used for the definition of colours and these defined colours are used for pixel labelling which are used to recognize colours in the picture and using the K-Means Algorithm for separating a particular colour or having more

commonly used colour in the multiple images.

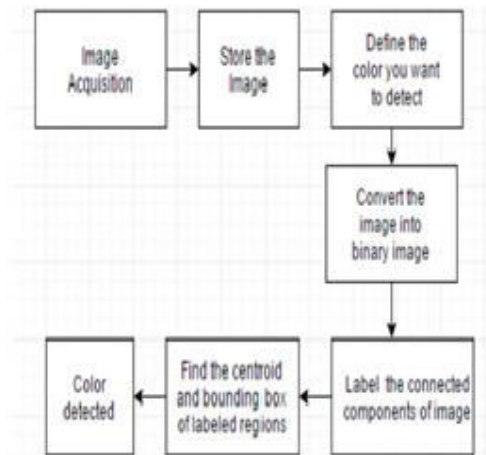


Fig: FlowChart of Detection of Colour

Steps involved in finding specific colour in image.

1. Read the Image.
 2. Display the Image.
 3. Select the position by Clicking mouse.
 4. Read the position.
 5. Get the pixel value of position.
 6. Map the pixel Value of current positions.
- *The first step is to collect the images, it might be from present the storage or from new snap with different pixels.
- * Next step is to store the images and display the image.
- *Select the position or segment of the colour or image by clicking the mouse.
- *Then read the position of the image.
- Here we having some xl sheets which is available in python which provide the 900 colours with values and pixels.
- *Read the position of the image.
- *Then lastly map the pixel value of current image.

RESULT

A set of 100 images of different resolution and clarity were used for testing of this algorithm and the results were mostly accurate. Red, green, blue, magenta, cyan, yellow and white and so on colours were successfully detected on these images. The result of this detection depends on

the threshold values that have been set for the images. Sometimes extra pixels are detected which are not the part of the desired region, and with increase in noise these errors increases. In the method of labelling of connected components for image segmentation. K-means clustering understand step by step.

To do this, will need a sample dataset (training set):

Objects	X	Y	Z
Obj1	1	4	1
Obj2	1	2	2
Obj3	1	4	2
Obj4	2	1	2
Obj5	1	1	1
Obj6	2	4	2
Obj7	1	1	2
Obj8	2	1	1

The sample dataset contains 8 objects with their X, Y and Z coordinates. Your task is to cluster these objects into two clusters i.e K=2

The algorithm works by:

- Taking any two centroids or data points (as you took 2 as K hence the number of centroids also 2) in its account initially.
- After choosing the centroids, (say C1 and C2) the data points (coordinates here) are assigned to any of the Clusters (let's take centroids = clusters for the time being) depending upon the distance between them and the centroids.
- Assume that the algorithm chose OB-2 (1,2,2) and OB-6 (2,4,2) as centroids and cluster 1 and cluster 2 as well.
- For measuring the distances, you take the following distance measurement function (also termed as similarity measurement function):

$$d=|x_2-x_1|+|y_2-y_1|+|z_2-z_1|$$

This is known as the Manhattan distance, where d is distance measurement between two objects, (x1,y1,z1) and (x2,y2,z2) are the X, Y and Z

coordinates of any two objects taken for distance measurement.

The following table shows the calculation of distances (using the above distance measurement function) between the objects and centroids (OB-2 and OB-6):

Obj	X	Y	Z	Distance from C1(1,2,2)	Distance from C2(2,4,2)
Obj1	1	4	1	3	2
Obj2	1	2	2	0	3
Obj3	1	4	2	2	1
Obj4	2	1	2	2	3
Obj5	1	1	1	2	5
Obj6	2	4	2	3	0
Obj7	1	1	2	1	4
Obj8	2	1	1	3	4

The objects are clustered based on their distances between the centroids. An object which has a shorter distance between a centroid (say C1) than the other centroid (say C2) will fall into the cluster of C1. After the initial pass of clustering, the clustered objects will look something like the following:

Cluster 1	Cluster 2
OB-2	OB-1
OB-4	OB-3
OB-5	
OB-7	OB-6
OB-8	

Now the algorithm will continue updating cluster centroids (i.e the coordinates) until they cannot be updated anymore (more on when it cannot be updated later). The updation takes place in the following manner:

$$\frac{\sum_{i=1}^n x_i}{n}, \frac{\sum_{i=1}^n y_i}{n}, \frac{\sum_{i=1}^n z_i}{n}$$

(Where n = number of objects belonging to that particular cluster) So,

Following this rule the updated Cluster 1 will be
 $((1+2+1+1+2)/5, (2+1+1+1+1)/5, (2+2+1+2+1)/5)$
 $= (1.4, 1.2, 1.6)$.

Cluster 2 will be

$((1+1+2)/3, (4+4+4)/3, (1+2+2)/3) = (1.33, 4, 1.66)$.

After this, the algorithm again starts finding the distances between the data points and newly derived cluster centroids. So the new distances will be like following

OBJE CT	X	Y	Z	C1(1.4,1.2, 1.6)	C2(1.33,4,1. 66)
OB-1	1	4	1	3.8	1
OB-2	1	2	2	1.6	2.66
OB-3	1	4	2	3.6	0.66
OB-4	2	1	2	1.2	4
OB-5	1	1	1	1.2	4
OB-6	2	4	2	3.8	1
OB-7	1	1	2	1	3.6
OB-8	2	1	1	1.4	4.33

The new assignments of the objects with respect to the updated clusters will be:

Cluster 1	Cluster 2
OB-2	OB-1
OB-4	OB-3
	OB-6
OB-5	
OB-7	
OB-8	

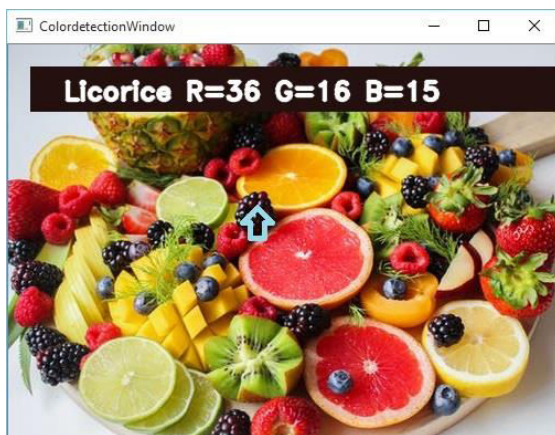
This is where the algorithm no longer updates the centroids. Because there is no change in the current cluster formation, it is the same as the previous formation.

Screenshots of Results :

Screen Shot1: Detected the Dark Tangerine colour in multiple colours by clicking mouse on that segment.



Screen Shot2: Detect the Licorice colour in multiple colours by clicking mouse on that segment.



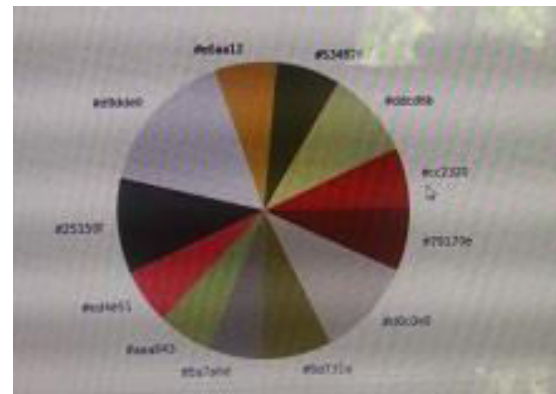
Screen Shot3: Detect the Deep Carmine colour in Multiple colours by clicking mouse on that segment.



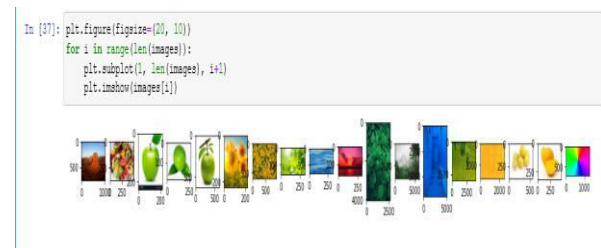
Screen Shot4: Detected the Brass colour in multiple colours by clicking mouse on that segment.



Screen shot5: Maximum colours in above image.



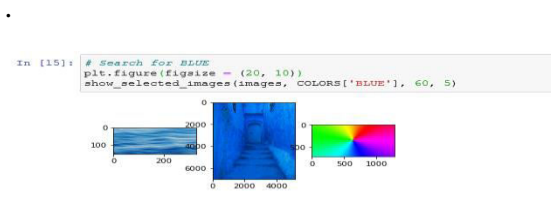
Screen Shot6: Collection of Multiple Images.



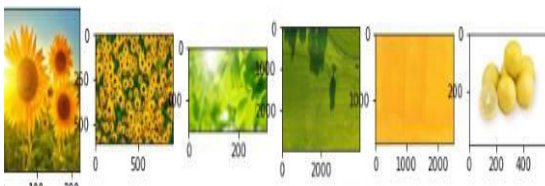


Screen Shot7: Separated the green colour based image from multiple images.

Screen Shot8: Separated Blue colour images from multiple images.



Screen Shot9: Separated Yellow colour images from Multiple Images.



CONCLUSION

With the help of image processing, machine learning and computer vision using python the program had been made which can detect the colour almost 900 colours. Also, the coloured object is being enclosed inside a bounded region along with the centroid of that region.

FUTURE SCOPE

1. Fill the noisy gaps in the images to find the hidden colours or objects in the images.
2. Object Tracking- A moving object can be tracked on the basis of its colour.

REFERENCES

[1] Color Managing the third edition of Billmeyer and Saltzman's principles of color Technology. Roy S.Berns, D.Mark Reiman

[2] Specific Color Detection in Images using RGB Modelling in MATLAB, Visheh Goel, Truan Jain, Sahil Singhal, Silica Kole

[3] N. Otsu, "Threshold Selection Method from Gray-level Histograms", IEEE Transactions on Systems, Man, Cybernetics, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[4] Rafael C. Gonzalez (University of Tennessee), Richard E. Woods (MedData Interactive) and Steven L. Eddins (The MathWorks, Inc.), in „Digital Image Processing Using MATLAB“ Second Edition, 2009 by Gatesmark, LLC.

[5] Alasdair McAndrew, in „An Introduction to Digital Image Processing with Matlab, Notes for SCM2511 Image Processing1“, School of Computer Science and Mathematics, Victoria University of Technology.

[6] Digital image processing using Matlab - Gonzalez woods & Eddins

[7] R. S. Berns, "Principles of Color Technology" (3rd edition New York: Wiley, 2000)

[8] G. Wyszecki and W. S. Styles, "Color Science: Concepts and Methods, Quantitative Data and Formulae" (2nd edition New York: Wiley, 1982)

[9] Devi, H.K.A., (2006). Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script. Ancient Asia. 1, pp.161–165.

[10] The Multi-stage Approach to Grey-Scale Image Thresholding for Specific Applications, Van Solihin and C. G. Leedham

[11] Document Image Analysis by Rangachar Kasturi, Louis Lam, Seong - Whan Lee & Ching Y. Suen.

[12] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation", Journal of Electronic Imaging, vol. 13, no. 1, pp. 146–168, Jan. 2004.

[13] N. R. Pal and D. Bhandari, "On Object Background Classification", International Journal Syst. Science, vol. 23, no. 11, pp. 1903–1920, Nov. 1992.

[14] J. Kittler and J. Illingworth, "Minimum error Thresholding", Pattern Recognition, vol. 19, no. 1, pp. 41–47, 1986.

[15] L. 10, nR. Guo and S. M. Pandit, "Automatic Threshold Selection based on Histogram Modes and a Discriminant Criterion", Machine Vision Applications, vol. 5–6, pp. 331–338, Apr. 19

[16] Haralick, Robert M., and Linda G. Shapiro. "Image segmentation techniques." Computer vision, graphics, and image processing 29.1 (1985): 100-132.