International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319–9598, Volume-1, Issue-3, February 2013

An Enhanced Technique for Red-Eye Detection and Correction using Neural Network

Rashmi Chandra, Rohit Raja

Abstract— Redeye is a common problem in consumer photography. When a flash is needed to illuminate the scene, the ambient illumination is usually low and a person's pupils will be dilated. Light from the flash can thus reflect off the blood vessels in the person's retina. In this case, it appears red in color and this reddish light is recorded by the camera. Though commercial solutions exist for red-eye correction, all of them require some measure of user intervention. A method is presented to automatically detect and correct red-eye in digital images. The algorithm contains a redeye detection part and a correction part. The detection part is modeled as a feature based object detection problem. Adaboost is used to simultaneously select features and train the classifier. A new feature set is designed to address the orientation-dependency problem associated with the Haar-like features commonly used for object detection design. For each detected redeye, a correction algorithm is applied to do adaptive desaturation and darkening over the redeye region. . The experimental results indicate that, the system can remove the red-eye automatically and effectively in the digital photo and has good robustness and rapidity.

Index Terms— Redeye detection, redeye correction, face detection, image processing, neural network.

I. INTRODUCTION

Everyday millions of photographs are taken from all around the world. Photography is a profession. It is a hobby. However, to most people it is a means to capture and relive memories. Great advances have taken place in the world of photography, with the transition to color being one of the most prominent. In spite of all the advancements, one problem is still very much evident. The problem is the reddish coloration of the pupils of the human eyes that can occur in photographs. This phenomenon is known as the Red-Eye effect.

Red-eye is a result of using flash photography. The red color of the pupil is due to the flash reflecting from blood vessels at the back of the eye. The effect is worsened as the angle of reflection between the flash and the lens decreases.

Red-eye has become increasingly more prevalent in recent years particularly with the advent of digital cameras. Digital cameras do not require any film, as a result the cameras are easier to maintain and can be built to very compact dimensions. Also the facility to transfer images directly to a computer has made digital cameras very popular. It is not surprising that there has been a huge growth in digital camera sales in recent years.

Digital cameras do not suffer the problem of running out of film. The cameras do not incur the cost of having to replace film or the cost to develop the pictures in order to see them, as images can simply be viewed on the computer. People as a result are less hesitant in taking lots of photos and also less inclined to take the necessary precautions that would otherwise lessen the effect of red-eye. These precautions are as simple as taking the photo in an area that has appropriate lighting, or making use of the camera's pre-flash.

The diminishing size of the camera has contributed significantly to the problem. As the size of a camera shrinks, so too does the distance between the flash and the lens, decreasing the angle of reflection. In short red-eye is a large and widespread problem.

Fig.1 shows the angle of reflection. The red-eye cone shines from the flashed eye back at the flash with an angle α. Its red color is caused by the reflection of the flash off the blood vessels of the retina; the camera will record this red hue if the angle β between the flash and the camera is not greater than α.

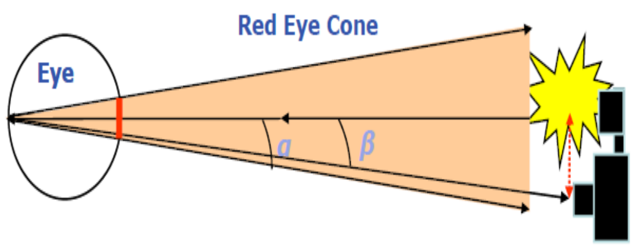

FIG. 1: ANGLE OF REFLECTION

The appearance of red-eye in flash photography is one of the most prevalent and disturbing artifacts that affects image capture. This problem has long bothered camera users, and is especially common with compact cameras, due to the inherently small angle between the lens and the flash. Fig.2 illustrates the red-eye beacon that shines from a subject's eye in the presence of a flash. This beacon is a cone of radius α shining back at the flash. Its apparent red color is caused by the reflection of the flash off the blood vessels of the subject's retina. The camera will record this red hue if the angle between the flash and camera, β,□ is not greater than α.

All algorithmic solutions for fixing red-eye can be segmented into two parts: detection and correction. By far the more difficult analytical and computational problem is the detection phase, that is, the process of correctly identifying the precise pixels in the input image that contain red-eye artifacts. Detection solutions are measured by percent correctly identified and false positive rate. Once the red-eye regions are identified, the second phase is correction where the offending pixels are to be changed.

A variety of solutions have been proposed for detection. Several methods [1,10] use a series of classifier-based modules that perform initial candidate selection and verification. Another [2] uses face detection as a starting point for looking for red-eyes, using a fast, robust face detector [3]. A

Manuscript received on February 11, 2013.

Rashmi Chandra, Department of Computer Science and Engineering, Shri Shankaracharya College of Engineering & Technology, Bhilai (C.G.), India.

Rohit Raja, Department of Computer Science and Engineering, M..M.. College of Technology, Raipur (C.G.), India.

wavelet-based face-detector approach [9] has also been presented. Instead of staring explicitly with faces, skin detection can be employed to narrow the search for red-eye artifacts [5,8]. These approaches have trade-offs in success rates, and detection solutions continue to evolve.

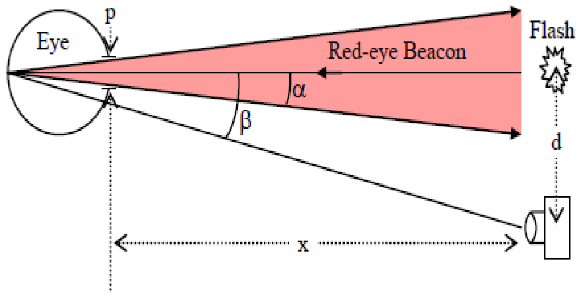

Fig. 2: The geometry of photo-flash red-eye.

The proposed system is divided into part:
- ➢ Red eye detection and
- ➢ Red eye correction

Red Eye Detection:
The red eye detection is divided into two parts:-
1 Training phase
2 Understanding phase

1 Training Phase
In the training phase we require to train a system by taking known image that is image that has no red eye, by preprocessing and cropping its remove the noise, image enhancement etc. In feature selection and extraction we calculate the threshold value and store in the database.

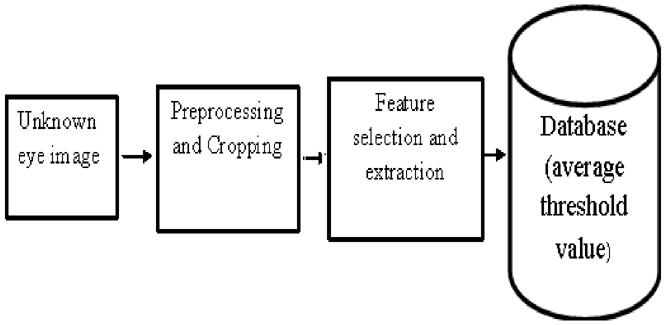

Fig.3:Flow chart of the training phase

- **Input eye image:** In this known input image is inputted i.e. without red eye image**.**
- **Preprocessing:** The purpose of preprocessing is to improve the quality of the image being processed. Preprocessing tasks that solve frequent system problems, such as interfering noise, low dynamic range, out-of-focus optics, and the difference in color representation between input and output devices.

The theme of the technique of magnification is to have a closer view by magnifying or zooming the interested part in the imagery. By reduction, we can bring the unmanageable size of data to a manageable limit. For resampling an image Nearest Neighborhood, Linear, or cubic convolution techniques are used.

I. Magnification
This is usually done to improve the scale of display for visual interpretation or sometimes to match the scale of one image to another. To magnify an image by a factor of 2, each pixel of the original image is replaced by a block of 2x2 pixels, all with the same brightness value as the original pixel.

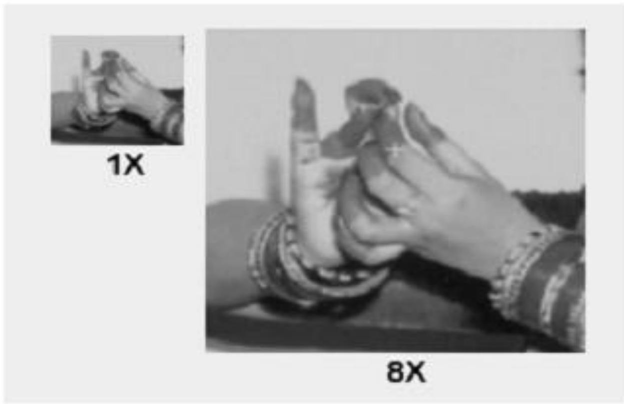

Fig.4: Image Magnification

II. Reduction
To reduce a digital image to the original data, every m^{th} row and m^{th} column of the original imagery is selected and displayed. Another way of accomplishing the same is by taking the average in 'm x m' block and displaying this average after proper rounding of the resultant value.

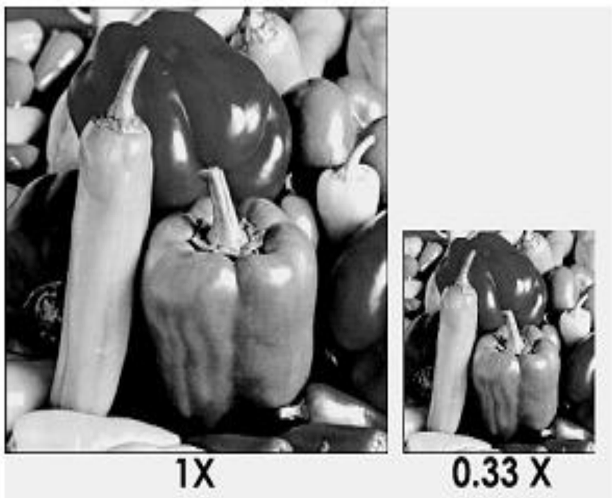

Fig.5: Image reduction

- **Cropping:** Cropping creates a new image by selecting a desired rectangular portion from the image being cropped. The unwanted part of the image is discarded. Image cropping does not reduce the resolution of the area cropped. Best results are obtained when the original image has a high resolution. A primary reason for cropping is to improve the image composition in the new image.
- **Feature selection:** Pupil part should be black or red colour, and outer part is white colour. Pupil shape must be oval or round.
- **Extraction:** We extract that part of image which is oval or round and black in colour.
- **Database :** average threshold value is stored in the database.
- **Threshold Value:** In black area of pupil there may be many pixel we take the average of that all point and that value is called threshold value which is to be stored in the database for each image.

1 Understanding phase

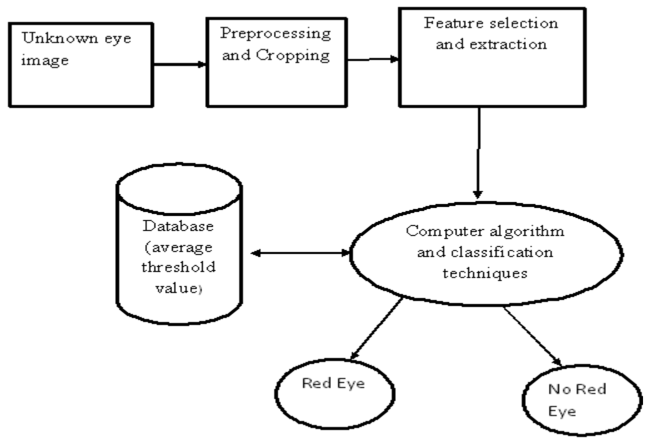

Fig.6: Flow chart of the understanding phase.

- **Input eye image:** We input eye image i.e. unknown image it may be red eye or non red eye.
- **Preprocessing:** The purpose of preprocessing is to improve the quality of the image being processed. In preprocessing image noise is remove.
- **Cropping:** Cropping creates a new image by selecting a desired rectangular portion from the image being cropped. The unwanted part of the image is discarded. Image cropping does not reduce the resolution of the area cropped. Best results are obtained when the original image has a high resolution. A primary reason for cropping is to improve the image composition in the new image.
- **Feature selection:** Pupil part should be black colour, and outer part is white colour. Pupil shape must be oval or round.
- **Extraction:** We extract that part of image which is oval or round and black in colour.
- **Classification techniques:** it is a machine learning technique used to predict group of membership for data instances For classification we use Bayes classification technique to classify the eye.
- **Database :** It matches with threshold values of known images stored in the database training phase.
- **Red Eye:** If the threshold value matches with the threshold value stored in the database by the known image, then unknown image is red eye.
- **Not Red Eye:** If the threshold value does not matches with the threshold value of unknown image then it is not red eye

In the understanding phase we input the image which is unknown i.e. we don't know whether it is red eye or not. In this phase also preprocessing and cropping is done to remove noise and cropping is done to select particular part of the image i.e. eye part. Then classification technique is applied, in this some algorithm is applied to calculate the threshold value. Then it is matched with the previously stored threshold value of known images. If the threshold value matches of known images with the unknown threshold value then it is not red eye. If it does not match then it is red eye.

Red eye correction:

In understanding phase it is clear that whether it is red eye or not. If red eye is found then it should be corrected. If it red eye through photography then it is done by selecting that part of area of eye which is red we have calculated in the threshold value of red eye and changing the red color into black color. And if it is due to medical problem then it is treated by the doctor.

Face detection is widely applied in all facial analysis, content-based image retrieval, video conferencing and intelligent human computer interaction (HCI). Evidently, face detection is the first step for the above applications. So whether the face detection is accurate is the most important process[24].

The goal of face detection is to determine whether or not there are any faces in the image and if present, return the image location and extent of each face. It is a very easy task for human vision, but for computer many variations in scale, location, orientation, pose, facial expression, light condition, presence of glasses, facial hair and makeup etc make it difficult.

How to evaluate the performance of the detection methods? There are many metrics such as learning time, execution time, the number of samples required in training, and the ratio between detection rates and false alarms. Here detection rate is the ratio between the number of faces correctly detected and the number faces determined by a human[31,33]. And an image region is correctly detected as a face when the region covers more than a certain percentage of a face in the image. Generally, there are two types of errors of face detection: false negatives in which faces are missed because of low detection rates and false positives in which an image region is detected as face but it is not. There have been a lot of datasets for test and learning, but unfortunately, there is a lack of uniformity in evaluation.

There have been hundreds of reported approaches to face detection in the past decades. Yang et al. [25] classified the various methods into four major categories: knowledge-based, feature invariant, template matching and appearance-based methods. Knowledge-based methods use rules based on human knowledge of faces to localize faces. Feature invariant approaches locate faces by finding structural features of faces. Template matching methods use predefined or parameterized face templates to locate and detect face. Appearance-based methods use learned models from training images which should capture the facial appearance for detection. Generally, appearance-based methods had been showing superior performance to the others [24,25] emphasized Viola-Jones face detector as the most impact in the past decade, because it has been used in the real world applications such as digital cameras, digital photo management softwares.

Red-eye Correction first left eye is corrected then right eye is corrected.

1) Face detection
2) Eye crop
3) Region is selected
4) Red-eye correction

1) Face Detector Algorithm

Viola-Jones Face Detector has three distinguished key contributions Integral Image, variant AdaBoost learning algorithm and Cascade structure to achieve high processing speed and detection rates [30,31]. The detection rates of Viola-Jones face detector are comparable to the best previous systems. So it has been used in real-time applications.

a) Haar-like Features

They used three kinds of features: two-rectangle feature, three rectangle feature and four-rectangle feature.

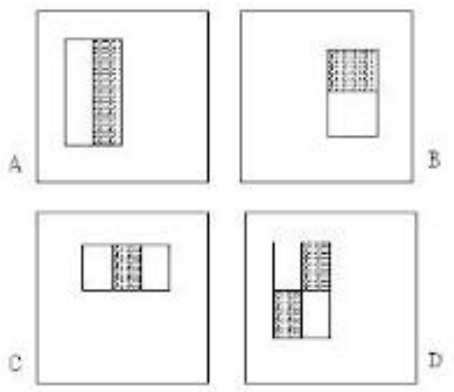

Fig.7: Haar like rectangle feature A-D

The value of a Haar-like feature is the difference from the sum of pixels in the grey rectangles subtracting the sum of the pixels in the white rectangles. The grey or white regions have the same size and shape in one feature. Given a detector with the base resolution 24x24, the exhaustive set of features is over 180,000.

b) Integral Image

In order to compute these Haar-like features very rapidly at many scales the integral image representation is introduced. Any Haar-like feature can be computed at any scale or location very quickly. The integral image at location x,y contains the sum of the pixels above and to the left of x,y in the original image as bellows.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Where ii(x,y) is the integral image at picture location (x,y) and i(x',y') is the original image. Any rectangular sum can be computed in four array references. Like the example in bellowing figure, to compute the sum of D which is equal to 4 + 1 - 2 - 3, so we need the value at 1, 2, 3 and 4 in integral image.

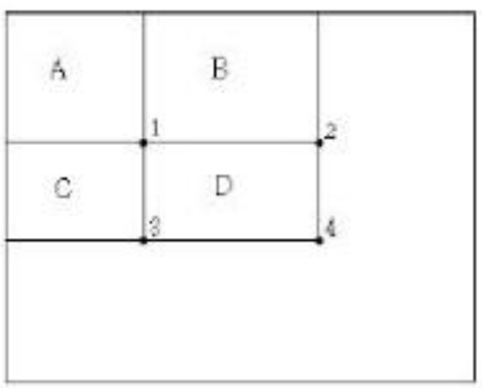

Fig.8 Integral image

c) Learning Algorithm based on AdaBoost

Within any image subwindow the total number of Haar-like features is very large. In order to ensure fast classification, the learning process must focus on a small set of critical features. Here, a modified AdaBoost is used not only to select a small set of features but also to train the classifier as belows.

Input:Given training feature vectors(x_1,y_1),……,(x_n,y_n), where y_i=0,1 for negative and positive examples respectively.

Boosting Algorithm:

I. Initialize weight $w_i^{(0)} = \frac{1}{2m}, \frac{1}{2l}$ for yi=0,1 respectively, where m and l are the number of negative and positive examples respectively.

II. For t=1,…,T:

1) Normalize weights: $w_i^{(t)} = w_i^{(t-1)} / (\sum_{k=1}^{n} w_k^{(t-1)})$.
2) For each feature j, train a classifier $h_j^{(t)}()$ that is restricted to using a single feature that minimizes the classification error $e_j^{(t)}=\sum_i w_i^{(t)} \backslash h_j(x_i) - y_i \backslash$.
3) Choose the weak learner function as the $h_j^{(t)}$ with the lowest error $e_j^{(t)}$, e.g., $H_t() = h_{j_0}^{(t)}(), e_t = e_{j_0}^{(t)}$, where j_0=arg $\min_j(e_j^{(t)})$.
4) Update the weights: $w_i^{(t)} = w_i^{(t)} \beta_t^{1-\epsilon_i}$, where ϵ_i=0 if example x_i is classified correctly ; ϵ_i=1 otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

Output: the final classifier is:

$$H(x) = \begin{cases} 1 \sum_{t=1}^{T} \alpha_t H(x) > 0.5 \sum_{t=1}^{T} \alpha_t \\ 0 \quad \text{otherwise} \end{cases}$$

Where $\alpha_t = -\log \beta_t$.

Using the AdaBoost we need a simple learning algorithm which selects the single rectangle feature which best separates the positive and negative examples.

A simple classifier $h_j(x)$ thus consists of a feature f_j, a threshold θ_j and parity p_j indicating the direction of the inequality sign[3]:

$$H_j(\infty) = \begin{cases} 1 \text{ if } p_j f_j(\infty) < p_j \theta_j \\ 0 \quad \text{otherwise} \end{cases}$$

Where x is a 24x 24 pixel sub-window of an image.

d) Cascade Structure

Cascade structure increases the speed of the detector by focusing on face-like regions of the image. It is a sequence of more and more complex classifiers that sub-windows which are not rejected by the last classifier will be processed by the next more complex classifier. The cascade structure contains many classifier stages. In each stage there is a strong classifier trained by using modified AdaBoost on increasingly features until the target detection and false positives rates are met. In the test of original Viola-Jones Detector [3], the cascade has 38 stages with over 6000 features.

2) Eye cropped

In this the eye part is cropped

3) Region is selected

After eye part is cropped then region is selected where the eye part is red then it is corrected using the following algorithm

If red>80 && if green>100 && if blue>100
Then cur=cur+ 40
Cug=cug+10
Cub=cub+10

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In Table 1 shows the values of the area cropped in non red eye image using Jimage software. In this area, mean min, max, xm, ym, circ., ar, round and solidity value is calculated.

Table 1

s.no .	Area	Mean	Min	Max	XM	YM	Circ.	AR	Solidity
1	390	136.72	19	255	11.93	9.022	0.981	1.365	1
2	944	66.84	1	255	24.75	11.642	0.85	2.237	1
3	660	103.06	10	255	19.17	0.03	0.90	1.89	1
4	253	104.98	9	255	12.501	6.48	0.89	1.91	1
5	860	79.39	4	255	26.85	10.31	0.82	2.47	1
6	1544	97.37	6	255	29.68	18.09	0.95	1.6	1
7	560	125.98	5	255	22.16	8.21	0.79	2.72	1
8	776	105.69	0	255	18.83	12.30	0.96	1.45	1
9	610	107.33	3	255	18.78	10.80	0.95	1.58	1
10	1069	125.49	6	255	22.46	16.39	0.99	1.23	1

In Table 2 shows the values of the area cropped in red eye image using Jimage software. In this area, mean min, max, xm, ym, circ., ar, round and solidity value is calculated.

S.No.	Area	Me	Min	Max	XM	YM	Circ.	AR	Solidity
1	226	120	19	255	11.308	6.423	0.94	1.676	1
2	79	139	49	255	4.601	5.449	1	1.188	1
3	170	127	22	255	10.687	4.698	0.901	2.125	1
4	210	151	22	255	8.684	7.177	0.982	1.352	1
5	104	138	27	255	5.587	5.909	1	1.134	1
6	161	121	20	255	5.579	9.127	0.911	1.696	1
7	124	152	21	255	6.419	5.987	1	1.071	1
8	79	138	28	255	4.55	5.114	1	1.188	1
9	190	122	19	255	7.007	8.701	1	1.182	1
10	102	155	24	255	5.139	6.281	0.982	1.315	1

In fig.9 shows the removal of red eye.

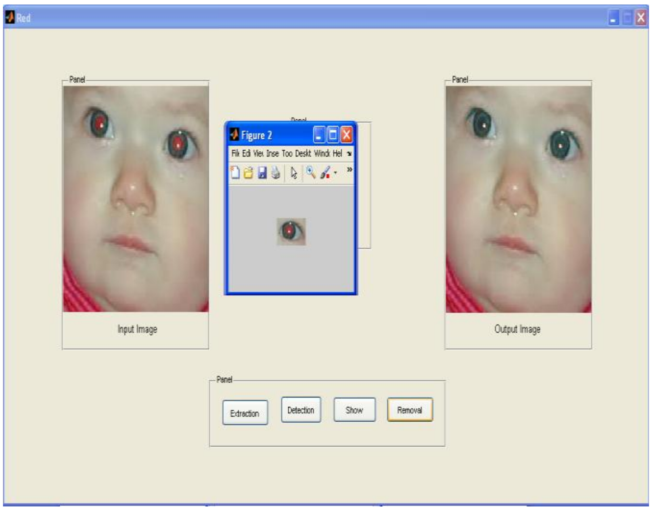

Fig.9: red eye removal

II. PERFORMANCE MEASURED

For estimation of detection quality, a quantitative quality criterion is proposed which was compared with existing automatic solutions. This algorithm is tested over data set, composed of 16 color images of different sizes (from 120 x 160 to 2274 x 1704 pixels), resolutions and quality. These images where downloaded from web-pages. Single red-eyes, as well as high variability of red-eyes colors, poses and shapes have been considered in building the dataset, the hit-rate of the proposed red-eyes detector is 80%.

III. CONCLUSION

In this approach an automatic redeye detection and correction algorithm has been presented. Faces are detected using face detection based on adaBoost algorithm, and then the red-eyes are located using kinds of image processing method. At last the located red-eyes are finished corrected. The main problem in an automatic procedure applied to digital photos is to avoid the correction of false positives, while maintaining high correction rates and quality. The method is fast, computationally inexpensive, reliable.

The red-eye removal has been tested on a dataset of 16 images in which 8 red-eyes have been manually labeled. The dataset has been collected from various sources, compact cameras, personal collections and Internet photos. Single red-eyes, as well as high variability of red-eyes colors, poses and shapes have been considered in building the dataset, the hit-rate of the proposed red-eyes detector is 80%. The experimental result on 16 redeye images show that the proposed red-eye reduction solution is robust and efficient.

ACKNOWLEDGMENT

The authors are thankful to Dr.P.B.Deshmukh, Director, Shri Shankaracharya Group of Institutions, Bhilai(C.G.), India and Prof. Samta Gajbhihe, Head, Department of Computer Science and Engineering, Shri Shankracharya Group of Institute, Bhilai (C.G.), India for providing the necessary facilities. One of the author (R.C.) is grateful to Dr. Shamita Chakraborty, Director, Maha Maya College of Technology , Raipur(C.G.), India and Prof.(Mrs.) Kiran Chhabra, Head, Department of Computer Science and Engineering, Maha Maya College of Technology, Raipur(C.G.), India for their help and co-operation.

REFERENCES

1. Jon Y. Hardeberg (2001), "Red Eye Removal using Digital Color Image Processing", Proc. IS&T Image Processing, Image Quality, Image Capture Systems (PICS), pp: 283-287.
2. Matthew Gaubatz and Robert Ulichney (2002), "Automatic red-eye detection and correction", International Conference on Image Processing (ICIP-2002), Vol. 1, pp: 804-807.
3. Jay S. Schildkraut and Robert T. Gray (2002), "A Fully Automatic Redeye Detection and Correction Algorithm", Proceedings of the 2002 International Conference on Image Processing (ICIP 2002), Rochester, New York, USA, 22th-25th Sept, Volume 1. IEEE, pp: 801-803.
4. B. Smolka, K. Czubin, J.Y. Hardeberg, K.N.Plataniotis, M. Szczepanski, K. Wojciechowski (2003), "Towards automatic redeye effect removal", Pattern Recognition Letters, Vol. 24, No.11, 2003, pp. 1767-1785.
5. Sergey Loffe (2003), "RED EYE DETECTION WITH MACHINE LEARNING", Proceeding of International Conference on Image Processing (ICIP 2003), Barcelona, Catalonia, Spain, 14th-18th Sept ,Vol. 2, pp: 871-874.
6. J. Wan and X.P. Ren (2004), "Automatic red-eyes detection based on AAM" In Proceedings of International Conference on Systems, Man and Cybernetics, vol. 7, pp: 6337–6341.

7. Huitao Luo, Jonathan Yen and Dan Tretter, (2004), "An Efficient Automatic Redeye Detection and Correction Algorithm", IEEE International Conference on Pattern Recognition 2004 (ICPR 2004). Proceedings of the 17th International Conference, Vol.-4, 23th-26th Aug, Cambridge, UK, pp: 883-886.
8. Lei Zhang, Yanfeng Sun, Mingjing Li, Hong-Jiang Zhang (2004), "Automatic red eye detection and correction in digital photograph" Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Vol. 2.
9. X.P. Miao and T. Sim (2004) "Automatic red-eye detection and removal" Proceedings of International Conference on Multimedia and Expo, pp: 1195–1198.
10. Francesca Gasparini and Raimondo Schettini (2005), "Automatic red eye removal for smart enhancement of photos of unknown origin", VISUAL '05 Proceeding of the 8th international conference on Visual Information and Information Systems, 2005, pp: 226-233.
11. Robert Ulichney (2005), "PERCEPTUAL-BASED CORRECTION OF PHOTO RED-EYE" Proceding of the IASTED International Conference, pp: 532-537.
12. Yi Wang and Fuhuei Lin (2007), "A novel automatic red eye detection and removal method", Communication and Information Technologies,(ISCIT'07) Sydney, NSW, 17-19 Oct 2007, pp: 759-762.
13. Ilia V. Safonov (2007), "Automatic red eye detection" Proc. of 17th International Conference on Computer Graphics and Vision. GRAPHICON-*2007*, pp. 112-119.
14. Sebastiano Battiato, Mirko Guarnera, Tony Meccio, and Giuseppe Messina (2009), "Red Eye Detection through Bag-of-Keypoints Classification" Image Analysis and Processing ICIAP 2009 15th International Conference, Sept 2009, pp: 528-537.
15. Francesca Gasparini and Raimondo Schettini, (2009), "A Review of Redeye Detection and Removal in Digital Images Through Patents", Recent Patents on Electrical Engineering, Vol. 2 Issue-1 ISSN: 1874-4761.
16. S. Battiato, G. M. Farinella, M. Guarnera, G. Messinaz, D. Ravi (2010), "Boosting Gray Codes for Red Eyes Removal". Proceesing of International Conference on Pattern Recognition (ICPR 2010), 23th-26th Aug, pp: 4214 – 4217.
17. YueLi Cui, ZhiGang Chen, AiHua Chen (2010), "Study of Fully-automatic Red-eye Removal Algorithm", Image and Signal Processing (CISP), 2010 3rd International Congress ,16th-18th Oct, pp: 2673 – 2676.
18. S. Battiato, G. M. Farinella and M. Guarnera, G. Messina, D. Ravi (2010), "Red-eyes removal through cluster based linear discriminant analysis". Proceedings of the IEEE International Conference on Image Processing (ICIP 10), September 2010, pp: 2185–2188.
19. Yanfang Wang and Bo Han (2010), "A Novel Red-eye Removal Approach in DigitalColor Photos" Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing (PIC 2010), Vol.2, 10th-13th Dec, 0, Shanghai, China, pp: 733-737.
20. T Ali, S Khattak and I Kim (2011), "Automatic red-eye effect removal using combined intensity and colour information". Imaging Science Journal, Vol. 59, No. 1, 2011, pp: 8-16.
21. Ilia V. Safonov (2007), "Automatic red eye detection" Proc. of 17th International Conference on Computer Graphics and Vision. GRAPHICON-*2007*, pp: 112-119.
22. Raimondo Schettini, Francesca Gasparini, Fadi Chazli "A modular procedure for automatic redeye correction in digital images" Proceedings of the SPIE, Volume 5293, pp: 139-147.
23. Flavien Volken, Johann Terrier, Patrick Vandewalle "Automatic Red-Eye Removal based on Sclera and Skin Tone Detection" Proc. IS&T Third European Conference on Color in Graphics, Imaging and Vision (CGIV), pp: 359-364.
24. Shanshan Wang and Amr Abdel-Dayem (2012),"Improved Viola-Jones Face Detector" Proceedings of Taibah University International Conference on Computing and Information Technology (ICCIT 2012), pp:123-128.
25. Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja,"Detecting Faces in Images: A Survey" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 24, No. 1, January 2002.
26. Xiaoyi Jiang and Yung-Fu Chen "Facial Image Processing", Applied Pattern Recognition, Springer, 2008, pp: 29-48.
27. D.Sidibe, P.Montesinos, S.Janaqi. Parc Scientifique G.Besse, France "A Simple and Efficient Eye Detection Method in Color Images", "International Conference Image and Vision Computing New Zealand, 2006 .
28. E. Hjelmas and B.K. Low (2001), "Face detection: A survey. Computer Vision and Image Understanding", pp: 237–274.
29. P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss (2002), "The FERET evaluation methodology for face-recognition algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp: 1090–1104.
30. M.-H. Yang, D. Kriegman, and N. Ahuja. (2002), "Detecting faces in images: A survey" IEEE Transactions on Pattern Analysis and Machine Intelligence, pp: 34–58.
31. Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J. Flynn, Horst Bunke, Dmitry B. Goldgof, Kevin Bowyer , David W. Eggert, Andrew Fitzgibbon, and Robert B. Fisher (1996), "An Experimental Comparison of Range Image Segmentation Algorithms", IEEE TRANSACTIONS ON PATTERN AN4LYSIS AND MACHINE INTELLIGENCE, VOL. 18, No. 7, July 1996.
32. Rashmi Chandra, and Rohit Raja, "A Comparative Survey of Automatic Red-Eye Detection and Correction", International Journal on Computer Technology and Electronics Engineering (IJCTEE), ISSN: 2249-6363, Vol. 2, Issue 4, pp: 90-98, August 2012.
33. Volker Blanz, Kristina Scherbaum, Thomas Vetter and Hans-Peter Seidel (2004), "Exchanging Faces in Images",The European Association for Computer Graphics 25th Annual Conference EUROGRAPHICS 2004, Blackwell, ISSN:0167-7055, Vol. 23, pp: 669-676.

AUTHORS PROFILE

Rashmi Chandra, obtained her MCA in 2009 from School of Studies in Computer Science, Pt. Ravi Shankar University, Raipur, India (C.G.). She is pursuing her M.E. in Computer Technology and Applications from Shri Shankaracharya College Of Engineering & Technology (SSCET), Bhilai (C.G.), India. Presently she is working as an Assistant Professor at Maha Maya College Of Technology, Raipur, (C.G.), India.

Mr. Rohit Raja, obtained his M.E. in Computer Technology and Applications in 2008 from Shri Shankaracharya College Of Engineering & Technology (SSCET), Bhilai (C.G.), India. He is pursuing Ph.D from CV Raman University, Bilaspur,(C.G.), India. Presently he is working as a Professsor at Shri Shankaracharya College Of Engineering & Technology (SSCET), Bhilai,(C.G.), India. His area of interest is image processing.