Published version DOI: 10.1109/ICTAI.2012.42

Mel-frequency Cepstral Coefficients for Eye Movement Identification

Nguyen Viet Cuong
Department of Computer Science
National University of Singapore
Email: nvcuong@comp.nus.edu.sg

Vu Dinh
Department of Mathematics
Purdue University, USA
Email: vdinh@math.purdue.edu

Lam Si Tung Ho
Department of Statistics
University of Wisconsin-Madison, USA
Email: lamho@stat.wisc.edu

***Abstract*—Human identification is an important task for various activities in society. In this paper, we consider the problem of human identification using eye movement information. This problem, which is usually called the eye movement identification problem, can be solved by training a multi-class classification model to predict a person's identity from his or her eye movements. In this work, we propose using Mel-frequency cepstral coefficients (MFCCs) to encode various features for the classification model. Our experiments show that using MFCCs to represent useful features such as eye position, eye difference, and eye velocity would result in a much better accuracy than using Fourier transform, cepstrum, or raw representations. We also compare various classification models for the task. From our experiments, linear-kernel SVMs achieve the best accuracy with 93.56% and 91.08% accuracy on the small and large datasets respectively. Besides, we conduct experiments to study how the movements of each eye contribute to the final classification accuracy.**

***Keywords*-Biometric method, eye movement identification, Mel-frequency cepstral coefficients**

I. Introduction

Human identification is an important task for various activities in society. With the development of technology, it is now possible to record a person's biometric information with high quality and use this information for automatic human identification. For instance, some popular biometric methods that have been successfully used for human identification include fingerprint verification [1], iris recognition [2], and hand geometry verification [3].

The main disadvantage of the above traditional biometric methods, as pointed out in [4], is that they are only based on the physical characteristics of the human body. Such biometric methods do not require a person to be conscious during the identification process. Thus, the identification systems can be tricked by using an unconscious person or even a dead body. Moreover, a forger can technically prepare models of a finger, a retina, or a hand of a person and use them to bypass the identification systems.

In this paper, we consider a biometric method called *eye movement identification* (EMI) that is based on both physical and behavioral characteristics of a person and therefore can overcome the above drawbacks of the traditional biometrics. In this method, eye movements are used to identify people. The idea of using eye movements to identify people has been previously studied in [4]–[11]. According to [4], since the eye movement biometric is based on both physical and behavioral characteristics of a person, it requires the person to be conscious during the identification process. Besides, eye movements are also difficult to be forged because they are produced mostly by a person's brain, which is hard to be imitated.

A common approach for EMI is the machine learning approach. In this approach, we are given a training set of eye movement recordings and their labels. The label of one eye movement recording is the person that the eye movement belongs to. Our task is to identify the labels of the eye movement recordings in an unknown testing set. For this problem, a typical method is to train a multi-class classification model from the training set and use this model to predict the labels of the examples in the testing set.

We propose using Mel-frequency cepstral coefficients (MFCCs) [12] to encode various useful features for EMI. The idea of MFCCs is to model various short and overlapping signals obtained from applying a Hamming window function to the eye position, eye difference, or eye velocity signals. The use of MFCCs as features for EMI has many advantages: (1) MFCCs can represent signals in a compact and meaningful way, (2) MFCC features can be easily computed from the data even if the user only has little domain knowledge of EMI, and (3) classifiers with MFCC features can achieve a good accuracy without the need to build complex classification models. Although MFCCs have been successfully used in many applications such as speech recognition [13], music modeling [14], or emotion recognition [15], to the best of our knowledge, they were not previously explored as features for EMI.

In this work, we show how to extract the MFCC features from the eye position, eye difference, or eye velocity signals and use them to train a multi-class classifier for EMI. We conduct an experiment to show that using MFCCs to represent these useful features would result in a much better accuracy than using raw representation or other previously proposed representations such as cepstrum [4] and Fourier transform [6]. To find a suitable model for EMI, we train and compare the accuracy of four different models using the MFCC features: decision tree [16], k-nearest neighbor [17], Bayesian network [18], and support vector machine (SVM)

[19]. The experimental result suggests that the linear-kernel SVMs achieve the best accuracy with 93.56% and 91.08% accuracy on the small and large datasets respectively.

We also study the effects of one-eye identification, in which we use the features from only one eye for EMI. In this study, we conduct an experiment to show how the movements of each eye contribute to the classification accuracy. This result is important because we may not be able to obtain the movements of both eyes in real applications. Thus, if we are to select only one eye for EMI, it is more preferable to choose the eye that provides better classification accuracy.

II. Related works

The first papers using eye movements for human identification are [4]–[6]. In [4], Kasprowski et al. propose the jumping point stimulation experiment for recording the eye movements of the participants. They extract the cepstrum features from the eye movements and train one binary classifier for each participant using the extracted features. Their method can achieve an average error rate as low as 24% on a small dataset with nine participants.

In a following work [6], Kasprowski et al. also employ the same jumping point stimulation method to record eye movements. They convert each eye movement recording into several vectors of attributes such as average velocity, eye difference, discrete Fourier transform of eye positions, etc. and extract the features from these vectors by applying Principle Components Analysis (PCA) [20]. For each participant, they train several binary classifiers and use a voting algorithm to make decisions for this particular person. The average error rate of their method is about 16%.

In both papers [4] and [6], the authors do not attempt to solve the EMI problem fully. Instead, they only try to verify whether an eye movement recording belongs to a given person. Thus, they do not specify how to combine the trained binary classifiers to obtain a full classifier for EMI. Our work, on the other hand, focuses on the EMI problem and builds a single multi-class classifier for all the participants using the MFCCs of eye movement, eye difference, and eye velocity information.

In [5], Bednarik et al. use a different stimulation method for recording eye movements. More specifically, their stimulation consists of several tasks such as text reading, tracking a moving or a static cross, and watching a static image. Since the authors use a different eye tracker for recording, their data also include the pupil diameters of the participants besides the eye movements. They extract features by applying Fourier transform and PCA on the data and train a k-nearest neighbor classifier for EMI. Their method can achieve up to 90% accuracy using static features and up to 60% accuracy using dynamic features.

Another work considers task-independent human authentication using eye movements [7]. In this work, Kinnunen et al. assume little or no prior knowledge about the stimulation task. Their method uses the histograms of eye angles and local velocity as features. Then, they model the eye movements of each participant using Gaussian mixture model and universal background model [21]. Their method can achieve around 30% equal error rate on a dataset of 17 participants.

In [8], Holland et al. propose using eye movement scanpaths in reading for human identification. The authors extract various scanpath features from the fixation and saccades computed from the eye movements. Then, they score the similarity of these scanpath features and use the similarity scores for identification. This method can achieve an equal error rate as low as 27%. Besides scanpaths, oculomotor plant mathematical models computed from eye movements are also used for human identification [9], [10]. These models are reported to achieve as low as 38% error rate. EMI using graph matching techniques is also explored in [11], which is reported to achieve about 30% equal error rate.

Eye movements have also been studied in applications other than human identification. For example, Cowen et al. [22] use eye movements to analyze web-page usability. In this paper, the authors study the relationships between eye movements and users' performance on various online tasks. Eye movements are also used to provide information for the visual mental imagery process [23]. In the paper, Johansson et al. describe the experiments that present evidences about the relationships between eye movements and the mental images formed by the brain during visual mental imagery. Besides, eye movements are also used in other fields such as medicine and technology [24].

III. Materials and Methods

In this section, we first give a short introduction to MFCCs and the eye movement recording process. Then, we describe in detail our approach for EMI.

A. Mel-Frequency Cepstral Coefficients

MFCCs are short-term spectral-based representations of signals that have been successfully used in many applications such as speech recognition [13], music modeling [14], or emotion recognition [15]. The main reason for their success is that they can represent spectrum signals in a compact form and thus can capture the most important features of the signals [14]. In this paper, we show that MFCCs are also useful for modeling eye movements and they can be used to encode features for EMI classifiers.

MFCCs of a signal are computed by a sequence of steps described in figure 1. Given an input discrete signal, we divide the signal into overlapping frames by applying a Hamming window function on the signal at fixed-length overlapping intervals. We compute the MFCCs of the short-term signal in each frame by first computing its discrete Fourier transform and then take the magnitudes (or powers) of the result. Next, we map the magnitudes obtained above

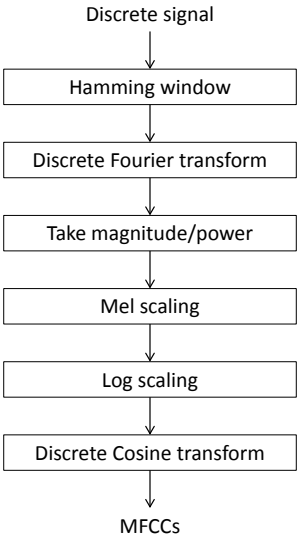

Figure 1. Steps for computing MFCCs of a discrete signal.

onto the Mel scale using a Mel filter bank and take their logarithm at each Mel frequency. Finally, we treat the Mel-log magnitudes as a discrete signal and apply discrete Cosine transform to obtain the MFCCs. The details of the whole process are given in [12].

For each overlapping frame, we can also use the energy and the derivatives of the MFCCs computed by the process above as features [13]. For example, if the above process generates 12 MFCCs for each overlapping frame, we can add the energy term and the first derivatives to obtain a total of 26 MFCCs for each frame. We can also adjust the rate of applying the Hamming window function to control the total number of overlapping frames for the whole signal.

B. The Eye Movement Recording Process

We now briefly describe the eye movement recording process from which the experimental data are obtained. This recording process was first used in [4]. Generally, the eye movement recordings are obtained by using a *jumping point* stimulation experiment. In the experiment, the participants are required to look at a stimulus point on the screen. The stimulus point changes its position at pre-specified time steps during the experiment. There are totally nine possible placements for the stimulus point on the screen, creating a 3×3 matrix. Figure 2 shows these nine possible positions of the stimulus point.

At various time steps during an experiment, the eye positions of the participants are recorded using an OBER2 eye tracker [25]. The timings are set so that the durations between any two consecutive recording points are equal. Thus, we can obtain the discrete eye positions of the participants at equally separated time steps.

After the recording experiments, each example of the final eye movement data contains: (1) the X-Y positions

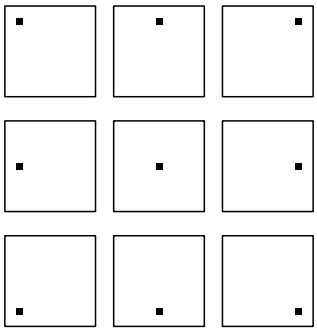

Figure 2. Nine possible placements for the stimulus point during the experiments.

of the stimulus points at equally separated time steps, (2) the corresponding X-Y positions of the left eye at the same time steps, and (3) the corresponding X-Y positions of the right eye at the time steps. In the datasets used in this paper, the whole recording session for one example contains 2,048 equally separated time steps, and we record the X-Y positions of the stimulus point and the eyes at each time step. A sample graph of the X positions of the stimulus point and the left eye in one recording session are shown in figure 3. In this work, we do not use the positions of the stimulus point for EMI since they are almost the same for all examples. So, we will ignore them in the later sections.

C. Eye Movement Identification

Our approach for EMI is to train a classifier on the training data and use this classifier to predict the human identity on the testing data. Unlike previous approaches which train various binary classifiers for verification [4], [6], we model EMI by a single multi-class classifier. This choice of modeling has many advantages: (1) we can easily utilize the known methods for multi-class classification problem and use their implementations without concerning about the details; (2) we do not need to explicitly maintain many binary classifiers and manually combine them; and (3) we also do not need to process the data into many training sets to facilitate the training of the binary classifiers. The idea of using multi-class classifiers for EMI was also employed in [5].

To train a good multi-class classifier, we need to compute the useful information from the eye movement signals described in section III-B and encode them as features for the classifier. Previous works have shown that information such as eye position, eye difference, and eye velocity can be useful for EMI [4]–[6]. Thus, we use these information to compute the features for our classifier. However, our eye velocity is different from the average eye velocity proposed in [6]. Specifically, we compute the instant eye velocity and treat it as a signal rather than compute its average. In our opinion, a signal of instant eye velocity gives more information about the dynamics of the eye movement than

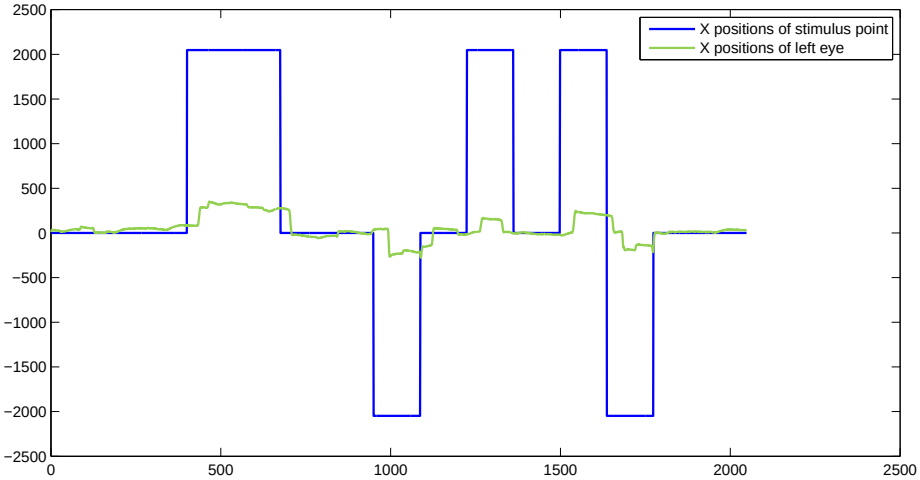

Figure 3. The X positions of the stimulus point and the left eye in one recording session (approximately 8 seconds). The stimulus point changes its positions 9 times during the recording session. After each change, the left eye also adjusts its positions accordingly.

a single eye velocity average. In section III-C1, we describe in detail how to compute the useful information from the eye movement signals.

After obtaining the useful information from the eye movement signals, we need to encode them as features for the classifier. In this work, we propose using MFCCs to represent the information and use them as features. In section III-C2, we describe how to compute the MFCCs features and use them for our classifier.

1) Useful information for eye movement identification: Each eye movement recording described in section III-B contains the eye positions of a person at equally separated time steps during a recording session. More specifically, each eye movement recording contains four vectors $\mathbf{lx}, \mathbf{ly}, \mathbf{rx}, \mathbf{ry}$ which are the X positions of the left eye, the Y positions of the left eye, the X position of the right eye, and the Y position of the right eye respectively. Let

$$\mathbf{lx} = (lx_1, lx_2, \ldots, lx_n)$$
$$\mathbf{ly} = (ly_1, ly_2, \ldots, ly_n)$$
$$\mathbf{rx} = (rx_1, rx_2, \ldots, rx_n)$$
$$\mathbf{ry} = (ry_1, ry_2, \ldots, ry_n)$$

where n is the number of recording points in a recording session. It is important to note that we can treat the above vectors as discrete time-series signals. We now describe how to compute the useful information for EMI from these signals.

- *Eye position (EPos)*: Eye position simply consists of the four vectors $\mathbf{lx}, \mathbf{ly}, \mathbf{rx}, \mathbf{ry}$ themselves. This is the simplest type of information that can be readily obtained from each record without any processing. This type of information was used for EMI in [4].
- *Eye difference (EDiff)*: Eye difference is the difference in X positions and Y positions between the left eye and the right eye. This type of information was used for EMI in [5] and [6]. More specifically, we compute two vectors $\mathbf{XDiff}$ and $\mathbf{YDiff}$. The vector $\mathbf{XDiff}$ is the difference in X positions between the eyes and the vector $\mathbf{YDiff}$ is the difference in Y positions between the eyes. Formally, these two vectors are computed as follows

$$\mathbf{XDiff} = (lx_1 - rx_1, lx_2 - rx_2, \ldots, lx_n - rx_n)$$
$$\mathbf{YDiff} = (ly_1 - ry_1, ly_2 - ry_2, \ldots, ly_n - ry_n)$$

- *Eye velocity (EVel)*: Eye velocity is the instant velocity vectors of the X positions and the Y positions of the eyes. More specifically, we compute four vectors $\mathbf{LXVel}$, $\mathbf{LYVel}$, $\mathbf{RXVel}$, and $\mathbf{RYVel}$, in which $\mathbf{LXVel}$ and $\mathbf{LYVel}$ are the velocity vectors of the left eye in X and Y directions respectively, while $\mathbf{RXVel}$ and $\mathbf{RYVel}$ are the velocity vectors of the right eye in X and Y directions respectively. This type of information is different from the average velocity used in [6]. Formally, we compute these four vectors as follows

$$\mathbf{LXVel} = (lx_2 - lx_1, lx_3 - lx_2, \ldots, lx_n - lx_{n-1})$$
$$\mathbf{LYVel} = (ly_2 - ly_1, ly_3 - ly_2, \ldots, ly_n - ly_{n-1})$$
$$\mathbf{RXVel} = (rx_2 - rx_1, rx_3 - rx_2, \ldots, rx_n - rx_{n-1})$$
$$\mathbf{RYVel} = (ry_2 - ry_1, ry_3 - ry_2, \ldots, ry_n - ry_{n-1})$$

2) Training the classifier with MFCC features: After obtaining the vectors of eye position, eye difference, and eye velocity information as in section III-C1, we compute the MFCC features for the classifier from these vectors. The computation can be done by treating these vectors as discrete time-series signals and applying the process described in section III-A to compute the MFCCs of the signals. For

Table I
THE TRAINING SET SIZE, THE TESTING SET SIZE, AND THE NUMBER OF CLASSES FOR THE TWO DATASETS IN THE EXPERIMENTS.

DATASET	TRAINING	TESTING	CLASSES
A	652	326	48
B	2,778	1,390	79

eye position and eye difference signals, we use the MFCCs and their energy as features. For eye velocity signals, we use the MFCCs, their energy, and their first derivatives as features for our classifier. With these features, we can train a multi-class classifier for EMI. Note that we do not include the MFCCs' derivatives of the eye position and eye difference signals as features because they do not help increase the classification accuracy. There are many multi-class classification models that can be used. In this work, we focus on four well-known models: decision tree [16], k-nearest neighbor [17], Bayesian network [18], and multi-class SVM [19].

IV. EXPERIMENTS

In this section, we describe various experiments to test our method and discuss the results of these experiments. The first experiment is used to test the usefulness of the MFCC features. In the second experiment, we compare different classification models for EMI using the MFCC features. In the last experiment, we compare the performance of the classifiers trained using only the features from left or right eye with the classifier trained using features from both eyes.

For the experiments, we use the first two datasets obtained from the Eye Movement Verification and Identification Competition (EMVIC 2012)[1]. The statistics of the datasets are given in table I. Dataset A contains 652 training examples and 326 testing examples with a total of 48 different classes. Dataset B is a more difficult dataset which contains 2,778 training examples and 1,390 testing examples with 79 classes. The main measurement used in the experiments is the accuracy of the classifiers, which is the percentage of the correct predictions on the testing set.

A. The usefulness of MFCC features

1) Experiment settings: In this experiment, we train a classifier using MFCC features and compare it with three other baselines. Each of the baselines represents a different encoding method for the information obtained in section III-C1. The baselines that we use to compare with our method are:

- **RAW**: This baseline uses all the raw vectors obtained in section III-C1 as features without any processing. This is the simplest baseline for EMI.

[1]http://www.emvic.org/

Table II
NUMBER OF FEATURES IN AN EXAMPLE FOR EACH TYPE OF INFORMATION WITH RESPECT TO RAW, FT, CEPS, AND MFCC ENCODING METHOD.

FEATURES	RAW	FT	CEPS	MFCC
EPos	8,192	8,192	800	1,612
EDiff	4,096	4,096	400	390
EVel	8,188	8,188	800	1,456

- **FT**: In this baseline, we treat the vectors obtained in section III-C1 as discrete time-series signals and compute their Fourier transforms. Then, we use the resulting sequences as features for our training classifier. The idea of using Fourier transform for EMI was used in [5] and [6].
- **CEPS**: For this baseline, we treat the vectors in section III-C1 as discrete time-series signals and compute their cepstrum, which is a sequence of cepstral coefficients. Note that these cepstral coefficients are different from the MFCCs. The cepstral coefficients of a signal are computed by determining the logarithm of the magnitude of the Fourier transform of x, then applying inverse Fourier transform on the resulting sequence. In this baseline, we use the first M coefficients of each signal as features for the classifier. The idea of using the first M cepstral coefficients as features is common in speech processing [12]. This idea was first applied for EMI in [4]. In our experiment, we fix $M = 200$ since this value of M gives the best accuracy for the classifiers.

In table II, we give the number of features for each type of information with respect to the encoding methods. RAW and FT have the same number of features for all the information types, while CEPS has the least number of features. For our MFCC method, we adjust the rate of applying the Hamming window function so that the number of features is moderate, as shown in table II.

In this experiment, we consistently use the multi-class linear-kernel SVMs (with parameter $C = 100$) for all the classifiers. The SVM classifiers are trained by using libSVM library [26]. We can also obtain similar results with other models such as decision tree, k-nearest neighbor, or Bayesian network. However, the accuracy of these models is lower than that of SVMs. We use MATLAB [27] to compute the Fourier transform and the cepstrum of signals. For the MFCC features, we use the VOICEBOX toolbox [28] in our implementation.

2) Results: The results of the experiments on dataset A and B are given in tables III and IV respectively. With only eye position features, our MFCC method has already achieved 91.41% accuracy on dataset A and 85.25% on dataset B. Adding eye difference features improves the

Table III
THE ACCURACY (%) OF THE RAW, FT, CEPS, AND MFCC CLASSIFIERS WITH DIFFERENT SETS OF FEATURES ON DATASET A.

FEATURES	RAW	FT	CEPS	MFCC
EPos	80.67	78.83	65.64	**91.41**
EPos+EDiff	82.52	80.06	65.64	**93.25**
EPos+Evel	81.60	78.53	73.01	**93.56**
EPos+EDiff+EVel	82.52	80.06	71.78	**93.56**

Table IV
THE ACCURACY (%) OF THE RAW, FT, CEPS, AND MFCC CLASSIFIERS WITH DIFFERENT SETS OF FEATURES ON DATASET B.

FEATURES	RAW	FT	CEPS	MFCC
EPos	71.37	63.31	50.65	**85.25**
EPos+EDiff	69.86	62.16	53.24	**87.77**
EPos+Evel	71.30	63.45	62.01	**90.43**
EPos+EDiff+EVel	69.78	62.01	62.01	**91.08**

accuracy of our method by approximately 2% on both datasets, while adding eye velocity features improves the accuracy of our method to 93.56% and 90.43% on datasets A and B respectively. Overall, our MFCC method with all the features can achieve 93.56% accuracy on dataset A and 91.08% accuracy on dataset B.

On dataset A, the RAW and FT baselines achieve the best accuracy with EPos and EDiff features, while the CEPS baseline achieves the best accuracy with EPos and Evel features. For dataset B, the RAW baseline achieves the best accuracy with only EPos features, while the FT and CEPS baselines achieve the best accuracy with EPos and Evel features. Our MFCC method can achieve a good accuracy using only EPos and Evel features. Further adding EDiff features can help to improve the accuracy of our method on dataset B slightly.

Among the three baselines, RAW achieves the best accuracy, while CEPS performs worse than the other two baselines. The accuracy of CEPS using only EPos features is consistent with the results reported in [4]. On both datasets, our method outperforms all the baselines significantly. On dataset A, the MFCC method performs better than the best baseline by more than 10%. While on dataset B, it performs better than the best baseline by more than 13%.

Thus, we can see from the results that MFCC features are much better than Fourier transform, cepstrum, or raw features for EMI. The experiment also shows that Fourier transform and cepstrum features are not good for multi-class eye movement classifiers since they perform worse than even the raw features.

B. Comparisons of different models

In this experiment, we evaluate and compare the accuracy of different multi-class classification models to determine the most suitable model for the EMI problem. In particular, we train and compare four models: decision tree (J48), k-nearest neighbor (kNN), Bayesian network (BayesNet), and Support Vector Machine (SVM). These models have been successfully used in many classification problems. We use all the EPos, EDiff, and EVel features (with MFCC representation) to train the models in the experiment.

We use the implementations of J48, kNN, and BayesNet in Weka 3.6 [29] and the implementation of SVM in libSVM [26]. For J48, we choose the confidence factor $C = 0.25$ and the minimum number of objects in the leaf nodes $M = 2$. For kNN, we choose $k = 3$, following the discussions in [4]. For BayesNet, we use a simple estimator for estimating the conditional probability tables once the structure has been learned [18]. For the search algorithm in BayesNet, we use the K2 hill climbing algorithm [30]. The parameters for SVM classifiers are the same as in section IV-A.

Figure 4 shows the performance of the four models on datasets A and B. On both datasets, SVM classifiers achieve the best accuracy compared to J48, kNN, and BayesNet classifiers. BayesNet performs reasonably (about 85% accuracy) on dataset A but performs poorly (about 60% accuracy) on dataset B. On the other hand, J48 performs poorly on both datasets with only about 55% accuracy on dataset A and 30% accuracy on dataset B. The performance of kNN is moderate with about 80% accuracy on both datasets.

By comparing the accuracy of the classifiers on dataset A and dataset B, we see that the effect of increasing the number of labels is more severe for J48 and BayesNet than for kNN and SVM. For J48 and BayesNet classifiers, their accuracy on dataset B is lower than their accuracy on dataset A by more than 20%. In contrast, for kNN and SVM classifiers, the accuracy on dataset B is only slightly lower than the accuracy on dataset A.

The results from this experiment suggest that support vector machine is more suitable for EMI than decision tree, k-nearest neighbor, or Bayesian network. In practice, we may use cross-validation to select the best model among all the possible models and parameter settings for the task.

C. One-eye classification

In this experiment, we study the effects of one-eye classification. Specifically, we compare the accuracy of the classifiers trained using features from only left or right eye with the classifier trained using features from both eyes. This experiment is useful for us to understand how much the information from each eye contributes to the classification accuracy and which eye is dominant for identification.

The result from this experiment is useful for the limited resource settings where we can only use the movements of one eye for human identification. Such situations may

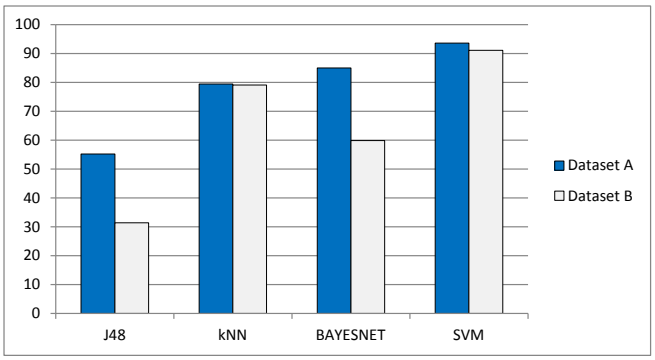

Figure 4. The accuracy of the decision tree (J48), k-nearest neighbor (kNN), Bayesian network (BAYESNET), and support vector machine (SVM) classifiers on datasets A and B.

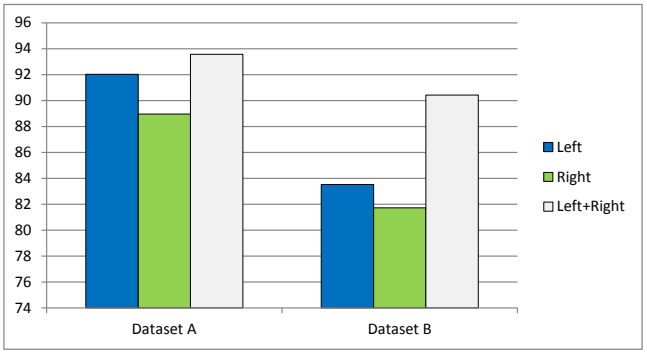

Figure 5. The accuracy of the SVM classifiers on datasets A and B using features from left eye, right eye, and both eyes.

happen if the eye tracker can only record the movements of one eye instead of two eyes as in the experiments in sections IV-A and IV-B. In these cases, we may need to decide which eye to use for classification.

Since we assume that we only know the movements of either left eye or right eye but not both, we can only use the EPos and EVel features in this experiment. We use the MFCC representation of these features and SVM model to train the classifiers. There are a total of 3,068 features from both eyes, in which each eye contributes 1,534 features. The parameters of the SVMs are the same as in sections IV-A and IV-B.

The results for this experiment on dataset A and B are shown in figure 5. From the result, using left eye for EMI is significantly better than using right eye. This result is somewhat consistent with [31], in which the authors use the OBER2 eye tracker calibration to recognize the dominant eye. For both datasets, using features from both eyes for classification can achieve better accuracy than using features from only one eye.

V. Discussions

In our experiments, one reason for MFCC features to perform better than RAW and FT features is that the number of RAW and FT features is much larger than the number of MFCC features. Thus, it is hard to learn a good model with that many features considering the size and the number of classes of the datasets. Another reason why RAW features perform worse than MFCC features is that RAW features cannot filter out the noise in the signals. MFCC features, on the other hand, can represent the signals in a more compact way and therefore can filter out the noise in the signals.

An important advantage of MFCC features compared to the baselines is that the MFCC features can model short-term signals by applying the Hamming window function at overlapping intervals and computing the MFCCs of the short-term signals within the intervals. This is a better way to encode the information in a signal than using the Fourier transform or the cepstrum of the whole long-term signals.

In comparison with previous approaches, our MFCC method can achieve a comparable accuracy with one of the best reported results [5] for EMI. In that model, an accuracy of up to 90% can be obtained if we assume that only eye movements are given in the data (i.e. we exclude the pupil diameter information). However, in their work, the authors use a different stimulation method from ours to record the eye movements.

In terms of model complexity, our model is much simpler than the previous approaches in [4], [6]. For example, in [4], one binary classifier is trained for each person. Thus, the number of models required is equal to the number of people to be identified. In [6], 264 binary classifiers are trained and the best 72 classifiers are combined using a voting algorithm. Our method, on the other hand, only needs to train one single multi-class classifier.

Our method also does not require much domain knowledge about EMI as in [9], [10]. In these works, the models need to compute various oculomotor plant characteristics and use them to build a complex model. This task requires a lot of domain knowledge about EMI. In contrast, out method only needs to compute some simple information such as eye difference or eye velocity and uses their MFCCs as features.

VI. Conclusions

In this paper, we have introduced the use of MFCCs to encode features such as eye position, eye difference, and eye velocity for EMI. We show that using MFCCs to represent these features is significantly better than using the raw representation or other representations such as cepstrum and Fourier transform. We also show that using cepstrum or Fourier transform to model long-term eye movement signals is not useful for EMI. Besides, we give the performance of four different models trained with the MFCC features on two datasets. Among these models, linear-kernel SVMs can achieve a very high accuracy. This result suggests that using linear-kernel SVMs with MFCC features is a good choice for the EMI problem. We also study the effects of using one-eye features for identification. The study shows that using

left-eye features is better than using right-eye features for human identification.

ACKNOWLEDGMENT

We would like to thank Wynne Hsu and Wee Sun Lee for their support, Pawel Kasprowski for letting us use the data, and the reviewers for their constructive suggestions.

REFERENCES

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

[2] R. Wildes, J. Asmuth, G. Green, S. Hsu, R. Kolczynski, J. Matey, and S. McBride, "A machine-vision system for iris recognition," *Machine Vision and Applications*, vol. 9, no. 1, pp. 1–8, 1996.

[3] A. Ross, "A prototype hand geometry-based verification system," *M.S. Project Report, Department of Computer Science & Engineering, Michigan State University*, 1999.

[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. Fränti, "Eye-movements as a biometric," *Image Analysis*, pp. 16–26, 2005.

[6] P. Kasprowski and J. Ober, "Enhancing eye-movement-based biometric identification method by using voting classifiers," in *Proceedings of SPIE*, vol. 5779, 2005, p. 314.

[7] T. Kinnunen, F. Sedlak, and R. Bednarik, "Towards task-independent person authentication using eye movement signals," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 2010, pp. 187–190.

[8] 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.

[9] 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.

[10] O. Komogortsev, T. San Marcos, A. Karpov, L. Price, and C. Aragon, "Biometric Authentication via Oculomotor Plant Characteristics," in *Proceedings of the IARP/IEEE International Conference on Biometrics*, 2012.

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

[12] L. Rabiner and B. Juang, *Fundamentals of speech recognition*. Prentice hall, 1993, vol. 103, no. 58.

[13] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University Engineering Department*, vol. 3, 2002.

[14] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *International Symposium on Music Information Retrieval*, vol. 28, 2000, p. 5.

[15] N. Sato and Y. Obuchi, "Emotion recognition using mel-frequency cepstral coefficients," *Information and Media Technologies*, vol. 2, no. 3, pp. 835–848, 2007.

[16] S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[17] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[18] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2, pp. 131–163, 1997.

[19] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proceedings of the 7th European Symposium on Artificial Neural Networks*, vol. 4, no. 6, 1999, pp. 219–224.

[20] I. Jolliffe and MyiLibrary, *Principal component analysis*. Wiley Online Library, 2002, vol. 2.

[21] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[22] L. Cowen, L. Ball, and J. Delin, "An eye movement analysis of web page usability," *People and Computers*, pp. 317–336, 2002.

[23] R. Johansson, J. Holsanova, and K. Holmqvist, "What do eye movements reveal about mental imagery? Evidence from visual and verbal elicitations," in *Proceedings of the 27th Cognitive Science Conference*, 2005, p. 1054.

[24] J. Ober, J. Hajda, J. Loska, and M. Jamicki, "Application of eye movement measuring system OBER 2 to medicine and technology," in *Proceedings of SPIE*, vol. 3061, 1997, p. 327.

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

[26] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[27] M. Guide, "The mathworks," *Inc., Natick, MA*, vol. 5, 1998.

[28] M. Brookes, "VOICEBOX: Speech processing toolbox for Matlab," *Software, available [Mar. 2011] from www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html*, 1997.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[30] M. Singh and M. Valtorta, "Construction of Bayesian network structures from data: a brief survey and an efficient algorithm," *International Journal of Approximate Reasoning*, vol. 12, no. 2, pp. 111–131, 1995.

[31] Z. Mikrut and P. Augustyniak, "Dominant eye recognition based on calibration of the OBER2 eyetracker," *IFMBE Proceedings*, vol. 3, pp. 394–395, 2002.