Facial Attributes for Active Authentication on Mobile Devices

Pouya Samangouei[a], Vishal M. Patel[b], Rama Chellappa[a]

[a] *Electrical and Computer Engineering Department, University of Maryland, College Park, MD, USA*
[b] *Electrical and Computer Engineering Department, Rutgers University, 94 Brett Road, Piscataway, NJ, USA*

Abstract

We present a method using facial attributes for continuous authentication of smartphone users. We train a bunch of binary attribute classifiers which provide compact visual descriptions of faces. The learned classifiers are applied to the image of the current user of a mobile device to extract the attributes and then authentication is done by simply comparing the calculated attributes with the enrolled attributes of the original user. Extensive experiments on two publicly available unconstrained mobile face video datasets show that our method is able to capture meaningful attributes of faces and performs better than the previously proposed LBP-based authentication method. We also provide a practical variant of our method for efficient continuous authentication on an actual mobile device by doing extensive platform evaluations of memory usage, power consumption, and authentication speed.

Keywords: attributes, security, active authentication, face, mobile

1. Introduction

Most probably the first time that a password was used for authenticating users of a computer was in 1961 on the famous Compatible Time-Sharing System (CTSS) which was developed at MIT's computing center [1]. However, passwords turned out to be hard to remember and maintain, and they need to be kept secure. Soon researchers began investigating less vulnerable and easier to maintain forms of authentication. One of the most studied type of methods is biometric identification [2], such as fingerprints, retinal scans or facial image matching. These methods are appealing because while they are unique for each person and hence more secure, they are more intuitive, hence requiring less user effort either for remembering or entering them.

URL: pouya@umiacs.umd.edu (Pouya Samangouei), vishal.m.patel@rutgers.edu (Vishal M. Patel), rama@umiacs.umd.edu (Rama Chellappa)

Mobile devices are one of many categories of devices in which the password-based approaches are widely used as the sole authentication method. Smartphones, tablets, and wearable devices fall into this category. These devices have become an inseparable part of people's lives. They contain a lot of valuable information, from bank account details to emails and other private content. Therefore, these devices are being increasingly targeted by different kinds of attacks [3]. Typical devices incorporate no mechanisms to verify that 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. Biometrics-based algorithms have emerged as a solution for continuous authentication on these devices [4, 5, 6]. See [7] for a comprehensive survey of recent mobile continuous authentication systems.

Modern mobile devices come with a variety of built-in sensors and accessories such as cameras, microphone, gyroscope, accelerometer and pressure sensor. These sensors can be used to extract the biometric data for the user [8], [9], [10], [6]. For instance, faces can be captured using the front-facing camera of a mobile device and can be used to continuously authenticate a mobile device user [6]. Also, the gyroscope touchscreen and accelerometer can be used to measure biometrics such as gait, touch gestures and hand movement.

One of the most popular active authentication methods is based on the face biometric. Authentication methods such as the ones reviewed in [9], [10], [6] use camera sensor images to detect the face of the user, extract low-level features and apply patter recognition algorithms on these features to authenticate the user. The common drawbacks of these approaches are that the low-level features can vary significantly in different environmental conditions and head pose changes. Also as emphasized in [10], the tradeoff between verification accuracy and mobile performance is an important challenge of active authentication. Many of the methods with good accuracies explored in [9] and [10] have either costly enrollment phase or test phase, in terms of computation or memory.

We propose to use a large number of facial attributes, like gender, race, ethnicity, etc. as intermediate representations to authenticate the user of the mobile device. The overview of our method is shown in Figure 1. These attributes give a compact and discriminative representation for the task of continuous authentication.

There are several benefits for using facial attributes as features. First, the training of attribute classifiers can be done offline on large datasets of images which can embody various conditions. As a result, we achieve robustness to changes that can make low-level features ineffective for authentication. Secondly, as we show later in Section 5, the attribute models can be run efficiently, authenticating more than four frames per second on an average mobile device with low power and memory consumption. This is very important since the algorithm must seamlessly run continuously alongside other applications. Furthermore, the attributes are compact representations. Suppose we have n binary attributes, the probability that two people having the same attribute is bounded

above by $1/2^n$. This probability can be very low if attribute scores are continuous like age or skin tone. Therefore, just by comparing the derived attributes with the enrolled ones, one can detect with a high probability whether the current user of the phone is the registered user or not. Furthermore, enrollment of attributes can be done in different ways, they either can be asked directly from the user or they can be captured on one device and be used on some other device within the same network. Finally, if stricter security is needed, we show that traditional low-level features can be fused with attribute features to give better performance.

The contributions of this paper are three fold. First, we introduce the use of facial attributes to the task of active authentication and show with extensive experiments on MOBIO [11] and AA01[9] datasets that this approach produces promising results. Second, we present evaluations based on an implementation of our algorithm on an actual mobile device. These evaluations are necessary for all mobile continuous authentication systems to prove their feasibility. Lastly, we also present the labels for UMDAA dataset of 50 subjects each having 44 attributes.

2. Related Work

2.1. Attributes

In computer vision, almost in all problems, the very first step is to extract features from a given visual signal. The first use of attributes as higher order features was introduced in content-basedd image retrieval where they are presented as a solution to decrease the semantic gap [12, 13, 14]. Attributes were also referred to as a kind of "intermediate features". This term initially appeared in [14] referred to the features that are "low-level" semantic features but "high level" image features.

Subsequent applications of attributes were in object recognition domain and human identification. Ferrari et al. [15] learned visual attributes for objects such as "dotted" or "striped". In [16] Farhadi et al. used L1-regularized logistic regression to learn object attributes such as "has wheels" or "metallic" from images of PASCAL VOC 2008 [17] and then used them to describe objects in the image. In [18] Lampert et al. learned object attributes via kernel Support Vector Machines (SVMs)[19] in two learning paradigms, Direct and Indirect Attribute Prediction and then used these to perform object recognition. They demonstrated good results on their Animal with Attributes dataset. There are several other areas where attribute features have been shown to be useful: zero-shot learning [20], scene classification [21], and action recognition [22].

Human attributes or "soft biometrics" such as age and gender suggested in [23], have been successfully used for identity recognition/verification in many applications. In [23], Jain et al. combined height, race and gender information with fingerprint to improve the recognition accuracy on an in-house dataset. Face image retrieval solely based on attributes was investigated in [24] by Kumar et al., and in [25] by Park et al. For face verification, Kumar et al. in [26]

extracted attribute feature vectors. Zhang et al. [27] used attributes to improve face clustering in the presence of pose and illumination variations. Klare et al.[28] defined 46 facial attributes to perform suspect identification task. In [29] Layn et al. showed that attributes such as "jeans", "headphones","sunglasses" etc. can help re-identifying people seen on different cameras of a distributed camera network. Vaquero et al. [30] developed a method for searching with attributes in surveillance environments using Viola-Jones attribute detectors. In [31] Thornton et al. used attribute profiles to search in large datasets of surveillance video data. In [32] Jain et al. fused fingerprints, a few soft biometrics, and low-level features for face recognition on an in-house dataset of 263 users having 10 images each.

Detecting the presence of each attribute has been the focus of many researchers. These algorithms can be roughly divided into two groups, those which learn a specific model per attribute and those which present a general framework to learn all the target attributes together at once. Our focus in this paper is on the second group of attributes. Bourdev et al. [33] defined poselets-based on Histogram of Oriented Gradients (HOGs) [34] features and train SVMs on them. Zhang et al. [35] trained a Convolutional Neural Network (CNN) on parts extracted from full body person images using [33] to detect attributes, they achieved good results on Berkley Attributes of People dataset and Attributes25k [33] dataset. Berg et al. [36] learned one SVM per class pairs and part pairs to take into account the class relationship and part relationship and then created a feature vector out of all the SVMs. These features then were used to learn classifiers for each attribute. Kumar et al. [24] trained their local SVMs and let Adaboost to optimize for best ones for ten attributes and showed the performance on FaceTracer [24] dataset. In [26] Kumar et al. concatenated different low-level features extracted from face components and incrementally learn SVMs for each attribute and tested them on PubFig [26]. We present two approaches, one consisting of model selection between different SVMs, and another simpler one which gives efficient linear SVMs for platform implementation.

2.2. Active authentication

Early research to find alternatives for password-based authentication were focused on extracting unique characteristics from users' keystrokes. In [8], Spillane et al. suggested to use timing between key presses and the pressure patterns of keystrokes to identify users. Then in [37] Monrose et al. created a method using pseudorandom polynomials to generate a secure sequence based on keystroke time interval of users to increase password security. In [38], Klosterman et al. introduced the first continuous face verification system implemented in Linux. They also presented a comprehensive set of differences between biometric and password-based authentication systems. The next biometric based continuous authentication system design was introduced by Carrillo [39] to secure aircraft cockpit against unverified access. This was followed many studies on continuous authentication mostly for desktop computers like [40, 41, 42, 43, 44].

With the exponential growth in the use of mobile devices, active authentication on them has become the focus of many researchers. Various biometrics have been proposed to continuously authenticate the users. In [4] Frank et al. proposed a set of 30 behavioral touch features and then used a k-nearest neighbor classifier and Gaussian kernel SVMs for horizontal and vertical strokes of the user to perform authentication. [45], [46] also use touchscreen gestures for this purpose. Gait as well as device movement patterns measured by the smartphone accelerometer were used in [5], [47] for continuous authentication. Stylometry, GPS location, web browsing behavior, and application usage patterns were used in [48] for active authentication.

Face-based continuous user authentication has also been under study by researchers. In [6] Hadid et al. used Haar-like features and Adaboost [49] is employed for part detection and, Local Binary Pattern (LBP) [50] followed by nearest neighbor thresholding for identification. In [9], Fathy et al. extracted two intensity features for images, one from the whole face and one from face components. Then they compared four still image algorithms and five convex hull image set comparison methods for the AA01 dataset and compared their recognition rates. Lastly, [10] Gunther et al. provide an overview of methods that depend on low-level features for this task such as [51],[52],[53],[54] and their results on the MOBIO [11] dataset.

Multi-modal methods have always been of interest when it comes to biometrics. Fusion of speech and face was proposed in [11] by McCool et al., based on LBP features and nearest neighbor thresholding for faces. [55] fused face images with the inertial measurement unit data to continuously authenticate the users. A low-rank representation-based method was proposed in [56] for fusing touch gestures with faces for continuous authentication. Finally, a domain adaptation method was proposed in [57] for dealing with data mismatch problem in continuous authentication.

The rest of the paper is organized as follows: the details of training our attributes classifiers are presented in Section 3. Section 4 presents the results of extensive experiments on two publicly available datasets, MOBIO [11] and AA01 [9] dataset. Section 5 details the implementation details of the algorithm on the Android OS and performance evaluation of the implemented algorithm on an actual device.

Finally, Section 6 concludes the paper with a brief summary and discussion.

3. Attribute-based Authentication

In this section, we present the details of the proposed attribute-based authentication system. In particular, we describe the training data used to learn the attribute classifiers, how different classifiers are trained for each attribute and how verification is performed using the attributes.

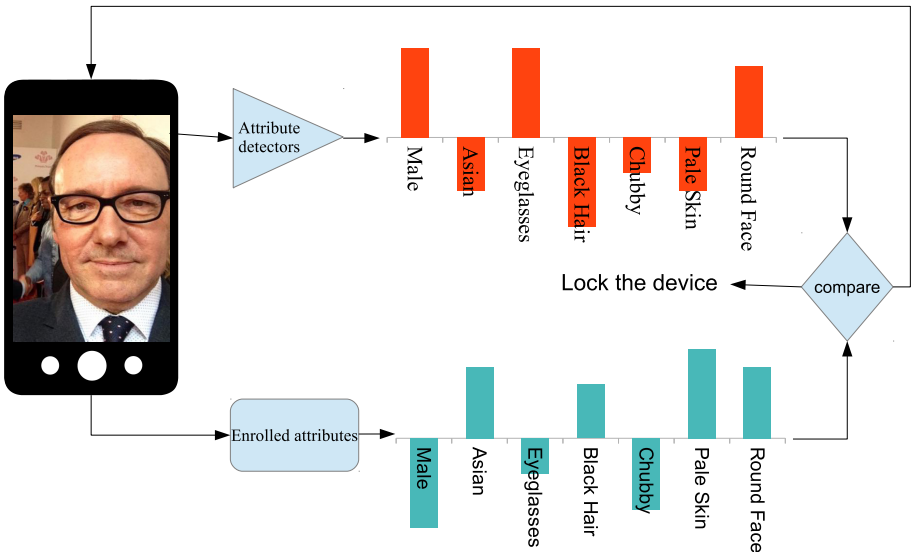


Figure 1: Overview of our attribute-based authentication method.

3.1. Training Data

PubFig dataset [26] is one of the few publicly available datasets that provides facial attributes along with face images. We use this dataset to train our attribute classifiers. PubFig dataset consists of unconstrained faces collected from the Internet by using a person's name as the search query on a variety of image search engines, such as Google Images and flickr. However, there are several challenges have to be overcome before this dataset can be effectively utilized for our application. Since the release of this dataset in 2009, many links to the images in this dataset are broken. Hence, not all the images listed in this dataset are available for downloading. As a result, we use a subset of this dataset where we could establish proper links to the images. Furthermore, the true attribute labels of the images are not provided, instead the output of their attribute classifiers are provided. As a result, we used a proper threshold to get the labels for each attribute of the available images to ensure that the classifier is certain enough about the label given to the image. Finally, rather than using all the 73 binary attributes in the PubFig dataset, we selected a more meaningful subset of 44 attributes in our implementation.

FaceTracer [24] is another publicly available dataset that has face images with 18 attributes. This dataset is smaller than the PubFig dataset and again a several hyperlinks to the images in this dataset are broken. Also, only a subset of attribute labels has been provided.

3.2. Attributes Classifiers

Each attribute classifier $Cl_i \in \{Cl_1, ..., Cl_N\}$ is trained by an automatic procedure of model selection for each attribute $A_i \in \{A_1, ..., A_N\}$, where N

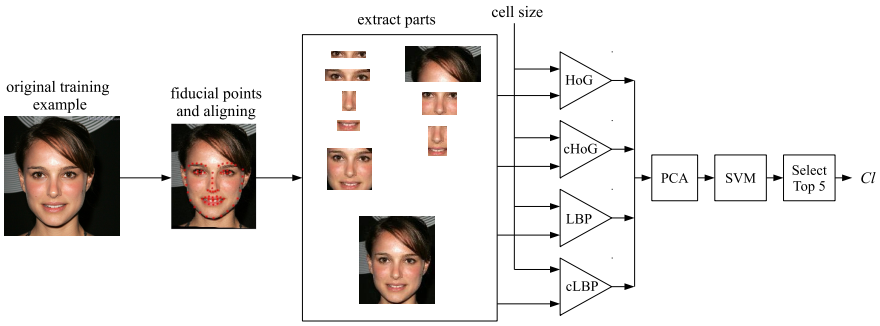


Figure 2: Training phase pipeline for each attribute classifier. Landmarks are first detected on a given face. Different facial components are then extracted from these landmarks. Then for each part, features are extracted with different cell sizes and the dimensionality of features is reduced using principle component analysis. Classifiers are then learned on these low-dimensional features. Finally, top five *Cl*s are selected as our attribute classifier.

is the total number of attributes. Automatic selection is necessary since each attribute needs a different model. Our models are indexed as follows:

1 **Facial parts:** For each attribute, a set of different facial components can be more discriminative. The face components considered for training are: *eyes*, *nose*, *mouth*, *hair*, *eyes&nose*, *mouth&nose*, *eyes&nose&mouth*, *eyes&eyebrows*, and the *full* face. In total, nine different face components are considered.

2 **Features:** For different attributes, different types of features may be needed. For example, for the attribute "blond hair", features related to color can be more discriminative than features related to texture. The following features are considered in this paper: *LBP*[50], *ColorLBP*, *HoG*[34], and *ColorHoG*. ColorLBP and ColorHOG are obtained by concatenating the HoG/LBP feature of each RGB channel. In total, four types of features are extracted using the VLFeat toolbox [58].

3 **Locality of features:** In order to capture the local information, we consider different cell sizes for the HOG and LBP features. In total, six different cell sizes, $6, 8, 12, 16, 24, 32$, are used.

The implementation of the algorithm for this section is done in Matlab [59]. We use a state-of-the-art publicly available fiducial point detection method [60] to extract the different facial components. Furthermore, the detected landmarks are also used to align the faces to a canonical coordinate system. After extracting each set of features, the Principal component analysis (PCA) is used with 99% of the energy to project each feature onto a low-dimensional subspace. An SVM with the RBF kernel is then learned on these features. This process is run exhaustively to train all possible models. For each attribute classifier, 80% of

the available data is used for training the SVMs and 20% of the data is used for model selection. The face images in the test set do not overlap with those in the training set. The total number of negative and positive classes are the same for both training and testing. Finally, among all 216 SVMs, five with the best accuracies are selected.

For a given test face image F, a feature vector $[f_{a_1}...f_{a_N}]$ is calculated by

$$f_{a_k} = \frac{\sum_{i=1}^{5} w_k^i Cl_k^i(F)}{\sum_{i=1}^{5} w_k^i}, \tag{1}$$

where $Cl_k^i(F) \rightarrow \{0, 1\}$ is the output of the ith accurate classifier for the kth attribute A_k on face image F, and w_i is the accuracy of Cl_k^i. The entire training pipeline of our method is shown in Figure 2.

3.3. Verification

We consider the continuous authentication problem as a verification problem in which given two pairs of videos or images, we determine whether they correspond to the same person or not. The well-known receiver operating characteristic (ROC) curve, which describes the relations between false acceptance rates (FARs) and true acceptance rates (TARs), is used to evaluate the performance of verification algorithms. As the TAR increases, so does the FAR. Therefore, one would expect an ideal verification framework to have TARs all equal to 1 for any FARs. The ROC curves can be computed given a similarity matrix.

We use the proposed framework to extract the attribute vector from each image in a given video. We then simply average them to obtain a single attribute vector that represents the entire video. Then, the (i, j) entry of the similarity matrix S_{attrs} is calculated as

$$s_{i,j} = \frac{1}{\|\mathbf{e}_i - \mathbf{t}_j\|_2}, \tag{2}$$

where $\mathbf{e}_i$ is the ith attribute vector representing the gallery (or enrollment) video, and $\mathbf{t}_j$ is the jth attribute vector representing the probe video.

4. Experimental Results

We evaluate the performance of the proposed attribute-based authentication method on two publicly available mobile video datasets - MOBIO [11] and AA01 [9]. In addition to the ROC curves, the Equal Error Rate (EER) is used to measure 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 authentication system.

We use an LBP-based method as a baseline for comparison. In this method, each detected face is represented by the histogram of LBP features. The same

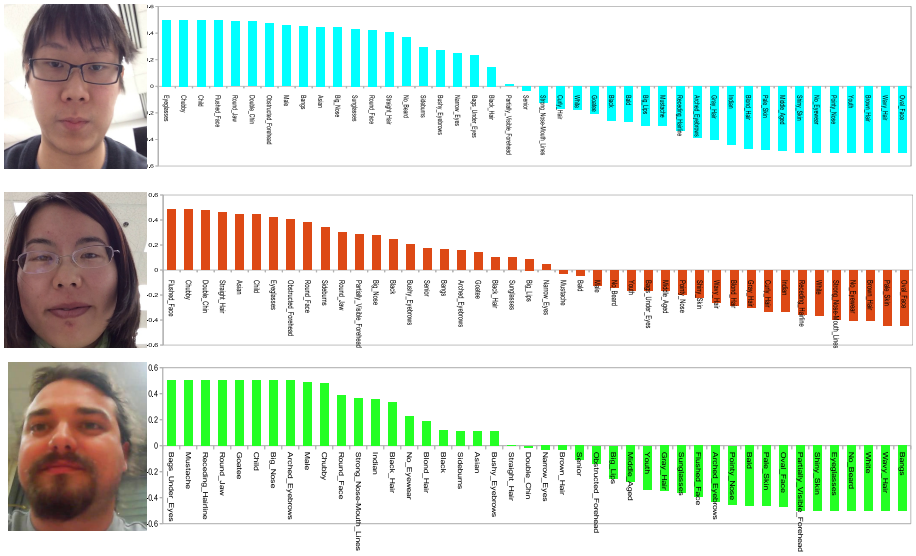


Figure 3: Illustration of our attribute classifiers on sample face images from the AA01 (first two images) and the MOBIO (last image) datasets.

aligned faces that are used for attribute feature extraction are also used to extract the LBP features. Similar to the attribute features, the LBP features from each image in a video are extracted and averaged to represent a single video. The LBP features are extracted using the VLfeat toolbox. The similarity matrix, S_{LBP}, is then built by comparing two feature vectors. This LBP-based method has been used for mobile face authentication in [11] and [6]. A third fusion score matrix, $S_{fusion} = \tilde{S}_{LBP} + \tilde{S}_{attrs}$, is calculated by z-score normalization

$$\tilde{s}_{i,j} = \frac{s_{i,j} - \bar{S}}{\sigma(S)}, \tag{3}$$

where $\bar{S}$ and $\sigma(S)$ are the mean and the standard deviation of the entries in similarity matrix S, respectively.

4.1. Attribute Classifiers

In Tables 1 and 2 the accuracies of the attribute classifiers trained using our method on PubFig and FaceTracer datasets are given. As can be seen from these tables, most of the accuracies are high. Also the accuracies for AA01[9] are provided in Table 3. The attributes for this dataset is labeled by four volunteers. It consists of 50 subjects and for each of 44 binary attributes; if 3 out of 4 people agreed on the presence of the attribute then it is set to one else zero.

Furthermore, in Figure 3 we show some sample outputs of our attribute classifiers. Results of the classifiers are scaled to be between -0.5 to 0.5. For the

Attribute	Accuracy	Attribute	Accuracy
Blond Hair	0.9089	Child	0.9538
Partially Visible Forehead	0.8645	Narrow Eyes	0.7777
Round Face	0.9156	Big Nose	0.8039
Indian	0.9714	Male	0.9451
Gray Hair	0.9091	Pointy Nose	0.816
Bags Under Eyes	0.8986	Asian	0.9225
Obstructed Forehead	0.8913	White	0.6992
Shiny Skin	0.9532	Youth	0.7299
No Eyewear	0.8875	Brown Hair	0.6725
Middle Aged	0.929	Bald	0.7909
Senior	0.8867	Wavy Hair	0.9357
Eyeglasses	0.9397	Straight Hair	0.7408
Sunglasses	0.9701	Bangs	0.9397
Mustache	0.8606	Arched Eyebrows	0.6462
Chubby	0.8815	Strong Lines	0.9308
Receding Hairline	0.8164	Pale Skin	0.793
Round Jaw	0.9357	Flushed Face	0.7819
Big Lips	0.7578	Double Chin	0.9727
No Beard	0.7766	Black Hair	0.8029
Goatee	0.9775	Curly Hair	0.8746
Black	0.7818	Bushy Eyebrows	0.836
Sideburns	0.8756	Oval Face	0.82

Table 1: Accuracies of the 44 attribute classifiers proposed in this paper on the PubFig dataset [26].

Attribute	Accuracy	Attribute	Accuracy
Asian	0.8786	middle aged	0.7321
eyeglasses	0.7214	black	0.808
sunglasses	0.89	female	0.88
smiling false	0.8	senior	0.7933
no eyewear	0.7481	hair color blond	0.7875
child	0.8276	white	0.763
mustache	0.815	youth	0.692

Table 2: Accuracies of the attribute classifiers proposed in this paper on available attributes on the FaceTracer dataset [24].

Attribute	Indoor	Lights off	Outdoor	Attribute	Indoor	Lights off	Outdoor
Asian	0.64	0.62	0.54	Bags Under Eyes	0.96	0.96	0.96
Bald	0.98	0.98	0.98	Bangs	0.88	0.88	0.88
Big Lips	0.80	0.80	0.80	Big Nose	0.90	0.90	0.92
Black	0.98	0.98	0.98	Black Hair	0.62	0.62	0.72
Blond Hair	0.96	0.96	0.96	Brown Hair	0.96	0.96	0.96
Bushy Eyebrows	0.94	0.94	0.94	Child	0.74	0.76	0.78
Chubby	0.74	0.74	0.76	Curly Hair	0.96	0.96	0.96
Double Chin	0.92	0.94	0.94	Eyeglasses	0.60	0.58	0.58
Flushed Face	0.98	0.98	0.98	Goatee	0.96	0.96	0.96
Gray Hair	0.96	0.96	0.96	Indian	0.86	0.86	0.86
Male	0.82	0.82	0.84	Middle Aged	0.96	0.96	0.96
Mustache	0.86	0.86	0.86	Narrow Eyes	0.64	0.68	0.62
No Beard	0.58	0.56	0.58	No Eyewear	0.74	0.74	0.74
Obstructed Forehead	0.84	0.84	0.88	Oval Face	0.78	0.78	0.78
Pale Skin	0.98	0.98	0.98	Partially Visible Forehead	0.70	0.70	0.70
Pointy Nose	0.98	0.98	0.98	Receding Hairline	0.86	0.86	0.90
Round Face	0.96	0.96	0.96	Round Jaw	0.76	0.74	0.76
Senior	0.98	0.98	0.98	Shiny Skin	0.98	0.98	0.98
Sideburns	0.98	0.98	0.98	Straight Hair	0.72	0.72	0.74
Strong Nose-Mouth Lines	0.82	0.82	0.82	Sunglasses	0.98	0.98	0.98
Wavy Hair	0.96	0.96	0.96	White	0.90	0.90	0.90

Table 3: Accuracy of the attribute classifiers for AA01[9] dataset.

first face, eyeglasses, chubby, round jaw, Asian, male, no beard, sideburns, bangs classifiers give high scores. This clearly matches with the image shown on the left. For the second face, it is interesting to see that the Male classifier produces a negative score since the image corresponds to a female subject. Finally, for the last face, "mustache", "goatee", "chubby" and "bags under eyes" produce high positive scores which clearly match with the image shown on the left.

4.2. MOBIO Dataset

The MOBIO dataset [11] consists of video data taken from 152 subjects. The dataset was collected in six different sites from five different countries. In total twelve sessions were captured for each subject - six sessions for phase 1 and six sessions for phase 2. The database was recorded using two mobile devices: a NOKIA N93i mobile phone and a standard 2008 MacBook laptop computer. The laptop was only used to capture videos of part of the first session. So the first session consists of data captured with both the laptop and the mobile phone. Figure 4 shows some frames from the MOBIO dataset.

In the MOBIO protocol, for each person, the data from one session is used for enrollment and the data from the remaining sessions are used for testing. In the first set of experiments with the MOBIO dataset, we do not consider the data from the laptop session. The first mobile session is considered as the enrollment session and the data from the next 11 sessions are considered for testing. The ROC curves corresponding to this experiment are shown in Figure 5 for the entire dataset. As can be seen from this figure, our attribute-based method performs comparably to the LBP-based methods. However, the best performance is achieved when the similarity matrices corresponding to the LBP and attribute features are fused. The EER values corresponding to this experiment are compared in Table 4.

Figure 4: Sample images from the MOBIO dataset. One can clearly see the different illumination conditions in this dataset.

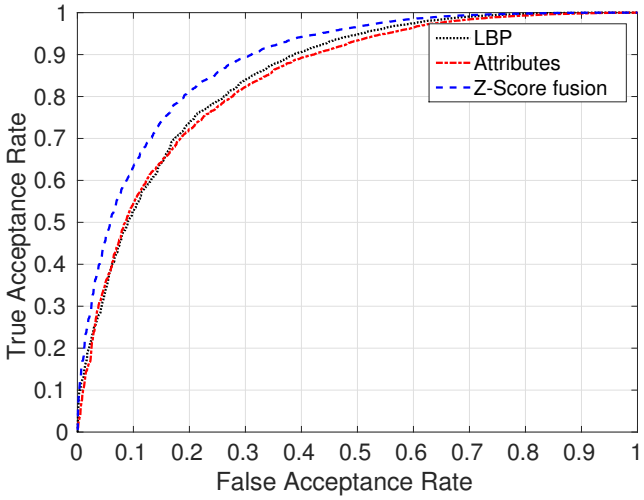


Figure 5: Performance evaluation on the MOBIO dataset.

Site	LBP	Attributes	Fusion
but	0.29	0.28	**0.25**
idiap	0.18	0.20	**0.14**
lia	0.31	**0.24**	0.25
uman	0.20	0.25	**0.18**
unis	0.24	0.28	**0.24**
uoulu	0.27	0.24	**0.23**
All together	0.22	0.23	**0.19**

Table 4: The EER values for different methods on the MOBIO dataset.

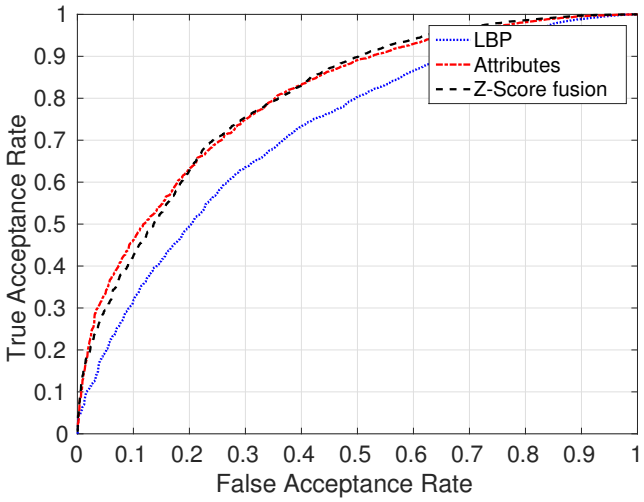


Figure 6: Cross device robustness. Laptop session videos are used for enrollment and the data from the remaining sessions are used for testing.

Enrollment	LBP	Attributes	Fusion
Laptop	0.33	**0.27**	0.27

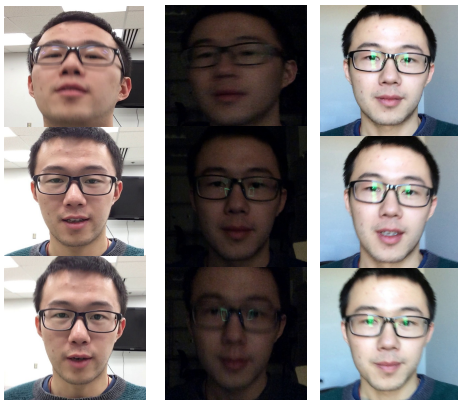
Table 5: The EER values corresponding to the cross-device experiment on the MOBIO dataset.

4.2.1. Cross-device Experiments

Images captured by different cameras have different characteristics. Since the MOBIO dataset has videos that were captured using different sensors, we conduct cross-session experiments in which the data from the laptop session are considered as the enrollment data and the data from the cell phone are used as the test videos. This experiment essentially allows us to study the robustness of different algorithms with respect to different image quality. Figure 6 and Table 5 show the ROC curves and the EER values corresponding to this experiment. As can be seen from this results, attributes are more robust to camera sensor change than LBP features. In this experiment, fusion does not necessarily improve the performance over the attributes since LBP features perform poorly.

4.3. AA01 Dataset

The AA01 dataset consists of 750 videos from 50 different individuals collected in three different sessions corresponding to three different illumination conditions. The UMDAA-01 dataset was collected using an app on an iPhone 5s. Each user performed five tasks in three sessions. The different tasks were enrollment task, document task, picture task, popup task and scrolling task. Figure 7 shows some sample images from the UMDAA-01 dataset where one can clearly see the different illumination conditions present in this dataset.



(a) (b) (c)

Figure 7: Sample images from the AA01 dataset. (a), (b) and (c) show some sample images from session 1, 2 and 3, respectively.

Enrollment	LBP	Attributes	Fusion
Indoor light	0.13	0.14	**0.10**
Low light	0.31	**0.18**	0.20
Natural light	0.19	0.16	**0.14**
AA01_all	0.34	**0.30**	0.30

Table 6: The EER values of different methods for the AA01 dataset.

In the first set of experiments using this dataset, we use the data corresponding to the enrollment task as gallery and the data from the remaining tasks for testing. Figure 8 and Table 6 show the ROC curves and the EER values, respectively corresponding to this experiment. As can be seen from these results, our attribute-based method performs much better than the LBP-based authentication system. Fusion of the LBP and the attribute similarity matrices results in performance comparable to our method as the LBP features do not perform well on this dataset.

Furthermore, we conducted several session-specific experiments on this dataset. We used the enrollment data as gallery and the data from other tasks from the same session as probe. The ROC curves corresponding to these experiments are shown in Figures 9(a)-(c). It can be seen from these figures that our attribute-based method works better than the LBP-based method, and fusion improves the result as expected. The reason that attributes work better here is that the sessions are all taken in the same day so the change in attributes are less severe than in the MOBIO dataset.

Finally, similar to the cross-device experiments on the MOBIO dataset, we conducted cross-session experiments on the AA01 dataset. We used the data from the enrollment task from one session as gallery and the data from the other sessions as probe. This experiment shows the robustness of our attribute-based

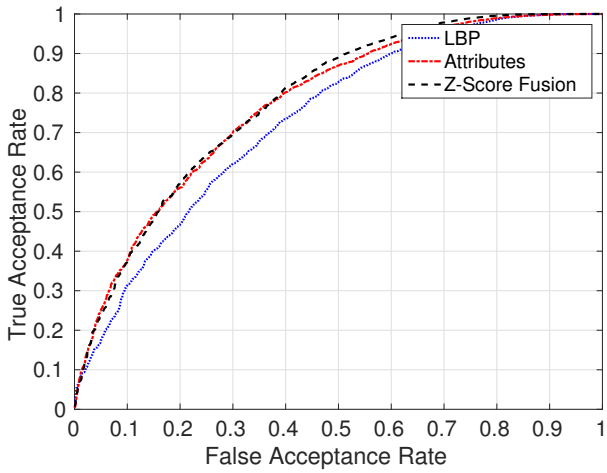


Figure 8: Performance evaluation for the AA01 dataset.

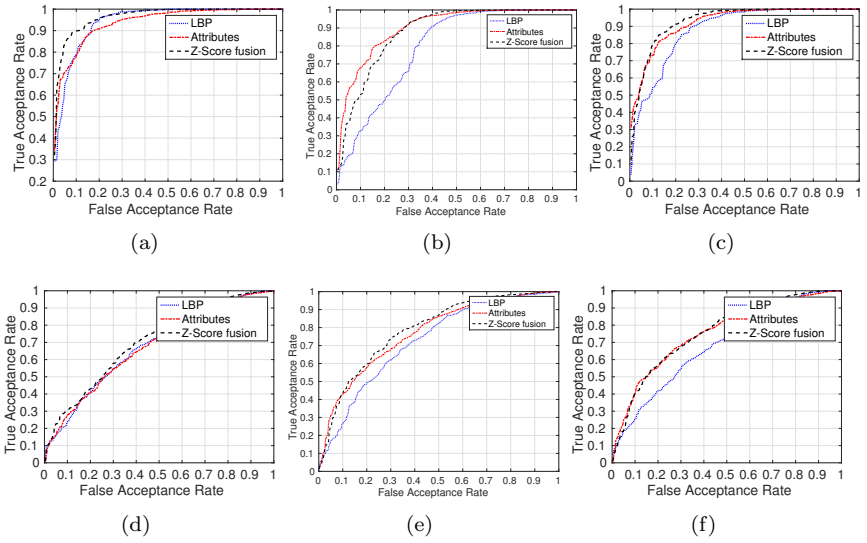


Figure 9: Session-specific performance evaluations for the AA01 dataset. (a) Gallery and probe data from session 1. (b) Gallery and probe data from session 2. (c) Gallery and probe data from session 3. (a) Gallery data from session 1 and probe data from sessions 2 and 3. (e) Gallery data from session 2 and probe data from sessions 1 and 3. (f) Gallery data from session 3 and probe data from sessions 2 and 1.

Gallery→Probe	LBP	Attributes	Fusion
$1 \rightarrow 2,3$	0.36	0.33	**0.32**
$2 \rightarrow 1,3$	0.35	0.31	**0.30**
$3 \rightarrow 1,2$	0.38	0.33	**0.31**

Table 7: The EER values corresponding to the cross-session experiments for the AA01 dataset. 1 is the office light session, 2 is the low light session, 3 is the natural light session.

method to different illumination conditions. From Figures 9(d)-(f), we see that even when the illumination conditions are different, our attribute-based method is more robust than the LBP feature-based method. From Figures 9(d)-(f) and Table 7 we see that in all cases, attributes performed better than LBP and the fusion of both gives the best results.

5. Platform implementation and evaluations

One set of the challenges of continuous mobile authentication is the computational complexity and memory usage of the algorithm. The limited computation capacity of a mobile phone is shared among many processes. So if the algorithm takes most of the CPU time, other processes will slow down. Also, computations consume energy. The more complex they are, the sooner the battery of the phone needs to be recharged. In addition, the memory capacity of the phones are limited. Algorithms with high memory usage, will force other running processes to go in the swap memory of the phone. This costly I/O operations results in both slow down and high power consumption.

As a consequence, algorithms with high complexity are run on a server and the mobile device is used just as a client that takes pictures, sends them to the server and waits for the response as suggested in [61]. This solution has two drawbacks for continuous authentication. The phone will get locked if the mobile device gets disconnected from the server. Furthermore, the system will be less secure since the communication between mobile and server can be interfered. This can result in either locking the device of the victim, or even worse unlocking it by creating a fake server which responds in a way that keeps the phone unlocked. Also, depending on the enrollment policy, we may need to re-enroll the user multiple times to account for changes in appearance or environment after the first enrollment to create a better template. It will also take time to re-enroll the user on the server again. This will be an unproductive experience for the user. In this section, we show that our approach allows enrollment and authentication of the user on the device.

Our implementation is tested on a Google Nexus 5 with 2GB of RAM and a quad core 2.2GHz CPU. The implementation is done on the Android operating system using the well-known OpenCV [62] library. Since the authentication should be done continuously, efficiency-accuracy trade-off will become very

Learning method	PCA+RBFSVM	RBFSVM	Linear SVM
Average memory usage	80MB	54MB	1MB

Table 8: Average memory usage per attribute classifier for full face

important. To explore this trade-off, we looked at three measures: memory, running time, and power consumption. Fully changing all the parameters and performing evaluations is out of the scope of the research, but we will present one pathway to platform implementation which highlights the decisions that impact the efficiency-accuracy trade-off.

5.1. Memory

Memory usage or spacial complexity has always been a challenge while implementing computer vision algorithms. The algorithm presented in Section 3 is not an exception. We changed the last two steps of our learning method to evaluate the memory usage of different models. The average test time memory requirement of each learning approach for attribute classifiers can be found in Table 8. The memory usage is calculated by first loading all the attribute classifiers and looking at the increase in memory usage and dividing that change by the number of classifiers. We use LBP features of intensity image and RGB channels in this experiment on 128×168 face images, with dimensionality of 76800 per face crop. In PCA, we keep 90% of energy. As can be seen from Table 8, since we have 44 classifiers at least, using PCA for dimensionality reduction or RBF kernel will need more than 2GB of memory in total. So we focus on linear SVMs for attributes.

5.2. Final attribute classifiers for platform

By looking at the memory usage per classifier given in Table 8, we have no choice but to simplify our classifier learning framework. For training the classifiers, we use the LFW[63] dataset. It has more subjects, hence containing more variations for each attribute. Also, the output of classifiers from [26] is available for the LFW dataset, so to train our classifiers, the same framework as in Section 3 is followed with some changes. Since we can not afford 5 PCA-RBFSVM per attribute, our goal is to train the least number of classifiers possible which gives us the desired accuracy.

We simplify our training procedure to learn one single linear SVM for each attribute while trying to consider challenges of learning attribute classifiers addressed in Section 3. We reran the experiments for the AA01 dataset from Section 4.3 with the linear classifiers learned with our simplified learning procedure. The resulting ROC curves can be seen in Figure 10 and the corresponding EER values in 9. The differences with classifiers of Section 3 and Figure 2 are:

- **Feature extraction:** In the approach discussed in Section 3, we extracted different types of features to capture the dependence of each attribute on

Method	LBP	AttrsSection3	L-SVM 1	L-SVM 0.7	L-SVM 0.5	L-SVM 0.3
Feature dim	19200	variable	76800	33280	16128	3840
Indoor lighting	0.13	0.14	0.16	0.19	**0.11**	0.22
Low lighting	0.31	0.18	0.20	0.20	**0.15**	0.24
Natural light	0.19	0.16	0.18	0.18	**0.11**	0.21
Altogether	0.33	0.30	0.29	0.29	**0.25**	0.37

Table 9: Comparison of EER values for LBP, attribute detectors of Section 3, linear models of Section 5. The scale L-SVM 1 is trained on images of size 128×168 and the rest are scaled by the indicated value. The best EER is gained from L-SVMs of Section 5 with scale 0.5.

color and scale. In the simplified model, we just extract LBP feature on gray scale image and the three channels and concatenate them together. We don't change the cell size of LBP to capture dependence on locality. Instead, we perform evaluation with different image sizes and choose the one that works best for all attributes together.

- **No PCA** No dimensionality reduction step is employed after feature extraction, because loading the PCA basis sets on a phone needs significant memory space.

- **No kernel** A linear classifier is learned, because from memory usage in Table 8 it is impractical to load the kernelized classifiers into memory.

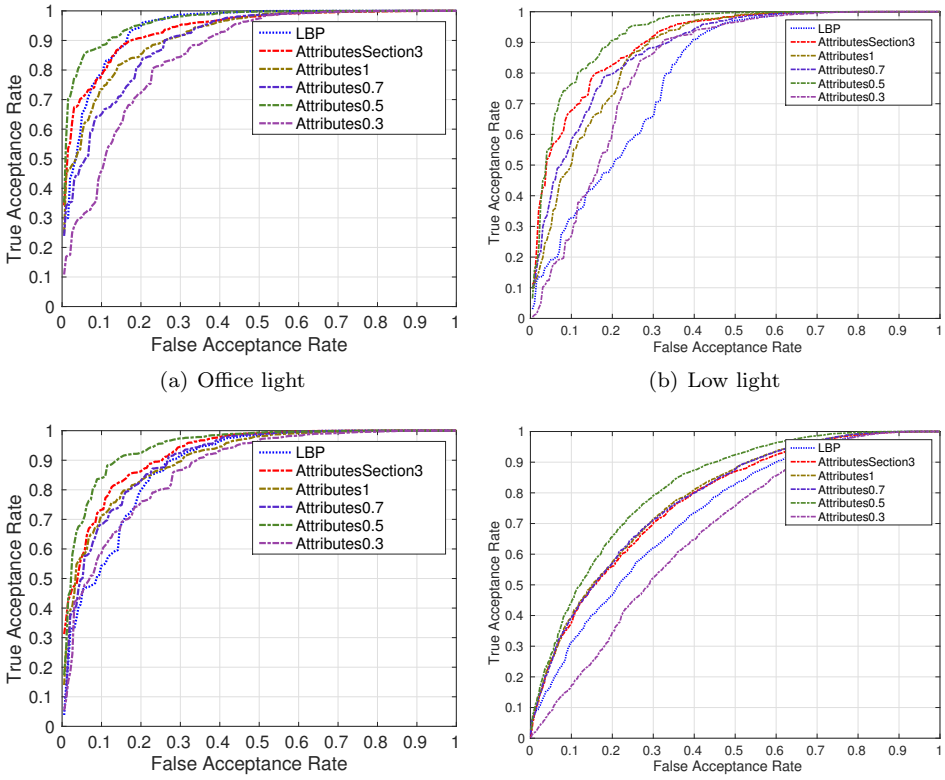
- **No part-based attribute classifier** We just train classifiers on the full face image in the simplified model. Extracting the face parts from face image is not a trivial task and adds to the complexity of the model. Also linear classifiers optimize the weights that are directly related to pixel values, so choosing face parts is taken care of by SVM optimization objective to some extent.

- **Attribute dimmension value** In Section 3, we took the weighted average of binary decision values of the top five attribute classifiers as the dimension value. In the simplified learning approach, we just use the distance from margin of each attribute classifier. This is valid since we trained the attribute classifiers with the same image size.

The interesting result is that the classifier with scale 0.5 performs better than the ones from Section 3. The most important one is probably the last step of the approach presented in Section 3. For the last step, we fuse the output of top the top five SVMs to get a score by taking the weighted average of binary decision values of each attribute classifier. This results in a discrete and finite range of scores. However, the scores of the simplified model are the distances from the margins of the linear classifiers which gives a continuous value and hence more discriminative range of value for different faces.

5.3. Frames per second and power consumption

Since linear attribute classifiers on the full face turned out to be the winner of accuracy-efficiency tradeoff, we test their speed and power consumption. For



(a) Office light

(b) Low light

(c) Natural light

(d) Altogether

Figure 10: Comparison of linear SVMs with model learned in section 3 and LBP. The best result among all is achieved with linear models of scale 0.5 i.e. face crop size of 64×80.

Scale	0.3		0.5		0.7		1	
Size/dim	$32 \times 48/3840$		$64 \times 80/16128$		$88 \times 112/33280$		$128 \times 168/76800$	
Detection/Alignment	W/O	W/	W/O	W/	W/O	W/	W/O	W/
FPS	114	29	31	16	13	8	5	4
Energy	26.8J	128.9J	93.5J	201.2J	207J	369.1J	524.9J	603J
Energy per frame	5.4mJ	25.8mJ	18.7mJ	40.2mJ	41.4mJ	73.8mJ	105mJ	120.6mJ
Endurance (hours)	16.6	16	16.2	15.6	15.6	14.7	14	13.6

Table 10: The speed and power consumption of different realization of the classifiers learned with the simplified training framework on Google Nexus 5 device. W/O column means our algorithm extract all the attributes given aligned and cropped face. In last row we assumed that we are doing authentication with the speed 1fps. W/ column first detects the face then extracts attributes. We can authenticate 17.6 hours every second employing our classifiers with best EER of Table 9 on a Nexus 5.

speed, we look at the number of frames that we can authenticate per second and for power consumption we use the power consumption profiler presented by Zhang et al. [64]. We extract the 44 dimensional feature vector for 5000 frames with each set of attribute classifiers indexed by scale. Android provides a mechanism to get the time in miliseconds, so we can measure the exact time up to miliseconds that it takes for each set of classifiers to process the 5000 frames. Also the power profiler provides the energy consumption in Joules up to $0.1J$ for each running application. The numbers for different setups of our algorithm are provided in 10. The landmark detection which was done with Asthana et al. [60] in Section 3 is replaced by the algorithm of Kazemi et al. [65] which is implemented using DLib [66]. This algorithm adds a 90MB to memory consumption but it is very fast. The evaluation for each scale is done in two settings, one with Haar face detection and DLib alignment and one without this step. From the table we see that face detection and alignment step add around 20mJ energy consumption per frame and reduces the FPS significantly. In the worst case with fastest available face and landmark detection methods, our algorithm can authenticate users at the speed of 4 frames per second. This is more than enough for authentication task which probably requires authenticating every couple of seconds.

Google Nexus 5 battery capacity is $2300mAh$ and the average working voltage is $3.8V$ which can be verified by running the power profiler of [64]. This means that it has in total $8740mWh$. If we run the profiler without our authentication program on the phone for more than 5 minutes, it shows the average power usage as $520mW$ which means the phone will last for 16.8 hours. The last row of Table 10 shows how many hours our algorithm can run in background if we do authentication once every second considering these numbers.

6. Conclusion and Future Directions

We presented a continuous face-based authentication method using facial attributes for mobile devices. We trained binary attribute classifiers and showed their effectiveness as feature vectors for active authentication with extensive experiments. We showed that attribute-based scores alone can improve the ver-

ification results. Furthermore, we showed that in situations where the low-level features such as LBP are reliable, verification results can be further improved by fusing the resulting scores with the attribute-based scores. We also evaluated the different realizations of our method on an actual cell phone and showed that the authentication algorithm can be implemented with low memory usage, power consumption and with the speed of more than four frames per second.

In the future, we are planning on exploring how attributes can be detected more reliably from mobile images using deep CNNs, and also how we can effectively adapt the attribute classifiers to changing attributes of the user, such as aging or facial hair change by exploiting classifiers with feedback.

Acknowledgement

This work was supported by cooperative agreement FA8750-13-2-0279 from DARPA.

References

[1] Wikipedia, Password — wikipedia, the free encyclopedia, [Online; accessed 30-September-2015] (2015).
URL https://en.wikipedia.org/w/index.php?title=Password&oldid=682883615

[2] A. Jain, L. Hong, S. Pankanti, Biometric identification, Communications of the ACM 43 (2) (2000) 90–98.

[3] CNBC, Smartphones: Hackers target for 2015 (2015).
URL http://www.cnbc.com/2014/12/31/smartphones-hackers-target-for-2015.html

[4] M. Frank, R. Biedert, E. Ma, I. Martinovic, 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) (2013) 136–148.

[5] M. Derawi, C. Nickel, P. Bours, C. Busch, Unobtrusive user-authentication on mobile phones using biometric gait recognition, in: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010, pp. 306–311.

[6] A. Hadid, J. Heikkila, O. Silven, M. Pietikainen, Face and eye detection for person authentication in mobile phones, in: ACM/IEEE International Conference on Distributed Smart Cameras, 2007, pp. 101–108. doi:10.1109/ICDSC.2007.4357512.

[7] V. M. Patel, R. Chellappa, D. Chandra, B. Barbello, Continuous user authentication on mobile devices, IEEE Signal Processing Magazine.

[8] R. Spillane, Keyboard apparatus for personal identification, IBM Technical Disclosure Bulletin 17 (3346) (1975) 3346.

[9] M. E. Fathy, V. M. Patel, R. Chellappa, Face-based active authentication on mobile devices, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2015.

[10] M. Gunther, A. Costa-Pazo, C. Ding, E. Boutellaa, G. Chiachia, H. Zhang, M. de Assis Angeloni, V. Struc, E. Khoury, E. Vazquez-Fernandez, et al., The 2013 face recognition evaluation in mobile environment, in: Biometrics (ICB), 2013 International Conference on, IEEE, 2013, pp. 1–7.

[11] C. McCool, S. Marcel, A. Hadid, M. Pietikainen, P. Matejka, J. Cernocky, N. Poh, J. Kittler, A. Larcher, C. Levy, D. Matrouf, J.-F. Bonastre, P. Tresadern, T. Cootes, Bi-modal person recognition on a mobile phone: using mobile phone data, in: IEEE ICME Workshop on Hot Topics in Mobile Multimedia, 2012.

[12] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, Pattern Recognition 40 (1) (2007) 262–282.

[13] R. Datta, J. Li, J. Z. Wang, Content-based image retrieval: approaches and trends of the new age, in: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, ACM, 2005, pp. 253–262.

[14] M. Obeid, B. Jedynak, M. Daoudi, Image indexing & retrieval using intermediate features, in: Proceedings of the ninth ACM international conference on Multimedia, ACM, 2001, pp. 531–533.

[15] V. Ferrari, A. Zisserman, Learning visual attributes, in: Advances in Neural Information Processing Systems, 2007, pp. 433–440.

[16] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1778–1785.

[17] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge 2008, 2008.

[18] C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (3) (2014) 453–465.

[19] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[20] M. Liu, D. Zhang, S. Chen, Attribute relation learning for zero-shot classification, Neurocomputing 139 (2014) 34–46.

[21] G. Patterson, J. Hays, Sun attribute database: Discovering, annotating, and recognizing scene attributes, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2751–2758.

[22] J. Liu, B. Kuipers, S. Savarese, Recognizing human actions by attributes, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3337–3344.

[23] A. K. Jain, S. C. Dass, K. Nandakumar, Soft biometric traits for personal recognition systems, in: Biometric Authentication, Springer, 2004, pp. 731–738.

[24] N. Kumar, P. N. Belhumeur, S. K. Nayar, FaceTracer: A Search Engine for Large Collections of Images with Faces, in: European Conference on Computer Vision (ECCV), 2008, pp. 340–353.

[25] U. Park, A. K. Jain, Face matching and retrieval using soft biometrics, Information Forensics and Security, IEEE Transactions on 5 (3) (2010) 406–415.

[26] N. Kumar, A. C. Berg, P. N. Belhumeur, S. K. Nayar, Attribute and Simile Classifiers for Face Verification, in: IEEE International Conference on Computer Vision (ICCV), 2009.

[27] L. Zhang, D. V. Kalashnikov, S. Mehrotra, Context-assisted face clustering framework with human-in-the-loop, International Journal of Multimedia Information Retrieval 3 (2) (2014) 69–88.

[28] B. F. Klare, S. Klum, J. C. Klontz, E. Taborsky, T. Akgul, A. K. Jain, Suspect identification based on descriptive facial attributes, in: Biometrics (IJCB), 2014 IEEE International Joint Conference on, IEEE, 2014, pp. 1–8.

[29] R. Layne, T. M. Hospedales, S. Gong, Q. Mary, Person re-identification by attributes., in: BMVC, Vol. 2, 2012, p. 8.

[30] D. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur, M. Turk, et al., Attribute-based people search in surveillance environments, in: Applications of Computer Vision (WACV), 2009 Workshop on, IEEE, 2009, pp. 1–8.

[31] J. Thornton, J. Baran-Gale, D. Butler, M. Chan, H. Zwahlen, Person attribute search for large-area video surveillance, in: Technologies for Homeland Security (HST), 2011 IEEE International Conference on, IEEE, 2011, pp. 55–61.

[32] A. K. Jain, K. Nandakumar, X. Lu, U. Park, Integrating faces, fingerprints, and soft biometric traits for user recognition, in: ECCV Workshop BioAW, Vol. 4, Springer, 2004, pp. 259–269.

[33] L. Bourdev, S. Maji, J. Malik, Describing people: A poselet-based approach to attribute classification, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1543–1550.

[34] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886–893.

[35] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L. Bourdev, Panda: Pose aligned networks for deep attribute modeling, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1637–1644.

[36] T. Berg, P. N. Belhumeur, Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 955–962.

[37] F. Monrose, M. K. Reiter, S. Wetzel, Password hardening based on keystroke dynamics, International Journal of Information Security 1 (2) (2002) 69–83.

[38] A. J. Klosterman, G. R. Ganger, Secure continuous biometric-enhanced authentication.

[39] C. M. Carrillo, Continuous biometric authentication for authorized aircraft personnel: A proposed design, Tech. rep., DTIC Document (2003).

[40] A. Altinok, M. Turk, Temporal integration for continuous multimodal biometrics, in: Proceedings of the Workshop on Multimodal User Authentication, Citeseer, 2003.

[41] T. Sim, S. Zhang, R. Janakiraman, S. Kumar, Continuous verification using multimodal biometrics, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (4) (2007) 687–700.

[42] K. Niinuma, A. K. Jain, Continuous user authentication using temporal information, in: SPIE Defense, Security, and Sensing, International Society for Optics and Photonics, 2010, pp. 76670L–76670L.

[43] K. Niinuma, U. Park, A. K. Jain, Soft biometric traits for continuous user authentication, Information Forensics and Security, IEEE Transactions on 5 (4) (2010) 771–780.

[44] R. Janakiraman, S. Kumar, S. Zhang, T. Sim, Using continuous face verification to improve desktop security, in: Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on, Vol. 1, IEEE, 2005, pp. 501–507.

[45] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, N. Nguyen, Continuous mobile authentication using touchscreen gestures, in: IEEE Conference on Technologies for Homeland Security, 2012, pp. 451–456.

[46] H. Zhang, V. M. Patel, M. E. Fathy, R. Chellappa, Touch gesture-based active user authentication using dictionaries, in: IEEE Winter conference on Applications of Computer Vision, IEEE, 2015.

[47] A. Primo, V. Phoha, R. Kumar, A. Serwadda, Context-aware active authentication using smartphone accelerometer measurements, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, 2014, pp. 98–105. doi:10.1109/CVPRW.2014.20.

[48] L. Fridman, S. Weber, R. Greenstadt, M. Kam, Active authentication on mobile devices via stylometry, gps location, web browsing behavior, and application usage patterns, IEEE Systems Journal.

[49] P. Viola, M. J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, International Journal of Computer Vision 63 (2) (2005) 153–161.

[50] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: Application to face recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (12) (2006) 2037–2041.

[51] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, J. Weng, Discriminant analysis of principal components for face recognition, in: Face Recognition, Springer, 1998, pp. 73–85.

[52] D. Cox, N. Pinto, Beyond simple features: A large-scale feature search approach to unconstrained face recognition, in: Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE, 2011, pp. 8–15.

[53] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 1, IEEE, 2005, pp. 786–791.

[54] L. Wiskott, J.-M. Fellous, N. Kuiger, C. Von Der Malsburg, Face recognition by elastic bunch graph matching, Pattern Analysis and Machine Intelligence, IEEE Transactions on 19 (7) (1997) 775–779.

[55] D. Crouse, H. Han, D. Chandra, B. Barbello, A. K. Jain, Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data, in: International Conference on Biometrics, 2015.

[56] H. Zhang, V. M. Patel, R. Chellappa, Robust multimodal recognition via multitask multivariate low-rank representations, in: IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, 2015.

[57] H. Zhang, V. M. Patel, S. Shekhar, R. Chellappa, Domain adaptive sparse representation-based classification, in: IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, 2015.

[58] A. Vedaldi, B. Fulkerson, Vlfeat: An open and portable library of computer vision algorithms, `http://www.vlfeat.org/` (2008).

[59] MATLAB, version 8.4.0.150421 (R2014b), The MathWorks Inc., Natick, Massachusetts, 2014.

[60] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Robust discriminative response map fitting with constrained local models, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3444–3451.

[61] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, B. Girod, Outdoors augmented reality on mobile phone using loxel-based visual feature organization, in: Proceedings of the 1st ACM international conference on Multimedia information retrieval, ACM, 2008, pp. 427–434.

[62] G. Bradski, Opencv, Dr. Dobb's Journal of Software Tools.

[63] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007).

[64] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, L. Yang, Accurate online power estimation and automatic battery behavior based power model generation for smartphones, in: Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, ACM, 2010, pp. 105–114.

[65] V. Kazemi, J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1867–1874.

[66] D. E. King, Dlib-ml: A machine learning toolkit, Journal of Machine Learning Research 10 (2009) 1755–1758.