Touch Gesture-Based Active User Authentication Using Dictionaries

Heng Zhang, Vishal M. Patel, Mohammed Fathy and Rama Chellappa
Center for Automation Research
University of Maryland, Collage Park 20742
{hzhang98,pvishalm,mefathy,rama}@umiacs.umd.edu

Abstract

Screen touch gesture has been shown to be a promising modality for touch-based active authentication of users of mobile devices. In this paper, we present an approach for active user authentication using screen touch gestures by building linear and kernelized dictionaries based on sparse representations and associated classifiers. Experiments using a new dataset collected by us as well as two other publicly available screen touch datasets show that the dictionary-based classification method compares favorably to those published in the literature. Experiments done using data collected in three different sessions corresponding to different environmental conditions show a drop in performance when the training and test data come from different sessions. This suggests a need for applying domain adaptation methods to further improve the performance of the classifiers.

1. Introduction

In the past few years, we have witnessed an exponential growth in the use of mobile devices such as smartphones and tablets. Mobile devices are becoming increasingly popular due to their flexibility and convenience in managing personal information such as bank accounts, profiles and passwords. With the increasing use of mobile devices comes the issue of security as the loss of a smartphone would compromise the personal information of the user.

Traditional methods for authenticating users on mobile devices are based on passwords or fingerprints. As long as the mobile phone remains active, existing devices do not incorporate any mechanisms for verifying if the user originally authenticated is still the user in control of the mobile device. Thus, unauthorized individuals may improperly obtain access to personal information of the user if a password is compromised or if a user does not exercise adequate vigilance after initial authentication on a device.

To deal with this problem, active authentication systems have been proposed in which users are continuously monitored after the initial access to the mobile device [6]. Screen touch gestures, as a kind of behavioral biometric, are basically the way users swipe their fingers on the screen of their mobile devices. They have been used to continuously authenticate users while users perform basic operations on the phone [4], [14], [8], [3]. In these methods, a behavioral feature vector is extracted from the recorded screen touch data and a discriminative classifier is trained on these extracted features for authentication. These works have demonstrated that touch gestures can be used as a promising biometric for active user authentication of mobile devices in the future.

In recent years, sparse representation and dictionary learning based methods have produced state-of-the-art results in many physiological biometrics recognition problems such as face recognition [16] and iris recognition [11]. These methods make the assumption that given sufficient training samples of certain class, any new test sample that belongs to the same class will lie approximately in the linear or nonlinear span of the training samples from that class. We assume that this assumption is also valid for behavioral biometric, like screen touch gestures.

Kernel sparse coding [5] and kernel dictionary learning [9] have been proposed and applied for image classification and face recognition. In this paper, we study the effectiveness of kernel dictionary learning-based methods in recognizing screen touch gestures for user authentication. Our method builds dictionaries for users, which can be viewed as biometric templates of users and more suitable to be incorporated into a biometric system to authenticate users actively and continuously. Application of kernel dictionary learning for touch gesture recognition and achieving very promising performance are the goals of this work.

The new dataset we have collected consists of not only video facial data but also screen touch data obtained from mobile platforms. Because we believe that in order to actively or continuously authenticate users on mobile devices, face and screen touch gestures are two very important modalities to consider. Due to space limitations, we only focus on the touch data component of the new dataset. Results on face recognition as well as the fusion of both screen touch data and face data will be presented later.

This paper makes the following contributions:

1. We propose kernel sparse representation and kernel dictionary learning-based methods for touch gesture-based active user authentication.
2. We introduce a new multi-modal dataset containing face data and screen touch data simultaneously captured from 50 subjects over 3 sessions. This dataset will be available to researchers to facilitate progress in this field.

1.1. Organization of the Paper

This paper is organized as follows. Section 2 describes the sparse representation and dictionary learning-based methods for screen touch gesture recognition. Details of the new dataset are given in Section 3. Experimental results on this new dataset as well as on two other publicly available screen touch datasets are presented in Section 4. Finally, Section 5 concludes the paper with a brief summary and suggestions for future work.

2. Methods

2.1. Sparse Representation-based Classification (SRC)[16]

Suppose that we are given C distinct classes and a set of N_c training samples per class. Let $\mathbf{Y}_c = [\mathbf{y}_1^c, \ldots, \mathbf{y}_{N_c}^c] \in \mathbb{R}^{d\times N_c}$ be the matrix of training samples from the cth class. Define a new matrix, $\mathbf{Y}$, as the concatenation of training samples from all the classes as

$$\begin{aligned}\mathbf{Y} &= [\mathbf{Y}_1, \ldots, \mathbf{Y}_C] \in \mathbb{R}^{d\times N}\\ &= [\mathbf{y}_1^1, \ldots, \mathbf{y}_{N_1}^1 | \mathbf{y}_1^2, \ldots, \mathbf{y}_{N_2}^2 | ......| \mathbf{y}_1^C, \ldots, \mathbf{y}_{N_C}^C]\\ &\triangleq [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N],\end{aligned}$$

where $N = \sum_c N_c$. We consider an observation vector $\mathbf{y}_t \in \mathbb{R}^d$ of unknown class as a linear combination of the training vectors as

$$\mathbf{y}_t = \sum_{c=1}^{C}\sum_{i=1}^{N_c} x_i^c \mathbf{y}_i^c \tag{1}$$

with coefficients $x_i^c \in \mathbb{R}$. Equation (1) can be more compactly written as

$$\mathbf{y}_t = \mathbf{Y}\mathbf{x}, \tag{2}$$

where

$$\begin{aligned}\mathbf{x} &= [x_1^1, \ldots, x_{N_1}^1 | x_1^2, \ldots, x_{N_2}^2 | \ldots | x_1^C, \ldots, x_{N_C}^C]^T\\ &\triangleq [x_1, x_2, \ldots, x_N]^T\end{aligned} \tag{3}$$

and $(\cdot)^T$ denotes the transposition operation. One can make an assumption that given sufficient training samples of the cth class, $\mathbf{Y}_c$, any new test sample $\mathbf{y}_t \in \mathbb{R}^d$ that belongs to the same class will approximately lie in the linear span of the training samples from the class c. This implies that most of the coefficients not associated with class c in (3) will be close to zero. As a result, assuming that observations are noisy, one can recover this sparse vector by solving the following optimization problem,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y}_t - \mathbf{Y}\mathbf{x}\|_2 \le \epsilon \tag{4}$$

or equivalently the following formulation,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{y}_t - \mathbf{Y}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1, \tag{5}$$

where λ is a parameter and $\|\cdot\|_p$ for $0 < p < \infty$ is the ℓ_p-norm defined as

$$\|\mathbf{x}\|_p = \left(\sum_{j=1}^{d} |x_j|^p\right)^{\frac{1}{p}}.$$

The sparse code $\mathbf{x}_t$ can then be used to determine the class of $\mathbf{y}_t$ by computing the following error for each class,

$$e_c = \|\mathbf{y}_t - \mathbf{Y}_c\mathbf{x}_t^c\|_2, \tag{6}$$

where, $\mathbf{x}_t^c$ is the part of coefficient vector $\mathbf{x}_t$ that corresponds to $\mathbf{Y}_c$. Finally, the class c^* that is assigned to the test sample $\mathbf{y}_t$, can be declared as the one that produces the smallest approximation error [16]

$$c^* = \text{class of } \mathbf{y}_t = \arg\min_c e_c. \tag{7}$$

2.2. Kernel Sparse Representation-based Classification (KSRC)

In kernel SRC, essentially the idea is to map data in the high dimensional feature space and solve (5) using the kernel trick [5] [9]. This allows one to deal with data which are not linearly separable in the original space [13]. Let $\mathbf{\Phi}: \mathbb{R}^d \to G$ be a non-linear mapping from d-dimensional space into a dot product space G. A non-linear SRC can be performed by solving the following optimization problem,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(\mathbf{Y})\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{8}$$

where

$$\mathbf{\Phi}(\mathbf{Y}) \triangleq [\mathbf{\Phi}(\mathbf{y}_1^1), \cdots, \mathbf{\Phi}(\mathbf{y}_{N_1}^1)| \cdots | \mathbf{\Phi}(\mathbf{y}_1^C), \cdots, \mathbf{\Phi}(\mathbf{y}_{N_C}^C)].$$

Denote the first term of (8) by $\mathcal{E}_\kappa$ as follows

$$\begin{aligned}\mathcal{E}_\kappa(\mathbf{x}; \mathbf{Y}, \mathbf{y}_t) &= \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(\mathbf{Y})\mathbf{x}\|_2^2\\ &= \mathbf{\Phi}(\mathbf{y}_t)^T\mathbf{\Phi}(\mathbf{y}_t) + \mathbf{x}^T\mathbf{\Phi}(\mathbf{Y})^T\mathbf{\Phi}(\mathbf{Y})\mathbf{x}\\ &\quad - 2\mathbf{\Phi}(\mathbf{y}_t)^T\mathbf{\Phi}(\mathbf{Y})\mathbf{x}\\ &= \boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{y}_t) + \mathbf{x}^T\boldsymbol{\mathcal{K}}(\mathbf{Y}, \mathbf{Y})\mathbf{x}\\ &\quad - 2\boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{Y})\mathbf{x},\end{aligned} \tag{9}$$

where $\mathcal{K}(\mathbf{Y},\mathbf{Y}) \in \mathbb{R}^{N\times N}$ is a positive semidefinite kernel Gram matrix whose elements are computed as

$$\begin{aligned}[\mathcal{K}(\mathbf{Y},\mathbf{Y})]_{i,j} &= [\langle \mathbf{\Phi}(\mathbf{Y}), \mathbf{\Phi}(\mathbf{Y})\rangle]_{i,j} \\ &= \mathbf{\Phi}(\mathbf{y}_i)^T \mathbf{\Phi}(\mathbf{y}_j) = \kappa(\mathbf{y}_i, \mathbf{y}_j),\end{aligned}$$

$\mathcal{K}(\mathbf{y}_t,\mathbf{y}_t) = \kappa(\mathbf{y}_t,\mathbf{y}_t)$, and

$\mathcal{K}(\mathbf{y}_t,\mathbf{Y}) \triangleq [\kappa(\mathbf{y}_t,\mathbf{y}_1), \kappa(\mathbf{y}_t,\mathbf{y}_2), \cdots, \kappa(\mathbf{y}_t,\mathbf{y}_N)] \in \mathbb{R}^{1\times N}$.

Here, $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the kernel function.

Note that the computation of $\mathcal{K}$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\mathbf{\Phi}$. Some commonly used kernels include polynomial kernels

$$\kappa(\mathbf{x},\mathbf{y}) = \langle(\mathbf{x},\mathbf{y}\rangle + a)^b$$

and Gaussian kernels

$$\kappa(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c}\right),$$

where a, b and c are the parameters. With the above definitions, the kernel version of the SRC optimization problem in (5) can be written as,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \mathcal{E}_\kappa(\mathbf{x}; \mathbf{Y}, \mathbf{y}_t) + \lambda\|\mathbf{x}\|_1. \qquad (10)$$

One can solve the optimization problem (10) by modifying the LARS algorithm [2]. When the ℓ_1-norm is replaced by the ℓ_0-norm in (10), kernel orthogonal matching pursuit algorithm can be used to solve the resulting optimization problem [9].

2.3. Dictionary-based Touch Gesture Recognition (DTGR)

Rather than finding a sparse representation based on training samples as is done in SRC, C touch specific dictionaries can be trained by solving the following optimization problem

$$(\hat{\mathbf{D}}_i, \hat{\mathbf{X}}_i) = \arg\min_{\mathbf{D}_i,\mathbf{X}_i} \|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 \ \text{ s. t. } \|\mathbf{x}_j\|_0 \le T_0 \ \forall j, \qquad (11)$$

for $i = 1, \cdots, C$, where $\|\mathbf{x}\|_0 := \#\{j : x_j \neq 0\}$, which is a count of the number of nonzero elements in $\mathbf{x}$. The optimization problem in (11) can be solved by the KSVD algorithm [1]. Given a test sample $\mathbf{y}_t$, first we compute its sparse codes $\mathbf{x}_i$ with respect to each $\mathbf{D}_i$ using Orthogonal Matching Pursuit(OMP)algorithm which is fast and efficient, and then compute reconstruction error as

$$\mathbf{r}_i(\mathbf{y}_t) = \mathbf{y}_t - \mathbf{D}_i\mathbf{x}_i. \qquad (12)$$

Since the KSVD algorithm finds the dictionary, $\mathbf{D}_i$, that leads to the best representation for each examples in $\mathbf{Y}_i$, one can expect $\|\mathbf{r}_i(\mathbf{y}_t)\|_2$ to be small if $\mathbf{y}_t$ were to belong to the i^{th} class and large for the other classes. Based on this, one can classify $\mathbf{y}_t$ by assigning it to the class, $d \in \{1, \cdots, C\}$, that gives the lowest reconstruction error, $\|\mathbf{r}_i(\mathbf{y}_t)\|_2$

$$\begin{aligned} d &= \text{identity}(\mathbf{y}_t) \\ &= \arg\min_i \|\mathbf{r}_i(\mathbf{y}_t)\|_2. \end{aligned} \qquad (13)$$

Note that similar methods have been used for face biometric in [10].

2.4. Kernel Dictionary-based Touch Gesture Recognition (KDTGR)

Linear dictionary learning model (11) can be made nonlinear so that non-linearity in the data can be handled better [9]. Kernel dictionary learning optimization can be formulated as follows

$$\begin{aligned}(\hat{\mathbf{A}}_i, \hat{\mathbf{X}}_i) &= \arg\min_{\mathbf{D}_i,\mathbf{X}_i} \|\mathbf{\Phi}(\mathbf{Y}_i) - \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{X}_i\|_F^2 \\ &\text{such that } \ \|\mathbf{x}_j\|_0 \le T_0 \ \forall j, \end{aligned} \qquad (14)$$

where we have used the following model for the dictionary in the feature space [9]

$$\mathbf{D} = \mathbf{\Phi}(\mathbf{Y})\mathbf{A}, \qquad (15)$$

where $\mathbf{A}$ is a coefficient matrix. This model provides adaptivity via modification of the matrix $\mathbf{A}$. Through some algebraic manipulations, the cost function in (14) can be written as

$$\begin{aligned}\|\mathbf{\Phi}(\mathbf{Y}_i) - &\mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{X}_i\|_F^2 = \\ &tr((\mathbf{I}-\mathbf{A}_i)^T)\mathcal{K}(\mathbf{Y}_i,\mathbf{Y}_i)(\mathbf{I}-\mathbf{A}_i)). \end{aligned} \qquad (16)$$

This problem can be solved by applying sparse coding and dictionary update steps in the feature space. Details of the optimization algorithm can be found in [9].

Once the dictionaries are learned, we use a sparse subspace approach for classification. In particular, let $\mathbf{D}_i = \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i$ denote the learned kernel dictionary for each class, where $i \in \{1, \cdots, C\}$. Given a query swipe sample $\mathbf{z} \in \mathbb{R}^d$, we first perform kernel sparse coding separately for each $\mathbf{D}_i$ to obtain the sparse code $\mathbf{x}_i$. The reconstruction error can then be computed as

$$\begin{aligned} r_i &= \|\mathbf{\Phi}(\mathbf{z}) - \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i\|_F^2 \\ &= \mathcal{K}(\mathbf{z},\mathbf{z}) - 2\mathcal{K}(\mathbf{z},\mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i + \mathbf{x}_i^T\mathbf{A}_i^T\mathcal{K}(\mathbf{Y}_i,\mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i \end{aligned} \qquad (17)$$

for all $i \in [1, \cdots, C]$. The test sample is simply then classified to the class that gives the smallest reconstruction error.

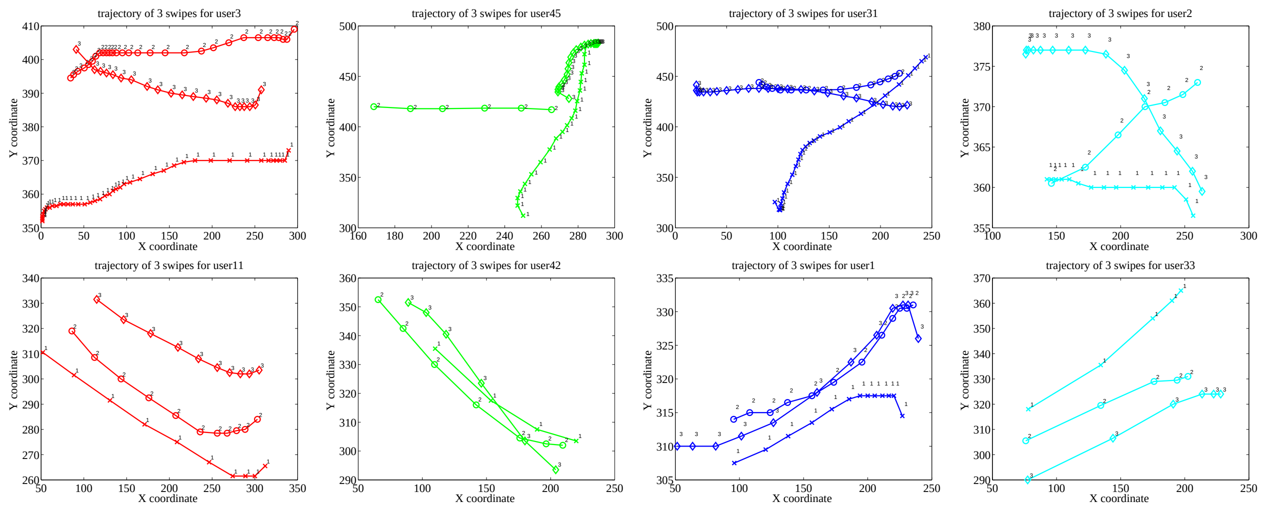


Figure 1. Samples of screen touch data from the new dataset. First and second rows respectively show touch data corresponding to four different individuals performing the same task. The figure is best viewed in color and 200% zoom in. It is interesting to see that even for the same task touch data of different users show significant differences. This information can be used to authenticate the different users.

3. Data Collection

In this section, we describe the details of the new touch dataset we have collected using an iPhone 5s in an application environment. We developed an iPhone application consisting of five different tasks. During each task, the application simultaneously records each users' face video from the front camera on the iPhone and the touch data sensed by the screen.

Our dataset differs from the other touch gesture datasets [14] and [4] in three aspects: a) data collection was done using the iOS platform, b) it is multi-modal dataset consisting of face and touch data, c) data were collected over three different sessions with different ambient conditions. Three settings were as follows: in a well-lit room, in a dim-lit room, and in a room with natural daytime illumination. The goal was to simulate the real-world scenarios to study whether ambient changes can influence the users' screen touch behavior. During data collection, users were free to use the phone in either orientation mode and hold the phone in any position.

3.1. Different Tasks

Enrollment Task——User would enroll his/her face by turning their head to the left, then to the right, then up, and finally down while being recorded by the front-facing camera on the iPhone. Following the enrollment task, the user would perform four tasks with both face and screen touch data being recorded simultaneously. The four tasks are described as follows.

Scrolling Task——User would view a collection of images that are arrayed horizontally and vertically. Each image would take up the whole screen and the user is required to swipe their finger on the screen left and right or up and down in order to navigate through the images.

Popup Task——15 images are positioned off screen in such a way that only a little bit of the image was shown. The user would be then required to drag the image and position it in the center of the iPhone to the best of their ability.

Picture Task——A large poster-like image displays 72 cars with different colors in a 12 by 6 table. Only a few cars could be seen at any given time on the screen. The user was then asked to count the number of cars that were of the color selected by the test proctor. The user was then required to scroll through the entire image in order to provide the correct number.

Documents Task——This task contains a PDF of a research paper which is 12 pages long. The user was asked to count the number of items indicated by the test proctor such as figure, tables etc.

On average it took about 0.5 to 2 minutes to collect facial and touch data per task per session. The new dataset consists of 50 users with 43 male users and 7 female users. All 50 users used the phone in the portrait mode and only one user also used the phone in the landscape mode. In total there are 750 videos recording facial data and 600 txt files recording screen touch data. Figure 1 shows samples of touch data in this dataset.

4. Experimental Results

In this section, we present several experimental results demonstrating the effectiveness of the proposed kernel dictionary-based methods for screen touch gesture recognition. In particular, we present experiments results on the dataset described in Section 3, the Touchalytics dataset [4] and the BTAS 2013 dataset [14].

4.1. Evaluation Metrics

Average Equal Error Rate (EER) and average F1 score are used to evaluate the performance of different methods. The EER is the error rate at which the probability of false acceptance rate is equal to the probability of false rejection rate. The lower the EER value, the higher the accuracy of the biometric system. The F1 score is defined as a harmonic mean of precision P and recall R

$$\text{F1 score} = \frac{2PR}{P+R},$$

where the precision P is the number of correct results divided by the number of all returned results and the recall R is the number of correct results divided by the number of results that should have been returned. The F1 score is always between 0 and 1. The higher the F1 score, better is the accuracy of the biometric system.

4.2. Feature Extraction

Every swipe on the screen is a sequence of touch data when the finger is in touch with the screen of the mobile phone. Every swipe $\mathbf{s}$ is encoded as a sequence of vectors

$$\mathbf{s}_i = (x_i, y_i, t_i, A_i, o_i^{ph}),$$

$i \in \{1, \cdots, N_c\}$ where x_i, y_i are the location points, t_i is the time stamp, A_i is the area occluded by the finger and o_i^{ph} is the orientation of the phone (e.g. landscape or portrait). Given these touch data, we extracted a 27 dimensional feature vector for every single stroke in our dataset using the method described in [4]. These features are summarized in Table 1.

Note that for the Touchalytics and the BTAS 2013 dataset, we extracted 28 dimensional features from each swipe. The additional feature for these datasets corresponds to the mid-stroke pressure. The new dataset described in Section 3 was collected using an iPhone 5s and it does not allow one to capture the pressure information. Whereas, the Touchalytics dataset and the BTAS 2013 dataset , were collected using Android phones which allow them to collect the pressure information. Also the 28 dimensional features extracted are different from what were extracted in [14], even though the dimension is 28 as well.

4.3. Implementation Details

The state of the art performance of touch gesture recognition achieved by kernel SVM with optimized parameter are shown in [4]and [8]. In [14], logistic regression is shown to perform better than kernel SVM, but their experiment setup is different. To be specific, authors of [14]trained classifiers on vertical swipes and horizontal swipes separately under each phone orientation mode and also during testing, for each user, they used 10 samples from each of the remaining users to perform imposter test.

FeatureID	Description
feature1	inter-stroke time
feature2	stroke duration
feature3	start x
feature4	start y
feature5	stop x
feature6	stop y
feature7	direct end-to-end distance
feature8	mean resultant length
feature9	up/down/left/right flag
feature10	direction of end-to-end line
feature11	20%-perc. pairwise velocity
feature12	50%-perc. pairwise velocity
feature13	80%-perc. pairwise velocity
feature14	20%-perc. pairwise acceleration
feature15	50%-perc. pairwise acceleration
feature16	80%-perc. pairwise acceleration
feature17	median velocity at last 3 points
feature18	largest deviation from end-to-end line
feature19	20%-perc. dev. from end-to-end line
feature20	50%-perc. dev. from end-to-end line
feature21	80%-perc. dev. from end-to-end line
feature22	average direction
feature23	length of trajectory
feature24	ratio end-to-end dist and length of trajectory
feature25	average velocity
feature26	median acceleration at first 5 points
feature27	mid-stroke area covered

Table 1. Description of the 27 dimensional feature vector.

For a fair comparison, with all the datasets available, we extracted the same features on all the datasets, fixed the experimental setups, optimized the parameters of every algorithm compared using cross validation, repeated the experiment multiple times by randomly splitting the data into training data and testing data and reported mean and standard deviation of the evaluation metrics above.

We performed two types of experimental setups. For the first type of experiments on the datasets, we combine data from different tasks and sessions. and then we randomly split data for training and testing. As we are also interested in investigating how the environmental changes can affect the users' screen touch behavior, thus for the second type of experiments on the datasets, we performed cross session recognition meaning that training and testing samples are from different sessions. During testing, each user has its own test samples for genuine test and all the other users' test samples for imposter test which means a much larger number of samples were used for imposter test. In all the experiments, we used the histogram intersection kernel for

KSRC and KDTGR methods. The Gaussian kernel with the optimized parameter was used for the kernel SVM. For all the experiments we performed, we repeated 11 times.

4.3.1 Single-swipe vs. Multiple-swipe Classification

The performances of recognition algorithms is influenced by the number of swipes combined to predict a class label. For K-swipe classification, we first perform a single-swipe classification for all the K swipes and obtain the corresponding K predicted labels. Then by voting, we choose the one that appears the most frequently as the final predicted class label. Here, we let K be an odd integer. As K becomes larger, all the algorithms achieve better performance than the methods based on single-swipe classification. However, large K implies longer time to collect swipes and predict the current class label. This is a tradeoff that one has to consider when designing an authentication system based on screen touch gestures.

4.4. Results on the new Dataset

For the first set of experiments using the new dataset, we randomly selected 80 swipes from each user to form the training matrix and used the remaining data for testing. Table 2 summarizes the results obtained by different methods on this experiment. For the single-swipe classification (row one in Table 2), rbfSVM performs the best. However, the average EER is very high for all the methods. This implies that authentication based simply on one swipe is very unreliable. As the number of swipes increase, KDTGR performs the best. This makes sense because by mapping the data onto a high-dimensional feature space and finding a compact representation by learning a dictionary in the feature space, one is able to find the common internal structure of the screen touch data. Classification based on the reconstruction error in the feature space is essentially improving the overall classification accuracy. Kernel SRC, does not use dictionary learning step. As a result, it does not provide the best results on this dataset.

Swipes	KSRC	KDTGR	rbfSVM
1	29.86 ± 0.37	28.03 ± 0.22	17.41 ± 0.13
3	15.82 ± 0.30	12.92 ± 0.34	14.00 ± 0.27
5	9.71 ± 0.33	7.53 ± 0.31	8.56 ± 0.25
7	7.50 ± 0.32	5.59 ± 0.20	6.15 ± 0.27
9	5.85 ± 0.41	4.12 ± 0.22	4.75 ± 0.29
11	4.55 ± 0.32	2.91 ± 0.21	3.58 ± 0.26
13	3.40 ± 0.32	2.16 ± 0.14	2.66 ± 0.28
15	2.55 ± 0.40	1.43 ± 0.23	2.11 ± 0.27
17	1.98 ± 0.20	1.05 ± 0.20	1.43 ± 0.24
19	1.54 ± 0.25	0.77 ± 0.21	1.13 ± 0.22

Table 2. Average EER values (in %) for different classification methods on the new dataset.

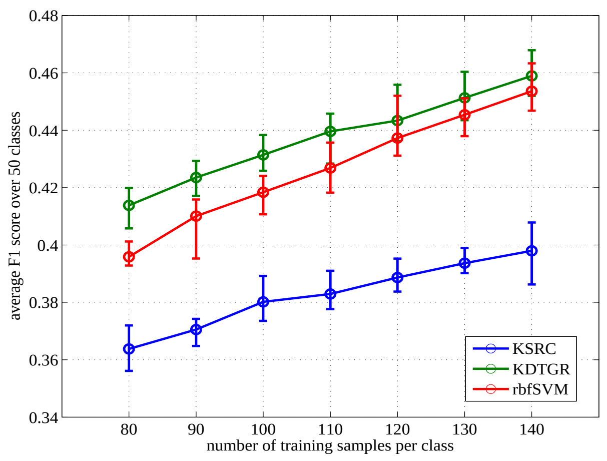
In the second set of experiments with the new dataset, we study the performance of different classification methods as we increase the number of training samples. The average F1 score values for different number of training samples corresponding to single-swipe and eleven-swipe classification are shown in Figure 2(a) and (b), respectively. As can be seen from these figures, KDTGR performs the best for both single-swipe and eleven-swipe classification. Furthermore, the average F1 score values increase as we increase the number of training samples for all the three classification methods. In particular, the average F1 score approaches 0.924, 0.913, 0.885 for the KDTGR method, rbfSVM and the KSRC method, respectively when 140 samples are used for training for eleven-swipe classification.

Finally, in the last set of experiment with the new dataset, we perform cross-session experiments. In particular, since the new dataset contains data from three different sessions with different environmental conditions, we train classifiers using the data from one session and test it on the data from the other session. We repeat this procedure for all the six different combinations of three sessions. For these experiments, we omitted 8 users who have less than 70 swipes in any one of the three sessions. Then, we randomly selected 70 swipes for each user in one session to form the training data and randomly selected 70 swipes for each user in another session to form the test data. The average EER values for different cases for eleven-swipe cross-session classification experiments are summarized in Table 3. The last row of the Table shows the result which were obtained by combining data samples from different sessions together and then splitting them into training and testing data.

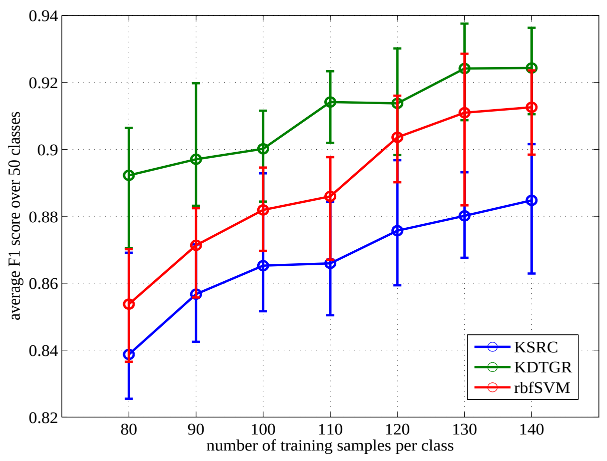
Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	12.05 ±1.21	9.90 ±0.61	11.04 ±1.13
$1 \rightarrow 3$	14.21 ±1.10	11.72 ±0.64	13.08 ±1.12
$2 \rightarrow 1$	14.42 ±0.79	11.69 ±1.12	11.65 ±1.07
$2 \rightarrow 3$	7.23 ±0.53	5.64 ±0.63	5.85 ±0.66
$3 \rightarrow 1$	13.94 ±1.60	11.60 ±0.91	11.75 ±0.97
$3 \rightarrow 2$	7.43 ±0.87	4.88 ±0.74	5.29 ±0.75
1 2 3→1 2 3	4.21 ±0.67	2.62 ±0.65	3.10 ±0.30

Table 3. Average EER values (in %) for the cross-session experiments with the new dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

As can be seen from Table 3, on average the KDTGR method performs the best. When samples from all three sessions are used for training, all three classification methods perform well. This can be seen from the last row of Table 3. However, when classifiers are trained on data from one session and tested on the data from another session, the performance of all the three methods degrade.



(a)



(b)

Figure 2. Average F1 score values (in %) for different classification methods on the new dataset as the number of training samples are increased. (a) Single-swipe classification. (b) Eleven-swipe classification.The figure is best viewed in color.

4.5. Results on the Touchalytics Dataset

Touchalytics dataset [4] consists of 41 users' touch data collected in two sessions separated by one week as described in the original paper. For each user, we randomly selected 80 swipes as training data and the remaining swipes as test data. Results are summarized in Table 4. As can be seen from this Table, on average, the KDTGR method performs the best. For a single-swipe classification, rbfSVM performs better than KSRC and KDTGR. As number of swipes are increased to make the final decision, KDTGR outperforms the other methods.

In the Touchalytics dataset, Session 2 contains touch data from only 14 users. As a result, we did not perform the cross-session experiments on this dataset.

Swipes	KSRC	KDTGR	rbfSVM
1	17.62 ± 0.45	17.69 ± 0.26	8.51 ± 0.13
3	6.13 ± 0.19	4.05 ± 0.097	4.16 ± 0.12
5	3.42 ± 0.13	2.29 ± 0.074	2.33 ± 0.11
7	2.19 ± 0.14	1.14 ± 0.13	1.25 ± 0.11
9	1.31 ± 0.099	0.60 ± 0.079	0.67 ± 0.088
11	0.85 ± 0.11	0.34 ± 0.084	0.36 ± 0.090
13	0.50 ± 0.082	0.16 ± 0.079	0.21 ± 0.074
15	0.35 ± 0.10	0.10 ± 0.062	0.16 ± 0.063
17	0.28 ± 0.082	0.051 ± 0.035	0.086 ± 0.043
19	0.18 ± 0.054	0.026 ± 0.025	0.060 ± 0.036

Table 4. Average EER values (in %) for different classification methods on the Touchalytics dataset.

4.6. Results on the BTAS 2013 dataset

The BTAS 2013 dataset [14] is a large dataset which consists of data in two parts: 138 users' mobile touch data in portrait mode over 2 sessions and 59 users' mobile touch data in landscape mode over 2 different sessions.

4.6.1 Portrait Mode Cross-Session Experiment

Only one user had data with less than 80 swipes in any one of the 2 sessions. We omitted this user for the cross-session experiments. We randomly selected 80 swipes for each user in one session to form the training data and randomly selected 80 swipes for each user in the other session to form the test data. For comparison, we have also added the case where 80 training data and 80 testing data for each user are selected from both sessions. Table 5 shows the average EER values for different cases when 11 swipes are combined to make the final decision (e.g. eleven-swipe classification).

It is interesting to see when data from both sessions are used, the EER values are the lowest. Similar to the observation we made in the experiments with the new dataset, as we train on the data from one session and test on the data from the other session, the performance of all the three classification methods degrade significantly.

Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	23.51 ± 0.70	19.78 ± 0.65	20.67 ± 0.53
$2 \rightarrow 1$	23.83 ± 0.49	19.20 ± 0.72	20.06 ± 0.62
$1\ 2 \rightarrow 1\ 2$	8.94 ± 0.62	5.00 ± 0.46	5.86 ± 0.45

Table 5. Average EER values (in %) for the portrait mode cross-session experiments with the BTAS 2013 dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

4.6.2 Landscape Mode Cross-Session Experiment

Like before, we omitted 6 users who have less than 80 swipes in any one of the two sessions. We applied the same experiment setup as we did for the touch data in the portrait mode. Table 6 shows the average EER values for different cases when 11 swipes are combined to make the final decision. Again, the KDTGR method outperforms the other methods on this dataset.

Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	14.25 ± 0.70	11.09 ± 0.98	13.19 ± 0.81
$2 \rightarrow 1$	13.70 ± 0.49	11.29 ± 0.54	12.04 ± 0.83
$1\ 2 \rightarrow 1\ 2$	4.06 ± 0.68	1.73 ± 0.44	2.18 ± 0.35

Table 6. Average EER values (in %) for the landscape mode cross-session experiments with the BTAS 2013 dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

5. Conclusion and Future Work

In this paper, we introduced a new multi-modal dataset for active user authentication containing face and touch data. Furthermore, we proposed sparse representation and dictionary learning-based classification methods for analyzing screen touch data only. Various experiments on screen touch data in the new dataset as well as two publicly available screen touch datasets showed that the proposed kernel dictionary-based method performed favorably over other compared methods. When considering incorporating the proposed method in mobile devices, training dictionaries can be done offline. For testing, the time complexity of our proposed method is $O(N)$, where N is the number testing data samples, and for each sample we perform OMP algorithm which reqires about $2d^{2.5}$ operations [12], where d is the dimension of the feature vector.

For the new dataset, cross-session experiments showed that there is a significant drop in the performance of all the methods. Similar result is observed in other dataset. This problem can be viewed as *domain adaptation* [7] or *dataset bias* problem [15] which have been studied in machine learning, natural language processing and computer vision. Our future work will address the domain adaptation problem for touch gestured-based active user authentication. Also, we are interested in modeling and recognizing multi-finger touch gestures in the future.

6. Acknowledgments

This work was supported by cooperative agreement FA8750-13-2-0279 from DARPA. The authors would like to thank Jacob Chen for his implementation of the data collection application.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transaction on Signal Process.*, 54(11):4311–4322, 2006. 3

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004. 3

[3] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *IEEE Conference on Technologies for Homeland Security*, pages 451–456, Nov 2012. 1

[4] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, Jan 2013. 1, 4, 5, 7

[5] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *ECCV (4)'10*, pages 1–14, 2010. 1, 2

[6] R. P. Guidorizzi. Security: Active authentication. *IEEE IT Professional Magazine*, 15(4):4–7, 2013. 1

[7] J. Jiang. *Domain adaptation in natural language processing*. PhD thesis, University of Illinois at Urbana-Champaign, 2008. 8

[8] L. Li, X. Zhao, and G. Xue. Unobservable re-authentication for smartphones. In *Network & Distributed System Security Symposium*, Feb 2013. 1, 5

[9] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, 2013. 1, 2, 3

[10] V. M. Patel, W. Tao, S. Biswas, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–965, 2012. 3

[11] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1877–1893, Sept 2011. 1

[12] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. 8

[13] B. Scholkopf and A. J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. 2

[14] A. Serwadda, V. Phoha, and Z. Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–8, Sept 2013. 1, 4, 5, 7

[15] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 8

[16] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009. 1, 2