Towards usable authentication on mobile phones: An evaluation of speaker and face recognition on off-the-shelf handsets

Rene Mayrhofer
University of Applied Sciences Upper Austria
Softwarepark 11, A-4232 Hagenberg, Austria
rene.mayrhofer@fh-hagenberg.at

Thomas Kaiser
University of Applied Sciences Upper Austria
Softwarepark 11, A-4232 Hagenberg, Austria
thomas.kaiser@students.fh-hagenberg.at

ABSTRACT

Authenticating users on mobile devices is particularly challenging because of usability concerns: authentication must be quick and as unobtrusive as possible. Therefore, biometric methods seem well suited for mobile phones. We evaluate both speaker and face recognition methods on off-the-shelf mobile devices concerning their accuracy, suitability for dealing with low-quality recordings, and running with limited resources. Our results show that speaker and face recognition can realistically be used on mobile phones, but that improvements – e.g. in the form of combining multiple methods – are still necessary and subject to future work.

1. INTRODUCTION

Authentication of users to their own mobile phone is a challenging task: on the one hand, it should be sufficiently secure to prevent unauthorized use even with physical access to the device (e.g. after a theft) and be performed frequently enough to reflect common usage patterns (e.g. leaving the phone on one's desk for a few minutes should not enable a passer-by to unlock it); on the other hand, most forms of authentication require conscious user effort and are therefore obtrusive. The difficulty is therefore to construct authentication methods that are sufficiently unobtrusive for users to perform frequently.

Biometric authentication methods have been considered as a viable compromise between usability and security, and they seem especially suitable for authenticating to mobile phones when considering their array of sensors available in off-the-shelf devices. Two such methods that have seen significant research in the past are speaker and face recognition, and both seem compelling because human users are used to recognizing each other based on their voices and faces. However, no single biometric authentication method has so far been shown to be both secure and usable on mobile devices in practical settings. Therefore, we evaluate both speaker and face recognition algorithms with a focus on using them for authenticating users to their own mobile phones; the future aim is to combine them dynamically based on the recognition rates of each of the methods and/or the context of use. Important evaluation criteria are consequently not only the recognition rate, but also recognition speed on limited processors and robustness when dealing with noisy sensors such as built-in microphones and low-quality front cameras in off-the-shelf devices. Based on this evaluation, a proof of concept for speaker and for face recognition was implemented on the Android platform and the results of a preliminary field study are presented.

2. SPEAKER RECOGNITION

Speaker recognition is a technique to identify a person based on their voice. There are two different variants of this task: *speaker verification* and *speaker identification*. In the case of speaker verification, the subject claims an identity, which the system tries to verify as correct or reject. In speaker identification, there is no claim of identity and the recorded speech signal has to be matched with all known subjects to determine the speaker's identity (which may be unknown, depending on a chosen minimum match quality). For authentication, the focus will typically lie on speaker verification because mobile phones are assumed to be single-user devices. However, within the scope of the current paper, we study speaker identification as the general (and more difficult) problem and leave speaker verification as a trivial specialization for concrete implementations.

Speaker recognition systems can be divided in *text-dependent* and *text-independent* ones. In text-dependent systems the user has to choose to speak one or more of some specified phrases or speak a given sentence. The user may also be required to speak out loud a sequence of randomly generated numbers. This also serves to prevent *replay attacks* on the system where a pre-recorded audio signal of an authorized speaker is played back to gain illegal access. We focus on text-independent systems for usability reasons.

2.1 Feature Extraction

Different features for speaker recognition have previously been described in literature, including low-level features based on the signal spectrum and high-level features based on the spoken language [10]. Low-level features are generally easier and faster to compute and can be used for real time recogni-

tion, whereas higher level features are more robust against noise and channel variations, but also vastly more complex to compute and require more training data. For performance reasons, we rely on low-level features for efficient matching on mobile devices.

As in most machine learning setting, speaker recognition is split into a training and a verification phase. During training, speakers give their identity and provide a sample of their voice in the form of a speech signal. We currently use short-term spectral features of the speech signal. Since the human voice apparatus changes constantly during articulating speech, those features must be frequently re-calculated. This means that the audio signal is broken down into short frames of 30 to 60 milliseconds duration. Within this interval, the spectral attributes of speech are considered to remain stationary.

Within each time window, we use so-called *Mel frequency cepstral coefficients (MFCC)*, which are triangular bandpass filter banks that attempt to simulate the human hearing system. These features were introduced in the early 1980's and have been proven successful in practice since then [4]. The sum of each filter output is compressed logarithmically to the target range and a discrete cosine transform (DCT) is applied to these sum values, creating a vector of MFCCs (the size corresponds to the number of Mel filters applied) [6]. Often the first coefficient is discarded as it may be regarded as a measure of the average energies in each frequency band.

The resulting vector then used as one dimension of the feature vector – one for each time frame. The set of all feature vectors (for all time frames of the recording) may be used to compare speakers to each other.

2.2 Speaker models

Based on previous results on speaker recognition [10], we use a template based model with one template for each known subject and Vector Quantization (VQ) for comparing features vectors to these templates during the verification phase. During training, we rely on a simple k-means clustering to compute the centroids of all known samples belonging to each of the subjects. These centroids are used as codebook after training. In the verification phase, the minimum average distance of test feature vector to all codebooks is calculated according to:

$$D_Q(I,V) = \frac{1}{T}\sum_{t=1}^{T} \min_{1\le k\le K} d(i_t, v_k) \qquad (1)$$

where D_Q is the average quantization distortion between the identification feature vector I and the speaker model codebook V. $d(x,y)$ is a multi-dimensional distance function such as the Euclidean (L2) distance.

The codebook with the minimum average distortion to the test feature vector is declared as the identity of the speaker. This method has one obvious weakness: if the identification sample is from a previously unknown speaker, the system will still choose the speaker model with the least quantization error as the winner. To give some more certainty to the verification process, a so-called *background* or *cohort* speaker model may be used [9]. With the background speaker model, an average distortion from the verification feature vector to all existing speaker models is calculated simultaneously. Only when the minimum average distortion to a single speaker model is less by some threshold amount than the average distortion to all speakers, a match is declared.

2.3 Evaluation

Recording the training and test samples as well as all the processing and calculation were done on-device on the mobile phone, specifically a *Desire HD* by *HTC*[1] using Android version 2.2. The recordings were taken using the built-in main microphone of the device.

The study was done with 8 participants with 6 male and 2 female speakers between the ages of 22 and 52. For the training phase each subject recorded a 60 s voice sample by reading the same (German) text[2] with controlled environment noise (low noise level, single office), which were used to train the speaker models. Additionally, each subject recorded a 15 s sample reading a different text for each speaker. This second sample was used as identification sample that was matched against the trained speaker models. The result of the study was to determine how well speakers could be identified when recording all audio and doing all processing on a mobile phone. The audio recording part was limited to a sample rate of 8000 Hz and 8 bit quantization resolution, which compares to the quality level of telephony voice transmission and was also used in the SpeakerSense project to simulate practical recordings [11]. We used an FFT window length of 512 samples with 50% overlap to compute 13 MFCCs based on 15 Mel filters for a speaker model codebook size of 64.

Table 1: Comparison of each speaker's sample to all codebooks (best match highlighted)

30.6	35.8	40.5	44.8	37.0	41.8	38.5	41.4
36.9	30.0	39.8	55.5	33.9	34.2	34.5	36.2
40.7	39.2	27.8	37.0	41.3	43.5	37.5	45.3
44.6	52.3	38.5	31.2	56.4	64.9	57.7	61.8
32.3	29.9	36.1	54.3	28.6	30.9	30.1	31.7
38.5	33.6	39.6	57.7	34.0	31.1	34.2	34.6
37.0	32.3	35.8	51.1	34.0	34.1	30.9	35.5
35.5	33.8	39.3	54.6	32.9	35.3	34.6	28.0

Table 1 shows our end results as a matrix of each speaker's verification sample compared to all speaker's codebooks, i.e. the average quantization error of the calculated feature vector to the speaker's codebook template. The difference between the lowest found average quantization error and the average error to all codebooks gives a measure of the confidence that can be used to determine if a match is close enough for authentication. This will form a basis for probabilistic reasoning on authentication and is subject to future work. We see in the main diagonal that the average quantization error from a speaker to their own codebook is 29.78

[1] http://www.htc.com/europe/product/desirehd/overview.html
[2] http://de.wikipedia.org/wiki/Spracherkennung

and the average quantization error of the speakers to all other speakers is 40.26. This means that the quantization error for other speakers is higher by a factor of 1.35 than the error for the same speaker. The table also shows that in each row, the same speaker has the lowest average quantization error, which is the desired result.

These results show that it was possible in every case to identify the correct speaker. The margin between correct speaker and all other speakers is however quite small. This makes it difficult to distinguish between a recognized, existing speaker and someone unknown to the system trying to access the system. Setting a meaningful threshold value to make the call between a known and an unknown speaker may prove challenging, but is not required for the purpose of speaker verification, i.e. for single-user system authentication. In this simplified case, a threshold range of factor 1.35 seems sufficient to distinguish between successful and unsuccessful authentication. However, future work will consider multiple users on shared mobile devices such as a tablet in shared in a family.

3. FACE RECOGNITION

Face recognition is the identification of people based on image data. To verify someone's identity, at first the image has to analyzed to find the face (called face detection) before any features of the individual face can be extracted. We focus on two different algorithms for feature extraction: the well-known Eigenfaces method as a baseline and a discrete cosine transform (DCT) based algorithm as the most promising with low-quality images (such as taken by the front camera of a mobile phone). To this end, a small database of face images was created specifically to study the effect of using different cameras for training of the system and identification and especially to create images taken by a mobile phone's camera.

As for speaker recognition, we can distinguish between verification and identification [8, 15]; again, our focus for mobile device authentication is on verification, but we implement identification among different subjects as the more general problem.

For face recognition algorithms to work correctly, it is essential that the face detection part of the system delivers the input data (faces cropped from images containing background and other content) reliably and consistently in the same dimensions and aligned equally. Especially the Eigenfaces algorithm described later is very sensitive to errors in face registration and alignment. Usually faces are also converted to grayscale images prior to processing, to reduces the effects of changing hair colors, different colors in illumination, etc [16].

3.1 Algorithms

Eigenfaces. Eigenfaces is one of the most well-known face recognition algorithms and one of the first examples of practical face recognition [14, 2]. We currently use the implementation from OpenCV [3] on different face databases as a baseline for evaluating potential improvements.

The principal component analysis used for reducing image complexity only needs to be performed in the training phase and as such this method is very fast in the testing phase [1] and therefore well suited for authentication on resource limited devices such as mobile phones.

DCT-based face recognition. As a compromise between robustness and recognition speed, a literature review pointed to a DCT-based face recognition algorithm [5] as a promising candidate for our intented scenarios, which we implemented in Java and ported to run on Android handsets. Figure 1 shows an overview of the DCT-based face recognition.

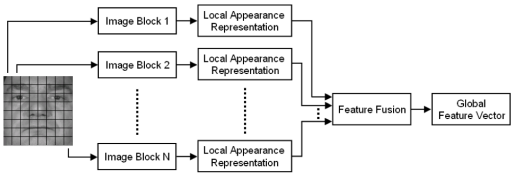

Figure 1: Building a representative value vector from a subject's face image [5]

Each image is divided into two-dimensional blocks of specific size. This is a trade-off between data reduction and fine granularity of representation. Each block is transformed using DCT, resulting in a matrix of DCT coefficients with the same dimensions as the chosen image block size. Typically, only a certain small number of coefficients (between 5 and 25 in our experiments) are chosen to represent a single image block. To best retain this representative power and discard less relevant coefficients, a zig-zag readout is applied in order to directly get a one-dimensional array of the important values. The first coefficient containing the DC content of the signal is discarded, because it represents the average intensity value of this image block and has no descriptive power.

At this point, we have a representation of the appearance of a single image block consisting of a number of real-valued coefficients. The process is repeated for each image block and the coefficients from each block are concatenated to construct the overall feature vector that is used to represent the whole image, and thus the face contained within. This feature vector is stored with an identification of the subject for future comparisons. The feature vector may be normalized with different techniques prior to storage.

In the online testing phase during face identification, we build a feature vector representing all newly presented images as described above. Then the unknown feature vector is classified using a nearest neighbor classification: a feature vector of length N is compared to all stored feature vectors using a chosen distance metric. In this work, the L1 and L2 norm were used. The closest match is found, and may be reported or confirmed as positive identification depending on a threshold value.

Parameters. A number of parameters need to be optimized for this DCT-based face recognition:

- Image size: The same size was used always for both training and testing, and we tried both $64x64$ and $128x128$ pixels.
- Block size: The size of the blocks that are transformed at once with the DCT were varied from $8x8$ up to $128x128$ depending on the currently used image size.
- Number of coefficients per block: The amount of DCT coefficients that were taken from each block to build the complete feature vector was varied from 5 to 25. The first coefficient containing the DC content of the current image block was always discarded.
- Distance metric: For nearest neighbor classification during the testing phase, the L1 or L2 norm were used.

Normalization of coefficients. Two normalization functions inspired by [5] are applied to the DCT coefficients. Consider that the DCT is a transformation that conserves all energy in the signal. Thus blocks with different brightness (which equals grayscale intensity values in the used images) levels have a different impact on the classification phase. Also the value range of the DCT coefficients differs significantly from the first few (large values) to the later ones; the first few coefficients have a larger impact on classification but do not have more descriptive power.

First, to overcome the problem of differing intensity/brightness, the coefficients of each block are normalized to 1, i,e. each coefficient is divided by the magnitude of the whole feature vector. The normalized coefficient f_i^U is calculated as $f_i^U = f_i / \|f\|$. Second, each coefficient is divided by their standard deviation across all training samples, i.e. all blocks. The normalized coefficient, $f_{i,j}^S$, is calculated as $f_{i,j}^S = f_{i,j}/\sigma(f_i)$ where $f_{i,j}$ is the j^{th} DCT coefficient in image block number i and $\sigma(f_i)$ is the standard deviation of the j^{th} coefficient across all image blocks.

3.2 Evaluation

There is a large number of face databases available[3] with widely differing number of subjects, number of images, varying conditions and, most importantly for researchers, varying availability and licenses. For the preliminary evaluation, we chose two databases that were freely available. The ORL database contains 40 subjects with 10 images each, for a total of 400 subjects with consistent illumination, while the Caltech database contains (in the version we used) 22 subjects with 259 images in total and a differing amount of images per subject with different settings.

Additionally, we took a small set of pictures to evaluate the effects of different cameras during training and testing. This database contains 12 subjects with 4 images per subject: two were taken with a Nikon digital single-lens reflex (DSLR) camera and two were taken by the subjects themselves using a Samsung Nexus S front camera. All images are taking from the front with the subject facing the camera directly; the self-shot images are naturally taken from a slightly lowered angle.

[3]See for example `http://face-rec.org/databases/`

Figure 2: A single subject in the new face database; two images taken in a controlled environment and two images taken by the subject himself.

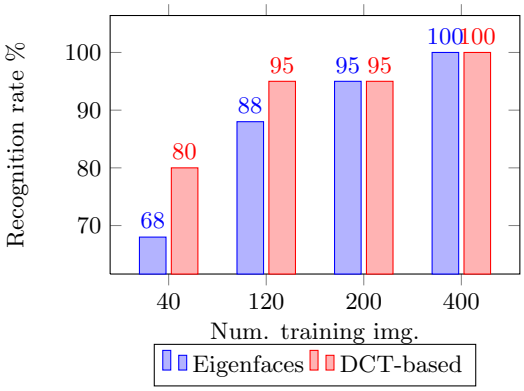

Figure 3: Results with ORL database

3.3 Experiment

For each test run, we used a specific image set for training and another for testing. Generally the results show that the performance of the face recognition improves with the number of training images used. From the ORL database, we used the *ORL5* image set for testing and all others for training, while from the Caltech database we used the *CAL4* set for testing and all others for training. As shown in Fig. 3 and 4, the last case is a test run with all images used for training and acts as a sanity check to confirm that algorithms achieve a 100% recognition rate when trained with all images they are supposed to recognize; for practical results, this last case should not be taken into account. We see that the DCT-based algorithm achieves sufficiently high recognition rates of over 90% when trained with at least around 60 images

4. CONCLUSIONS AND OUTLOOK

Our current implementation and results are work in progress. Although we have demonstrated that both speaker and face recognition can realistically be used for authenticating users on off-the-shelf mobile phones such as the various Android devices, recognition rates need to be improved for a better

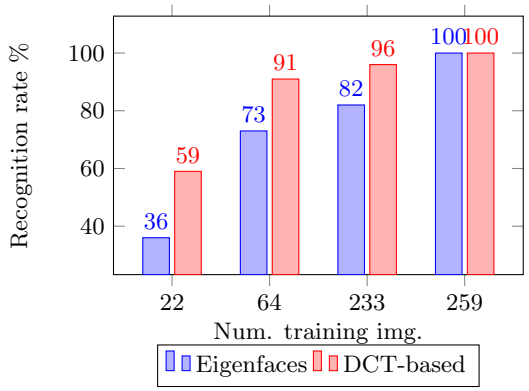

Figure 4: Results with Caltech database

compromise between security and usability. The immediate next step is to evaluate these (and potentially other) recognition algorithms on larger databases (e.g. [12, 7, 13]). We are currently in the process of creating a more extensive face and speech recognition database with both high-quality reference images and recordings and more realistic instances taken with off-the-shelf mobile phones at University of Applied Scienced Upper Austria and will make this database publicly accessible once it is complete (see Fig. 2 for an example, comparing standard face snapshots with ones taken by the subject himself using an off-the-shelf mobile phone camera).

The second step in our future research will aim at improving both security and usability of user authentication by directly combining the two biometric methods. First results towards fusing speaker and face recognition are promising but have not yet been evaluated quantitatively and are therefore subject to future work.

Our implementations of speaker and face recognition on Android and their combination in the form of a plugin-based authentication service for Android applications will be made available under an open source license at `http://openuat.org`.

5. REFERENCES

[1] A. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357 – 366, Aug. 1980.

[5] H. K. Ekene. *A Robust Face Recognition Algorithm for Real-World Applications*. PhD thesis, Universität Karlsruhe, 2009.

[6] Z. Fang, Z. Guoliang, and S. Zhanjiang. Comparison of different implementations of MFCC. *J. Comput. Sci. Technol.*, 16:582–589, November 2001.

[7] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):643–660, 2001.

[8] T. Heseltine, N. Pears, J. Austin, and Z. Chen. Face recognition: A comparison of appearance-based approaches. In *Proc. VIIth Digital Image Computing: Techniques and Applications*, volume 1, pages 59–68. Citeseer, 2003.

[9] ille Hautamäki, T. Kinnunen, I. Kärkkäinen, J. Saastamoinen, M. Tuononen, and P. Fränti. Maximum a posteriori adaptation of the centroid model for speaker verification. *IEEE Signal Process. Lett*, pages 162–165, 2008.

[10] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Commun.*, 52:12–40, January 2010.

[11] H. Lu, A. J. B. Brush, B. Priyantha, A. K. Karlson, and J. Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In K. Lyons, J. Hightower, and E. M. Huang, editors, *Proc. Pervasive*, volume 6696 of *Lecture Notes in Computer Science*, pages 188–205. Springer, 2011.

[12] A. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[13] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 947–954, Washington, DC, USA, 2005. IEEE Computer Society.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[15] 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*, pages 210–227, 2008.

[16] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.