Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (3) 395-398 (2012)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Eye State Recognition based on Machine Learning Techniques

T. ALI, L. HASAN*, H. ZAFAR*, T. NAWAZ**, S. AHMAD**, Y. MUHAMMAD**

Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Nethelands

Received 18th January 2012 and Revised 11th June 2012

Abstract: In this paper we propose a robust approach to determine eye state (open / close) in still images. The method is based on the assumption that an eye patch around the eye center is already found. There are several published works which can accomplish this including our earlier work on automatic eye detection that we will briefly review in this paper. After the eye patch is obtained, Artificial Neural Networks (ANN), Support Vector Machines (SVM) and AdaBoost with different weak learners are combined using a simple majority voting scheme to build a robust classifier for eye state identification. For experiments, a database of 600 images is used which include images with different illumination conditions, taken from different angles and subjects of different ethnicity. Experimental results are shown for each of individual classifier and their combination using majority voting. Self-consistency, Jackknife and independent dataset tests are performed for each individual classifier as well as for classifier obtained using majority voting scheme. When tested by 300 novel eye patches from the database, an accuracy of 100% can be achieved by our proposed majority voting scheme.

Keywords: ANN, SVM, AdaBoost, Eye patch, Eye state

1. INTRODUCTION

Robust eye states recognition has several applications including driver vigilance system, facial expression analysis and human-computer interface systems. Monitoring a driver's visual attention is very important for detecting fatigue, lack of sleep, and drowsiness. By automatically detecting eye state and subsequently drowsiness level of driver, an alarm can be activated to inform driver or other authority which can avoid a large number of road accidents. Eye state is very important in the driving fatigue detection especially when it has been proven that the fatigue has a very high correlation with the Perclos (Dinges and Perclos, 1998). Robust eye detection is crucial step for automatic eye state identification. After eye detection, information about eye state, gaze, eye blinking, and drowsiness can be determined by appending the eye detection algorithm with eye state identification algorithm. There are several works on automatic eye detection which achieve a very high level of detection rate (Ali and Kim, 2009) (Kawaguchi and Rizon, 2003) but the problem of automatic eye state identification is still far from being fully solved. After eye detection, several measures can be used to determine eye state and detect drowsiness. Many efforts have been made to detect drowsiness among drivers (Hamada *et al.*, 2003) (Ji and Yang, 2002). However, there are uncertain variances of illumination, pose, expression and size, so it is necessary to developing a robust eye state recognition method in view of the presence of such obstacles. In this paper, we use a still image to locate eye patch and then determine eye state by using a combination of different classifiers. We mainly focus on the classification stage of the system when an eye patch is already located from a real-time driver image. The basic purpose of this work is to improve the classification accuracy of the existing systems and make it more robust to illumination conditions, subjects' ethnicity, pose and partial closure of eyes etc.

2. MATERIALS AND METHIDS

Outline of the Proposed Method

The algorithm first detects the face and eye patch in an input image using the method described in our earlier work (Ali and Kim, 2009). In the next step, features are extracted using Principal Component Analysis (PCA) from detected eye patch. Five classifiers are then combined using a majority voting scheme to determine eye state. The main contribution of this work is to design a robust classifier for eye state recognition once we have an eye patch detected. **(Fig. 1)** shows the outline of the proposed system.

++Corresponding Author: laiqhasan@nwfpuet.edu.pk +92-3338666264

*Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, Pakistan

**Department of Electrical Engineering, COMSATS Institute of Information Technology, Abbottabad, Pakistan

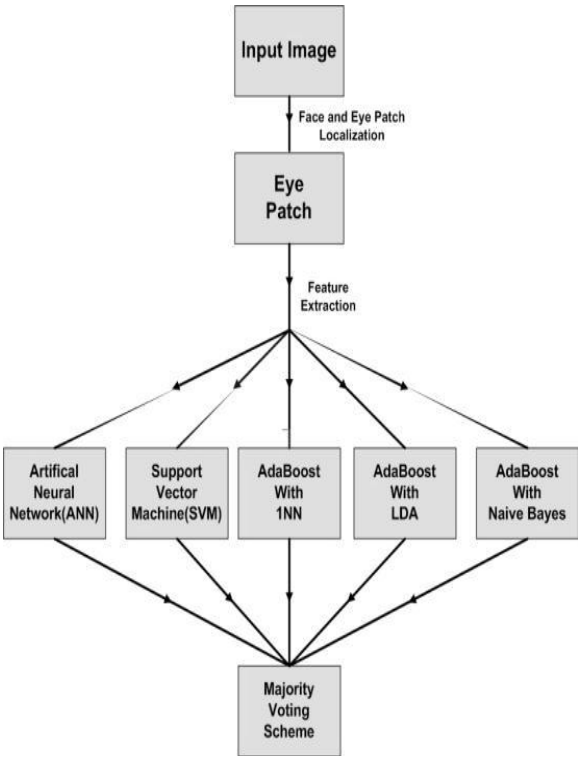


Fig. 1. Block diagram of the proposed system

Face and Eye Patch Localization

As a first step, the face in the input image is detected in order to restrict the background to the face. The face detection is based on popular Viola's method (Viola and Jones, 2001) which obtains a robust face classifier through supervised AdaBoost learning. Given a sample set of training data $\{x_i, y_i\}$, the AdaBoost algorithm selects a set of weak classifiers $\{h_j(x)\}$ from a set of Haar-like rectangle features and combines them into a strong classifier $g(x)$ defined as

$$g(x) = \begin{cases} 1 & \sum_{k=1}^{k\max} \alpha k h_k(x) \geq \theta, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where θ is the threshold that is adjusted to meet the detection rate goal and α is the weighting factor. A detailed description of this face detection process can be found in. (Hamada. *et al.*, 2003)

Several eye candidate points in the face region are detected using the same AdaBoost approach. An eye candidate detector is built by changing the training data and increasing the false-positive rate of the AdaBoost algorithm. A total of 7000 eye samples are used for training with the eye center being the center of the image and resized to 16×8 pixels. Because at this stage, the face region is already detected so the negative samples are taken only from the face images. By setting a low threshold in (1), more false positive are accepted. By properly adjusting the training parameters, the eye candidate points are restricted to a small number (e.g., 15). During the training process, special consideration is given to the fact that the classifier should equally detect open, partially open, and close eyes. The candidate points are then fine tuned and using several measures they are filtered and finally one candidate pair representing eyes is selected among the tuned candidate points. A complete description of this employed eye detection approach can be found in (Ali and Kim, 2009). The only modification in (Ali and Kim, 2009) is introduced at the candidate detection stage such that training data include enough number of close eye samples so that close eye is equally recognized as a candidate at the eye candidate detection stage.

Eye State Identification

i) **Database and Pre-processing:** A database containing a total of 600 eye patches is used for our experiments in classification stage. Some of eye patches are shown in **(Fig. 2)**. 300 patches are used for training while the 300 for testing the classifiers. These patches are taken from (BioID, 2012) and (Georgia Tech Face Database, 2000) and our own database by automatic detection of face and eyes in input image using the procedure discussed in last section. Size of each patch is restricted to 20×20. For feature extraction, Principle Component Analysis (PCA) is applied as a pre-processing step for classification

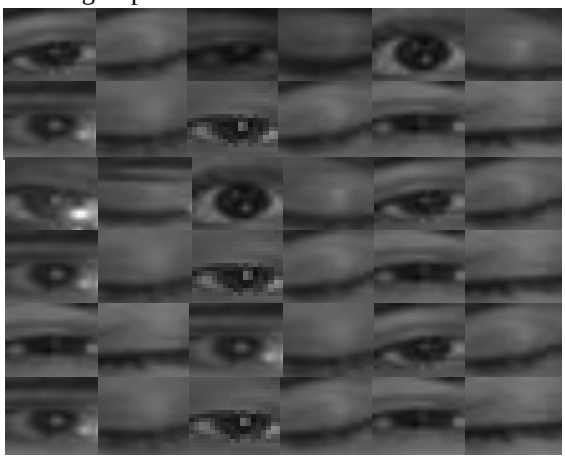


Fig. 2. Eye patches extracted from face images

ii) **ANN:** ANN based machine learning methodologies are widely used in pattern classification (Khan *et al.*, 2008). Their learning process is categorized as supervised and unsupervised learning. In our work, we have used a supervised ANN model in which *m* input training pairs (x_i, y_i) are presented to the ANN network, where $x_i \in R^N$ and $y_i \in Y = \{1, 2\}$. We use a multi-layered feed-forward back propagation based neural network for our two class eye state data. The number of neurons in input layer is equal to the total number of features. Similarly, the number of output neurons is equal to the number of classes. Two

hidden layers are created with the number of sigmoid neurons equal to 14 and 7, respectively followed by an output layer of linear neurons. This configuration is set empirically. Back-propagation learning algorithm is used during training phase. The error *e*, for output neuron *j* is computed as follows:

$$e_j(t) = z_j(t) - y_j(t) \quad (1)$$

where $z_j(t)$ and $y_j(t)$ represents the actual and target output for neuron *j* for iteration *t*. The average squared error of the ANN is then calculated as:

$$\xi_{avg}(t) = \frac{1}{m}\sum_{t=1}^{m}\xi(t) \quad (2)$$

where $\xi(t) = \frac{1}{2}\sum_{j\in P} e_j^{\,2}(t)$ represents instantaneous sum of squared errors and *P* represents the number of neurons in the output layer. The output neurons of the network compete between themselves so that only the winning neuron is fired in response to a given test sample.

iii) **SVM:** For a given training set of instance-label pairs *(xi, yi), i = 1,….m* where $X_i \in R^n$ and $y_i \in \{1,-1\}$, the SVM (Boser *et al.*, 1992) requires the solution of the following optimization problem:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to} \quad y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (3)$$

Here training vectors $\boldsymbol{x}_i$ are mapped into a higher dimensional space by the function ϕ. The SVM then finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term and $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is called the kernel function, for which there exists a space of inner product.

iv) **AdaBoost Algorithm:** AdaBoost algorithm basically works by calling a weak classifier repeatedly in a series of iterations (Freund and Schapire, 1997). For each call, a distribution of weights is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of incorrectly classified samples are increased so that the new classifier focuses more on those samples. The pseudo-code of AdaBoost is as follows:

Pseudo-code of AdaBoost Let's consider a training set *S* of labeled instances: $S = \{(x_i, y_i), i = 1, 2, ..., m\}, y_i \in Y = \{0,1\}$ and given the Base Learner (weak classifier) *B, Q* is the number of iteration, and *m* denotes the number of samples.

Procedure:

Initialize for all *i*: $w_1(i) = 1/m$

For *q* = 1 to *Q*

$$P_q(i) = \frac{w_q(i)}{\sum_i w_q(i)}$$

$$C_q = B(P_q)$$

$$\varepsilon_i = \frac{1}{m}\sum_{x_j \in S' : C_i(x_j) \neq y_j} \text{weight}(x)$$

If $\varepsilon_i > \frac{1}{2}$ then

Q = q-1

Exit

$$\beta_q = \frac{\varepsilon_i}{1-\varepsilon_i}$$

for all $i : w_{q+1}(i) = w_q(i)\beta_q^{\,1-[C_i(x_i)\neq y_i]}$

End of For loop

$$\text{Predicted Class} = \arg\max \sum_{q=1}^{Q}\left(\log\frac{1}{\beta_q}\right)[C_q(x) = y]$$

When the weak classifier B is provided weights, it has two options. Some classifiers such as gradient based algorithms make use of the weights directly. Some classifiers, however, cannot directly use weights such as K-Nearest Neighbors (NN). In the later case, we re-sample the training set and generate a new set of training examples that is distributed according to the weights. In this process, the samples with larger weights have more chances of being selected in the new set, even multiple times, while the samples with fewer weights might get lost. In this work, three types of weak classifiers are used with AdaBoost: NN, Linear Discriminant Analysis (LDA) and Naïve Bayes.

v) Majority voting scheme: A simple majority voting scheme is adopted using the combination of five individual classifiers: ANN, SVM, AdaBoost with LDA as a week learner, AdaBoost with NN as a week learner and AdaBoost with NaiveBayes as a week learner. Each classifier vote for a class and the final selected class is the one for which majority of classifiers vote.

3. RESULTS AND DISCUSSION

(Fig. 3) shows some of the face images with detected eye patches and correct eye state identified by the proposed approach. Input images are taken from our own database as well as from two public databases: (BioID, 2012) and (Georgia Tech Face Database, 2000). This allows for enough variation in terms of illumination conditions, ethnicity, eye size and facial expressions. Subjects wearing glasses and partial closure of eyes are also taken into consideration while developing database for training and testing classifiers.

First the face detection and eye patch extraction algorithm is run on above mentioned three databases. From a complete set of eye patches detected from images in these databases, 600 patches which exhibit greater variation are chosen for training and testing classifiers. 300 eye images are used for testing while 300 are used for training. For each of the classifiers, the prediction quality was examined by three test methods: the self-consistency test, the jackknife test and the independent data set test. In the self-consistency test, the eye state of each sample in the dataset is predicted using the classifier trained on the same dataset.

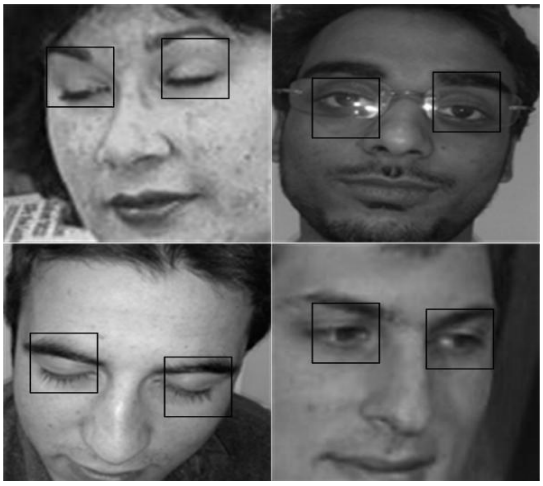


Fig. 3. Some images with detected eye patch and correct eye state identified by the proposed approach

In the jackknife test, each sample in the dataset is singled out in turn as a 'test' sample and the classifier is trained using the remaining $N-1$ samples, which builds up the training dataset. Jackkinfe test is considered one of the most effective and objective methods for cross-validation in statistics. For the demonstration of practical application, the independent data set test is also conducted. In this test, two independent datasets, the training dataset and the testing dataset, are set up. None of the samples contained in one dataset is included in the other. **(Table 1)** shows experimental results obtained using each of the individual classifiers as well as by the proposed majority voting scheme.

Table I. Performance of individual classifiers on using different tests

Classifier/Test type	Self-Consistency	Jackknife	Independent data set
ANN	100 %	92.5 %	93.3 %
SVM	100 %	93.3 %	96 %
AdaBoost (LDA)	100 %	88.7 %	90.7 %
AdaBoost (NN)	100 %	88.5 %	88.7 %
AdaBoost (NaiveBayes)	100 %	87.8 %	88.3 %
Majority Voting	100 %	100 %	100 %

4. CONCLUSION

In this paper we have proposed a robust approach for automatic eye state identification based on different machine learning techniques. A majority voting scheme is proposed based on ANN, SVM and AdaBoost with three different weak classifiers. The classifier is tested on 300 novel images with different illumination conditions, taken from different angles and subjects of different ethnicity. Experimental results are obtained for each individual classifier and their combinations using majority voting. It is noted that an accuracy of 100 % can be achieved by our proposed majority voting scheme on novel images.

REFERENCES:

Ali T. and I. Kim (2009) An Improved Eye Localization Algorithm Using Multi-cue Facial Information. 2nd IEEE

BioID (2012) http://www.bioid.com/ access 2012.

Boser E., I.Guyon and V. Vapnik (1992) A Training Algorithm for Optimal Margin Classifiers. Proc. Fifth Annual Workshop 1998. on Computational Learning Theory, ACM Press: 144-152.

Dinges D. F. and G.R. Perclos (1998) A Valid Psychophysiological Measure of Alertness as Assessed by Psychomotor Vigilance. Tech. Rep. MCRT-98-006, Federal Highway Administration,, Indianapolis.

Freund Y. and R.Schapire (1997) A Decision-theoretic Generalization of On-line Learning and An Application to Boosting. Journal of Computer and System Sciences, 55 **(1):** 119-139.

Georgia Tech Face Database (2000) ftp://ftp.ee.gatech.edu/pub/users/hayes/facedb/

Hamada T., T. Ito K. Adachi T. Nakano and S.Yamamoto (2003) Detecting Method for Drivers' Drowsiness Applicable to Individual Features. Proc. Intelligent Transportation Systems, **(2):** 1405–1410.

Ji Q. and X.Yang (2002) Real-time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. Real-Time Imaging, 8 **(5):** 357–377.

Kawaguchi T. and M. Rizon (2003) Iris Detection Using Intensity and Edge Information. Pattern Recognition, **(36):** 549–562.

Khan A., S.F. Tahir and A. Majid (2008) Machine Learning Based Adaptive Watermark Decoding in View of Anticipated Attack. Pattern Recognition, **(41):** 2594-2610.

Viola P. and M. Jones (2001) Rapid Object Detection Using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition Conference: 511–518. China.