CS4089 Project

Report

Security Implementation using Biometric

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

B130253CS Shrimadhav U K

Under the guidance of
Dr. Vinod Pathari

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India – 673 601

Winter Semester 2017

Declaration

I, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Place: NIT Calicut
Date: May 1, 2017
Name: Shrimadhav U K
Reg.No: B130253CS

Signature:

Department of Computer Science and Engineering

National Institute of Technology Calicut

Certificate

This is to certify that this is a bonafide record of the project presented by the students whose names are given below during Monsoon Semester 2016 in partial fulfilment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

B130253CS Shrimadhav U K

Dr. Vinod Pathari
(Project Guide)

Ms. Anu Mary Chacko
(Course Coordinator)

Acknowledgments

¡Acknowledgements here¿

Shrimadhav U K

May 2017
National Institute of Technology Calicut

Abstract

The project aims to develop a biometric security system, which can protect the user's device(s) from unauthorized or unauthenticated access. The idea is inspired from Microsoft Windows Hello and Google Now, which allows us to speak our mind and the machine does it, through the profound advancement in machine learning and artificial intelligence. I plan to implement an Android application which can recognize the face and the voice of the user, and accordingly allow or deny access to the system.

Contents

Chapter 1

Introduction

Biometric Security is gaining more and more attention recently. This project attempts to implement an application which can take the voice input from a microphone, face input from a camera, and verify the authenticity of the user accessing the system.

1.1 Motivation

Human beings have reached a stage where it is no longer convenient to type the password when they want to be authenticated. This was the basic motivation of this project, i.e., to replace the password input using a keyboard, and instead ask the user to smile in front of their personal computer, and talk interactively to it. Then that personal computer unlocks, if it recognizes the integrity of the user.
Currently, no fool-proof solution exists which attempts to do both these tasks. There exists individual solutions for each of these individual tasks. But, these solutions are proprietary and requires specific licenses to use the offered services.

Chapter 2

Problem Statement

To design a security system for GNU/Linux operating system using biometric of the user, i.e., the face and the voice of the user, that would replace the traditional password input using a keyboard.

2.1 Related Works

1. Google Now `https://www.google.com/search/about/learn-more/now/`

2. Microsoft Windows Hello `https://support.microsoft.com/en-in/help/17215/windows-10-what-is-hello`

Chapter 3

Literature Survey

3.1 Face Recognition

3.1.1 Background and Related Work

Much of the work in computer recognition of faces have been approached by characterizing a face by a set of geometric parameters and performing pattern recognition based on the parameters.
Kanade's face identification system [?] was the first system in which all steps of the recognition process were automated, using a top-down control strategy directed by a generic model of expected feature characteristics. His system calculated a set of facial parameters from a single face image and used a pattern classification technique to match the face from a known set. This approach was a statistical based approach, which depended primarily on local histogram analysis and absolute gray-scale values.

3.1.2 The EigenFace Approach

Much of the previous work on automated face recognition has ignored the issue of just what aspects of the face stimulus are important for identification. This suggested that an information theory approach of encoding and decoding face images may give insight into the information content of face images, emphasizing the significant local and global features. Such features may or may not be directly related to our intuitive notion of face features such as the eyes, nose, lips, and hair.
In the language of information theory, the relevant information in a face image should be extracted, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly.

A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.
In mathematical terms, we wish to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as a vector in a very high dimensional space. The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images.
These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that the eigenvector is displayed as a sort of ghostly face, which is called an eigenface (see Figure 3.1).
The approach to face recognition using the eigenface approach involves the following initiation operations:

1. Acquire an initial set of characteristic face images (the training set). This set should include a number of images for each person, with some variation in expression and in the lighting.
   (say **4** images of **10** people, so $\mathbf{M} = \mathbf{40}$.)

2. Calculate the eigenfaces from the training set, keeping only the M images that correspond to the highest eigenvalues.
   These M images define the *face space*. As new faces are experienced, the eigenfaces can be updated or re-calculated.

   (a) Calculate the (M x M) matrix L, find it's eigenvectors and eigenvalues, and choose the M' eigenvectors with the highest associated eigenvalues.
   (Let $M' = 10$ in this example.)

   (b) Combine the normalized training set of images (according to equation (3.1)) to produce the (say, $M' = 10$) eigenfaces u_k.

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad , l = 1, 2, \ldots, M \tag{3.1}$$

   (c) For each known individual, calculate the class vector Ω_k by averaging the eigenface pattern vectors Ω (from equation (3.2)) calculated from the original images (four, in this example) of the individual.

$$\epsilon_k^2 = ||(\Omega - \Omega_k)||^2 \tag{3.2}$$

Choose a threshold θ_ϵ that defines the maximum allowable distance from any face class, and a threshold θ_ϵ that defines the maximum allowable distance from face space. (according to equation (3.3))

$$\epsilon^2 = ||(\Phi - \Phi_f)||^2 \tag{3.3}$$

(d) For each new face image to be identified, calculate it's pattern vector Ω, the distance ϵ_i to each known class, and the distance ϵ to face space.
If the minimum distance $\epsilon_k < \Theta_\epsilon$ and the distance $\epsilon < \Theta_\epsilon$, classify the input face as the individual associated with the class vector Ω_k.
If the minimum distance $\epsilon_k > \Theta_\epsilon$ but distance $\epsilon < \Theta_\epsilon$, Then the image may be classified as unknown, and optionally used to begin a new face class.

3. If the new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated (steps 1-2). This gives the opportunity to modify the face space as the system encounters more instances of known faces.

4. Calculate the corresponding in M-dimensional weight space for each known individual, by projecting the face images onto the face space.

The above initialization operations can be performed from time to time whenever there is free excess computational capacity, available in the system. Having initialized the system, the following steps are then used to recognize new face images:

1. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.

2. Determine if the image is a face (whether known or unknown) by checking to see if the image is sufficiently close to face space.

3. If it is a face, classify the weight pattern as either a known person or as unknown.

4. Update the eigenfaces and/or weight pattern.

Figure 3.1: Mean Eigen Face

5. If the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into the known faces.

In the prototype implemented using the eigenfaces approach, calculation of the eigenfaces is done as part of the training process.
The recognition, using the eigenfaces approach, takes about 90 seconds implemented in Python on an Intel Core i5, using face images of size 132 x 132.

3.1.3 The Local Binary Pattern Histogram Approach

It is observed that if the above program is run without any alterations, trained with a specific person, and another untrained face is introduced, then it will be recognized as the trained person.
The methods to improve the accuracy of the Face Recognizer have been made more stringent. The threshold used to control unknown faces, in the case of the EigenFaceRecognizer, from the calculated distance can be adjusted to allow better accuracy. A default of 2000 is used but by increasing this to 5000, for example, will mean it will be less likely to allow a false match.

3.1.4 Principle Component Analysis

The EigenFaceRecognizer class applies PCA on each image, the results of which will be an array of Eigen values that a neural network can be trained to recognize.
The LBPHFaceRecognizer uses Local Binary Patterns (LBP) to create a feature vector using a Support Vector Machine or some other machine learning algorithm.

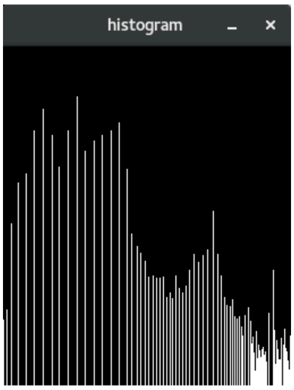

Figure 3.2: Local Binary Pattern

3.1.5 The Local Binary Pattern Histogram Classifier

The LBPH recognizer takes five variables:

radius The radius used for building the Circular Local Binary Pattern.

neighbors The number of sample points to build a Circular Local Binary Pattern from. A value suggested by OpenCV documentation is eight sample points. The more the number of sample points, higher will be the computational cost.

grid_x The number of cells in the horizontal direction. Eight is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.

grid_y The number of cells in the vertical direction. Eight is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.

threshold The threshold applied in the prediction. If the distance to the nearest neighbor is larger than the threshold, the method returns **-1**.

In the prototype implemented using the LBPH Approach (see Figure (3.2)), calculation of the LBP is done as part of the training process.
The recognition, using the LBPH approach, takes about 30 milliseconds implemented in Python on an Intel Core i5, using face images of size 132 x 132.

3.2 Voice Recognition

3.2.1 Background and Related Work

Speaker recognition is the identification of the person who is speaking by characteristics of their voices (voice biometrics), also called voice recognition.
Speech is a kind of complicated signal produced as a result of several transformations occuring at different levels: semantic, linguistic and acoustic. Differences in these transformations may lead to differences in the acoustic properties of signals. The recognizability of speaker can be affected not only by the linguistic message but also the age, health, emotional state and effort level of the speaker.
Background noise and performance of recording device also interfere the classification process.
Speaker recognition is an important part of Human-Computer Interaction (HCI). As the trend of employing wearable computer reveals, Voice User Interface (VUI) has been a vital part of such computer. As these devicesare particularly small, they are more likely to lose and be stolen. In these scenarios, speaker recognition is not only a good HCI, but also a combination of seamless interaction with computer and security guard when the device is lost. The need of personal identity validation will become more acute in the future. Telephone banking and Telephone reservation services will develop rapidly when secure means of authentication are available.

3.2.2 Algorithm Used

1. An utterance of a user is collected during the enrollment procedure.

2. Voice Activity Detection is performed: Signals must be first filtered to rule out the silence part, otherwise the training might be seriously biased.

3. Feature Extraction

    - Mel-Frequency Cepstral Coefficient (MFCC) is a representation of the short term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non-linear mel-scale of frequency . MFCC is the most widely used features in Automatic Speech Recognition, and it can also be applied to speaker

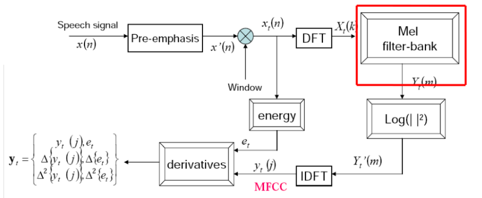

Figure 3.3: MFCC Feature Extraction Process

recognition task. The process to extract MFCC feature is demonstrated in Figure 3.3.

- Linear Predictive Coding (LPC) is a tool used in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a Linear predictive Model.
  The basic assumption in LPC is that, the n th signal is a linear combination of the previous p signals. Therefore, to estimate the coefficients ai, we have to minimize the squared error. This optimization can be done by Levinson-Durbin algorithm.

4. Gaussian Mixture Model (GMM) is used in acoustic learning task such as speaker recognition, since it describes the varied distribution of all the feature vectors. Therefore, GMM is merely a weighted combination of multivariate Gaussian distribution which assumes feature vectors are independent. We use diagonal covariance since the dimensions of the feature vector is independent to each other. GMM can describe the distribution of feature vector with several clusters, as shown in Figure 3.4.

   After training, the model can give the score of fitness for every input feature vector, measuring the probability that the vector belongs to this model. Therefore in the task of speaker recognition, we can train a GMM for every speaker. Then for a input signal, we extract lists of feature vectors for it, and calculate the overall likelihood that the vector belongs to each model. The speaker whose model fits the input best will be chosen as the answer.
   Moreover, an enhancement has been done to the original GMM method. The training of GMM first requires a random initialization of the means of all the components. However, we can first use K-means algorithm to

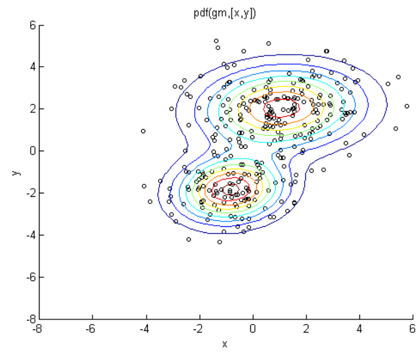

Figure 3.4: A Two Dimensional GMM with Two Components

perform a clustering to all the vectors, then use the clustered centers to initialize the training of GMM. This enhancement can speed up the training and also give a better training result.

5. Joint Factor Analysis (JFA) is a typical method which behave very well in classification problems, due to its ability to account for different types of variability in training data. Within all the factor analysis methods, JFA was proved to outperform other methods in the task of speaker recognition.
   JFA models the user by supervector, i.e., a C x F dimension vector, where C is the number of components in the Universal Background Model, trained by GMM on all the training data, and F is the dimension of the acoustic feature vector. The supervector of an utterance is obtained by concatenating all the C means vectors in the trained GMM model.

Chapter 4

Design

1. Design a function which takes the user voice through the microphone, and the name of the user and returns True or False, accordingly.

2. Design a function which takes an image of the user, using the camera, and the name of the user and returns True or False, accordingly.

3. Finally, design a system which unifies the functions designed above. The system should be able:

   - to override the default login screen in a GNU/Linux system.
   - to ensure the integrity of the confidential details created using the above functions.

Chapter 5

Implementation, Result, and Analysis

5.1 The Biometric System

5.1.1 Advantages of Face Recognition

1. Face recognition systems are the least intrusive from a biometric sampling point of view because they neither require contact nor the awareness of the subject.

2. The biometric works with legacy photograph databases, video tape and other image sources.

3. It is a fairly good biometric identifier for small scale verification application.

5.1.2 Disadvantages of Face Recognition

1. A face needs to be well-lit by controled light sources in automated face authentication systems.

2. Face is a poor biometric for use in a pure identification protocol, it performs better in verification.

5.1.3 Advantages of Voice Recognition

1. Voice is natural biometric (one that people use instinctively to identify each other) under certain circumstances and machine decisions can be verified by relatively unskilled operators.

2. The voice biometric requires only inexpensive hardware and is easily deployable over existing, ubiquitous communications infrastructure. Voice is therefor very suitable for pervasive security management.

3. Voice allows incremental authentication protocols. For example, the protocol prescribes waiting for more voice data when a higher degree of recognition confidence is needed.

5.1.4 Disadvantages of Voice Recognition

1. Speech characterstics can drift away from models with age.

2. With the improvement of text-to-speech technology improving, it becomes possible to create non-existent identities with machine voices and trainable speech synthesis may make it possible to create an automatic system that can imitate a given saying anything.

3. Voice recognition is dependent on the quality of the captured audio signal. Speaker identification systems are susceptible to background noise, channel noise, and unknown channel or microphone characterstics.

Chapter 6

Conclusion

Face recognition uses low-power infrared illumination to obtain robust images under poor lighting conditions, its systems are the least intrusive from a biometric sampling point of view and it is a fairly good biometric identifier for small-scale verification applications.
Speaker recognition is attractive because of its prevalence in human day-to-day communication and conversational biometrics provides higher accuracy and flexibility.
Hence, a combination of the above two biometric identification system is suitable for a good security implementation. However, several aspects remain to be researched and extended. For instance, since face to face meeting encompasses several modalities, such as speech and gesture, these capabilities need to researched and implemented.

References

[1] Microsoft Windows Hello, https://support.microsoft.com/en-in/help/17215/windows-10-what-is-hello

[2] UCSD Computer Vision, http://vision.ucsd.edu/content/yale-face-database

[3] Google Now, https://www.google.com/search/about/learn-more/now/

[4] The CMU Audio Databases, http://www.speech.cs.cmu.edu/databases/

[5] P. Jonathon Phillips, A. Martin, C.l. Wilson, M. Przybocki, An Introduction to Evaluating Biometric Systems, http://www.face-rec.org/databases/151436.pdf