Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 5, Issue. 11, November 2016, pg.57 – 64

A Novel Method for Color Image Recognition

Dr. Ghazi. M. Qaryouti, Prof. Ziad A.A. Alqadi, Prof. Mohammed K. Abu Zalata

Albalqa Applied University

Jordan-Amman

Abstract: RGB Color images are now widely used as an important data type for data transmission via the internet, and by many small l mobile robots which have little on-board processing power for time-consuming vision algorithms. This paper produces a simple method to extract very dense yet highly useful information from color images quickly. An RGB color image is treated to create a small feature victor which can be used later as a signature to recognize the image. A data base of image signatures will be created and passed for artificial neural network(ANN) for recognition purposed, ANN will be trained using the created data base, then will be used after on to recognize a selected color image.

Keywords: RGB image, HSV image, ANN, Feature victor, image signature.

1- Introduction

1-1 RGB and HSV images

Many digital units such as mobile robots must process many images per second to react in time [1]. RGB color image pre-processor subsystem should be able to quickly extraction a small size and compact informative descriptor (signature) of the current image, to be fed into the digital unit. Hence the need is so urgent to a fast algorithms that often (but not necessarily always) produce image descriptions containing all the information necessary for decent image recognition.

RGB color image is a 3 dimensional matrix[2], [3], and it is an additive color model. It means that different proportions of Red, Blue and Green light can be used to produce color. The RGB color model was created specifically for display purposes (display screens, projectors etc).

HSV color system is based on the **H**ue shift, **S**aturation and **V**alue. Unlike the RGB color system, which has to do with "implementation details" regarding the way RGB displays color, HSV has to do with the "actual color" components. Another way to say this would be RGB is the way computers treats color, and HSV try to capture the components of the way we humans *perceive* color. The main reason to work on the HSV version of an image is because using Hue component makes the algorithms less sensitive (if not invariant) to lighting variations. Because HSL and HSV are simple transformations of device-dependent RGB models, the physical colors they define depend on the colors of the red, green, and blue primaries of the device or of the particular RGB space[4], [5].,[16]

RGB to HSV conversion formula

To convert the *R,G,B* of the color image values are divided by 255 to change the range from 0..255 to 0..1 we can apply the following formulas:

$$R' = R/255$$
$$G' = G/255$$
$$B' = B/255$$
$$Cmax = \max(R', G', B')$$
$$Cmin = \min(R', G', B')$$
$$\Delta = Cmax - Cmin$$

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = Cmax$$

Table 1 shows an example of RGB values and there equivalents in HSV

1-2 Artificial neural network

Artificial neural network is powerful tool which can be used for many application such as classification or object recognition. ANN [15] model is consisted of a set of neurons which are fully connected [6], [7]. The neurons are organized in layers(one input layer, 1 or more hidden layer and one output layer). Each neuron is a computational element which finds the summation of the results of multiplication each input with its weight. The summation then to be used to generate the output depending on the activation function used for the neuron [8], [9]. To use ANN as a tool for recognition we have to follow the following steps:

1- Define the input data set 2- Define the targets(outputs) 3- Select ANN architecture(Number of layers, number of neurons in each layer activation function for each layer). 4- Define ANN parameters(error, training cycles). 5- Train ANN 6- Implement ANN.

Table 1: RGB and there equivalents in HSV

Color name	(R,G,B)	(H,S,V)
Black	(0,0,0)	(0°,0%,0%)
White	(255,255,255)	(0°,0%,100%)
Red	(255,0,0)	(0°,100%,100%)
Lime	(0,255,0)	(120°,100%,100%)

Blue	(0,0,255)	(240°,100%,100%)
Yellow	(255,255,0)	(60°,100%,100%)
Cyan	(0,255,255)	(180°,100%,100%)
Magenta	(255,0,255)	(300°,100%,100%)
Silver	(192,192,192)	(0°,0%,75%)
Gray	(128,128,128)	(0°,0%,50%)
Maroon	(128,0,0)	(0°,100%,50%)
Olive	(128,128,0)	(60°,100%,50%)
Green	(0,128,0)	(120°,100%,50%)
Purple	(128,0,128)	(300°,100%,50%)
Teal	(0,128,128)	(180°,100%,50%)
Navy	(0,0,128)	(240°,100%,50%

1-3 Features extraction

A color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors. Color histograms are flexible constructs that can be built from images in various color spaces, whether RGB, VHS or any other color space of any dimension [13] and [14].

The main drawback of histograms for classification is that the representation is dependent of the color of being studied, ignoring its shape and texture. Color histograms can potentially be identical for two images with different object content which happens to share color information. Conversely, without spatial or shape information, similar objects of different color may be indistinguishable based solely on color histogram comparisons.

2- The proposed method of image recognition

The proposed method can be implemented in two phases:

Phase 1: Creating a vector (1,32) indicating the features extracted from HSV color space

- Get the original RGB color image.
- Convert the original image to HSV image.
- Split image into h, s and v planes.
- Quantize each H, S, V equivalently to 8x2x2 (8 quantizing levels for H and 2 for each of S and V)
-

The following matlab function computes a special signature to be used later for image recognition:

```
function hsvColorHistogram = hsvHistogram(image)
% input: image to be quantized in hsv color space into 8x2x2 equal bins
% output: 1x32 vector indicating the features extracted from hsv color
% space

[rows, cols, numOfBands] = size(image);
% totalPixelsOfImage = rows*cols*numOfBands;
image = rgb2hsv(image);
```

```
% split image into h, s & v planes
h = image(:, :, 1);
s = image(:, :, 2);
v = image(:, :, 3);
 % quantize each h,s,v equivalently to 8x2x2
% Specify the number of quantization levels.
% quantize each h,s,v to 8x2x2
% Specify the number of quantization levels.
numberOfLevelsForH = 8;
numberOfLevelsForS = 2;
numberOfLevelsForV = 2;

% Find the max.
maxValueForH = max(h(:));
maxValueForS = max(s(:));
maxValueForV = max(v(:));
 % create final histogram matrix of size 8x2x2
hsvColorHistogram = zeros(8, 2, 2);
 % create col vector of indexes for later reference
index = zeros(rows*cols, 3);
 % Put all pixels into one of the "numberOfLevels" levels.
count = 1;
for row = 1:size(h, 1)
   for col = 1 : size(h, 2)
      quantizedValueForH(row, col) = ceil(numberOfLevelsForH * h(row, col)/maxValueForH);
      quantizedValueForS(row, col) = ceil(numberOfLevelsForS * s(row, col)/maxValueForS);
      quantizedValueForV(row, col) = ceil(numberOfLevelsForV * v(row, col)/maxValueForV);
      % keep indexes where 1 should be put in matrix hsvHist
      index(count, 1) = quantizedValueForH(row, col);
      index(count, 2) = quantizedValueForS(row, col);
      index(count, 3) = quantizedValueForV(row, col);
      count = count+1;
   end
end
 % put each value of h,s,v to matrix 8x2x2
% (e.g. if h=7,s=2,v=1 then put 1 to matrix 8x2x2 in position 7,2,1)
for row = 1:size(index, 1)
   if (index(row, 1) == 0 || index(row, 2) == 0 || index(row, 3) == 0)
      continue;
   end
   hsvColorHistogram(index(row, 1), index(row, 2), index(row, 3)) = ...
      hsvColorHistogram(index(row, 1), index(row, 2), index(row, 3)) + 1;
end
 % normalize hsvHist to unit sum
hsvColorHistogram = hsvColorHistogram(:)';
hsvColorHistogram = hsvColorHistogram/sum(hsvColorHistogram);
 end
```

Phase 2: Creating ANN

A signature of each color image is to be passed to ANN which has the following features:

- The input data base is a matrix of 32 rows and n columns (number of the color images stored in a specified folder), one column for each image, which represents image signature.

- The targets is one row matrix with values indicating image numbers(1, 2, 3, ...,n)

- ANN has at least one input layer with 32 neurons and one output layer.
- The activation function for the input layer is tansig, while for the output layer is linear.

3- Implementation and results discussion

One hundred color with different sizes were selected and stored in a selected folder, each color image was treated by the function shown in phase 1, and the output signature was stored in the data base of signatures, table 2 shows the sizes the signatures for selected 8 images:

Table 2: Signatures for selected 8 images

384* 512 * 3 byte	480 *640 * 3	194 * 259 * 3	230 219 3	510 700 3	180 280 3	194 260 3	168 300 3
0.0028	0.0205	0.0011	0.0002	0.0433	0.0227	0.0352	0.0074
0.0057	0.0595	0.0017	0.0005	0.0510	0.0446	0.1177	0.0013
0.0011	0.2662	0.0029	0.0004	0.0240	0.0109	0.0997	0.0004
0.0001	0.0850	0.0157	0.0013	0.0215	0.0071	0.0046	0.0016
0.0002	0.0275	0.3896	0.0010	0.1469	0.2908	0.0001	0.0013
0.0007	0.0225	0.0955	0.0009	0.0086	0.0134	0.0001	0.0084
0.3694	0.0228	0.0006	0.0004	0.0035	0.0021	0.0004	0.3162
0.0159	0.0230	0.0014	0.0002	0.0016	0.0038	0.0034	0.1062
0.0216	0.0021	0	0.0004	0.0073	0.1526	0.0117	0.0077
0.0803	0.0242	0	0.0000	0.0007	0.1366	0.0782	0.0008
0.0009	0.1294	0.0001	0.0000	0.0019	0.0044	0.0439	0.0014
0	0.0198	0.0003	0.0026	0.0335	0.0001	0.0001	0.0048
0.0000	0.0043	0.0260	0.0005	0.0329	0.0011	0	0.0019

0.0000	0.0042	0.0359	0	0.0003	0.0008	0	0.0090
0.0715	0.0059	0.0000	0.0023	0	0.0007	0.0001	0.0530
0.0416	0.0115	0.0000	0.0002	0.0000	0.0019	0.0059	0.0212
0.0310	0.0008	0.0762	0.1124	0.0261	0.0326	0.0246	0.0071
0.0077	0.0044	0.0671	0.1040	0.0115	0.0329	0.0270	0.0003
0.0012	0.0352	0.0227	0.0382	0.0122	0.0111	0.0929	0.0004
0.0006	0.0097	0.0524	0.0623	0.0169	0.0136	0.0179	0.0015
0.0002	0.0197	0.1498	0.1243	0.1855	0.1333	0.0001	0.0002
0.0003	0.0488	0.0398	0.0308	0.0423	0.0150	0	0.0301
0.0326	0.1008	0.0026	0.1040	0.0086	0.0015	0.0014	0.0430
0.0030	0.0043	0.0183	0.0593	0.0053	0.0010	0.0274	0.0542
0.1581	0	0	0.1058	0.0061	0.0462	0.1754	0.0819
0.0619	0.0003	0	0.0696	0.0026	0.0190	0.0367	0.0000
0	0.0018	0	0.0086	0.0090	0	0.0097	0
0	0	0	0.0318	0.0533	0	0	0.0338
0	0	0	0.0663	0.2365	0	0	0.0014
0	0.0000	0	0.0010	0.0069	0	0	0.0932
0	0.0091	0	0.0161	0	0	0.0022	0.0966
0.0916	0.0368	0	0.0546	0.0001	0	0.1837	0.0139

The data base then was passed to ANN to be trained, figure 1 shows the data base for 8 images.

The trained ANN was created using 3 layer(input layer with 32 neurons, hidden layer with 32 neurons and output layer with 1 neuron), the activation functions were tansig for the input and the hidden layers, and linear for the output layer.

The goal(error) was reset to 0 .

p =

0.0028	0.0205	0.0011	0.0002	0.0433	0.0227	0.0352	0.0074
0.0057	0.0595	0.0017	0.0005	0.0510	0.0446	0.1177	0.0013
0.0011	0.2662	0.0029	0.0004	0.0240	0.0109	0.0997	0.0004
0.0001	0.0850	0.0157	0.0013	0.0215	0.0071	0.0046	0.0016
0.0002	0.0275	0.3896	0.0010	0.1469	0.2908	0.0001	0.0013
0.0007	0.0225	0.0955	0.0009	0.0086	0.0134	0.0001	0.0084
0.3694	0.0228	0.0006	0.0004	0.0035	0.0021	0.0004	0.3162
0.0159	0.0230	0.0014	0.0002	0.0016	0.0038	0.0034	0.1062
0.0216	0.0021	0	0.0004	0.0073	0.1526	0.0117	0.0077
0.0803	0.0242	0	0	0.0007	0.1366	0.0782	0.0008
0.0009	0.1294	0.0001	0	0.0019	0.0044	0.0439	0.0014
0	0.0198	0.0003	0.0026	0.0335	0.0001	0.0001	0.0048
0	0.0043	0.0260	0.0005	0.0329	0.0011	0	0.0019
0	0.0042	0.0359	0	0.0003	0.0008	0	0.0090
0.0715	0.0059	0	0.0023	0	0.0007	0.0001	0.0530
0.0416	0.0115	0	0.0002	0	0.0019	0.0059	0.0212
0.0310	0.0008	0.0762	0.1124	0.0261	0.0326	0.0246	0.0071
0.0077	0.0044	0.0671	0.1040	0.0115	0.0329	0.0270	0.0003
0.0012	0.0352	0.0227	0.0382	0.0122	0.0111	0.0929	0.0004
0.0006	0.0097	0.0524	0.0623	0.0169	0.0136	0.0179	0.0015
0.0002	0.0197	0.1498	0.1243	0.1855	0.1333	0.0001	0.0002
0.0003	0.0488	0.0398	0.0308	0.0423	0.0150	0	0.0301
0.0326	0.1008	0.0026	0.1040	0.0086	0.0015	0.0014	0.0430
0.0030	0.0043	0.0183	0.0593	0.0053	0.0010	0.0274	0.0542
0.1581	0	0	0.1058	0.0061	0.0462	0.1754	0.0819
0.0619	0.0003	0	0.0696	0.0026	0.0190	0.0367	0
0	0.0018	0	0.0086	0.0090	0	0.0097	0
0	0	0	0.0318	0.0533	0	0	0.0338
0	0	0	0.0663	0.2365	0	0	0.0014
0	0	0	0.0010	0.0069	0	0	0.0932
0	0.0091	0	0.0161	0	0	0.0022	0.0966
0.0916	0.0368	0	0.0546	0.0001	0	0.1837	0.0139

Figure 1: Input data set

After training ANN, it was tested for recognition and the obtained results gave a high recognition ratio, because images signatures are deferent and each signature can be used to select or recognize the desired color image.

Conclusions

Method of image recognition was proposed, this method is simple and effective and it can reach a recognition ratio of 100%.

The proposed method was tested using several images, and for each image a unique signature for each image was obtained and passed to ANN for recognition.

References

[1]: Shao, H., Svoboda, T., Van Gool, L.: ZuBuD — Z¨urich buildings database for image based recognition. Technical Report 260, Computer Vision Laboratory, Swiss Federal Institute of Technology (2003) Database downloadable from http://www.vision.ee.ethz.ch/showroom/.

[2]: Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. AbuJazar and Rushdi Abu Zneit, Optimized True-Color Image Processing, World Applied Sciences Journal 8 (10): 1175-1182, 2010.

[3]: Akram Mustafa and Ziad AlQadi, Color image, reconstruction using a new model. J. Comput. Sci., 5: 250-159, 2009.

[4]: Rafael C. Gonzalez and Richard Eugene Woods (2008). *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall. ISBN 0-13-168728-X. pp. 407–413

[5]: Monika Deswal1 , Neetu Sharma, A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm, International Journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014.

[6]: Ziad A.AlQadi and others, Investigation and Analysis of ANN Parameters, European Journal of Scientific Research 121(2):217-225 · January 2014.

[7]: Ziad A.AlQadi and others, Speech Fingerprint to Identify Isolated Word-Person, World Applied Sciences Journal 31(10):1767-1771 · January 2014.

[8]: Haykin, S. 1994. Neural Networks: A Comprehensive Foundation. New York, NY: Macmillan College Publishing Company, Inc.

[9]: C.-C. YANG1 , S.O. PRASHER , J.-A. LANDRY1 , H.S. RAMASWAMY and A. DITOMMASO, Application of artificial neural networks in image recognition and classification of crop and weeds, CANADIAN AGRICULTURAL ENGINEERING Vol. 42, No. 3 July/August/September 2000

[10]: Jayanta Kumar Basu , Debnath Bhattacharyya , Tai-hoon Kim, Use of Artificial Neural Network in Pattern Recognition, International Journal of Software Engineering and I International Journal of Software Engineering and Its Applications ts Applications Vol. 4, No.2, April 2010.

[11]: Cheng-Yuan Tang, Yi-Leh Wu, Maw-Kae Hor, Wen-Hung Wang. Sift descriptor for image matching under interference. In: Machine learning and cybernetics international conference. Vol. 6; 2008. p. 3294–300.

[12]: E. Rosten, R. Porter, T. Drummond FASTER and better: a machine learning approach to corner detection IEEE Trans Pattern Anal Mach Intel, 32 (2010), pp. 105–119

[13]: Jinxia L, Yuehong Q. Application of SIFT feature extraction algorithm on the image registration. In: Tenth international conference on electronic measurement & instruments IEEE; 2011.

[14]: H. Stokman, T. Gevers, Selection and fusion of color models for image feature detection Pattern Anal Mach Intel IEEE Trans, 29 (March 2008), pp. 371–381

[15]: Akram A. Moustafa1, Ziad A. Alqadi and Eyad A. Shahroury. Performance Evaluation of Artificial Neural Networks for Spatial Data Analysis, WSEAS TRANSACTIONS on COMPUTERS, Issue 4, Volume 10, April 2011.

[16]: Waheeb Abu Ulbeh , Akram Moustafa , Ziad A. Alqadi , Gray Image Reconstruction, European Journal of Scientific Research ISSN 1450-216X Vol.27 No.2 (2009), pp.167-173.

