Human Identification using Linear Multiclass SVM and Eye Movement Biometrics

Namrata Srivastava
Department of Information Technology
IIIT Allahabad
Allahabad, India
Email id: namratasri01@gmail.com

Utkarsh Agrawal
Department of Information Technology
IIIT Allahabad
Allahabad, India
Email id: utkarsh3914@gmail.com

Soumava Kumar Roy
Department of Information Technology
IIIT Allahabad
Allahabad, India
Email id: soumava.roy91@gmail.com

U. S. Tiwary
Department of Information Technology
IIIT Allahabad
Allahabad, India
Email id: ustiwary@gmail.com

***Abstract*— The paper presents a system to accurately differentiate between unique individuals by utilizing the various eye-movement biometric features. Eye Movements are highly resistant to forgery as the generation of eye movements occur due to the involvement of complex neurological interactions and extra ocular muscle properties. We have employed Linear Multiclass SVM model to classify the numerous eye movement features. These features were obtained by making a person fixate on a visual stimuli. The testing was performed using this model and a classification accuracy up to 91% to 100% is obtained on the dataset used. The results are a clear indication that eye-based biometric identification has the potential to become a leading behavioral technique in the future. Moreover, its fusion with different biometric processes such as EEG, Face Recognition etc., can also increase its classification accuracy.**

Keywords—Linear Multiclass SVM; Eye Position; Eye Difference; Eye Velocity; Biometric Eye Verification and Identification

I. INTRODUCTION

Biometrics is a branch that deals with the identification and authentication of users on the basis of their physical or/and behavioral traits. With the advancement of technology, we can now measure biometric information of users with great accuracy and quality. Some of the popular biometric techniques include fingerprint verification [1], iris recognition [2], face recognition [3] etc.

The major drawbacks with these traditional biometric techniques are that they only depend on the physical characteristics of the user, hence can be easily generated or forged. Such systems cannot even identify whether the user is dead or alive while performing identification.

Eye movement biometrics is one of the growing fields in the biometric research area. It depends on both physical and behavioral characteristics of the user. The information used for biometric measures is mainly produced by brain, so forging this kind of information is very difficult task. Moreover, covert identification is also possible by this technique.

The common approach to perform identification can be done by using machine learning techniques. In this approach, we are provided with a dataset of eye movement recordings of users involved in the identification process. The recording consists of their eye position values at different time interval along with labels. The label denotes the subject id of the user whose eye movement is taken into consideration. The main focus of the paper is to determine how users may be identified on the basis of their eye movement characteristics. Our task is to predict the labels of unknown users in the testing data-set, by using a proper classification model.

The biometric consists of two phases- verification and identification. Verification is done by using a binary classification model, but in case of identification we require a multi-class classification model. We propose using Multi-Class SVM to train the data, and use this model to predict the unknown labels in the testing set.

Eye movements contain a lot of information about an individual. Our work intends on proposing a technique for the exploitation of these eye movement dynamics in the field of biometrical identification. The features from the eye movements are extracted when the user tries to fixate on a 'jumping point' visual stimuli [4], and use them to train a multi-class classifier.

The structure of the paper proceeds in the following format, in Section II the previous work of researchers are inspected, highlighting the most important aspects of each work. The theory used in the paper is explained in Section III. Next the suggested methodology in detail is presented in Section IV. The results are discussed in Section V. The paper concludes with final remarks in Section VI.

II. LITERATURE SURVEY

Eye movement biometric was introduced by Kasprowsi et al. in [4]. They conducted the Jumping Point Stimulation experiment for recording participant's eye movements. The eye recordings were converted into corresponding features like average velocity direction calculation, distance to stimulation, eye difference, Fourier and Wavelet transform of the eye signal, and then these features were minimized by applying Principal Component Analysis (PCA). For each participant, the test was performed multiple number of times, and data was trained using a binary classifier. The average error rate was 16% using their method.

In [5], Bendarik et al. use a different eye tracker and incorporate new features like pupil diameter during the experiment. The stimulation consists of series of tasks like text reading, following a moving cross, looking at a static image. The eye recordings were divided into static and dynamic features. They have performed Fourier transform and PCA on the data, and used k-nearest neighbor classification model. The accuracy of their method was upto 90% for static features and upto 60% for dynamic features.

Another work, in [6] Holland et al. considers a different technique for user identification. They focused on basic eye movements and their aggregated scanpath characteristics of the user while reading. Fixation and saccades were computed from the eye movements. The method was able to identify subjects with an equal error rate as low as 27%.

In [7], Komogortsec et al. introduced a different approach for user identification through eye movements by using oculomotor plant mathematical model (OPMM). The method achieved an error rate of 38%. In [8], identification was performed using graph-matching techniques, it achieved about 30% equal error rate.

In [9],Rigas et al. compares the distribution of dynamic eye movement features, like velocity and acceleration, and trained it with a simple k-nearest neighbor classifier. Their method achieved an accuracy upto 91%.

III. THEORY

A. Feature Extraction and Combination

We have incorporated three features in our work namely Eye Position, Eye Velocity and Eye Difference. The Eye Movement's Verification and Identification Competition (EMVIC) 2012 dataset [12] has the information about eye movements in both the directions. Four arrays are created namely **Lx, Ly, Rx, Ry** which represents the horizontal direction movement of the left eye, the vertical direction movement of the left eye, the horizontal direction movement of the right eye and the vertical direction movement of the right eye respectively.

Eye Position: Eye Position are the coordinates of eye which are itself provided in the dataset. We can use the values of array **Lx, Ly, Rx** and **Ry** for this information.

$$Lx= [lx_1, lx_2, \dots\dots, lx_n] \tag{1}$$

$$Ly= [ly_1, ly_2, \dots\dots, ly_n] \tag{2}$$

$$Rx= [rx_1, rx_2, \dots\dots, rx_n] \tag{3}$$

$$Ry= [ry_1, ry_2, \dots\dots, ry_n] \tag{4}$$

where n is the number of sample points for which data is recorded.

Eye Difference: Eye difference gives the measure of difference between horizontal direction positions and vertical direction positions of left eye and right eye. Two arrays namely **Xd** and **Yd** are constructed

$$Xd= [lx_1 - rx_1, lx_2 - rx_2, \dots\dots, lx_n - rx_n] \tag{5}$$

$$Yd= [ly_1 - ry_1, ly_2 - ry_2, \dots\dots, ly_n - ry_n] \tag{6}$$

where Xd is the horizontal direction difference between the eyes and Yd is the vertical direction difference between the eyes.

Eye Velocity: Instantaneous velocity in horizontal direction and vertical direction is computed. Four arrays namely **LxV, LyV, RxV** and **RyV** are quantified.

$$LxV= [lx_2 - lx_1, lx_3 - lx_2, \dots\dots, lx_n - lx_{n-1}] \tag{7}$$

$$LyV= [ly_2 - ly_1, ly_3 - ly_2, \dots\dots, ly_n - ly_{n-1}] \tag{8}$$

$$RxV= [rx_2 - rx_1, rx_3 - rx_2, \dots\dots, rx_n - rx_{n-1}] \tag{9}$$

$$RyV= [ry_2 - ry_1, ry_3 - ry_2, \dots\dots, ry_n - ry_{n-1}] \tag{10}$$

where LxV is the horizontal direction velocity of left eye, LyV is the vertical direction velocity of left eye, RxV is the horizontal direction velocity of right eye and RyV is the vertical direction velocity of right eye.

The features are now combined in the following ways:

$$FV1= [Lx, Ly, Rx, Ry] \tag{11}$$

$$FV2= [Xd, Yd] \tag{12}$$

$$FV3= [LxV, LyV, RxV, RyV] \tag{13}$$

$$FV4= [FV1, FV2] \tag{14}$$

$$FV5= [FV2, FV3] \tag{15}$$

$$FV6= [FV1, FV3] \tag{16}$$

$$FV7= [FV1, FV2, FV3] \tag{17}$$

FV1 is the combination of Eye Position features, FV2 is the combination of Eye Difference features, FV3 is the combination of Eye Velocity features, FV4 is the combination of FV1 and FV2, FV5 is the combination of FV2 and FV3, FV6 is the combination of FV1 and FV3, FV7 is the combination of FV1, FV2 and FV3.

All the seven feature vectors are trained and tested using SVM multi-class classifier.

B. Support Vector Machine (SVM)

SVM classification scheme is based on the theory of Structural Risk Minimization (SRM) [11]. This theory states that the test error rate, or structural risk R (α) cannot exceed the training error rate, along with VC confidence term. The VC – confidence term depends on the Vapnik-Chervonenkis (VC)-dimension 'h' of the classifier used. The VC dimension for a given function describes the ability of the classifier to classify the training data points. Higher value of VC-dimension will lead to overfitting and result to poor generalization, whereas on the other hand, a lower value will lead to a higher training error rate.

Support Vector Machines (SVM) is a part of supervised learning models, which analyzes data for pattern recognition and use the pattern for classification and regression. SVM models the data as points in the space such that instances of every disjoint categories have a clear gap (margin) which is made as wide as possible. This is achieved by constructing a set of hyperplane(s) in a high dimensional space such that distance between the hyperplane and the nearest training data of any class is as large as possible. In case of non-linearly separable data, a kernel function *k(x,y)* is used to map the non-linear data to a higher dimensional space, with an assumption that the separation will be easier in the higher dimensional space. We have used Linear SVM in our paper, which is explained below using a two-class classification problem.

In a linear SVM, the linear hyperplane decision boundary is given

$$f(x)= \mathrm{sign}((w*x)+b) \text{ [eq1]} \qquad (18)$$

where ‘x’ is the training data, ‘b’ the bias vector, ‘w’ is the weight vector and the norm of weight vector ‘w’ determines the VC dimension.

Let {(x_1, y1), (x_2, y2)… (x_n,y_n)} denote the ‘n’ training vectors such that $x_i \in R^n$, and $y_i \in$ {-1,1}. Thus for a linearly separable training data, we have

$$y_i\,(x_i*w +b)-1\geq 0 \qquad \forall i = 1..n \qquad (19)$$

The margin between the points belonging to the two different classes is defined by the two hyperplanes given as x*w + b=±1 where the eqn. 19 results to zero. This is presented in the figure below.

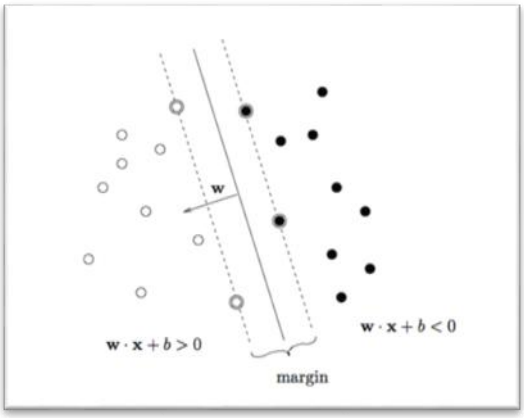

Fig. 1. *A diagrammatic illustration of a linear SVM*

No points can lie within the hyperplanes and width of the margin is give by 2/||w||, where ||x|| defines the norm of the vector x. The algorithms main aim is to maximize the value of margin by minimizing ||w||, such that at an optimal solution, generalization is maximal. As soon as the margin is maximized, the data points, which lie on the separating hyperplanes i.e. eqn. 19 amounts to zeros, are called as support vectors.

In order to simplify the calculation of solution, the problem is converted into Lagrangian framework whose details are given below:

$$L_D = \sum_{i=1}^{n} \alpha i - \frac{1}{2} \sum_{ij} \alpha_i \, \alpha_j \, y_i \, y_j \, x_i . x_j \qquad (20)$$

where

$$w=\sum_{i=1}^{n} \alpha i \; yi \; xi \text{ [eq4]} \qquad (21)$$

$$\sum_{i=1}^{n} \alpha i \; yi =0 \text{ [eq5] and } \alpha_i \geq 0 \; ¥i =1 \dots n \qquad (22)$$

Each α_i (¥i =1 … n) are Lagrangian multipliers that are to be determined. New examples are classified using {eq1}

IV. METHODOLOGY

The proposed method consists of mainly five phases- Data gathering, Feature Extraction, Dividing dataset into train-set and test-set, developing a classification model to classify all unknown test-set samples and Error calculation. These five phases have been pictorially represented in the diagram below

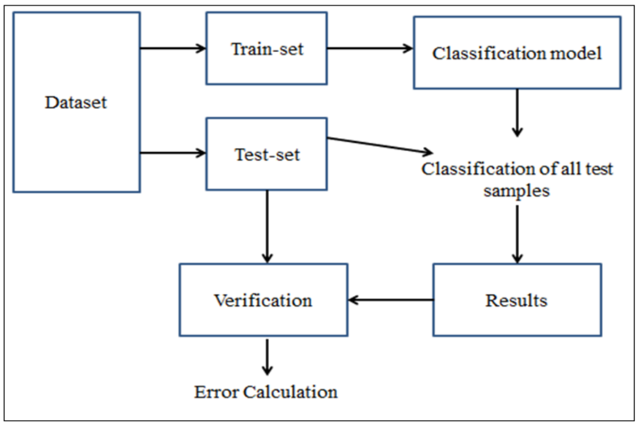

Fig. 2. *A diagramatic representaion of methodology*

A. Data Gathering

The dataset is created by performing the eye movement recording process. This process was first used in [4].The samples were taken with 250Hz frequency using Ober2 eye tracker [10]. A ‘jumping point’ kind of stimulation was done with the same points order for all the participants. In the experiment, the participants are shown a point on the screen, they are required to follow the point movement. The screen is divided into 3x3 matrix, and the point is shown in these nine possible positions. The stimulus point changes its position after a fixed interval of time. The user is attracted towards the flashing point even without its will.

After the recording experiment, each sample consists of three major values- X-Y positions of stimulus point, X-Y positions of the left eye, X-Y position of the right eye at fixed interval of time.

The figure 3 shows possible placement positions for the visual jumping point stimuli.

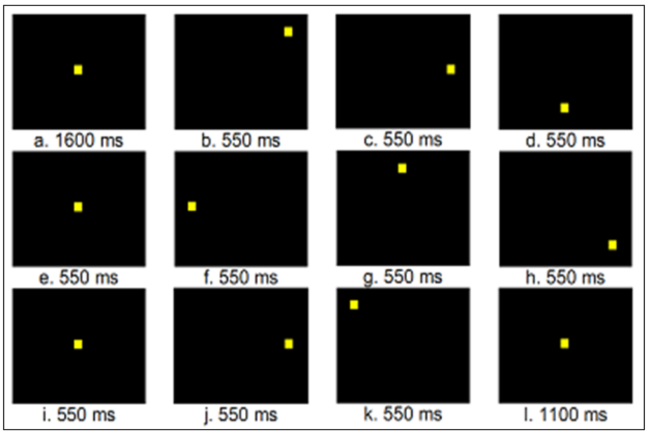

Fig. 3. *Possible placement position for the stimulus during the experiment*

B. Feature Extraction

The dataset consists of only eye movement recordings of all the participants. Three main features calculated from the dataset are Eye position, Eye difference and Eye velocity according to the equations 1-10. These extracted features are combined to form feature vectors in accordance to equations 11-17, which have been used for training the linear SVM.

C. Divide Dataset into Train-set and Test-set

The dataset is divided into train-set and test-set in the ratio of 7:3, i.e. 70% data of the whole dataset is used for training our classifier, rest 30% is used for the testing process.

D. Developing a classification model to classify all unknown test-set samples

The data from the train-set is used to create a classification model with a linear multiclass SVM. The same model is used to predict the unknown class labels of the test-set.

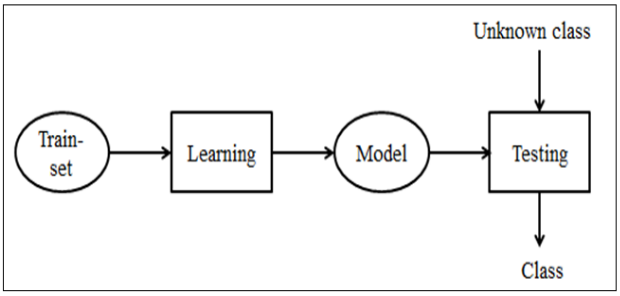

Fig. 4. *A diagrammatic illustration of a Classification Process*

E. Error Calculation

After the training process is completed, the predicted class labels is compared with the actual class labels for the test sets, and the classification accuracy (A) is calculated as follows:

$$A = \frac{\text{(Number of Correctly Classified Data)}}{\text{(Total Number of Test Data)}} \times 100\% \qquad (23)$$

V. Results and Discussion

During the testing process, the feature vectors selected have obtained a classification accuracy of 100% for 6 classes, and achieve a maximum accuracy of 91.67% for 12 classes.

TABLE I. Accuracy Comparision of 3-14 Classes

No. of User	FV1	FV2	FV3	FV4	FV5	FV6	FV7
3	100	100	77.8	100	88.9	88.9	88.9
4	100	100	83.3	100	83.3	83.3	83.3
5	100	100	66.7	100	86.7	86.7	86.7
6	100	100	72.2	100	88.9	88.9	88.9
7	95.2	95.2	57.2	95.2	85.7	81	81
8	91.7	87.5	50	91.7	66.7	79.2	79.2
9	92.6	85.2	44.4	92.6	63	70.4	74.1
10	83.3	80	46.7	83.3	63.33	73.3	76.7
11	81.8	78.8	39.4	84.9	57.6	63.6	70
12	88.9	80.6	44.5	91.7	58.3	66.7	77.8
13	76.9	71.8	48.7	82.1	53.9	64.1	64.1
14	81	69.1	31	83.3	47.6	52.3	59.5

The above table exhibits the accuracies obtained of 14 different classes for all the seven feature vectors taken into account. It is evident from the table that feature vectors FV1, FV2 and FV4 give 100% accuracy when 6 users (classes) are taken into consideration. They also have a superior performance over other feature vectors when the numbers of users are increased beyond six. Thus, it can be inferred that Eye Position and Eye Difference have a more positive influence on the classification accuracy than Eye Velocity. Therefore feature vectors FV3, FV5, FV6 and FV7, which incorporated the Eye Velocity feature in them have a lesser accuracy than FV1, FV2 and FV4 that have not incorporated the same. A graphical representation of above table is given below.

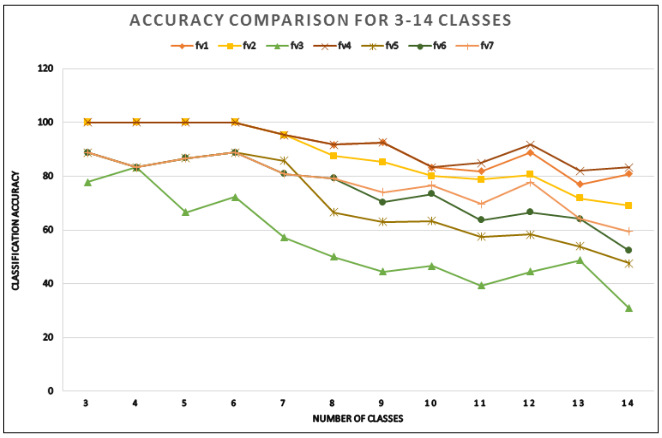

Fig. 5. *Accuracy comparision for 3-14 clasees*

It clearly shows that FV1, FV2, and FV4 have a superior performance over FV3, FV5, FV6 and FV7 across all users.

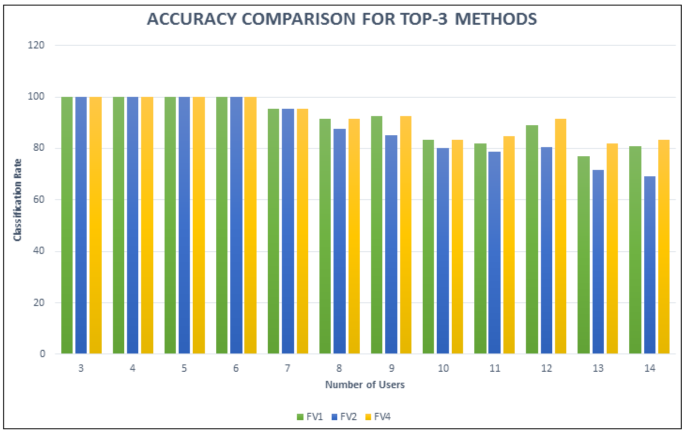

Fig. 6. *Accuracy comparision for top-3 methods*

The above bar chart depicts the classification accuracy across all users for feature Vectors FV1, FV2, and FV4. FV1 and FV4 have almost similar performance over all the users, but FV4 slightly has a better accuracy when numbers of users are increased beyond 6. FV4 has the combination of the two features, while FV1 and FV2 have just a single feature respectively, thus it gives a better accuracy over the two.

VI. Conclusion

In this paper we have embedded the use of Eye Position, Eye Difference and Eye Velocity. We have proved that a combination of these features have produced good results in human identification. We have also incurred that the use of Eye Position and Eye Difference as features have produced better accuracy than Eye Velocity.

Acknowledgment

The authors are grateful to the Indian Institute of Information Technology, Allahabad for providing the facilities and necessary equipment for the successful conduct of the research.

References

[1] Jain, A.; Hong, L.; Bolle, R., "On-line fingerprint verification," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.19, no.4, pp.302,314, Apr 1997

[2] Iris Recognition for Human Identification, Alandkar, Lajari and Gengaje, Sachin, AIP Conference Proceedings, 1324, 229-234 (2010)

[3] Turk, M.A.; Pentland, A.P., "Face recognition using eigenfaces," Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on , vol., no., pp.586,591, 3-6 Jun 1991

[4] P. Kasprowski and J. Ober, “Eye movements in biometrics,” Biometric Authentication, pp. 248–258, 2004..

[5] R. Bednarik, T. Kinnunen, A. Mihaila, and P. Fr¨anti, “Eyemovements as a biometric,” Image Analysis, pp. 16–26, 2005..

[6] C. Holland and O. Komogortsev, “Biometric identification via eye movement scanpaths in reading,” in International Joint Conference on Biometrics (IJCB). IEEE, 2011, pp. 1–8.

[7] O. Komogortsev, S. Jayarathna, C. Aragon, and M. Mahmoud, “Biometric identification via an oculomotor plant mathematical model,” in Proceedings of the 2010 Symposium on Eye- Tracking Research & Applications. ACM, 2010, pp. 57–60

[8] I. Rigas, G. Economou, and S. Fotopoulos, “Biometric identification based on the eye movements and graph matching techniques,” Pattern Recognition Letters, 2012.

[9] Rigas, I.; Economou, G.; Fotopoulos, S., "Human eye movements as a trait for biometrical identification," Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on , vol., no., pp.217,222, 23-27 Sept. 2012

[10] J. Ober and J. Loska, “Function of eye movement measurement system OBER2,” in Proceedings of the Conference on Medical Informatics and Technologies, MIT, 2000.

[11] V. Vapnik. Estimation of Dependences Based on Empirical Data [in Russian]. Nauka, Russia (English Translation: Springer Verlag, New York,1982), 1979.

[12] KASPROWSKI P., OBER J. 2005. Enhancing eye movement based biometric identification method by using voting classifiers. In SPIE Defence & Security Symposium, SPIE Proceedings, Orlando, Florida.