Bulletin of Electrical Engineering and Informatics
Vol. 7, No. 3, September 2018, pp. 367~376
ISSN: 2302-9285, DOI: 10.11591/eei.v7i3.1216

Creating Color Image Features Using Local Contrast Method

Ayman Al-Rawashdeh, Ziad Al-Qadi
Al-Balqa Applied University/Faculty of Engineering Technology, Department of Computor Engineering, Amman, Jordan

Article Info

Article history:

Received May 04, 2018
Revised Aug 12, 2018
Accepted Aug 21, 2018

Keywords:

Artificial Neural Network
Feature extraction
Image generation
Image processing
Image quality

ABSTRACT

Digital color images are now one of the most popular data types used in the digital processing environment. Color image recognition plays an important role in many vital applications, which makes the enhancement of image recognition or retrieval system an important issue. Using color image pixels to recognize or retrieve the image, but the issue of the huge color image size that requires accordingly more time and memory space to perform color image recognition and/or retrieval. In the current study, image local contrast was used to create local contrast victor, which was then used as a key to recognize or retrieve the image. The proposed local contrast method was properly implemented and tested. The obtained results proved its efficiency as compared with other methods.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Ayman Al-Rawashdeh,
Al-Balqa Applied University/Faculty of Engineering Technology,
Department of Electrical Engineering, Amman, Jordan
Email: dr.ayman.rawashdeh@bau.edu.jo

1. INTRODUCTION

1.1. Color Image Features

Color image features play an important role in image manipulation, as they can be used as a finger print for fast image retrieval or image recognition [1], [2]. The process of image retrieval depends on the features that can be automatically extracted from the images themselves [3]. Feature extraction process generates features to be used in the selection and classification tasks for image retrieval process. It transforms rich content of images into various content features. This process involves the selection of features that assist in discrimination of an image. Feature selection reduces the number of features provided to the classification task. Feature extraction is considered to be most critical task among other image processing tasks. This is because the selection of the particular features for discrimination directly influences the efficacy of the image recognition and identification. Feature extraction process ends with a set of features, commonly referred to as the feature vector, which constitutes the unique representation of the image [3]. The feature victor is usually small in size as compared to the original image size, which means that training time of artificial neural network (ANN) will be reduced when image features is used to identify the image [4-7]. The ANN image identification is illustrated in Figure 1.

Reducing the data needed to identify the image leads to the optimal ANN construction [7]. The main requirements needed for an optimal ANN construction includes:

- Minimum memory space to store ANN.
- Minimum ANN input data set size.
- Minimum ANN architecture.
- 100 % recognition or identifying ratio.
- Minimum ANN training time.
- Minimum image retrieving time.

Colors image are usually represented by a histogram, which is a three columns victor [1], [2]. The histogram can be used as a signature to identify color image. Each histogram victor is a features array that is very small in size comparing with color image size (256*3). Figure 2 shows a colored image with its corresponding histogram victor.

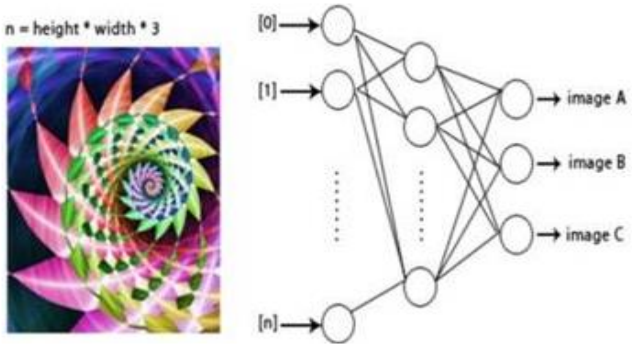


Figure 1. The ANN to identify the image pixel by pixel

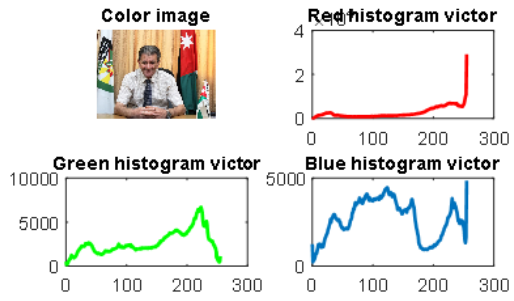


Figure 2. Histogram victors for color image

Digital color image is a 3D matrix as shown in Figure 3. Each column represents a color channel, for instance, the first column represents the red color, and the second represents the green color, while the third one represents the blue color [8-11].

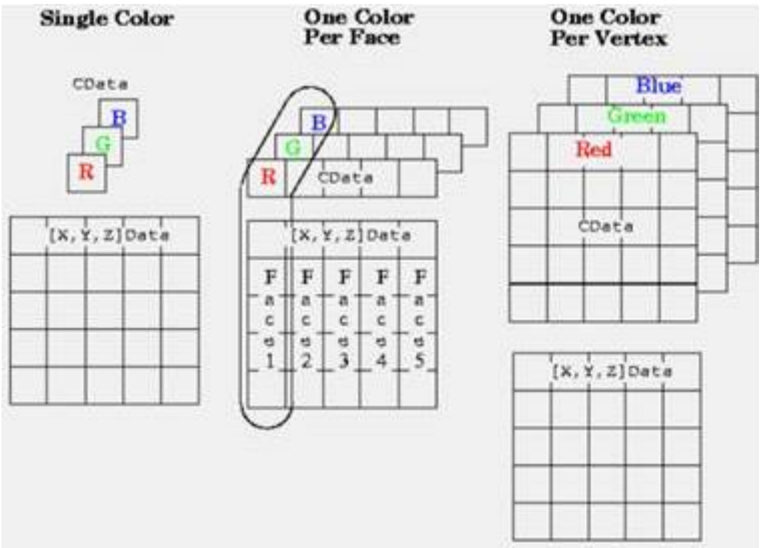


Figure 3. 3D color matrix

Each color component can be processed by a 2D matrix as shown in Figure 4 [12], [13]. Color images usually have big sizes and the identification of image pixel by pixel needs big efforts and might be considered time consuming process. This suggests the need for a more efficient method to identify color image depending on the extraction of features with a small size [8-10].

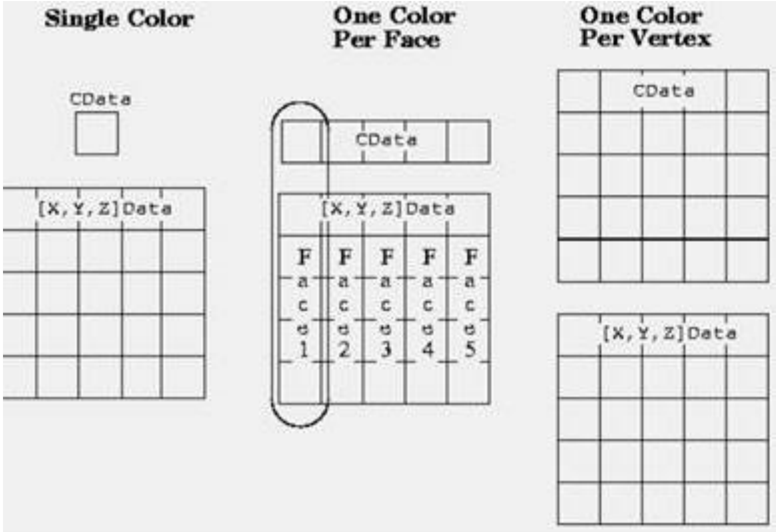


Figure 4. 2D matrix for each color

Color image can be represented by color histograms. Again, each histogram is a one column array that contains 256 elements. The histogram can be used as image identifier, and accordingly, image identification and retrieval might be enhanced leading to a quick system response time [14], [15] as could be deduced from Figure 5.

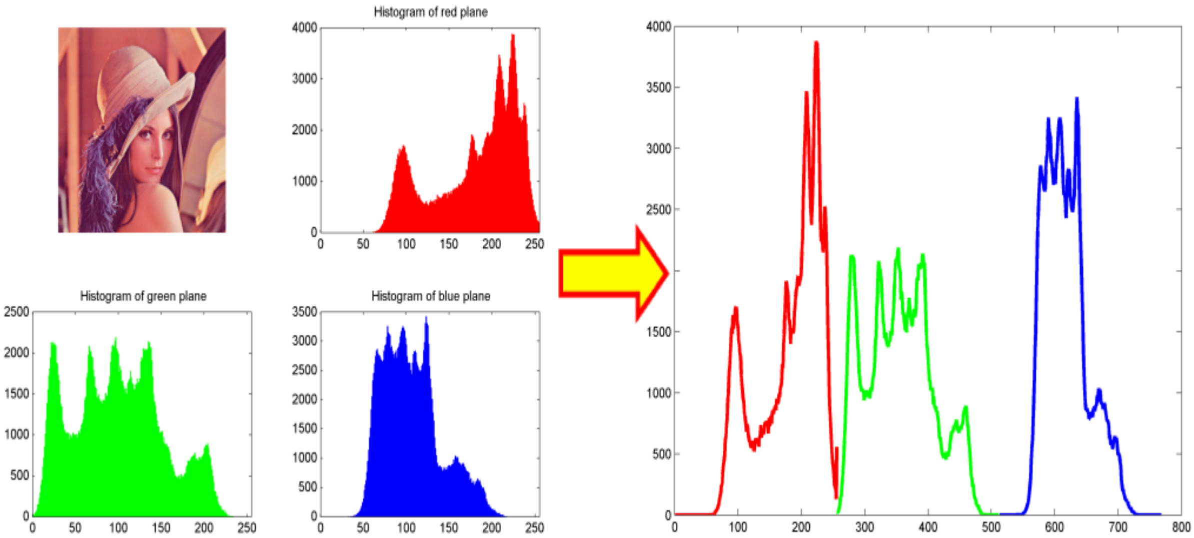


Figure 5. Color image histograms

1.2. Color Imaage Feature Extraction Using CSLBP Method

Calculation LBP operators depend on the neighbor's pixels values [9]. In the LBP method, an array of 256 elements (index) will be generated representing the image features. The LBP 256 operator can be calculated using the values of the neighbors of a pixel in the image as shown in Figure 6.

100	120	200		0	0	1
160	150	70		1	x	0
30	180	50	→	0	1	0
LBP calculation Compare neighbours with 150				LBP=01010010binary Decimal =82		

Figure 6. LBP calculation using Neighbor pixels

However, LBP method does not reduce the histogram victor size (256 elements), and therefore it might not be suitable to extract color image features [8-11]. The CSLBP method creates a repetition value (from 0 to 15) for each pixel in the image; these repetitions can be used as a feature array to identify the image [17]. CSLBP method is used to generate features array for each pixel in the region [23], [24]. In CSLBP, center-symmetric pairs of pixels are compared to produce more compact binary patterns as shown in Figure 7. It decreases the number of features, thus increases the system efficiency.

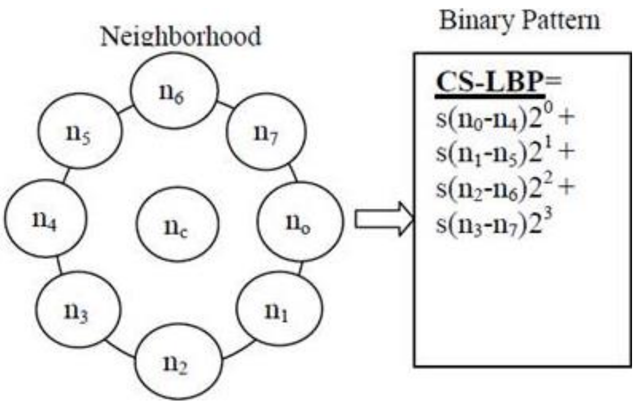
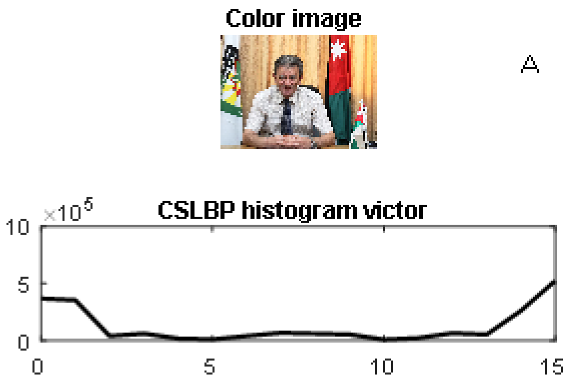


Figure 7. CSLBP calculation

An example is shown in Figure 7, in which 8 neighbors (n0 - n7), CS-LBP produced 16 different binary patterns. A Small threshold value was then used to obtain the robustness on flat image regions as shown in Figure 8(a).

Here, a matlab code was used and implemented to generate image features using CSLBP method. The results indicated that each feature array for each individual image is unique; accordingly, an array can be used as a key to identify its associated image. Figure 8(b), shows samples of the image features obtained as a result of implementing the matlab code (one column array for each color image).



A

B

44920	28694	9089	211947	94738	579040	364147	164497	78080	...56
40941	33681	5747	102697	53709	394427	249463	97468	40005	16992
4395	2670	1421	30977	15849	98198	64822	21299	14466	5258
9594	3643	1802	56873	23831	156400	94780	52290	16406	6570
2474	839	911	35009	14684	43683	37068	20676	10699	3764
1982	819	671	17718	7132	34615	29842	6602	7350	2465
4728	2210	1377	27596	14504	95740	59677	26028	12154	4388
10099	3937	1967	66963	25675	185344	114437	52700	20997	7054
7301	3808	1884	70998	23013	141004	88772	47852	19376	6303
6346	3505	1352	31735	15473	111151	71703	25894	12605	4301
1441	652	639	17522	7167	30515	25521	7304	7689	2574
3201	934	951	34502	14022	63340	47506	20374	11597	3990
7897	3677	1814	59995	24361	134328	91243	60323	17031	6253
6284	3638	1626	36351	18119	126714	85834	25260	15673	5510
29455	19944	5932	98374	51057	349087	233008	100417	41143	14286
87612	37428	10949	221933	133670	844435	694821	195524	103913	31585

Figure 8. A: CSLBP histogram victor; B: Image feature samples obtained by CSLBP

1.3. Image Locat Contrast Victor

Image local contrast (LC) was proposed in 1992 by Hunt [25]. LC is an average difference between neighboring pixels, which can be used to generate local contrast array of 9 elements. Calculation of the LC involves the following steps:

a. Calculation of the scaled and corrected values of linear luminance Equation l:

$$l = \left(\frac{k}{255}\right)^{\gamma} \quad (1)$$

Where: k is the pixel value (0 to 255), γ correction is 2.

b. Calculation of the perceptual luminance L using Equation 2:

$$L = 100 * \sqrt{l} = 100 * \sqrt{\left(\frac{k}{255}\right)^{\gamma}} \quad (2)$$

c. Calculating l_{Ci} for each of the following resolution levels (Resolutions=[1 2 4 8 16 25 50 100 200] using Equation 3.

$$\boldsymbol{l_{ci}} = \frac{|L_i - L_{i-1}| + |L_i - L_{i+1}| + |L_i - L_{i-w}| + |L_i - L_{i+w}|}{4} \quad (3)$$

Figure. 10 explains the process of abstaining LC using 4 neighbors [22].

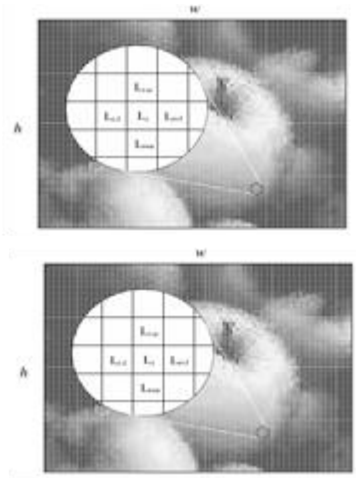


Figure 10. Using 4-neighbores to calculate LC

2. IMPLEMENTATION

To calculate the LC and to use LC array as an image features, the following steps were performed [9], [10]:

a. Get the original color image.
b. The color image matrix was reshaped from 3D to 2D.
c. Steps used to calculate the LC array were applied.
d. The obtained LC array was saved as a key for image identification.

2.1. Experimental Results

The following matlab function was written and implemented using various color images with different types and sizes:

```
function [ LC]=localcontrast( im )
% Input:
%     im - 2D image
%% Output:
%     LC - Local contrast victor
  resolutions=[1 2 4 8 16 25 50 100 200];
    LC=zeros(size(resolutions));
  W=size(im,2);
  H=size(im,1);
  rIm=im;
    for i=1:length(resolutions)
        %attempt at resizing as in the paper
    if i>1
      rIm=imresize(im, 1/(2^(i-1)), 'bilinear');
    end
    W=size(rIm,2);
    H=size(rIm,1);
    rL=zeros(size(rIm));
    % compute linear luminance l
    l=(double(rIm(:,:))/255) * 2.2;
    % compute perceptual luminance L
    rL(:,:)=100 * sqrt(l);
        % compute local contrast for each pixel
    lc=0.0;
    for x=1:H
      for y=1:W
                  if (x==1) && (x==H)
          if (y==1) && (y==W)
```

```
          lc=lc + 0;
        elseif (y==1)
          lc=lc + abs(rL(x, y) - rL(x,y+1));
        elseif (y==W)
          lc=lc + abs(rL(x, y) - rL(x,y-1));
        else
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                abs(rL(x, y) - rL(x,y+1)) )/2;
        end
      elseif (x==1)
        if (y==1) && (y==W)
          lc=lc + abs(rL(x, y) - rL(x+1,y));
        elseif (y==1)
          lc=lc + ( abs(rL(x, y) - rL(x,y+1)) + ...
                abs(rL(x, y) - rL(x+1,y)) )/2;
        elseif (y==W)
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                abs(rL(x, y) - rL(x+1,y)) )/2;
        else
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                abs(rL(x, y) - rL(x,y+1)) + ...
                abs(rL(x, y) - rL(x+1,y)) )/3;
        end
              elseif (x==H)
        if (y==1) && (y==W)
          lc=lc + abs(rL(x, y) - rL(x-1,y));
        elseif (y==1)
          lc=lc + ( abs(rL(x, y) - rL(x,y+1)) + ...
                abs(rL(x, y) - rL(x-1,y)) )/2;
        elseif (y==W)
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                abs(rL(x, y) - rL(x-1,y)) )/2;
        else
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                  abs(rL(x, y) - rL(x,y+1)) + ...
                  abs(rL(x, y) - rL(x-1,y)) )/3;
        end
else % x > 1 && x < H
        if (y==1) && (y==W)
          lc=lc + ( abs(rL(x, y) - rL(x+1,y)) + ...
                abs(rL(x, y) - rL(x-1,y)) )/2;
        elseif (y==1)
          lc=lc + ( abs(rL(x, y) - rL(x,y+1)) + ...
                  abs(rL(x, y) - rL(x+1,y)) + ...
                  abs(rL(x, y) - rL(x-1,y)) )/3;
        elseif (y==W)
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                  abs(rL(x, y) - rL(x+1,y)) + ...
                  abs(rL(x, y) - rL(x-1,y)) )/3;
        else
          lc=lc + ( abs(rL(x, y) - rL(x,y-1)) + ...
                  abs(rL(x, y) - rL(x,y+1)) + ...
                  abs(rL(x, y) - rL(x-1,y)) + ...
                  abs(rL(x, y) - rL(x+1,y)) )/4;
        end
      end
    end
  end
      % compute average local contrast c
```

```
c(i)=lc/(W*H);
w(i)=(-0.406385*(i/9)+0.334573)*(i/9)+ 0.0877526;
  % compute local contrast factor
LC(i)=c(i)*w(i);
  end
end
```

Different images were processed, and for each image in the dataset the LC array was obtained. The implementation results are shown in Table 1. From Table 1, it can be noticed that each LC array is unique, and thus it can be used as a key to identify the image.

Table 1. LC Array for Different Color Images

Image	Local contrast
1	0.7029
2	0.7521
3	1.0355
4	0.9834
5	1.4120
6	0.8420
7	0.6520
8	1.2015
9	1.2783
10	1.7832

Again and as abovementioned, LC array is unique and very sensitive to any changes in the original image, for example in the case the original image is changed, another and different LC array will be generated, which proves the uniqueness of LC array for each image, or for same image with slight changes and updates. Table. 2 shows different LC arrays for the same image with different versions (slight changes have been applied to the original image).

Table 2. Images Changes Lead to Changes in LC array

Image	Features								
Origin	0.1346	0.2283	0.3534	0.5632	0.8658	1.0041	0.8788	1.4651	0
Changing pixel	0.1349	0.2284	0.3533	0.5630	0.8654	1.0041	0.8788	1.4651	0
Changing 2 pixels	0.1351	0.2285	0.3533	0.5631	0.8654	1.0041	0.8788	1.4651	0

The calculation time needed to generate the LC array was obtained by implementing the previous matlab function. Results are shown in Table 3.

Table 3. LC Calculation Time

Image	Size (Pixels)	Calculation time (Seconds)
1	270948	0.631241
2	151875	0.125924
3	49152	0.101759
4	1125600	0.841033
5	540000	0.678468
6	3396069	1.451037
7	2359296	1.176620
8	928800	0.784822
9	432000	0.690299
10	151353	0.128947

3. RESULTS COMPARISON

A matlab code was written to create color images features using CSLBP method. Table 4 shows the extraction time for both methods. The comparison shows that the extraction time using CSLBP is lower compared to the time when LC extraction method was used.

Table 4. Features Extraction Time

Image	CSLBP extraction time (seconds)	LC calculation time (seconds)
1	0.0883	0.631241
2	0.0481	0.125924
3	0.0194	0.101759
4	0.3390	0.841033
5	0.1549	0.678468
6	0.9680	1.451037
7	0.6615	1.176620
8	0.2636	0.784822
9	0.1231	0.690299
10	0.0477	0.128947
Average	0.2714	0.6610

A dataset of 10 images features for each method was built, the ANN for each method was created and trained to achieve 100% recognition ratio and then tested to identify the image. Table. 5 summarizes comparison results. The results in Table. 5 clearly indicate the efficiency of the proposed local contrast LC method which exceeds the efficiency of CSLBP method.

Table 5. Results Comparisons

Factor	LC method	CSLBP method
Extraction time (average)	0.6610	0.2714
ANN training time (seconds)	0.997369	9.311232
Retrieving time (seconds)	0.104247	0.120581
ANN architecture	3 layers with 9, 3, 1 neuron	3 layers with 16, 4, 1 neuron
ANN activation functions	Tansig, tansig, linear	Tansig, tansig, linear
Number of features	9	16
Memory size to save ANN	288 KB	304 KB
Input data base size(10 images)	720 byte	1280 byte

4. CONCLUSION

In the current study, a local contrast method of color image features extraction was proposed. For each image, the obtained feature victor is unique, thus it can be used as a key to identify or retrieve the image. The proposed method was implemented and tested. The results showed that LC method can increase the system efficiency and performance compared with CSLBP method.

REFERENCES

[1] Uma S, Srujana B. *Feature Extraction for Human Detection using HOG and CS-LBP methods*. International Journal of Computer Applications (0975 – 8887) National Conference Electronics, Signals, Communication and Optimization, NESCO-2015: 11-14.

[2] Yang S, Liao X, Borasy U. A Pedestrian Detection Method Based on the HOG-LBP Feature and Gentle AdaBoost. *International Journal of Advancements in Computing Technology*. 2012; 4(19): 553-560.

[3] Choras, R. Image feature extraction techniques and their applications for CBIR and biometrics systems. *International Journal of Biology and Biomedical Engineering*. 2007; 1(1): 6-16.

[4] Matrouk K, Alasha'ary H, Al-Hasanat A, Al-Qadi Z, Al-Shalabi H. Investigation and Analysis of ANN Parameters. *European Journal of Scientific Research*. 2014; 121(2): 217-225.

[5] Kohonen T. An Introduction to Neural Computing. *Neural Networks*. 1988; 1(1): 3-16.

[6] Hecht-Nielsen R. *Counter-Propagation Networks*. Proceedings of the IEEE First International Conference on Neural Networks. Ssan Diego, CA, USA. 1987; 19-32.

[7] Lu T, Chang C. Color image retrieval technique based on color features and image bitmap. *Information processing & management*. 2007; 43(2): 461-72.

[8] Hong X, Zhao G, Pietikainen M, Chen X. Combining LBP difference and feature correlation for texture description. *IEEE Transactions on Image Processing*, 2014; 23(6): 2557-2668.

[9] Qaryouti G, Khawatreh S, Alqadi Z, Abu Zalata M. Optimal Color Image Recognition System (OCIRS). *International Journal of Advanced Computer Science and Technology*, 2017; 7(1): 91-99.

[10] Gupta R, Patil H, Mittal A. *Robust order-based methods for feature description*. IEEE conference on computer vision and pattern recognition (CVPR). San Francisco, CA, USA. 2010.

[11] Rami H, Hamri M, Masmoudi L. Objects Tracking in Images Sequence Using Center-Symmetric Local Binary Pattern (CS-LBP). *International Journal of Computer Applications Technology and Research*. 2013; 2(5): 504 – 508.

[12] Kong H, Wang L, Teoh E, Li X, Wang J, Venkateswarlu R. Generalized 2D principal component analysis for face image representation and recognition. *Neural Networks*. 2005; 18(5-6): 585-594.

[13] Abu Zneit R, AlQadi Z, Abu Zalata M. A Methodology to Create a Fingerprint for RGB Color Image. *Internationa Journal of Computer Science and Mobile Computing*. 2017; 6(1): 205 – 212.

[14] Murala S, Maheshwari R, Balasubramanian R. Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Transactions on Image Processing*. 2012; 21(5): 2874-2886.

[15] Leon K, Mery D, Pedreschi F, Leon J. Color measurement in L* a* b* units from RGB digital images. *Food Research International*. 2006; 39(10): 1084-1091.

[16] Yam K, Papadakis S. A simple digital imaging method for measuring and analyzing color of food surfaces. *Journal of Food Engineering*. 2004; 61(1): 137-142.

[17] Heikkilä M, Pietikäinen M, Schmid C. Description of Interest Regions with Center-Symmetric Local Binary Patterns. *ICVGIP-LNCS*, 2006; 4338: 58-69.

[18] Petruk V. Application of Local Binary Patterns to Face Recognition Problem Solving. Joint Advanced Student School, Bauman Moscow State Technical University- Moscow, Russia, 2011.

[19] Gehler P, S. Nowozin. *On feature combination for multiclass object classification*. IEEE International conference of Computer Vision. Kyoto, Japan. 2009: 221–228.

[20] Henry C, Peters J. Perceptual image analysis. *International Journal of Bio-Inspired Computation*. 2010; 2(3/4): 271-281.

[21] Li J, Zhao Y, Quan D. *The combination of CSLBP and LBP feature for pedestrian detection*. Computer Science and Network Technology (ICCSNT), 3rd International Conference, Dalian, India. 2013: 543-546.

[22] Matković K, Neumann L, Neumann A, Psik T, Purgathofer W. *Global Contrast Factor - a New Approach to Image Contrast. Computational Aesthetics in Graphics, Visualization and Imaging*, 2005; 18-20.

[23] Baber J, Satoh S, Keatmanee C, Afzulpurkar N. *Improving the performance of SIFT and CSLBP for image copy detection*. Telecommunications and Signal Processing (TSP), 36th International Conference. Rome, Italy. 2013; 803-807.

[24] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*. 2002; 24(7): 971-987.

[25] Hunt R. Measuring Color”, 2nd edition. Ellis Horwood, 1996.