Automatic Redeye Correction in Digital Photos

Navid Razmjooy
Young Researchers club,
Majlesi branch, Isfahan, Iran

S. Shahram Naghibzadeh
Department of Environment,
Collage of Agriculture and
Natural Resources, Velayat
University, Iranshahr, Iran

B. Somayeh Mousavi
Department of Electrical
Engineering, Hatef Higher
Education Institute, Zahedan,
Iran

ABSTRACT

A methods to correct the artifact known as "red eye" is proposed by means of digital color image processing and classification procedure. First, skin like regions is detected with a pixel-based support vector machine processing; morphological operations are then used to eliminate the extra areas. In the second step 6 new features include geometric and color metrics are proposed to better classification of artifacts. Finally a support vector machine is used to classify the output of skin detected images by the use of presented features. 30 custom images are used to accuracy evaluation and results show the high performance of the proposed method toward some other methods.

Keywords

Redeye correction, HSI color space, Skin detection, Classification, Support Vector Machines, Morphological operations, Color and Geometry features

1. INTRODUCTION

The red eye effect has been a major problem in photographers since they first started including flash guns in their cameras; it happens when using a photographic flash very close to the camera lens (as with most compact cameras); the extent use of compact devices with built-in flash, involving cell phones and handheld computers, causes a large amount of digital photographs that are possibly affected with red eye. As shown in fig.1, an extreme flash enters the pupil from a direct angle, mirrors the blood vessels in the back of the eye, and leaves the pupil headed straight back to the camera. The light picks up most of its red color from blood in the choroid, a vascular part of the eye which supplies blood to the retina causes the exiting light red. To decrease these artifacts, most cameras have a red-eye flash mode which fires a series of pre-flashes prior to picture capturing [1]; despite detection and correction of red eye is a regular characteristic on most digital cameras today, the technology is not one-size-fits-all. Its efficiency changes based on many causes, including lighting conditions, topic and camera angle. Understanding what causes red eye effect is a good first step toward building a better camera for the makers. Fast pre-flashes make pupil contraction, thus, minimizing the area of reflection; such operation does not entirely remove the red-eye effect, but reduces it. Power consumption is the major drawback of the pre-flash approach in order to the most power consuming of flash. Besides, repeated flashes usually cause uncomfortable feeling. Various attempts have been made to reduce this problem at the time the photograph is taken.

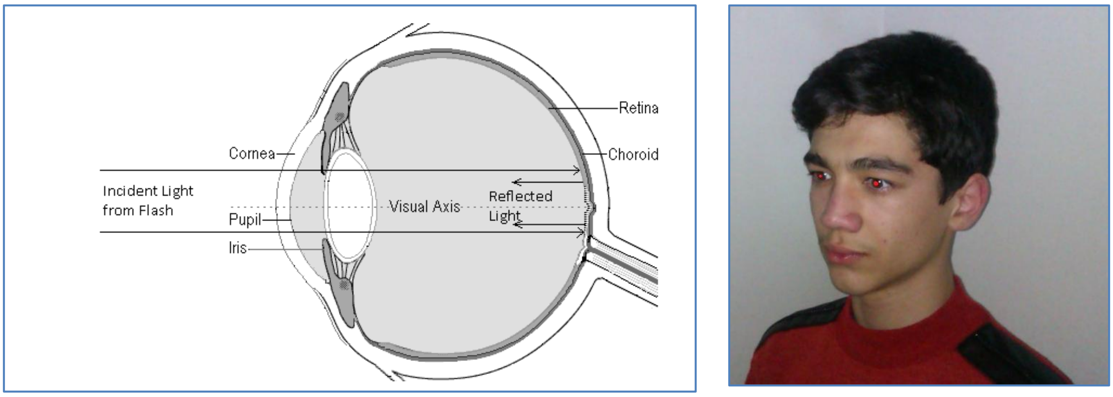


Fig.1 Flash-gun light cone produced by reflection off the retina and an example of the reflected image; the light enters the pupil and bounces off the back wall of the eye, picking up its characteristic red color along the way.

Most of them are semi-automatic or manual approaches which the user has to define the red eye pixel or draw a box containing the red eye before the red eye removal algorithm is exerted. Another method which is used to correct red eye, is automatic approaches; the advantage of such approaches is in the automatic state, which means the user need no selection of eye selection to red eye correction.

2. RELATED WORKS

Dobbs and Goodwin were apparently the first to recommend an approach for re-coloring a selected region of a digital image, and especially a red eye [2]. Benati et al. [3] and [4] regarding to the earlier work, decrease the user interface as only the general region of interest is required. In late 2003, Ulichney et al. [16] improved a one-click system called RedBot which is ready for use free on line [5]. Substantial work in red eye detection and correction was done by Hardeberg and collaborators [6-10]. At the beginning Hardeberg, [8], [9], proposed a semi-automatic method where the eye region had to select manually. After that, Hardeberg et al. developed this semiautomatic approach and improved a fully automatic method [7] which was used a preliminary color segmentation based on thresholding in different color models to identify the skin regions [10, 6]. Held proposed an intensity conversion method of these regions performed to increase the redness of the red eye defects and based on a gray level conversion of a red-enhanced image [11]. The red eye search space sequentially reduced by taking into consideration semantical information about the image, and the detection of the eye centers is foundation of the Hough transform. An automatic red eye detector based mainly on computer vision and machine vision developed by Joffe developed [12]. Described red eye detection system can be integrated by a red eye correction module to obtain a fully automatic red eye remover procedure. In 2004, Luo et al proposed a two module approach include: red eye detection and its correction [13]. Another procedure based on image classification techniques developed by Zhang et al. [14]. So that automatically characterize red eyes, a heuristic algorithm is first employed searching to a group of red like regions. Afterwards, an eye classifier utilized to confirm whether each candidate region is eye or not. A manual red eye detector is also proposed by them. In 2006 Gasparini and Schettini improved a modular approach for automatic detection and correction of human red eye in images of unknown origin [15]. Composition the results of a color-based face detector and of a face detector based on classification, the facial-like regions described. Red eyes are inspected just for within these regions, searching areas with high redness satisfying different geometric constraints. Some other techniques not based on face detection were developed by Wan et al. [16] and Volken et al. [17]. In 2004 Wan et al. proposed a procedure based on Active Appearance Models (AAM). An AAM is a computer vision algorithm to mach a statistical model of object shape and appearance to a new image. Participating color information and a deformable model they defined red eyes as deformable objects. Volken et al. in 2006 based their work on image processing and heuristics used the basic knowledge that an eye is characterized by its shape and the white color of the sclera. Combining this intuitive approach with the detection of "skin" around the eye, they obtained a high performance. In 2006 Willamowski and Csurka presented a probabilistic-based approach [18]. This algorithm was based on stepwise filtration of a pixel-wise red eye probability map. The correction part implemented a soft red eye correction establish on the resulting probability map. In 2007, Marchesotti et al. [19] have closely considered the problem of red eye correction, mostly less analyzed than red eye detection. They proposed three correction methods and compared regarding to their image degradation risk and their expected perceptual quality development. Based on those examinations an adaptive system is designed which selects the correction strategy dependent on those measures and the detection confidence.

3. NEW ALGORITHM DESCRIPTION

Two main part of the proposed system is divided to skin and red eye detection which both are detected using Support Vector Machines (SVMs). Since, a brief explanation about the SVMs is represented below.

Image classification is one of the significant tasks for many aspects of texture change studies and environmental applications in machine vision. There is several classification procedures have been improved from a simple k-NN classifier to ANN classifiers [20]. The SVM is a classification procedure based on a linear optimization theory. It efficiently trains a linear learning machine in a kernel-induced feature space, while its generalization capability is assured and overtraining is evaded. SVMs use the statistical learning and optimization theories results in order to maximize its generalization capability to train samples. Again, the SVM has been implemented successfully to different applications such texture classification [20, 21]. Therefore, we anticipate that SVM could improve performance on our skin and red eye detection system. The main purpose of SVM is to find a decision space that best classifies the data points into two classes.

The decision function in SVM is as below:

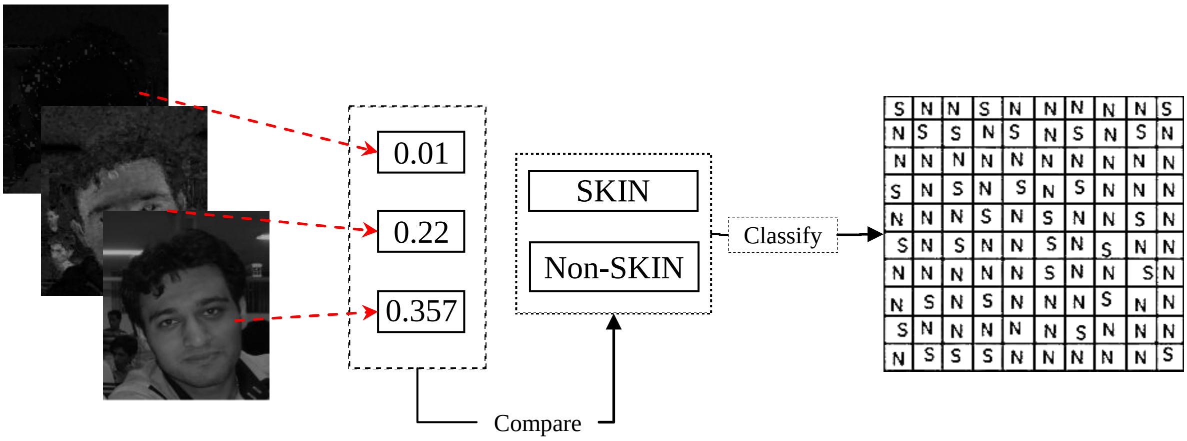
$$y = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b \right) \tag{1}$$

where x is the d-dimensional vector of a test example, y∈{-1,+1} is a class label, xi is the vector for the ith training example, N is the number of training examples, K(x, xi) is a kernel function, α={α1,…, αN and b are the parameters of the model [20, 22].

Proposed approach is organized as follows: Section 3.1 describes Detection of the skin-like areas in the color image with fast segmentation algorithms based on Support Vector Machines (SVMs) in RGB color spaces. Section 3.2 describes the whole works of the red eye detection which includes: some new features to find a proper area which can reduce the image domain to be searched to find red eye artifacts; morphological operations are then applied to eliminate the extra areas not including the red eye. Detection and segmentation of the image parts affected by the redeye effect in the skin regions is also described in this section. In section 3.3 red eye correction method is presented, Finally discussions and conclusion are described in section 4 and section 5 respectively.

3.1. Skin Detection

Skin detection in a single image is a challenging task because the overall appearance of skin color ranges. In the study presented here, main feature of the skin for classification is the color; color is a significant feature used to deduce and recollect the contents within a scene. In our approach we use color as the key tool in detecting and locating the skin regions in different scenes with complex background. To detect skin regions, a classifier is need. Classification uses to categorize all pixels in a digital image into one of several classes; classification make easy to recognize of the Information classes (i.e., skin type) of interest in the image. Skin color is one of the cases which can consider as a classification problem. In this state, skin color pixel classification uses to describe whether a color pixel is a skin color or not. Good skin color pixel classification should supply coverage of different types of skin from like blackish, yellowish, etc. This type of problem is well-suited to Support Vector Machines, which have been proven as an efficient tool for feature classification purposes. The image is then classified by attempting the performance for each pixel and making a decision about which of the pixels it resembles most; fig. 2 shows the steps of classification.



S	N	N	S	N	N	N	N	N	S
N	S	S	N	S	N	S	N	S	N
N	N	N	N	N	N	N	N	N	N
S	N	S	N	S	N	S	N	N	N
N	N	N	S	N	S	N	N	S	N
S	N	S	N	N	S	N	S	N	N
N	N	N	N	N	S	N	N	S	N
N	S	N	S	N	N	N	S	N	N
S	N	N	N	N	N	S	N	N	N
N	S	S	S	N	N	N	N	N	S

Fig. 2: Steps in Supervised Classification.

A two region classification as skin or non skin is used in this paper; we consider *pixel-based* skin detection SVM, which classify each pixel, independently from its neighbours. The proposed skin detection approach is applied to different images by various sizes under Matlab environment [23]. HSI (Hue-Saturation-Intensity) color mode as perceptual features of color is a very good choice for skin classification, since the transformation of RGB to HSI is favourably invariant to high intensity at white lights, ambient light and surface orientations relative to the light source [24]. The relationship among RGB and HS colors is shown below:

$$H^{/} = \begin{cases} undefined, if C = 0 \\ \dfrac{G-B}{C} \bmod 6, if M = R \\ \dfrac{B-R}{C} + 2, if M = G \\ \dfrac{R-G}{C} + 4, if M = B \end{cases} \tag{2}$$

(3)

$$H = 60^{\circ} \times H^{/}$$

$$S = \begin{cases} 0 & If, C = 0 \\ \dfrac{C}{V}, & O.W. \end{cases} \tag{4}$$

$$I = \frac{1}{3}(R + G + B) \tag{5}$$

The structure of the classifiers is a vector of 3×n pixel (H, S and I) coefficient vectors from each image either skin or non skin image. it will be producing the output between 0 and 255 (uint8 mode). Thus, the output of the neural network needs to be modified so that it is either 0 or 1 (double mode).

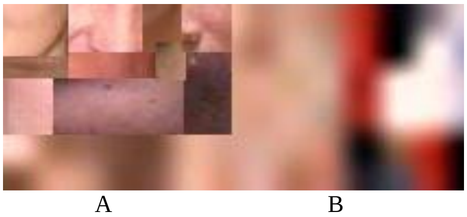


Fig.3. Test Set: (A) skin and (B) non skin sets.

The support vector machine classifies all skin regions. This is often imprecise and additional processing is needed to describe the actual skin. To this end, morphological operators are applied to the recognized skin pixel regions, morphological processing including: filling holes, and opening followed by closing are applied to the result images [25]. Region filling had to be adapted. Region filling is a synthetic algorithm including set dilation, complementation, and intersections as shown in:

$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1,2,3\ldots \tag{6}$$

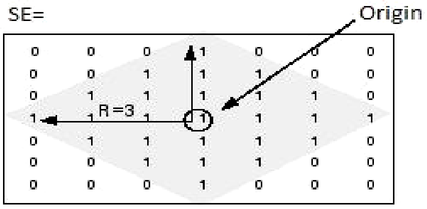
where *A* is a set of boundary and *B* is structuring Element. The algorithms terminates at iteration step *k* if $X_k = X_{k-1}$. The opening of *A* by *B* is achieved by the erosion of *A* by *B*, followed by dilation of the resulting image by *B*. it is shown as below:

$$A \circ B = (A \Theta B) \oplus B \tag{7}$$

Opening uses to remove small area noises which can't become a part of skin. Closing typically fuses narrow breaks, makes counters smooth and long thin gulfs, remove small holes, and fills gaps in the contour. The closing of set A by structuring element B, denoted A●B, is defined and mixing thin distances as below:

$$A \bullet B = (A \oplus B) \Theta B \tag{8}$$

Closing might result in amalgamations of disconnected components, which generated new holes. In this work, a flat, diamond-shaped by radius equal to 4 (this shape is good for all size of images) is selected as the structuring element.



0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

Fig.4. diamond-shaped structuring element

The Present work has implemented SVMs with features of image HSI color pixel values taken as input values, and skin or non skin pixels as the output. SVM has been applied to different applications such color classification. In this work, *C* (optimal parameter) is equal to 100, and end-accuracy is 10^{-4}. Fig.5 shows some results of this method.

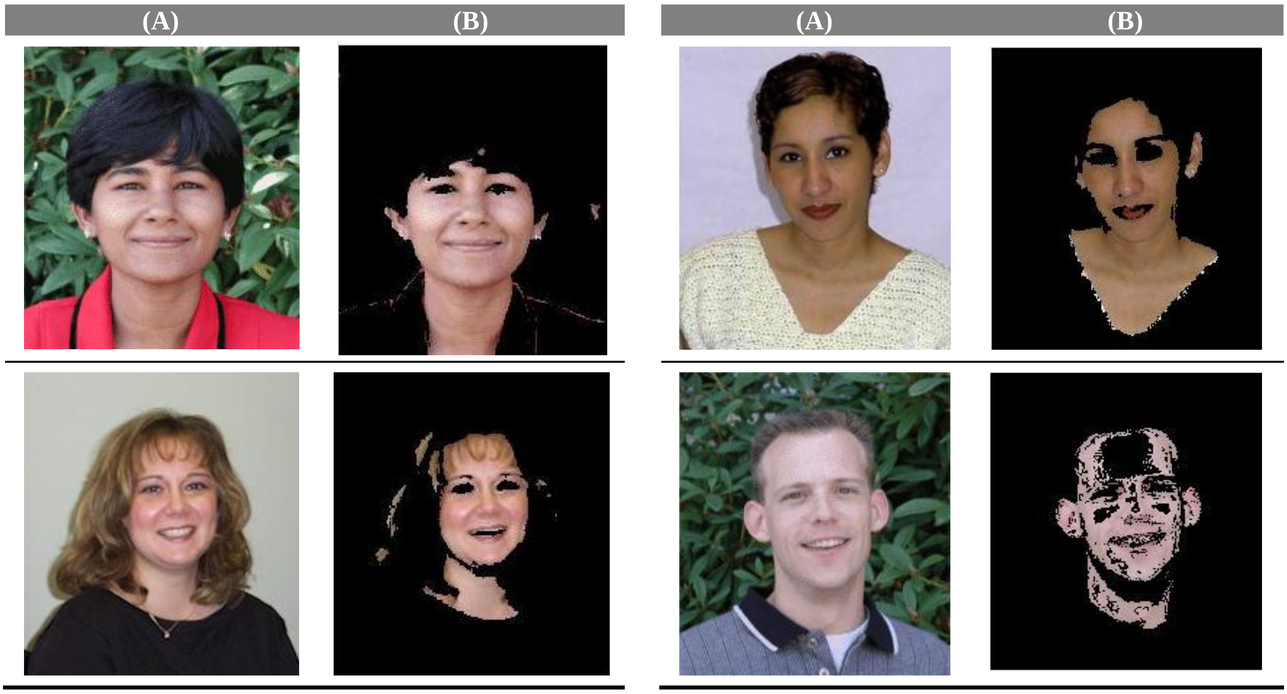


Fig.5. Skin detection: (A) Input image and (B) Output after SVM classification and mathematical morphology.

3.2. Red Eye Detection

In our procedure, the structure of the red-eye detector is like to the structure of the skin-detector. The red-eye detector comprises of six feature metrics to mask non-red-eye pixels which divided into two separate sections:

3.2.1. Color metric

The color metric emphasizes redness and several types of changes in luminance and redness. The resulting values are binarized to create a mask that eliminates most non-red-eye pixels. In this section, we are used the 3 following ratios a measure of the pixel's redness:

$$F1 = \cup\big((R-G),(R-B)\big) = 2R-G-B \qquad (9)$$

$$F2 = F1 - mean\{\max\{F1\}\} + mean\{\max\{F1\}\} \qquad (10)$$

$$F3 = Cr = 128 + \frac{112.439 \times R_D^{/}}{256} + \frac{94.154 \times G_D^{/}}{256} + \frac{18.258 \times B_D^{/}}{256} \qquad (11)$$

where the prime (′) symbols mean Gamma Correction is being used; thus R′, G′ and B′ nominally range from 0 to 1, with 0 representing the minimum intensity and 1 the maximum.

3.2.2. Geometric Constraints

To limit the number of false hits, the binary roundness constraint *F4* of the connected a definite region is verified as follows:

$$F4 = \begin{cases} True & |AP| \le 1 \\ False & |AP| > 1 \end{cases} \qquad (12)$$

Where

$$AP = 1 - \frac{Area}{Perimeter} \qquad (13)$$

F5 is an other feature for eliminate the big objects which wasn't removed by skin detection and also red-eye color detection.

$$F5 = \frac{\text{proportion of the labeled region}}{\text{proportion of the pixels in the whole image}} \le 0.25 \qquad (14)$$

F6 is finally used to remove the unlike to eye shape which have the same color and weren't removed in the previous steps.

$$F6 = \begin{cases} True & \dfrac{length..of.the.label.image}{width.of.the.label.image} \ge 1.5 \\ False & O.W. \end{cases} \qquad (15)$$

After achieving 6 definite features, they trained to an SVM. The output of this classifier is divided into two parts; 1 for red-eye regions and 0 for irrelevant objects. Morphological processing including: filling holes, and opening followed by closing are applied again to a better performance.Some of the results are shown in fig.6.

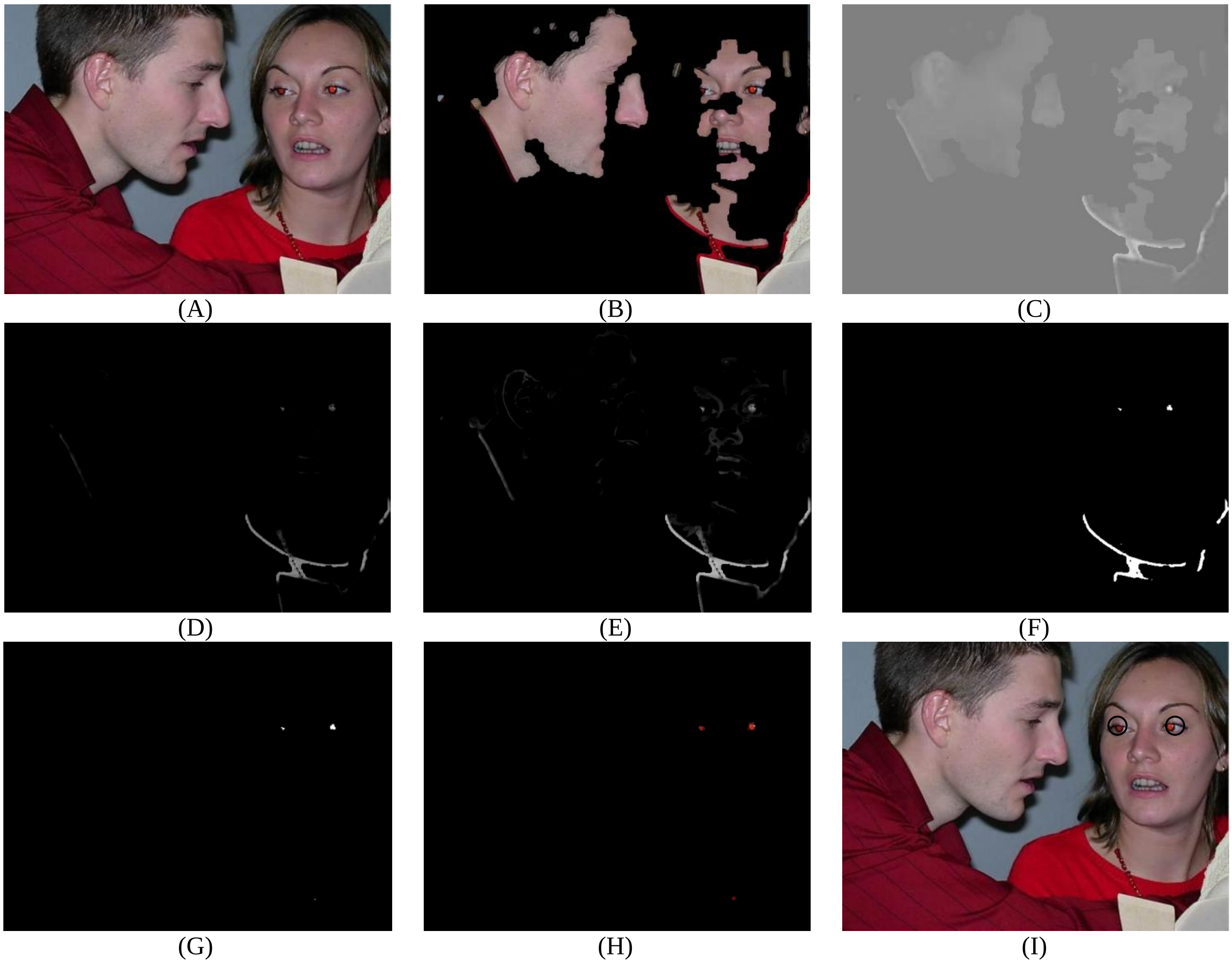

(A) (B) (C)
(D) (E) (F)
(G) (H) (I)

Fig.6. Red-Eye detection: (A) Input image, (B) Skin detection, (C), (D) and (E) are the color metric F3, F2 and F1 respectively, (F) is the image after red-eye classification, (G) is the red eye mask, (H) is locating red eye in the original image and finally (I) is located red eye artifacts.

3.3. Red Eye Correction

After red-eye detection, the correction step is performed. Generally the redeye artifact comprise of a glint red pupil. Red eye correction step is to absorb the light of this area and make it dark. A new color-based procedure is used to transform the red pupil to a dark region. The region of red pixels is employed to arrange the area that must be de-colored. In this situation, if a pixel has been detected as a red eye artifact, it is replaced by the following equation to all 3 dimensions from R, G and B:

$$\alpha R - \beta(G + B) \qquad (16)$$

.

where

$$\alpha = \min\{mean\{skin.like.pixels\}\}, \beta = \max\{mean\{skin.like.pixels\}\} \qquad (17)$$

After red eye detection, eq.16 is applied as a mask for just correcting red eye pixels. Because of using R, G and B as a main color features in the red eye detection, we decided to use them also to eliminate the artifacts by decreasing the red color as we can; since a less coefficient (α) is used to the red dimension whereas more coefficient (β) is used to enlarge other dimensions of the input image. Fig.7 shows some results of red eye correction

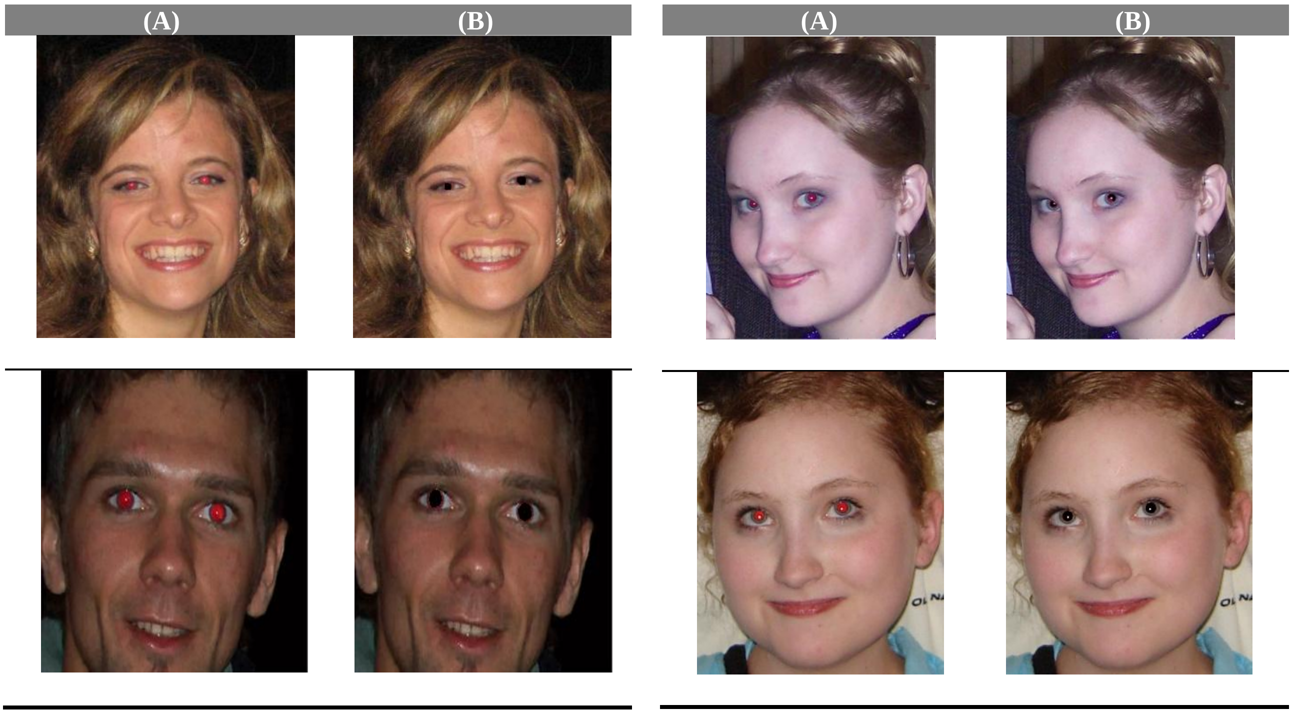


Fig.7. Skin Correction: (A) Input image and (B) Corrected image.

4. DISCUSSION

Two sections are investigated in this paper: two zone classification first as skin or non-kin, and second as red eye and non-red eye; we discuss *pixel-based* red eye detection classifier, which classify each pixel, independently from its neighbours and the structure of both classifiers is a vector of 3×*n* pixel coefficient vectors from each image which *n* is the number of input pixels for classification. The experiment result of the proposed approach is then compared by RediGone [26], Phixr [27] and Stop Red eye. To evaluate the performance of the proposed algorithm toward the other conditional methods, three performance metrics are defined: The first metric is the correct detection rate **(CDR)** and is given in Equation (18). The false acceptance rate **(FAR)** is the percentage of recognition moments in which false acceptance occurs. The false rejection rate **(FRR)** is the percentage of recognition moments in which false rejection occurs. The FAR and FRR are expressed in Equations (19) and (20), respectively; these methods are applied on a custom database which comprises different databases by 30 images.

$$CDR = \frac{No.of.Pixels.Correctly.Classified}{Total.Pixels.in.the.Test.Dataset} \tag{18}$$

$$FAR = \frac{No.of.non - potato.Pixels.Classified.as.potato.PixelsClassified}{Total.Pixels.in.the.Test.Dataset} \tag{19}$$

$$FRR = \frac{No.of.potatoPixels.Classified.as.non - potato.PixelsClassified}{Total.Pixels.in.the.Test.Dataset} \tag{20}$$

Tables 1 presents the performance of compared methods include: red eye gone, phixr and stop red eye in accuracy.

Table 1. Classification Comparison of Performance in the presented procedures

ACCURACY METRIC	METHOD			
	Proposed Method	Red Eye Gone [26]	Phixr [27]	Stop Red Eye!
CDR(%)	83.3	50.0	63.3	70.0
FAR(%)	10.0	30.0	23.3	20.0
FRR(%)	06.6	20.0	13.3	10.0

Our approach was tested on various images with different sizes. Some examples of the proposed procedure are shown in Figs 8-10.

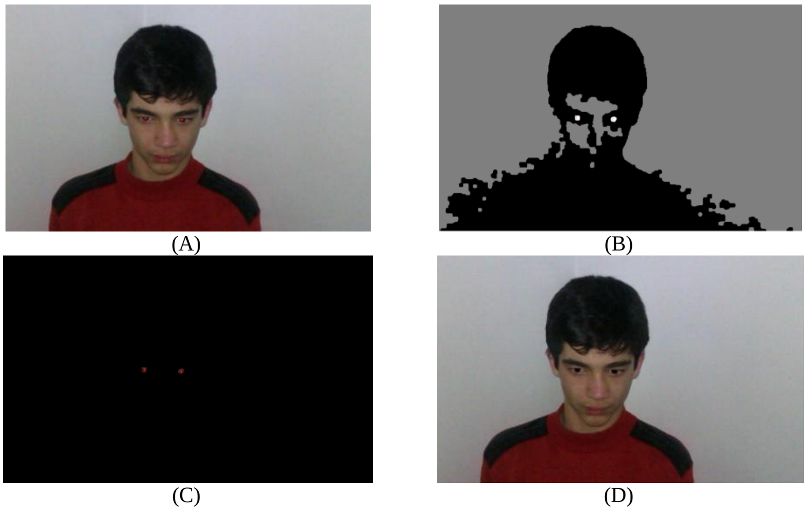


Fig.8. Skin detection step of the proposed approach enables to meaningfully decrease the probability of false redeye detection: (A) input image, (B) skin and red-like areas detection (black areas show the background, gray areas show the skin-like pixels, and white areas show the reddish pixels), (C) result with found red regions after skin detection, (D) final result of redeye correction.

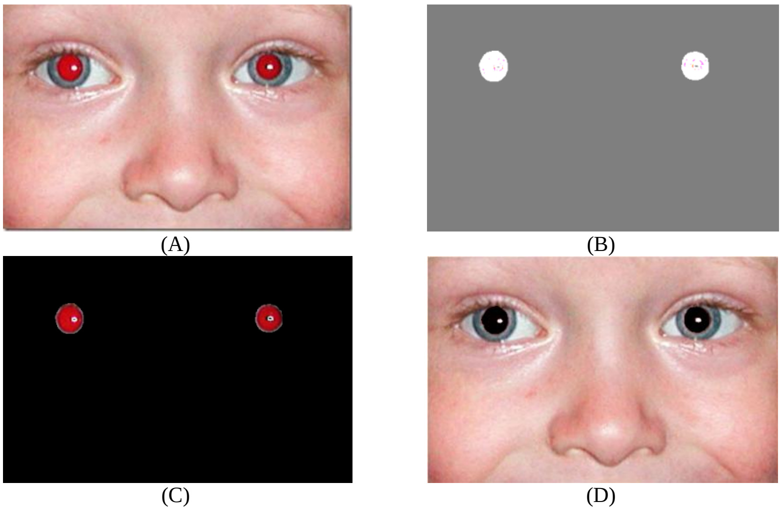
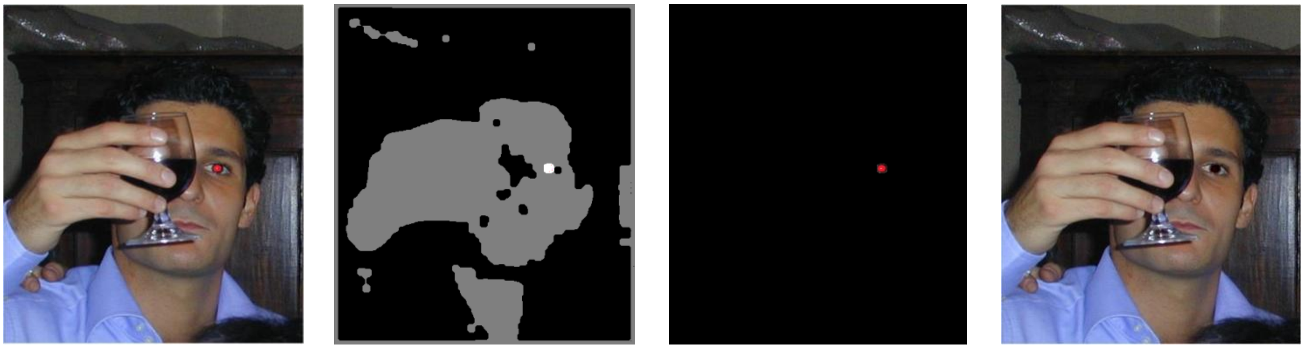
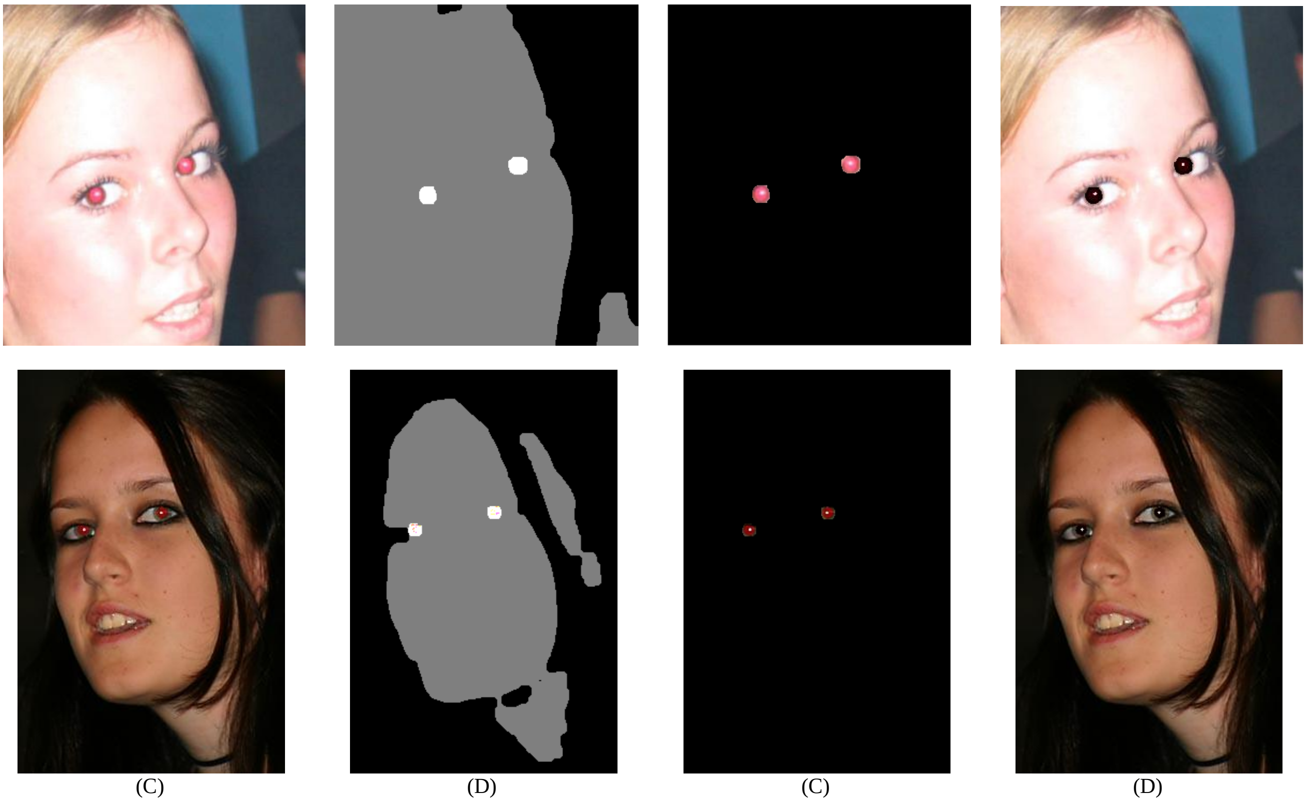


Fig.9. Skin detection step of the proposed approach enables to meaningfully decrease the probability of false redeye detection: (A) input image, (B) skin and red-like areas detection (black areas show the background, gray areas show the skin-like pixels, and white areas show the reddish pixels), (C) result with found red regions after skin detection, (D) final result of redeye correction.





(C) (D) (C) (D)

Fig.10. Skin detection step of the proposed approach enables to meaningfully decrease the probability of false redeye detection: (A) input image, (B) skin and red-like areas detection (black areas show the background, gray areas show the skin-like pixels, and white areas show the reddish pixels), (C) result with found red regions after skin detection, (D) final result of redeye correction.

5. CONCLUSION

This paper presented an algorithm to develop the detection of red eye artifacts using of image processing and classification technique. Our approach is using support vector machines to classify the red eye artifacts in two step: first, skin regions are detected by the SVMs, after that by using 6 new features (which contain both color and geometric features) and learning them to the SVM, red eye areas are detect. The results showed 83.3 percents accuracy, 10 percents FAR and 6 percents FAR.

6. REFERENCES

[1] Sebastiano Battiato, GiovanniMaria Farinella, Mirko Guarnera, GiuseppeMessina,and Daniele Rav`, Red-Eyes Removal through Cluster-Based Boosting on Gray Codes, Hindawi Publishing Corporation , EURASIP Journal on Image and Video Processing, doi:10.1155/2010/909043,Vol2010, 2010.

[2] C. M. Dobbs and R. Goodwin, “Localized image recoloring using ellipsoid boundary function,” US Patent 5130789, July 1992.

[3] P. Benati, R. Gray, and P. Cosgrove, “Automated detection and correction of eye color defects due to flash illumination,” US Patent 5432863, July 1995.

[4] P. Benati, R. Gray, and P. Cosgrove, “Automated detection and correction of eye color defects due to flash illumination,” US Patent 5748764, May 1998.

[5] “Redbot automatic red eye correction,” http://redbot.net/, Redbot Hewlett-Packard Labs.

[6] B.Smolka, K. Czubin, J. Y. Hardeberg, K. Plataniotis, M. Szczepanski, and K. Wojciechowski, “Towards automatic redeye effect removal,” Pattern Recognition Letters, vol. 24, pp. 1767–1785, July 2003.

[7] J. Y. Hardeberg, “Red-eye removal using color image processing,” US Patent 6728401, April 2004.

[8] Quality, Image Capture System Conference, (Montreal, Canacda), pp. 283–287, April 2001.

[9] J. Y. Hardeberg, “Digital red eye removal,” Journal of Imaging Science and Technology, vol. 46, pp. 375–381, July/August 2002.

[10] K. Czubin, B.Smolka, M. Szczepanski, J. Y. Hardeberg, and K. Plataniotis, “On the redeye effect removal algorithm,” in Proceedings of the First European Conference on Color in Graphics, Imaging and Vision (CGIV), (Poitiers, France), pp. 292–297, April 2002.

[11] A. Held, “Model-based correction of red eye defects,” in Proceedings of the IS&T/SID Tenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, (Scottsdale, Arizona), pp. 223–228, November 2002.

[12] S. Ioffe, “Red eye detection with machine learning,” in Proceedings of the IEEE International Conference on Image Processing (ICIP-2003), vol. 2, (Barcelona, Spain), pp. 871–874, September 2003.

[13] H. Luo, J. Yen, and D. Tretter, “An efficient automatic redeye detection and correction algorithm,” in Proceedings of the 17th International Conference on Pattern Recognition (ICPR-2004), vol. 2, (Cambridge, UK), pp. 883– 886, August 2004.

[14] L. Zhang, Y. Sun, M. Li, and H. Zhang, “Automated red-eye detection and correction in digital photographs,” in Proceedings of the IEEE International Conference on Image Processing (ICIP-2004), vol. 4, (Singapore), pp. 2363–2366, October 2004.

[15] F. Gasparini and R. Schettini, “Automatic redeye removal for smart enhancement of photos of unknown origin,” Lecture Notes in Computer Sciences, vol. 3736, pp. 226–233, 2006.

[16] J. Wan, X. Ren, and G. Hu, “Automatic red-eyes detection based on aam,” in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 7, (The Hague, The Netherlands), pp. 6337–6341, October 2004.

[17] F. Volken, J. Terrier, and P. Vandewalle, “Automatic red-eye removal based on sclera and skin tone detection,” in Proceedings of the IS&T Third European Conference on Color in Graphics, Imaging and Vision (CGIV), (Leeds, UK), pp. 359–364, June 2006.

[18] J. Willamowski and G. Csurka, “Probabilistic automatic red eye detection and correction,” in Proceedings of the 18th International Conference on Pattern Recognition (ICPR-2006), vol. 3, (Hong Kong), pp. 762–765, August 2006.

[19] L.Marchesotti, G. Csurka, and M. Bresssan, “Safe red-eye correction plug-in using adaptive methods,” in Proceedings of the International Conference on Image Analysis and Processing (ICIAP-2007), (Modena, Italy), In print 2007.

[20] S. Theodoridis, & K. Koutroumbas, Pattern Recognition, Third Edition, Academic Press, CA, 2006.

[21] R.G. Brereton, & G.R. Lloyd, Support Vector Machines for Classification and Regression, *Analyst*, 135(2), 2010, 230-267.

[22] K.I. Kim, K. Jung, S.H. Park, & H.J. Kim, Support Vector Machines for texture classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11), 2002,1542-1550.

[23] Matlab Tutorial, Mathworks, 2007.

[24] P. Kakumanu, S. Makrogiannis, N. Bourbaki, A survey of skin-color modeling and detection methods, Pattern Recognition 40 (3) (2007) 1106–1122.

[25] L. yang jun, W. Jiawen, *Graph and Image Processing based on MATLAB 7.0*, National Defense Industry Press, 2006, pp.226-276.

[26] The red eye remover tool: http//www.redigone.com

[27] The online photo editor: http://www.phixr.com/