future internet

MDPI

Article

Increasing Trustworthiness of Face Authentication in Mobile Devices by Modeling Gesture Behavior and Location Using Neural Networks

Blerim Rexha [1], Gresa Shala [2,*] and Valon Xhafa [3]

[1] Faculty of Electrical and Computer Engineering, University of Prishtina, Kodra e Diellit p.n., 10000 Prishtina, Kosovo; blerim.rexha@uni-pr.edu
[2] Department of Computer Science, Freiburg University, Georges-Köhler Alley 101, 79110 Freiburg im Breisgau, Germany
[3] Department of Informatics, Technical University of Munich, Boltzmannstraße 3, 85748 Garching bei München, Germany; valon.xhafa@tum.de
* Correspondence: gresa.shala@merkur.uni-freiburg.de

Received: 18 January 2018; Accepted: 2 February 2018; Published: 5 February 2018

Abstract: Personal mobile devices currently have access to a significant portion of their user's private sensitive data and are increasingly used for processing mobile payments. Consequently, securing access to these mobile devices is a requirement for securing access to the sensitive data and potentially costly services. Face authentication is one of the promising biometrics-based user authentication mechanisms that has been widely available in this era of mobile computing. With a built-in camera capability on smartphones, tablets, and laptops, face authentication provides an attractive alternative of legacy passwords for its memory-less authentication process, which is so sophisticated that it can unlock the device faster than a fingerprint. Nevertheless, face authentication in the context of smartphones has proven to be vulnerable to attacks. In most current implementations, a sufficiently high-resolution face image displayed on another mobile device will be enough to circumvent security measures and bypass the authentication process. In order to prevent such bypass attacks, gesture recognition together with location is proposed to be additionally modeled. Gestures provide a faster and more convenient method of authentication compared to a complex password. The focus of this paper is to build a secure authentication system with face, location and gesture recognition as components. User gestures and location data are a sequence of time series; therefore, in this paper we propose to use unsupervised learning in the long short-term memory recurrent neural network to actively learn to recognize, group and discriminate user gestures and location. Moreover, a clustering-based technique is also implemented for recognizing gestures and location.

Keywords: authentication; face; smartphones; gestures; location; LSTM; neural network

1. Introduction

The Internet's spread as a computer network, driven by its characteristics including its ability to include numerous devices and geographic areas, has changed the way information is spread. With the increase in functionality of mobile devices, we are transitioning from an Internet society towards a mobile one. Google's report "How People Use Their Devices" [1] shows that smartphone usage during the day dominates over computer usage, as seen in the graph in Figure 1. Based on the devices on which searches are made during the day, Google also found out that although the usage of computers becomes dominant from around 8 a.m. when people might start their workday, mobile usage takes the lead again in the late afternoon, and continues to increase in the evening, as presented in Figure 2.

However, besides offering a huge amount of information and other numerous functionalities, mobile devices also contain sensitive user data for personalization of user experience in applications or to offer more efficient services. Controlling access in computer devices through user authentication is one of the means of securing the data that these devices contain.

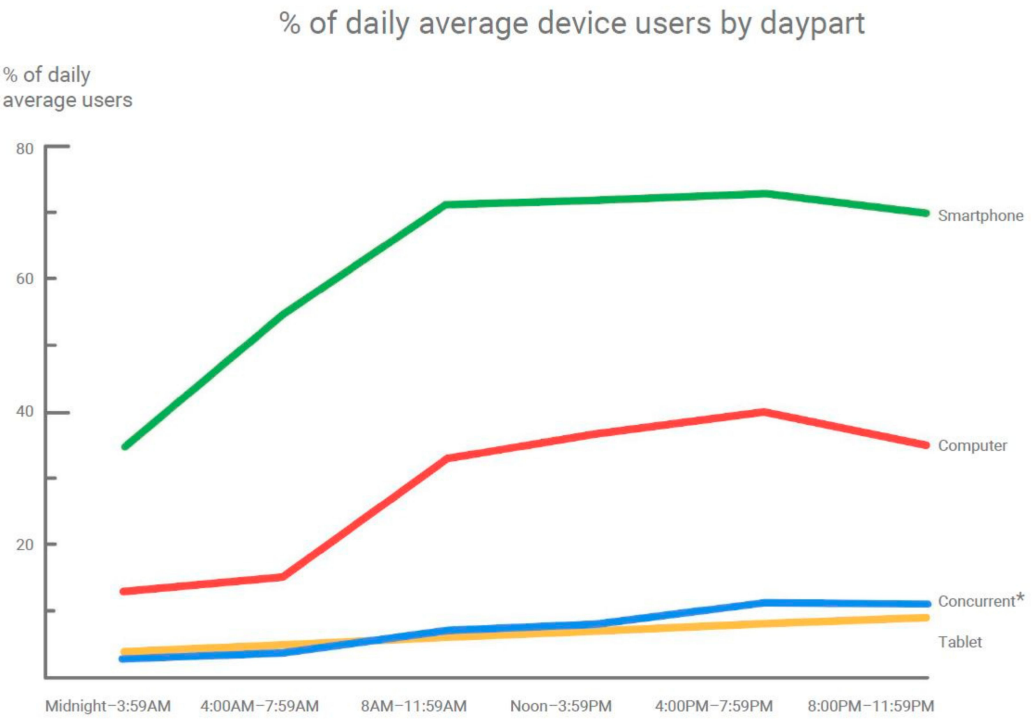

Figure 1. Daily device usage [1].

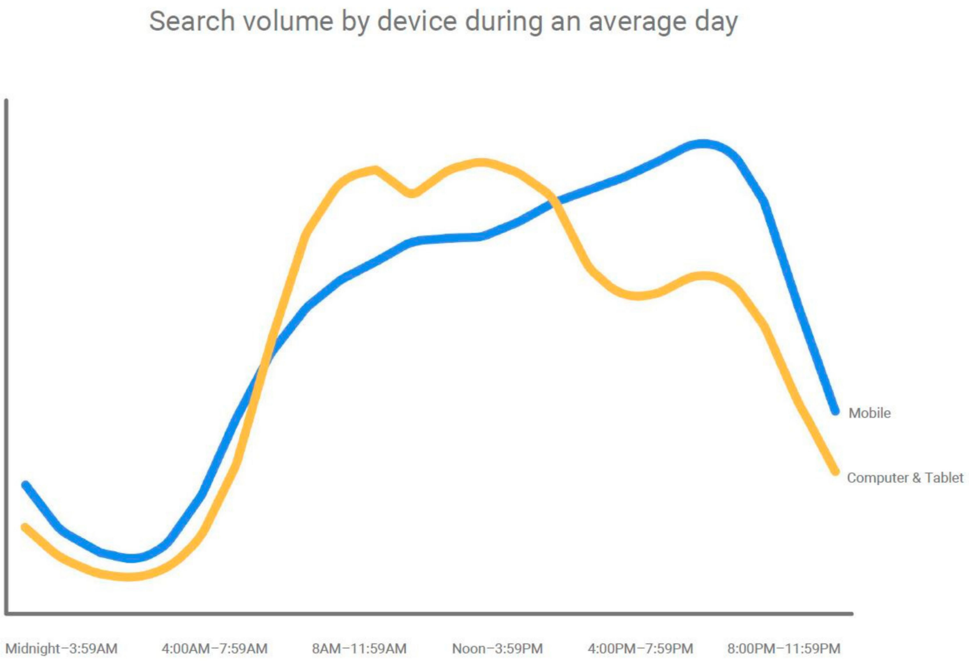

Figure 2. Search volume by device during an average day [1].

Secure authentication for smart phones is becoming important for many applications such as financial transactions. So far, Personal Identification Number (PIN) and password authentication are the most commonly used methods for smartphone access control. Specifically, for a PIN and limited length password, the level of security is low and thus can be compromised easily. The techniques of using the smartphone's front camera for face recognition is suggested. Shen et al. suggest an optimized Sparse Representation Classification for face authentication [2]. Sparse Representation Classification (SRC) is a state-of-the-art face-recognition algorithm, which has been shown to outperform many classical face-recognition algorithms. The success of SRC is due to its use of 1 optimization, which makes SRC robust to noise and occlusions [2]. Because the former technique requires another step of authentication, Kang et al. proposes a two-factor face authentication scheme using matrix transformations and a user password [3]. This scheme is designed with a secure cancellation feature, in that templates composed of permutation and feature vectors can be freely changed [3].

Numerous other authentication methods for smartphones have been explored [4]. Advancements in data analysis and machine learning have made it possible to explore other methods of achieving user authentication, such as authentication based on biometric signs. Biometric signs represent measurable biological characteristics of a human being. On the other hand, a biometric process represents the automated method of recognizing an individual based on the identification of biometric signs [5].

The human face plays an important role in daily social interaction. Face recognition is a natural process for humans. Even newborn babies have the ability to differentiate between known faces [6]. How hard could it be for computer devices? This question sets the foundation of the field of automatic face recognition, one of the most researched topics in computer vision and pattern recognition. In addition to applications related to identification and verification, such as access control and surveillance, face recognition has been useful in human-computer interaction, virtual reality and computer entertainment [7]. A study from Juniper Research [8] found that there would be a rapid growth in biometric authentication usage, with voice and facial recognition leading the way. The study predicts that biometric authentication usage would grow from 190 million mobile devices in 2016 to 600 million by 2021. Because of the simpler hardware implementation for voice and face recognition, it is predicted to overcome the usage of fingerprints for authentication. Furthermore, authentication by the use of multiple biometrics is predicted. Compared to other biometric systems based on fingerprint or iris recognition, face recognition has the advantage of being a non-intrusive process. It is a visual pattern recognition problem, where the face, being a three-dimensional object which is subject to lighting variations, pose variation and other factors, should be identified based on acquired images. Advancements in photographic devices and their presence in mobile devices have allowed further exploration of this problem. The aforementioned techniques introduced face recognition as a way of smartphone authentication; nevertheless, these techniques were proven vulnerable when used in real life scenarios. One of the vulnerabilities can be that in the most current implementations, a sufficiently high-resolution face image displayed on another mobile device will be enough to circumvent security measures and bypass the authentication. "Open With Your Face" [9] is the technology provided by Samsung on the next generation of Galaxy to unlock the phone using face recognition. The user raises the phone as if taking a selfie to use face recognition. This technique is a fast and effortless way to unlock the Galaxy 8 or S8+. However, using only frontal face images for authentication purposes can no longer be considered secure under the assumption of the easy availability of frontal snapshots of the respective device owners from social networks or other media. Back in 2011, Google added a "Face Unlock" system to Android 4.0 [10], and it had the same picture vulnerability that Samsung's solution has today. In Android 4.1, Google's Face Unlock added a "liveness check" that attempted to defeat the photo vulnerability by requiring the user to blink. This too was bypassed (rather hilariously) by grabbing a photo of someone, poorly Photoshopping a second copy of the picture with a set of closed "eyelids", and then switching between the "eyes open" and "eyes closed" pictures when the face unlock asks the user to blink. Because of the picture vulnerability, different other techniques for smartphone authentication requiring less user effort have been proposed. Gestures provide a faster and more convenient method of authentication compared to a complex password. Hong et al. [11] proposes Waving Authentication (WA), which is a motion gesture authentication system based on the accelerometer. WA utilizes eight distinguishing features hiding in the acceleration traces of motion gestures and exploits one-class Support Vector Machine for classification. It is insusceptible to shoulder-surfing attacks. Because gestures should be easy to remember, it is possible for users to record one gesture for every unique password. However, remembering which password corresponds to which gesture could get complicated as the number of unique passwords increases. Another similar suggestion is presented by Shrestha et al. [12]. They present the Wave-to-Access permission-granting approach that can be used to protect any sensitive mobile device service. It captures the user's intent to access the service via a lightweight hand waving gesture. This gesture is very simple, quick and intuitive for the user, but would be very hard for the attacker to exhibit without the user's knowledge. They also present the design and implementation of a hand-waving gesture recognition mechanism using an ambient light sensor, already available on most mobile devices. The results of Arif et al. [13] showed that the gesture techniques where more error prone than the digit lock technique.

In this paper, we propose three-factor authentication using face, gesture and location recognition. The focus is in building the gesture-recognition algorithm to learn actively from user gestures. User gestures are a sequence of time series; therefore, we will use unsupervised learning in the LSTM (Long Short-Term Memory) recurrent neural network to actively learn to recognize, group and discriminate user gestures. Using unsupervised learning in LSTM was shown by Klapper et al. [14]. They trained LSTM recurrent neural networks to maximize two-information objectives for unsupervised learning: binary information gain optimization and nonparametric entropy optimization. They also proved that LSTM is capable of learning to discriminate different types of sequences and group them to a variety of features. Using a user's location, frequent or trusted, can increase the confidence of two previous parameters—face and gesture—during the authentication process.

2. Improved Face Authentication System

In this paper, we suggest a face-authentication system consisting of face, gesture and location recognition as components. The proposed system is shown in Figure 3. For each component, necessary data for training and validation is collected and stored on the corresponding dataset. Training data from each dataset is used to train the corresponding model. After the face, gesture and location recognition model is trained and validated, they can be used for making corresponding predictions. At the end, three prediction scores corresponding to the three components yield the final authentication score.

The face-recognition component deals with image data of the user's face. This data is derived from the mobile device's front camera. Before performing face recognition, face detection is done to locate the face in the image data coming from the camera. Due to the accuracy and low time cost of detecting the face and extracting just the right amount of information from it, Google Face API was chosen for face detection. Having detected the face, recognition is then performed. The local binary patterns histograms (LBPH) algorithm is used for face recognition. Its performance is improved by taking into account location data of the device and improving the saved model of the user 's face in order to adjust to changes that can occur to the face due to aging, hair growth etc. By keeping track of the visited locations and the frequency of these visits, the recognition requirements are changed. Thus, for more frequently visited locations, such as home or work, the confidence requirements of the result of recognition by LBPH are more relaxed. For unvisited locations, or those less frequently visited, there are more strict requirements for authentication. Consequently, the lower the frequency of visits to a location, the higher the recognition result of LBPH needs to be in order to yield a positive match from the face-recognition component. This is done with the purpose of avoiding false negatives that can occur due to the influence of environmental factors in the image obtained from the mobile device's camera. Thus, by lowering the requirements on the recognition values of LBPH for frequently visited locations, authentication is possible in cases where, due to environmental factors, the standard LBPH algorithm would yield a low recognition score of the user's face. Another improvement done to the standard LBPH algorithm is the updating of the face model to adapt to changes to the user's face. The face is a biometric sign that is subject to change due to aging, beard growth, etc. Since the LBPH algorithm's face model consists of concatenated histograms of local binary patterns of all the images used for training, user images can be concatenated to this model, thus updating it. Considering this fact, the LBPH face model is updated dynamically even after it is constructed in order to adjust to changes occurring to the user's face. Thus, the chances for the face module to yield false negative results are significantly lowered. In the training phase, the images are utilized to construct a face model to be used for recognition. Once the face model is existent, input face images are used for recognition. Thus, the face images are compared to the face model. If the result of this comparison is as high as required by the authentication system based on the location of the device, the face module yields a positive match. User gestures are a sequence of time series; therefore, unsupervised learning in the LSTM recurrent neural network to actively learn to recognize, group and discriminate user gestures is suggested. By adding gesture recognition as a module in the overall authentication system, the reliability of the results derived from this system is further increased. Thus, the authentication system's final recognition scores depend on three factors: (i) user's face; (ii) gestures; and (iii) location.

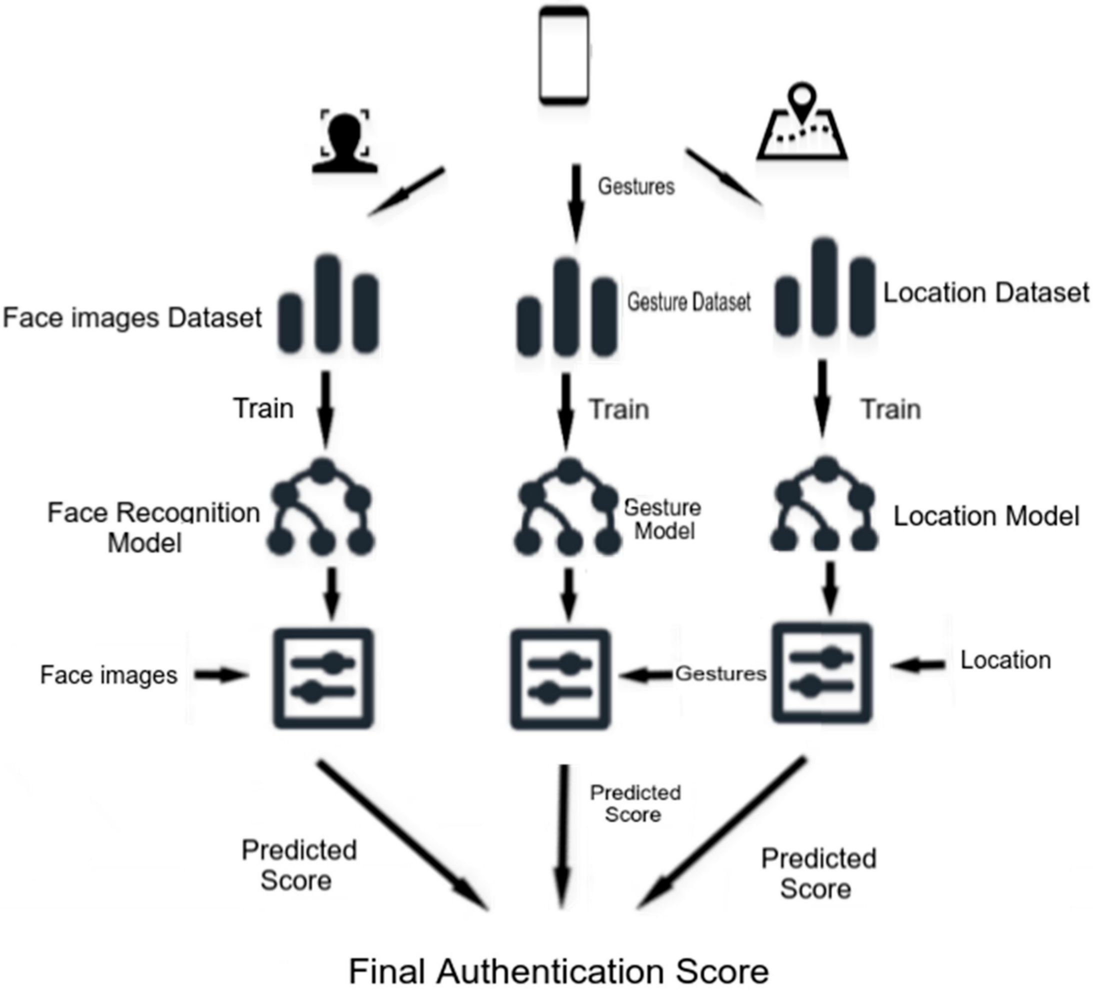

Figure 3. Authentication system schema.

3. Face Recognition System

A general face-recognition system, as presented Figure 4, consists of three main modules [15]:

- Face detection separates the area of the face from the image background. Face detection's purpose is to focus in a smaller image area for analysis, rather than analyzing the whole image.
- Face feature extraction is used to extract such information from the face image, which helps differentiate between faces of different individuals and is robust to photometric and geometric changes. The extracted face features are used to verify the identity of the face.
- Face recognition occurs when the extracted face features are compared to the saved data of one or more faces in the database. This module's output is "yes" or "no" for 1:1 verification; otherwise, when doing a 1:N identification, the output is the face's identity.

The purpose of this work is the development of an application that does authentication based on face recognition in Android mobile devices. Face recognition is not a problem belonging only to the field of computer vision. It is a relevant topic in pattern recognition, neural networks, computer graphics, image processing and psychology.

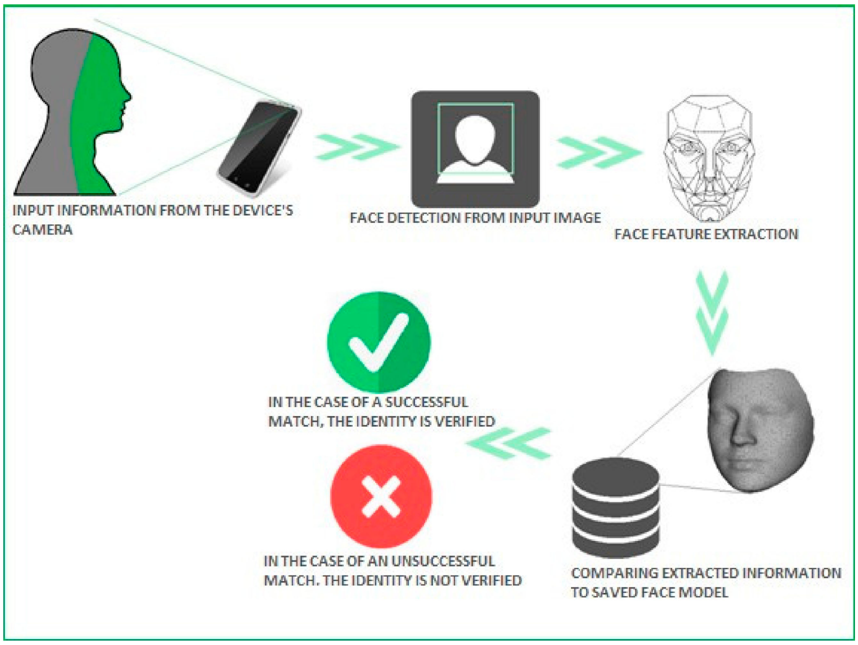
INPUT INFORMATION FROM THE DEVICE'S CAMERA
FACE DETECTION FROM INPUT IMAGE
FACE FEATURE EXTRACTION
IN THE CASE OF A SUCCESSFUL MATCH, THE IDENTITY IS VERIFIED
IN THE CASE OF AN UNSUCCESSFUL MATCH. THE IDENTITY IS NOT VERIFIED
COMPARING EXTRACTED INFORMATION TO SAVED FACE MODEL

Figure 4. Face-recognition system schema.

The three main problems that face recognition aims to solve are [15]:

- Verification (authentication): Am I who I pretend to be? The face recognition system decides whether the face in the acquired image belongs to the identity the person pretends to have.
- Identification (recognition): Who am I? The face recognition system matches the face from the acquired image with the face models saved in the database.
- Watch list: Are they looking for me? The acquired image is compared to the face models of people in the watch list saved in the database. If the face in the image belongs to a person in the watch list, that individual is identified.

3.1. Face Detection

Nowadays not all face-recognition applications need a face-detection module. In some cases, the input face images are already normalized. Therefore, there is no need for face detection. An example of such a face-recognition system is the criminal database. Law enforcement agencies save the models of the faces of individuals with criminal reports in the database. In order to uncover the identity and criminal records of an individual, all that is needed as input is the person's passport photo, which is compared to all the face models in the database. However, input images for a general face recognition system are not suitable for analysis, since they are likely to contain other objects besides faces. This is why there is a need for a face-detection module to be present. There are numerous challenges that face detection must overcome in order to accurately detect faces in an image. These challenges appear due to the uncontrolled environment where images are taken. These challenges can be attributed to factors such as:

- Pose variation: The ideal situation for face detection would be in cases where all the input images contain faces in frontal pose. However, this is hardly realizable in practice. Furthermore, pose variation affects face detection algorithms' performance.
- Obscuring face features: The presence of elements such as glasses, a hat or a beard is another challenge for face detection.
- Face gestures: Face features change by a considerable amount as a result of different face gestures.
- Factors of the environment where the image is acquired: Different cameras and different environmental conditions can greatly affect the image quality, thus also affecting the visibility of the face in the image.

3.2. Face Recognition Algorithms

3.2.1. Geometric Approach

The approach on face recognition based on geometric features of the face extracts features such as face landmarks' positions and their distances from one another. The feature vector consists of these positions of face features and their distances. Thus, each face is represented by such a feature vector. The decision of whether two faces match or not is made based on whether the feature vectors representing the two faces are more similar than a given similarity threshold. The main advantage of the geometric approach for face recognition is the speed of the comparison, taking into account the compactness of the representation. A main disadvantage of this approach is the difficulty of accurate automated detection of face landmarks. Whoever implements face recognition based on geometric properties of the face must make arbitrary decisions regarding the selection of face landmarks and face dimensions to be taken into account. This is important because face models will be built based on this choice, thus the better the selection of geometric face features, the easier it is for the face recognition system to differentiate between faces of different individuals.

3.2.2. Holistic Approach

While algorithms for face recognition based on the geometric approach make use of local features of the face, those based on a holistic approach analyze the face image as a whole. The main advantage of this approach is the spared time in face detection, since there is no need for time-consuming operations of accurately localizing face landmarks such as the mouth, eyes, or the nose. The whole face image is used for analysis, and as such, operations are performed to its pixels in order to create a face model. Depending on the conditions in which the images are taken, sometimes pre-processing is needed, such as histogram equalization in order to lower the influence of environmental factors in recognition results.

3.2.3. Principal Component Analysis (Eigenfaces)

Processing image data is a computationally expensive job. Principal component analysis (PCA) [16] aims to reduce the dimensionality of the image. This is achieved by projecting the images as a vector and only keeping the components that exhibit the biggest variance among the images in the database. Thus, d-dimensional data is projected in a k-dimensional space, where k < d. Since multi-dimensional data can often be described by a smaller number of inter-dependent variables, only a certain number of dimensions are important and contain most of the information in the image. The PCA method finds the dimensions that exhibit the biggest variance, called principal components.

3.2.4. Linear Discriminant Analysis (LDA)

Similarly, to the Eigenfaces algorithm, Fisherfaces [17] is a dimension-reduction technique. However, what makes it different from other holistic approach algorithms is that it considers the containing class information image data. This algorithm aims to project image data in a space with reduced dimensions while also preserving inter-class distance. This differentiation between classes is needed for the purpose of keeping information useful in distinguishing faces in the database from one another. This information would otherwise be very likely to be lost in the process of reduced dimensionality projection based on variance.

3.2.5. Local Binary Patterns Histograms (LBPH)

Local binary patterns histograms (LBPH) [18] differs from the above-mentioned algorithms because of the approach it takes on processing the images. Eigenfaces and Fisherfaces treat the image as a whole, thus being sensitive to the effect of lighting and other environment factors in image variance. As a result, variance in images caused by secondary factors is likely to yield unwanted results from the face-recognition system. LBPH concentrates on extracting local features of the image. The purpose of this is not in representing the image as a multi-dimensional vector, but rather in the description of local characteristics of objects in the image. In order to extract a summary of the local structure of an image, every pixel is compared to its neighboring pixels. First, the image is separated into several square cells or regions within which pixel comparisons are done. As described in Figure 5, when a pixel's value is being calculated, a comparison is made of each neighboring pixel's value to the one of the central pixel. Based on the result of the comparison, binary values are assigned next to each neighbor indicating whether its value is greater than the central pixel (the binary digit 1 is assigned) or not (the binary digit 0 is assigned). The binary value formed from the concatenation of all the binary digits of the surrounding neighbors characterizes the central pixel. Thus, each image pixel is characterized by a binary number. These binary numbers represent features of the region of the image they're calculated in.

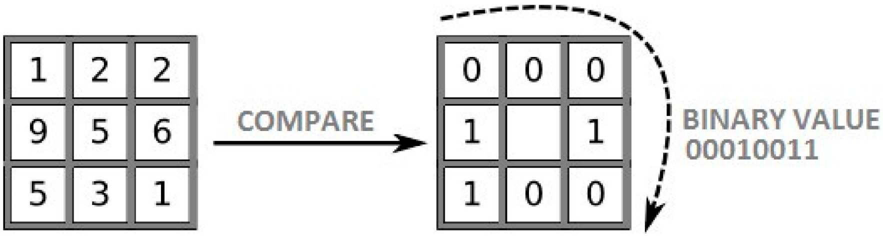

Figure 5. Local Binary Pattern calculation for a 3 × 3 area of 8 neighbors.

The LBP operator has been extended to use neighborhoods of different sizes. This can be done by changing the radius R of the neighborhood and the number of sampling points P used to represent the neighborhood of the central pixel. For a neighborhood of radius R and P neighbors, the texture of the region surrounding the center pixel can be described as:

$$\mathrm{T} \approx (\mathrm{s}(\mathrm{v}_0 - \mathrm{v}_\mathrm{c}), \mathrm{s}(\mathrm{v}_1 - \mathrm{v}_\mathrm{c}), \ldots, \mathrm{s}(\mathrm{v}_{\mathrm{p}-1} - \mathrm{v}_\mathrm{c})) \tag{1}$$

where v_0, v_1, ..., and $\mathrm{v}_{\mathrm{p}-1}$ represent the values of the corresponding neighbors of the central pixel, whereas v_c is the value of the central pixel.

For each sign $s(v_p - v_c)$ a binomial weight is assigned. The sum of the binomial weights represents the LBP value for the central pixel:

$$LBP_{P,R(Xc,Yc)} = \sum_{i-0}^{p-1} s(v_i - v_c)2^i \tag{2}$$

where x_c, and y_c are the coordinates of the central pixel, P is the number of neighboring pixels, R is the radius of the neighborhood, v_i is the value of the i-th neighbor and v_c is the value of the central pixel. After calculation of the local binary pattern for every pixel, using Equation (2), the feature vector of the whole image is constructed. For each square cell the image was separated in, a histogram of LBP values' frequency is built. This means that for each cell, the characteristic histogram contains all the possible LBP values the pixels can have as bins, and each bin contains the frequency of its occurrence in the cell. The feature vector is constructed by the concatenation of the frequency histograms of all the cells the image consists of. The feature vector of LBP characteristics can be used for classification of the images for the purpose of face recognition. The ability of LBPH to describe in detail the texture of an image has made this method attractive for research in the field of face recognition [19].

4. FaceAuth

4.1. Application Description

The purpose of the application is the authentication of the user through face recognition in the Android platform. In order to achieve that, the application works in two modes:

- Training mode: This is the mode in which information regarding the user's face is extracted. The output of the training mode is the built model of the face, which is used for recognition later.
- Recognition mode: The extracted information regarding the face taken from images of the camera device is compared to the saved face model. In the case of a match, the authentication is successful.

4.1.1. Training Mode

Training should be done in order for useful face information to be extracted, thus enabling the recognition model to be built. The face model, or recognition model, is built in order to serve the LBPH algorithm for face recognition. Training is done in two phases:

- Model construction, whereby user images are taken from the camera of the device, and the recognition model for the LBPH algorithm is built.
- Model testing, whereby an image of the user's face is taken from the camera of the device, and the face image is compared to the built recognition model, thus testing it, as presented in Figure 6. In case the result of the comparison is a match with a desirable confidence, we conclude the recognition model has satisfactorily described the face of the user. On the contrary, the training phase is repeated in order to update the model until it is able to describe the user's face to a desirable degree. When a satisfactory model has been built, it is saved. This model is used for comparison in order to achieve authentication in recognition mode.

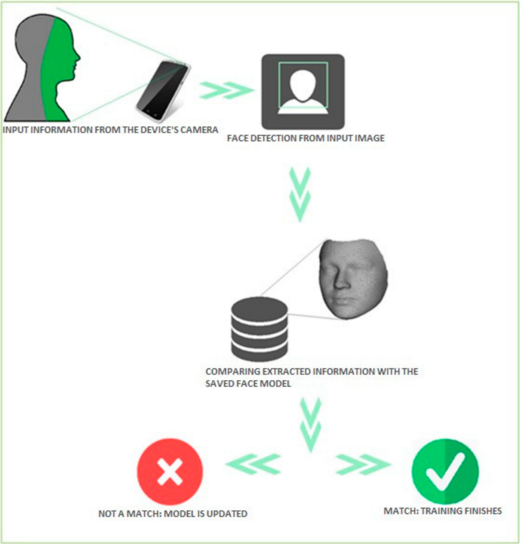
INPUT INFORMATION FROM THE DEVICE'S CAMERA
FACE DETECTION FROM INPUT IMAGE
COMPARING EXTRACTED INFORMATION WITH THE SAVED FACE MODEL
NOT A MATCH: MODEL IS UPDATED
MATCH: TRAINING FINISHES

Figure 6. Model testing phase.

4.1.2. Recognition Mode

User face recognition consists of a comparison between the extracted characteristics of the face from images taken with the camera with the saved recognition model, as presented in Figure 7. Depending on the result of the comparison, the recognition mode either results in successful user authentication or not. Recognition is achieved in two phases:

- Comparing the user's face to the recognition model: Information regarding the user's face is extracted from images taken with the device's camera and is compared against the recognition model by using the LBPH algorithm.
- Verification of face movement by tracing head movement: This includes head movement detection, thus aiming to avoid authentication by using 2D photos of the user, as presented in Figure 7.

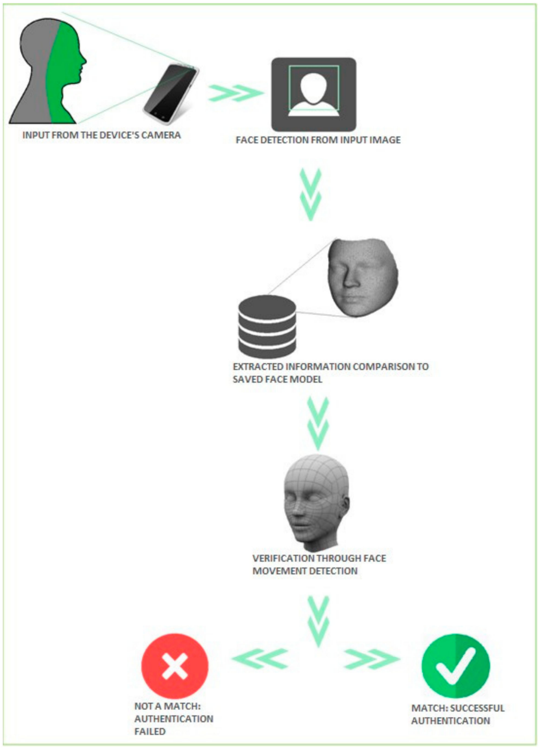
INPUT FROM THE DEVICE'S CAMERA
FACE DETECTION FROM INPUT IMAGE
EXTRACTED INFORMATION COMPARISON TO SAVED FACE MODEL
VERIFICATION THROUGH FACE MOVEMENT DETECTION
NOT A MATCH: AUTHENTICATION FAILED
MATCH: SUCCESSFUL AUTHENTICATION

Figure 7. Face-recognition mode.

4.2. Face Detection

The images FaceAuth gets from the camera of the mobile device are not normalized, in the sense that they do not only contain faces. These images also contain other objects that, in the case of our face recognition system, represent noise data. Thus, it is necessary to isolate the face from the image and only analyze the face data. For this purpose, the presence of a face-detection module is imperative. Besides detecting the location of the face in the image, FaceAuth requires the face detection module to provide some information about the face:

- The visibility of certain facial landmarks, such as the nose, eyes, mouth and cheeks.
- The angle that the detected person's facing direction forms with the axis perpendicular to the image plane.

This data is necessary to trace the movement of the person's face, in order to verify the presence of a three-dimensional face in the image, instead of a two-dimensional photo. During the implementation of the application, several solutions for providing the functionality of the face detection module were explored:

- OpenCV's Cascade Classifier [20]: A class of OpenCV Library that makes object detection possible. OpenCV is an open-source library that includes more than 2500 both classical and state-of-the-art computer vision algorithms. It is built with the purpose of offering a joint infrastructure for applications that are related to computer vision. OpenCV's algorithms can be used for the detection and recognition of faces, object detection, classifying human behavior in videos etc. OpenCV's Cascade Classifier uses Haar characteristics to detect faces in images. Face detection with a cascade of classifiers based on Haar characteristics [21] is an effective method of detection. It is a machine-learning approach where a cascade function is trained with groups of positive (images that contain faces) and negative (images that do not contain faces) images. Then, this function is used for face detection in other images. Though OpenCV's Cascade Classifier detects faces quickly, it is not able to extract more information regarding the detected face, thus it could not be used as a face-detection module for FaceAuth.
- Dlib's Face Detector [22]: This uses the histogram of oriented gradients (HOG) feature combined with a linear classifier. This type of object detector is fairly general and capable of detecting many types of semi-rigid objects in addition to human faces. Dlib is a modern C++ toolkit containing machine-learning algorithms and tools for creating complex software. Its face detector class can detect and locate 68 face landmarks. This detailed description of the face comes with the cost of time. Furthermore, since it gives a much more thorough description of the face, there is much more data regarding face landmarks than is needed for the function of the application.
- Google Face API [23]: The FaceDetector class localizes the position of 12 face landmarks (eyes, nose, mouth, cheeks and ears). Information is also extracted regarding the face's orientation, which is useful for the application. Due to the accuracy and low time cost of detecting the face and extracting just the right amount of information from it, Google Face API was chosen for the function of the face-detection module of FaceAuth's face-recognition system.

4.3. Face Detection

Describing the face by a feature vectors that consist of the face's geometric characteristics relies too much on the accuracy of the face-detection module. Furthermore, head movement causes face landmarks to be obscured, or their respective distances to change. Thus, representing the face by its geometric characteristics makes it more difficult for the recognition module to achieve authentication, unless in very controlled conditions.

Analysis of the Face-Recognition Algorithms

Holistic approach algorithms present a much better solution for the face-recognition module. Taking into account the fact that the nature of the recognition problem that FaceAuth aims to solve is face authentication, which represents a one-on-one comparison, the class notion of the Fisherfaces algorithm is ineffective. Considering that the training set consists of only one face, LBPH's property of individually comparing the local texture characteristics of the acquired face image to each of the face images in the training set has proven more useful in face authentication. On the other hand, the small size of the training image set proves to be an obstacle for the Eigenfaces algorithm. As can be concluded from the chart of recognition statistics in Figure 8, the small size of the training set as a consequence of the nature of the problem of face authentication causes Eigenfaces' performance to lack behind LBPH. To compare these two algorithms 100 recognition tests were done, from which half were tests of known faces (user faces that should have been authenticated) and the other half consisting of unknown faces (that should not have been authenticated). These tests concluded that Eigenfaces is very likely to authenticate unknown faces due to the nature of the training set (consisting of only one face). Thus, with the intention of avoiding false positives, LBPH proved to be the more fitting choice for a face recognition algorithm to solve the problem of face authentication.

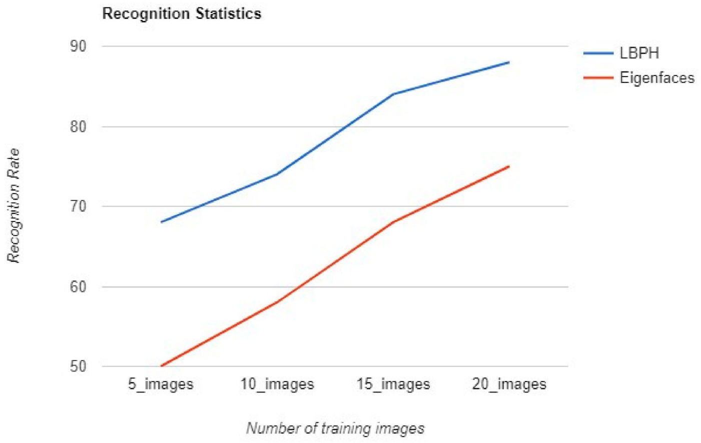

Figure 8. Recognition statistics' chart for Eigenfaces and Local Binary Patterns Histograms (LBPH)

The statistics in the chart of the False Accept Rate (FAR) and False Reject Rate (FRR) analysis of the LBPH algorithm's performance in recognition were obtained using 20 images of the user's face to create the model. As can be concluded from the chart, presented in Figure 9, the Equal Error Rate is at 6 with the threshold's value set at 60, thus achieving a recognition rate of 88%.

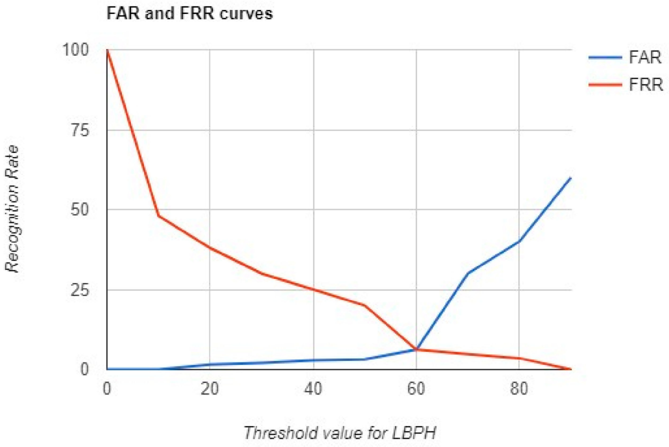

Figure 9. False Accept Rate (FAR) and False Reject Rate (FRR) analysis for Local Binary Patterns Histograms (LBPH) based on different threshold values.

5. Improving Recognition Performance

5.1. Face Model Evolution

Although the aforementioned face-recognition algorithms are able to recognize a trained face, it should be mentioned that the face model becomes outdated in the sense that it no longer represents the face of the user it is supposed to authenticate. This occurs because of the human face's tendency to change with time, which is manifested with aging signs, beard growth etc. With the intention of having a face model that adapts to the changes that occur to the user's face, FaceAuth makes sure that the model is updated. To achieve this, face images are appended to the face model dynamically. On the event successful authentication, if the confidence of the prediction by the face recognition algorithm is satisfactory, the face image is appended to the model. Thus, the face model is continuously updated to adapt to changes to the user's face.

5.2. Location-Based Recognition Adaption

FaceAuth adapts its recognition confidence requirements based on the location of the user. It achieves this by keeping track of the Wi-Fi networks the smartphone connects to and the connection frequency. This enables the recognition requirements, more specifically the required confidence of the recognition, to be adaptable based on the user's location. Thus, for more frequently visited locations, such as home or work, the confidence requirements of the recognition are more relaxed. This means that authentication is achieved even if the distance of the LBP feature vector of the face constructed from the camera view and the saved face model is near the set threshold. For unvisited locations, or those less frequently visited, there are more strict requirements for authentication. Consequently, the face image obtained from the camera has to match the saved face model, yielding a recognition result with a minimum demanded distance. This adaption differentiates between more frequently visited spaces and those spaces less frequently visited or not visited before. The more frequently visited spaces are considered to be safe because it can be implied that people located in those spaces are familiar to the user, which is why recognition requirements are more relaxed. On the other hand, when the user is at a new or less frequently visited location, that space is characterized as less safe because of its likelihood of containing people not familiar to the user, so recognition requirements are stricter for achieving authentication.

6. Modeling Gesture Behavior

Face authentication in the context of smartphones has proven to be vulnerable against attacks. A sufficiently high-resolution face image displayed on another mobile device will be enough to circumvent security measures and bypass the authentication. Therefore, in order to prevent such bypass attacks, we implemented an additional system. This system models user gesture to recognize such bypass attacks. In this section, we present the steps required to build such system. This process also assumes that users are familiar using face recognition to unlock their smart phones. User gestures are a sequence of time series; therefore, unsupervised learning in LSTM recurrent neural network to actively learn to recognize, group and discriminate user gestures is implemented. Moreover, an alternative system based on clustering technique is also implemented for recognizing gestures. We conducted an experiment to evaluate the LSTM-based gesture recognition against a cluster-based system.

6.1. Experiment

To test the performance of LSTM and cluster-based methods, we conducted an experiment. Because of the flexibility and feasibility, we implemented the LSTM and cluster-based methods on an Android-based mobile app. The process starts with collecting user gestures, which in foundation are time series signals. The signals are stored in the smartphone database. The datasets from the database are fetched to train the LSTM, which is also implemented on the mobile phone. The trained model is also stored on the aforementioned database together with the SRC model for face recognition. When the very first models are available, the gesture signals and face images are collected from the smartphone for the very first predictions. Scores from both predictions are combined into the final result, where it should be decided whether the user passed the authentication check or not. User gesture collection is conducted in four steps:

(1) User pushes the turn-on button of the smartphone (from this point, data is collected from the smartphone gyroscope, accelerometer and other sensors).
(2) User directs the smartphone front camera to her face (approximate distance front camera to face is estimated using front camera).
(3) Face recognition is done (at this point, data collection is stopped, and the collected data is analyzed).
(4) User behavior weights are derived from the collected data, which will be taken into consideration together with the weights from face recognition.

From the above description, unsupervised learning is used to train a model from the multi-dimensional smartphone movements collected data. Moreover, we do not need additional datasets from the description, because the data will be generated by the user herself and used only for that user. Until it generates enough data, the accuracy will be low, but in time, this accuracy increase.

6.2. Implementing LSTM

During this step, LSTM is implemented on a mobile app. This step includes implementing the environment for training the LSTM using the collected data from the local database on the app, and the implementation, where the model is used to recognize user gesture.

6.3. LSTM Evaluation

During this step, an evaluation of the trained model is performed. The developed app with the LSTM will be installed in a specific number of test users. The users will use the app in everyday life until enough data for training the LSTM is collected. At the end of the step when the model is built, each test user is asked to authenticate first with face recognition and then with the combination of face authentication and gesture recognition. Different scores are derived and checked, to establish which method scores better. After that, an image is first shown for each user to try to bypass the face authentication, then the same image is shown in front of the face-authentication and gesture-recognition system. The scores of the two methods are derived and compared. During the data collection, step 4 can be performed. As a result, an approximation of system accuracy is estimated.

6.4. Implementing Cluster-Based Gesture Recognition

During this step, cluster-based gesture recognition is implemented on a mobile app. This step includes implementing the environment for training the clusters using the collected data from the local database on the app, and the implementation, where the model is used to recognize user gesture. Evaluation can also be performed during this step, because the data has already been collected during the step 3.

6.5. Results

The performance comparison between the LSTM and cluster-based methods is presented in Figure 10.

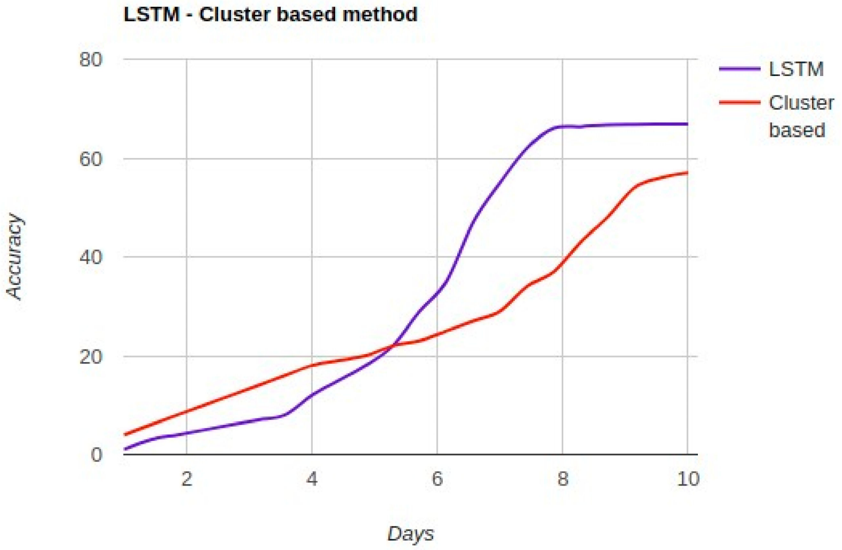

Figure 10. Performance comparison between LSTM (Long Short-Term Memory) and cluster-based methods for gesture recognition.

Figure 10 shows that both methods have a "cold start" with a low accuracy in the beginning. This can be justified based on the low amount of training data in the beginning. Moreover, in the beginning, the cluster-based method has a better performance than LSTM. With time passing, more data is collected on mobile phones from user interaction with their phones. After a couple of days, both methods have increased accuracy. Based on Figure 10, LSTM has a higher accuracy in comparison with the cluster-based method. After a specific time, the methods do not have any major increase on their accuracy, with the LSTM outperforming the cluster-based method on gesture recognition.

7. Conclusions

One of the facts that makes face recognition by computer devices attractive is its human-like approach. In this paper, Eigenfaces, Fisherfaces and Linear Binary Patterns Histograms were presented as some of the algorithms that enable face recognition. With further advancements in image processing, the sustainability of face recognition towards variations of environmental factors in images (e.g., lighting) is increased. Furthermore, the processing power of today's computer devices enables the aforementioned algorithms to perform better than humans in one-to-many face-recognition problems. However, when dealing with authentication by face recognition, which is a one-to-one matching problem (one face image is compared to the saved model of one face), recognition algorithms are not that effective. Since the face model is two dimensional, it could authenticate a two-dimensional photo of the user since it cannot see the difference between the user's face and a two-dimensional photo of the user. Furthermore, tracing the movement of the user's face when in recognition mode could still not see the difference between the user's face and a three-dimensional model of that face or a video of the user. Also, human faces are not that unique for the purposes of differentiating one human from another. For identical twins, face recognition would not be able to recognize the difference between the two individuals. Even recognition performed daily by humans would not be able to verify their identity based only on face data. The application of face-recognition systems in law enforcement, or entertainment apps such as games would increase their functionality. Taking into account the characteristics of the face as a biometric sign, the probability of face similarities between individuals and the limits of the cameras on computer devices, it can be concluded that face recognition is much more suitable as one of many modules of an authentication system rather than it being the only component of an authentication system. Thus, a face-authentication system consisting of face, gesture and location recognition as components was suggested. The proposed system utilizes the prediction scores from the three modules to yield the final authentication score. Considering the tendency of the face to change with time, the face model is updated dynamically even after it is constructed in order to adjust to these occurring changes. Thus, the chances for the face module to yield false negative results are significantly lowered. Furthermore, face recognition requirements are optimized based on location data of the user. By marking frequently visited locations as safer, the face-recognition requirements are optimized. Thus, the face-recognition algorithm is stricter when the user is located in unknown locations. Moreover, the observation of user gestures adds another component to the authentication system, making it less dependent on the face-recognition module. By adding gesture recognition as a module in the overall authentication system, the reliability of the results derived from this system is further increased.

Author Contributions: Gresa Shala and Valon Xhafa conceived and designed the experiments; Gresa Shala performed the experiments; Blerim Rexha and Valon Xhafa analyzed the data. Blerim Rexha, Gresa Shala and Valon Xhafa wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. How People Use Their Devices. Available online: https://storage.googleapis.com/think/docs/twg-how-people-use-their-devices-2016.pdf (accessed on 20 November 2017).
2. Shen, Y. Face recognition on smartphones via optimized Sparse Representation Classification. In Proceedings of the Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014; pp. 173–180.
3. Kang, J. Two-factor face authentication using matrix permutation transformation and a user password. *Inf. Sci.* **2014**, *269*, 1–20. [CrossRef]
4. Shafique, U. Face Description with Local Binary Patterns: Application to Face Recognition. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 1.
5. Pato, J.; Millett, L. *Biometric Recognition: Challenges and Opportunities*; The National Academies Press: Washington, DC, USA, 2010.
6. Otsuka, Y. Face recognition in infants: A review of behavioral and near-infrared spectroscopic studies. *Jpn. Phycol. Res.* **2014**, *56*, 76–90. [CrossRef]
7. Li, S.; Jain, A. *Handbook of Face Recognition IEEE Transactions on Pattern Analysis and Machine Intelligence*; Publishing House Springer: Secaucus, NJ, USA, 2011.
8. Mobile Biometrics: Consumer Markets, Opportunities & Forecasts 2016–2021. Available online: https://www.juniperresearch.com/press/press-releases/voice-and-facial-recognition-to-be-used-in-over-60 (accessed on 20 November 2017).
9. Samsung Face Recognition. Available online: http://www.samsung.com/global/galaxy/galaxy-s8/security/ (accessed on 20 November 2017).
10. Ice Cream Sandwich. Available online: http://www.android.com/about/ice-cream-sandwich (accessed on 20 November 2017).
11. Hong, F. Waving Authentication: Your Smartphone Authenticate You on Motion Gesture. *ACM* **2015**, *33*, 263–266.
12. Shrestha, B. Wave-to-Access: Protecting Sensitive Mobile Device Services via a Hand Waving Gesture. In Proceedings of the International Conference on Cryptology and Network Security, CANS 2013: Cryptology and Network Security, Paraty, Brazil, 20–22 November 2013; pp. 199–217.
13. Arif, A. A tap and gesture hybrid method for authenticating smartphone users. In Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services, Munich, Germany, 27–30 August 2013; pp. 486–491.
14. Klapper, M. Unsupervised Learning in LSTM Recurrent Neural Network. In Proceedings of the International Conference on Artificial Neural Networks ICANN 2001, Vienna, Austria, 21–25 August 2001; pp. 684–691.
15. Chellappa, R.; Sinha, P.; Phillips, P. Face Recognition by Computers and Humans. *IEEE Comput. Soc.* **2010**, 46–55. [CrossRef]
16. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neuro-Sci.* **1991**, *3*, 71–86. [CrossRef] [PubMed]

17. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen* **1936**, *7*, 179–188. [CrossRef]
18. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; Volume 1.
19. Zhang, G. *Boosting Local Binary Pattern (LBP)-Based Face Recognition*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; p. 3338.
20. OpenCV. Available online: opencv.org (accessed on 20 November 2017).
21. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Comput. Soc.* **2001**, *1*, 511–518.
22. Dlib Face Detector. Available online: http://dlib.net/face_detector.py.html (accessed on 20 November 2017).
23. Mobile Vision API. Available online: https://developers.google.com/vision/introduction (accessed on 20 November 2017).