Continuous Face Authentication Scheme for Mobile Devices with Tracking and Liveness Detection

Max Smith-Creasey*, Fatema A. Albalooshi†, and Muttukrishnan Rajarajan*
*School of Mathematics, Computer Science and Engineering, City, University of London, London, UK
{Max.Smith-Creasey, R.Muttukrishnan}@city.ac.uk
†College of Information Technology, University of Bahrain, Sakheer, Kingdom of Bahrain
Falbalooshi@uob.edu.bh

Abstract—We present a novel scheme for continuous face authentication using mobile device cameras that addresses the issue of spoof attacks and attack windows in state-of-the-art approaches. Our scheme authenticates a user based on extracted facial features. However, unlike other schemes that periodically re-authenticate a user, our scheme tracks the authenticated face and only attempts re-authentication when the authenticated face is lost. This allows our scheme to eliminate attack windows that exist in schemes authenticating periodically and immediately recognise impostor usage. We also introduce a robust liveness detection component to our scheme that can detect printed faces and face videos. We describe how the addition of liveness detection enhances the robustness of our scheme against spoof attacks, improving on state-of-the-art approaches that lack this capability. Furthermore, we create the first dataset of facial videos collected from mobile devices during different real-world activities (walking, sitting and standing) such that our results reflect realistic scenarios. Our dataset therefore allows us to give new insight into the impact of user activity on facial recognition. Our dataset also includes spoofed facial videos for liveness testing. We use our dataset alongside two benchmark datasets for our experiments. We show and discuss how our scheme improves on existing continuous face authentication approaches and efficiently enhances device security.

Index Terms—continuous authentication, face recognition, face tracking, liveness detection, biometrics

1 Introduction

Mobile devices are one of the most widely used technologies of our time, requiring users to store private and personal information to use features and applications. Whilst many devices incorporate a variety of security mechanisms such as a PIN, password, or pattern, recent research has shown that such security mechanisms are susceptible to a variety of forgery attacks, such as the smudge attack [1]. Additionally, such mechanisms are intrinsically limited in that they provide only inconvenient and one-time authentication; the user explicitly authenticates once for entire device access. These mechanisms for authentication leave the device vulnerable to attacks if it is left unlocked by the genuine user.

Recent research in mobile device security has sought to alleviate the issues with traditional security mechanisms by proposing continuous authentication (also known as active authentication) techniques [2]. These techniques typically collect biometric data from the device during use and compares the data to a user profile. Collected biometrics are either behavioural (e.g.: touch-screen gestures) or physiological (e.g.: fingerprint) [3]. Physiological biometrics often yield better results because they are not as susceptible to change. For this reason, facial recognition in continuous authentication schemes is an active research area.

Using transparently captured faces from mobile devices to authenticate was first proposed in studies such as [4] and [5]. Since then, however, the quality of cameras and computational power in devices has made facial recognition more feasible. Industry also has an interest in mobile face recognition with Google incorporating *Smart Lock*[1] into Android and Apple announcing *FaceID* for iPhone[2]. These approaches, however, use facial recognition in a one-time authentication process.

State-of-the-art research into continuous facial authentication sees schemes proposed that periodically (e.g.: every 30 seconds) capture facial images and authenticate them [6]. Such schemes leave windows of attack and can be seen as more periodic than continuous. Conversely, schemes that authenticate each available frame are computationally inefficient. Furthermore, state-of-the-art studies achieve results for robustness against attacks by testing the system using impostor faces only [7] and do not account for the possibility of facial spoof attacks [8]. We also find that such schemes do not account for variety in user activity during face recognition; a crucial area of exploration for real-world systems.

The main focus of this paper is producing novel components that form a facial authentication scheme that mitigates spoof attacks, properly continuously authenticates (rather than periodically) and provides insight into facial recognition in real-world scenarios. Our approach uses features extracted from a detected face to verify the liveness. We show the results of our face recognition approach on faces collected from different illumination conditions and different activities. We mitigate attack windows and improve efficiency by tracking authenticated faces rather than re-authenticating in subsequent video frames. The contributions of this paper are therefore threefold:

- We create a liveness detection component for use in continuous authentication schemes. It provides mitigation against 2D spoof attacks using printed faces or videos played in front of a mobile device camera. We test our liveness detection on different facial attributes.

Corresponding author: Max Smith-Creasey (email: Max.Smith-Creasey@city.ac.uk).

1. https://support.google.com/nexus/answer/6093922
2. https://www.apple.com/iphone-x/

- We present a new facial recognition scheme and experiment with different facial attributes, different attribute sizes, different classification techniques and different datasets. Our results uniquely show and explain face recognition scores during different user activities.
- We propose and show the effectiveness of a tracking algorithm for ensuring that the face authenticated is the user currently using the device. We show how this novel enhancement can efficiently and consistently maintain security after the user is authenticated.

In Section 2, we briefly summarise the previous work related to our study. Section 3 presents the general idea for our system and describes our novel approach to continuous authentication. Section 4 describes the experiments we performed on our system and discusses the results we obtained. Section 5 concludes our research and Section 6 discusses the future work that can be derived from our system.

2 Related Work and Motivation

Research into continuous authentication on mobile devices has attracted a lot of academic interest recently due to the added security concerns of devices. Sensors on mobile devices have led to schemes proposing a variety of biometrics for authentication, including touchscreen gestures [9], keystrokes [10], accelerometer data [11], location [12], facial features [13] and combinations of modalities [14].

The concept of face and facial feature authentication on mobile devices was first demonstrated in several earlier studies. In [5] the authors present a continuous authentication scheme on older Nokia N90 mobile devices. They used local binary patterns (LBP) [15] and skin tones from faces that were detected using the Viola-Jones detection technique [16]. Authentication rates of up to 96% were achieved. The researchers in [4] propose a transparent facial recognition scheme for mobile devices. They test a variety of different facial recognition algorithms and show that accuracy is increased when facial orientation is considered. However, the datasets used in these studies are limited because they do not consider user activity.

Advances in continuous facial authentication use improved mobile sensors and computational power to enhance accuracy and performance. In [6] the authors periodically collect faces with gyroscope, accelerometer and magnetometer data. They use the collected sensor data to align the face image to a neutral pose. The alignment method improved recognition performance by 6%. Results show that 96% of genuine users were never locked out during testing and 89% of impostors were detected within 2 minutes. Though the scheme achieves good accuracy, it only authenticates once every 30 seconds and little consideration is given to the potential attack window that this creates (in which an impostor could use the device). Furthermore, the dataset they use does not contain faces from different activities and thus their results lack realism.

Researchers in [13] present the results of different facial recognition algorithms (e.g.: Eigenfaces and Fisherfaces) on their own publicly available dataset containing videos of users collected on mobile devices in different illumination conditions. The study extracts the eyes, nose and mouth of faces for recognition. Results show that the different illumination conditions have a detrimental effect on the accuracy. However, the study does not include liveness detection which leaves it vulnerable to spoof attacks. Furthermore, the study recommends a data collection period of 10 seconds, creating a viable attack window.

In [17] the authors present a continuous facial attribute authentication scheme. Their scheme trains a set of Support Vector Machine (SVM) [18] classifiers on different facial characteristics (e.g.: moustache) such that they can produce a list of facial attribute scores for facial recognition. The authors show that their approach improves on the popular whole-face LBP method for facial recognition. However, no consideration is given to spoof attacks or captured faces during different user activities which reduces the practicality of the scheme. Additionally, authentication is done for each available frame which adds continuous requires computational overhead.

The facial part of [19] presents a scheme that uses a generalized version of multivariate low-rank regression for recognition. The scheme shows how combined individual areas of the face can achieve better recognition results than whole face or any one of the facial areas alone. When using all facial areas, they achieve accuracy results of 95.07%. The scheme, however, does not address minimizing attack windows or detecting spoof attacks. In the facial component of the continuous authentication scheme presented in [20], the researchers capture and authenticate a face when a touch-screen interaction occurs. Upon capturing a face, the eyes, nose and mouth were extracted. Histograms of oriented gradients (HOG) [21] were extracted and classified by 1 to 3 classifiers in a stacked classification scheme. The facial scheme achieves equal error rates (EER) as low as 4.76%. As with other related work, however, consideration was not given to attack windows between face collection or to the potential of spoof attacks.

Spoof attacks on mobile devices are a popular way to bypass the security mechanisms. To alleviate facial spoof attacks there have been investigations into techniques to detect spoofed faces. In [22] the authors collect a dataset of spoofed faces and present a mobile spoof detection framework that uses different intensity channels, different image regions and different feature descriptors to detect printed faces, videos and 3D masks. In the spoof detection scheme in [23] the authors collect spoofing attacks for PC and mobile devices. They propose a face spoofing detection algorithm based on Image Distortion Analysis (IDA). They use specular reflection, blurriness, chromatic moment and colour diversity as features extracted to form the IDA feature vector. Multiple SVM classifiers are then trained to detect different types of spoof attacks. However, a liveness detection component has not yet been proposed for continuous authentication contexts leaving state-of-the-art schemes vulnerable.

Object tracking is an area of computer vision that has been used to track faces. In [24] the authors use the Kanade-Lucas-Tomasi (KLT) to track faces. The authors show that their scheme can robustly track facial images even in cluttered backgrounds. In [25] the authors use a commercial eye detector and tracker in an iris recognition scheme achieving an EER of 11%. The literature on tracking, however, shows a lack of application on continuous face authentication schemes on mobile devices.

Based on the related work discussed we have identified areas that have not yet been explored alongside continuous facial authentication. Most schemes focus on scheme design and lack suitable realism or attack mitigation. We identify a lack of liveness detection within the continuous authentication context. Furthermore, we identify a lack of facial recognition components applied to realistic face data collected for different activities (e.g.: walking). Lastly, we see that there is a lack of research into truly continuous schemes that track the genuine user rather than only authenticate periodically.

3 Continuous Face Authentication

Here, we present our novel continuous facial recognition framework that enhances accuracy and robustness. We describe the

general concept, our data capture process, the flow and processing of data and how the classification setup we construct allows us to authenticate a face.

3.1 General Idea

This study introduces a novel scheme to continuously authenticate user identity using facial characteristics from mobile devices. We address a lack of realism and security concerns identified in previous work. We hypothesize that a novel continuous authentication scheme addressing these prevalent issues can yield a more robust system and set a new and important benchmark within this field.

Our proposed framework continually monitors video captured via a front-facing mobile device 2D-camera. Monitoring begins when the user unlocks the device and continues until the user ends the session by relocking the device. Frames from the front-facing camera are captured during this process. Each captured frame is adapted to make facial detection more effective and efficient (e.g.: normalisation and frame padding). If the face detection technique identifies a face in the frame, the framework then fits a facial model with which facial landmarks can be located. If no face is detected, the frames following sequentially up-to a specified limit are searched for a face until the user must explicitly re-authenticate (e.g.: with a PIN or password).

Once key facial landmarks have been identified they are extracted from the frame such that liveness detection may be performed. Features from the facial attributes are extracted and classified with a classifier trained on features from genuine and spoofed photos and videos. If the classifier detects the face is a spoof the user is locked out of the device. Facial attributes from a detected face with defined landmarks that pass liveness detection are extracted into separate feature images. Using well-known image processing techniques, features from each of the facial attributes are extracted into a feature vector. The feature vectors for these facial attributes are concatenated to form a feature vector representing the face. This feature vector can then be classified against previously collected faces from the genuine user using distance techniques.

When a face has been successfully authenticated we switch to tracking mode. The authenticated face is continuously tracked in subsequent video frames whilst the user uses the device. During this period, no re-authentication takes place. Only when the authenticated face being tracked is lost does our framework initiate immediate re-authentication. The re-authentication procedure follows the same described process and begins the facial detection method. This framework is shown in Figure 1.

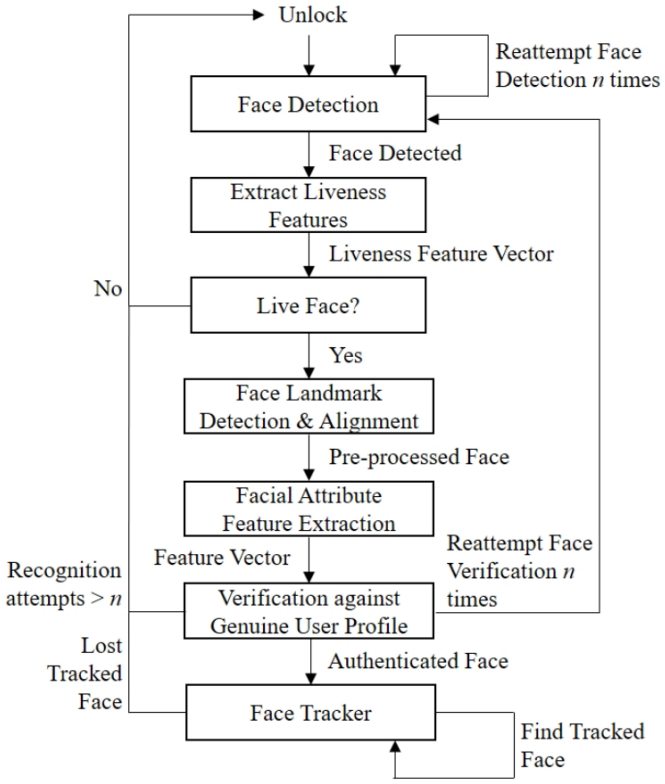

Fig. 1: The framework of our proposed continuous facial authentication scheme.

3.2 Face Processing

Face Detection. To detect faces in video frames we use the widely used Viola-Jones [16] algorithm. Whilst other techniques for face detection exist, e.g.:, we use Viola-Jones due to its relatively robust and efficient performance, making it computationally appropriate for mobile devices. We use the algorithm provided in OpenIMAJ [26], a Java-implemented multimedia analysis library. We build our scheme utilizing this library such that the system can be constructed using the Java programming language and thus making it possible to port the proof-of-concept to Android devices.

Faces captured from the front facing camera on mobile devices generally take up a significant portion of the frame which can impede detection. We therefore add padding to the frame by extending the width and height by continually repeating edge pixels (we discuss later the amount of padding applied). We find padding improves on the amount of faces detected. Additionally, as in [13], we set the minimum dimensions of the facial detection window such that redundantly small areas are not traversed and false positives are not as frequently recognized as faces. We find a minimum window size of 25% of the original frame width reduces the false positives and enhances performance, as in [13].

Face Landmarking. Detected faces need landmarks to be located such that facial attributes (e.g.: eyes) can be extracted. To achieve this, we use a popular Constrained Local Model (CLM) technique [27] provided in the OpenIMAJ library [26]. The CLM technique is passed an area in which a face has been detected. The CLM technique then fits a non-rigid Point Distribution Model (PDM) to the detected face. Landmarks are iteratively optimised within subspaces using mean-shifts and constrained to PDM shape limitations. The output of the technique is a facial model in which the landmarks are aligned to the face that was input. This technique was chosen due to the accurate performance as well as its open implementation availability.

Face Tracking. In order to track an authenticated face, we use template matching performed in the frequency domain using a fast Fourier transform. For efficiency, we select sub-image search area, I, of our frame in which it is probable the face will be located based on the previous frame. We then scan the search area using template T, the face region we identified in the previous frame. Each patch of I is compared with template T using the normalised correlation coefficient comparison method provided in [26]. A map of comparison results for each overlap permutation in the image I is computed by the following equation where w and h represent the template T width and height respectively and

$x' = 0...(w-1)$ and $y' = 0...(h-1)$.

$$R(x,y) = \frac{\sum_{x',y'} (T'(x',y') \times I'(x+x', y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \times \sum_{x',y'} (x+x', y+y')^2}} \quad (1)$$

Given the comparison map for the template T applied to image I we can identify the most likely location of the face in the frame as the patch that gives the highest comparison score. Given this image patch an attempt at recognising facial landmarks, using the discussed CLM method, can be attempted.

We chose this technique because, despite the limitations when scale and rotation adjusts, it is relatively fast and effective. Furthermore, we find that the spatial change of the face between two video frames does not change enough to impede our hypothesis that a face can be tracked in mobile device video for a suitable time period.

Face Liveness Detection. There is a critical requirement in continuous facial authentication schemes for a capable and solid system to identify and avoid spoofing threats. In the authentication process, it is important to distinguish between genuinely live faces and spoofed faces to eliminate false access to the device. The attacker can use printed faces and replayed videos from other devices of the genuine user, both of which have the same appearance and characteristics as the genuine face and result in a high false positive rate in systems without liveness detection. These false positive appearances make the conventional authentication process unreliable. Therefore, liveness detection (also known as spoofing detection) is an important factor for continuous facial authentication schemes on mobile devices [28]. Our focus in this paper is the liveness detection against 2D face spoof attacks, such as printed photos and video replays.

We analyse liveness detection using different facial regions (the entire face, eyes, nose, and mouth regions) and employ different textual features, namely LBP and HOG descriptors. As discussed localisation for the face regions is performed using the Viola-Jones method [16]. The attack set of facial videos contains genuine live facial video as well as the spoofed videos of printed faces and videos of faces displayed on device screens (see Section 4.1 for full dataset descriptions). The detected face patches from spoofed videos have an identifiable difference in local features (such as local texture and local histogram information) compared to face patches in genuine video images.

It has been widely proven that textural descriptors show successful outcomes in discriminating between live images and spoofed images in [29], [22] and [30]. The reflection generated from spoofing mediums like printed images and images generated from screens of other devices is different than that generated from real image faces. Moreover, printed face images and those generated by others displayed on screen devices usually contain artefacts and misrepresentation of colour and contrast degradation, which can be discriminated by textural descriptors.

As an example of textural description, we describe the Local Binary Pattern (LBP) technique, in [31] and [32], which measures surface texture by analysing micro-textural patterns in the face images. For an image having a g_p grey value of a sampling point in an equally divided rounded neighbourhood of P sampling points and radius R around point (x_c, y_c), the $LBP_{P,R}$ operator is defined as:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \quad (2)$$

Where g_c resembles the grey level of the centre pixel (x_c, y_c), and $s(z)$ is the thresholding step function.

$$s(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

LBP provides features that permit higher accuracy for liveness detection, which makes it an appropriate candidate for our liveness detection. Since textural characteristics of the live face images and attack face images are different, the LBP can therefore provide distinguishing features.

Our liveness detection method is based on single image analysis, referred to as a static approach. However, it can still be applied to a video sequence in the case where each video frame is analysed separately. The findings in the literature show that even this method can accomplish good performance with lower computational time compared to other dynamic systems [28].

Finally, an SVM classifier is learned in the LBP feature space for liveness detection. For a set of training data $D = ((\vec{x}_i), y_i)$, where each point is a pair of a vector point $(\vec{x}_i) \in R^d$ and a class label $y_i \in \{-1, +1\}$ corresponding to it, the classification function $f(\vec{x})$ can be expressed as:

$$f(\vec{x}) = sign(\vec{w}^T \vec{x} + b) \quad (4)$$

Here, w and b are parameters of the classification function. Our experimental results (see Section 4) demonstrate the effectiveness of our liveness classification framework.

Our spoofing detection scenario is shown in Figure 2, where we assume that a replayed video or a printed face picture is presented to the authenticating camera (on a mobile device). The figure shows the landmarking whole face, eyes, nose, and mouth regions to extract textural features from these areas. The SVM classifier is used to decide whether these features belong to a genuine face or an attack face.

Face Warping. After facial landmarks have been fitted to the face by the CLM technique and the face has been verified as live we perform face warping. Face warping uses the detected facial landmarks and uses them to form a warped version of the face standardized by size and pose. Whilst various complex techniques exist to account for extreme poses, as in [33] and [34], we find that for faces captured using a front facing mobile device camera the range of poses is limited due to the user explicitly looking at the device. The constrained local model is used to form triangles between the landmark points. In total 109 triangles are formed, each representing a patch of the face. We use Delaunay triangulation, as in [35], to warp these facial triangles to a standard pose. The effects of this facial warping aspect of our framework can be seen in Figure 3. We posit that this can help faces from different activities be recognised despite potentially having minor pose variations.

4 Experimental Results

In this section, we perform experiments on the proof-of-concept components that form our proposed continuous facial authentication framework. We evaluate the performance and robustness of our framework and assess facial recognition performance

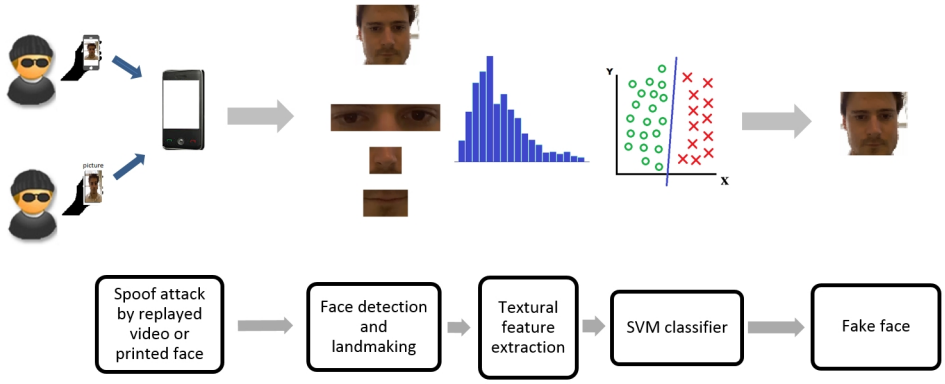

Fig. 2: This image shows the process taken to perform a face liveness detection test. In this scenario, the attacker has both a printed picture of the genuine user and a picture on a mobile device of the genuine user.

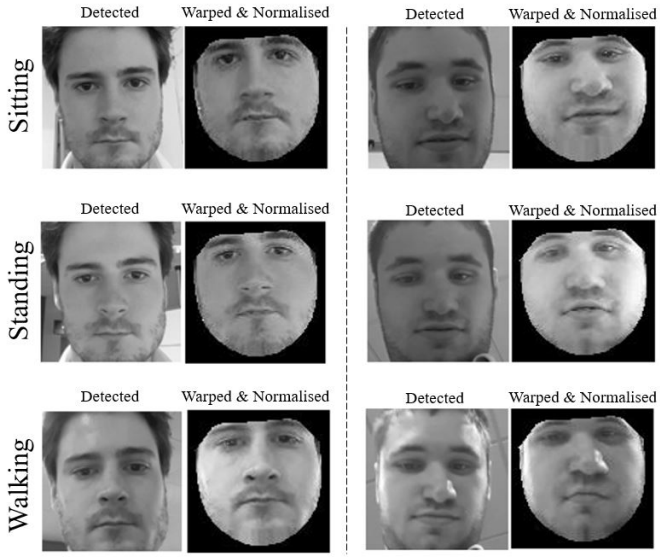

Fig. 3: Warped and normalised facial results for faces that have been detected in the CALF database from sitting (a), standing (b) and walking (c) activities.

during different user activities and different illumination scenarios. We discuss the pre-processing stages, implementation decisions, evaluation metrics and methodology for each experiment. We discuss and explain the results of each experiment.

4.1 Datasets

CALF (City Activity and Liveness Faces) Dataset. The City Activity and Liveness Faces Dataset is our dataset we have created. We used a free and publicly available background camera recorder application[3] with 20 volunteers. Our dataset is the first mobile faces dataset to contain data collected from different activities in which devices may be used. These activities are (1) sitting, (2) standing and (3) walking. The activities were chosen to replicate the most common use scenarios for mobile device users that could cause pose variation and consequently impact authentication. Our dataset contains video data from the front-facing camera for each user for these three different activities. Each activity was repeated 3 times (so that 3-fold cross-validation can be performed on same session data) and lasted more than 1 minute. This results in 9 videos for each participant and 180 videos in total. In each session, the participant was allowed to use the phone freely. We applied no constraints on the users other than maintaining the session activity.

Using the videos collected from activity sessions we also produce a dataset for liveness detection. This dataset is comprised of two spoof videos for each user. The first video is collected using a printed screenshot of the users face from the sitting scenario of the activity videos. The face is printed on A4 paper and held in front of the front-facing camera of a device of the same model as previously used and recorded for 30 seconds. The second video is collected using a live video of a sitting scenario video played on a mobile device screen. One of the two devices is held in front of the other such that the video playing on the screen of one device can be recorded using the front-facing camera of the other. As with the previous spoof video, this recording lasts 30 seconds.

The CALF dataset was produced using Samsung Galaxy A3 (2016) mobile devices. The front-facing camera on these devices records 1920×1080px videos at 30 frames per second.

UMDAA (University of Maryland Active Authentication) Dataset. This dataset was released by the authors of [13] and [7] and contains touchscreen gestures and face videos. The face part of the dataset contains the videos of 50 different users collected from the front-facing camera of the phone. The videos in this dataset were recorded on an iPhone 5S. The front-facing camera on this device records 1280×720 at 30 frames per second. For each user, videos are collected as the user carries out five tasks. These tasks are (1) enrolment, (2) document, (3) picture, (4) popup and (5) scrolling. The enrolment task requires the user to face the device and move their head up and down to collect pose variations. The remaining four tasks require a different form of touch interaction. The videos for each task are collected in 3 different illumination conditions: (1) indoor-light, (2) low-light, (3) natural-light. This benefits the realism of our study because different illumination environments would be expected in real-world scenarios. Our study, therefore, uses 750 face videos from this dataset.

MFSD (MSU Mobile Face Spoofing Database). The MFSD dataset was used in the spoofing study in [23]. It is one of the first datasets to include genuine and spoofed faces captured on a mobile device. The dataset is publicly available and consists of 280 video clips of photo and video attack attempts on 35 users. Two kinds of cameras were utilized as a part of collecting this dataset 1) a built-in camera on a MacBook Air and 2) a

3. https://play.google.com/store/apps/details?id=com.kimcy929.secretvideorecorder

front-facing camera in the Google Nexus 5 Android phone. Genuine faces were captured using both the MacBook and the Nexus camera. Spoofing attack videos were produced under the same conditions as in genuine face capture sessions. The dataset contains three types of spoof attacks: 1) high-resolution replay video attacks using an iPad Air screen, using a video captured on the MacBook and a video captured on the Nexus device 2) mobile phone replay video attacks using an iPhone 5S screen, using a video captured on the MacBook and a video captured on the Nexus device 3) printed photo attacks using an A3 paper with fully-occupied printed photo of the user face, using a face image captured on the MacBook and a face image captured on the Nexus device. Examples of real accesses and attacks from the MFSD database are shown in Figure 4.

4.2 Preprocessing

Regions containing faces or facial features are converted to greyscale to allow 1-dimensional evaluation of pixel values. Because of the different illumination conditions in the UMDAA dataset we perform normalization on the images extracted for facial recognition. This scales each pixel value between the highest and the lowest possible value.

4.3 Verification Techniques

We evaluate the face recognition component of our scheme using three different verifiers that allow us to produce a similarity score between a testing sample and collection of training samples. Our verification experiments use techniques that allow for anomaly detection as in [36] and [10]. We use such techniques because they require only one class of data unlike conventional binary verification techniques. We argue that anomaly detection techniques are more appropriate to biometric authentication schemes because it is not realistic to have samples from potential impostors in real-world scenarios. Furthermore, such similarity techniques do not require a considerable collection of training data or computational power, unlike more complex machine learning techniques. We use Euclidean distance, Chi-Squared distance and Cosine similarity (given by Equations 5, 6 and 7, respectively) to compute similarity scores between two facial feature vectors F_1 and F_2.

$$d(F_1, F_2) = \sqrt{\sum_{i=1}^{n} (F_{1_i} - F_{2_i})^2} \tag{5}$$

$$d(F_1, F_2) = \frac{1}{2} \times \sum\nolimits_{i=1}^{n} \left(\frac{(F_{1_i} - F_{2_i})^2}{F_{1_i} + F_{2_i}} \right) \tag{6}$$

$$d(F_1, F_2) = \frac{\sum_{i=1}^{n} (F_{1_i} \times F_{2_i})}{\sqrt{\sum_{i=1}^{n} ({F_{1_i}}^2)} \times \sqrt{\sum_{i=1}^{n} ({F_{2_i}}^2)}} \tag{7}$$

4.4 Evaluation Metrics

We test the effectiveness of our system by using the following five common biometric evaluation metrics:

1) False Acceptance Rate (FAR): This is the rate that an impostor is wrongly classified as the genuine user. The rate is calculated as in Equation 8.

$$FAR = \frac{ImpostorSamplesAccepted}{NumberofImpostorSamples} \tag{8}$$

2) False Rejection Rate (FRR): This is the rate that the genuine user is wrongly classified as an impostor. The rate is calculated as in Equation 9.

$$FAR = \frac{GenuineSamplesRejected}{NumberofGenuineSamples} \tag{9}$$

3) True Rejection Rate (TRR): This is the rate that the impostor user is correctly classified as an impostor. The rate is calculated as in Equation 10.

$$TRR = \frac{ImpostorSamplesRejected}{NumberofImpostorSamples} \tag{10}$$

4) True Acceptance Rate (TAR): This is the rate that the genuine user is correctly accepted as the genuine user. The rate is calculated as in Equation 11.

$$TAR = \frac{GenuineSamplesAccepted}{NumberofGenuineSamples} \tag{11}$$

5) Equal Error Rate (EER): The rate at which FAR and FRR are equal. FAR and FRR sets are usually obtained as an acceptance threshold is adjusted. FAR and FRR pairs are correlated such that if one increases the other decreases. For the FAR and FRR with the smallest difference, we define EER in Equation 12.

$$EER = \frac{FAR + FRR}{2} \tag{12}$$

4.5 Liveness Detection

In order to examine our liveness detection, we considered both the MSU MFSD dataset [24] and our CALF dataset. As discussed, both datasets contain various spoofing mediums, image qualities, and capturing devices which help to evaluate the liveness detection performance.

While most previous works on liveness detection are based on analysing the whole image frame of the face images, we consider only the inclusion of facial attributes (eyes, nose, and mouth) into our liveness detection method. We do this because the impostor might use a printed picture or replay a video that might not cover the whole area of the detected face or camera frame. Therefore, given a scenario where a spoofed face is presented to the authentication camera, the distortion features that allow us to recognise a spoofed face will only appear in the facial region and not in the surroundings of the facial area. Because the distortion features in the spoofed face might cover only a small portion of the testing area and not the whole area it could affect the performance of liveness detection if the whole frame is considered. This is particularly an issue if we consider small facial images such as a passport sized printed picture. For this reason, we will only consider facial landmarks like eyes, nose, and mouth in our approach to liveness detection.

Thus, we prepared eight test methods for liveness detection, namely: (1) LBP with the eyes region, (2) LBP with the nose

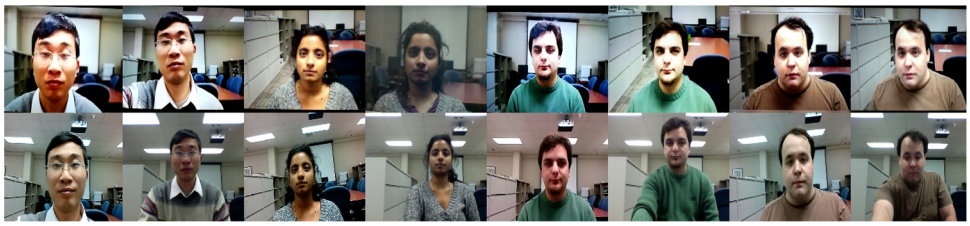

Fig. 4: Examples from MSU MFSD database. In the top row, samples from attack scenarios. In the bottom row, samples from real access scenarios.

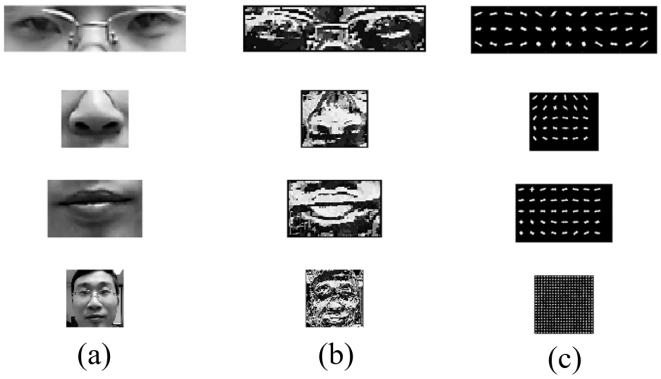

Fig. 5: Extracted textural features from different face regions including full face, eyes, nose and mouth regions. Images in (a) represent the original greyscale face regions, (b) shows a visualisation of LBP features and (c) shows a visualisation of HOG features.

region, (3) LBP with the mouth region, (4) LBP with the whole face, (5) HOG with the eyes region, (6) HOG with the nose region, (7) HOG with the mouth region, and (8) HOG with the whole face.

The performance of both LBP and HOG features in liveness detection is studied by first extracting these features from the different facial regions discussed. Figure 5 depicts those extracted features from different image regions. The liveness detection problem is formulated as a classification problem (following the approach in Section 3). SVM classifiers are trained on the LBP and HOG feature vectors extracted from the datasets for liveness detection. We use an SVM classifier with a radial basis function (RBF) kernel and apply a threshold of 0.5 to decide whether the score of the test image (between 0 and 1) is attributed to the live or spoofed class. These experiments use 10-fold cross-validation to verify the results. We compare the results of the algorithm to the true image labels to produce accuracy results. In the results we note that the higher the TRR and TAR the better the performance, and the lower the FRR and the FAR the better the performance.

Table 1 describes the percentage metrics computed from the test set utilizing both the MSU and the CALF datasets. We show the TRR, FAR, TAR and FRR for each textual extraction method and facial region. We notice that for both datasets the LBP textual descriptor appears to be superior at identifying spoofed facial images than the HOG textual descriptor. We find, for example, the best performing TRRs exceed 98.0% for both datasets when LBP is used whereas the best TRRs for HOG approaches only exceed 73.60%.

Moreover, when taking into consideration the need for a high TRR but also a high TAR we must find a method and facial region combination that yields a good compromise between the two metrics. Whilst LBP Eyes perform well in the CALF dataset they only achieve a TRR of 53.90% in the MSU dataset. The LBP Nose offers a better compromise with high TRR and TAR results for both datasets. The best compromise, however, is observed when LBP Mouth is used. In both datasets the TAR exceeds 90% (ensuring the genuine user is permitted access), whilst the TRR results are 91.58% and 77.61% for MSU and CALF datasets, respectively (ensuring security against spoof attacks). We find this to be the best compromise between TRR and TAR in Table 1.

The average elapsed time for all methods on both datasets has been measured. It is found that the mean time taken to make a decision is 41.9ms. The time differences between different attributes and texture extraction methods is minimal and all combinations perform fast enough to be performed transparently in our scheme. The results indicate the feasibility of the proposed solution.

From the results of our experiments we use the mouth region with LBP textual description features in the liveness detection component of our scheme due to its high accuracy, relatively quick classification time and compromise between TRR and TAR when applied to both datasets.

4.6 Facial Recognition and Scenario Cross-Comparison

In this section, we explore the facial recognition component of our continuous authentication scheme. We show that our facial recognition scheme is capable of improving on the accuracy of state-of-the-art continuous facial authentication methods. We show the accuracy and efficiency trade-offs of our scheme through varying different verification techniques and attribute parameters. We also show the results of cross-session tests for different faces recorded during scenarios.

We first test different combinations of attributes, feature types and verification techniques using our scheme on the UMDAA dataset and our CALF dataset. For each video, we extract every 10th frame and use the Viola-Jones algorithm to perform face detection. Where faces are detected within frames we extract and warp the face to a standardised pose and appropriate size for feature extraction using facial fiducial points identified via CLM. From each aligned face, we extract a sub-image for the left-eye, right-eye, nose and mouth.

Uniform LBP and HOG features are then extracted from each of the facial attributes. Each LBP cell provides a histogram of 59 uniform LBP features and each HOG cell provides a histogram of 9 HOG gradient features. For this experiment, all facial attributes are resized to 24×24px and 2×2 feature cells are utilized (variations of these parameters are discussed later). Features from the cells of a facial attribute are concatenated into a feature vector. Where applicable, multiple feature vectors are concatenated into a larger feature vector representing combined facial attributes. As in [7], for the UMDAA dataset we use the

	MSU Dataset				CALF Dataset				
Method and Facial Attributes	TRR (%)	FAR (%)	TAR (%)	FRR (%)	TRR (%)	FAR (%)	TAR (%)	FRR (%)	Avg Time (ms)
LBP Eyes	53.90	46.07	78.71	21.28	98.95	1.05	74.16	25.84	68
LBP Nose	77.58	22.42	90.16	9.84	94.89	5.11	69.97	30.03	50
LBP Mouth	91.58	8.42	93.47	6.53	77.61	22.39	98.51	1.49	24
LBP Whole Face	98.58	1.42	74.08	25.92	80.27	19.73	54.01	45.99	48
HOG Eyes	51.94	48.06	98.49	1.51	45.91	54.09	70.48	29.52	38
HOG Nose	64.33	35.67	79.85	20.15	77.61	22.39	97.01	2.99	17
HOG Mouth	73.60	26.40	62.82	37.18	69.94	30.06	88.08	11.92	71
HOG Whole Face	67.46	32.54	70.77	29.23	39.19	60.81	62.11	37.89	19

TABLE 1: The evaluation metrics for liveness detection when different feature extraction techniques are used on different facial attributes taken from faces detected in the MSU and CALF datasets.

Enrolment video from each session for training. For the CALF dataset, we use 3-fold cross validation in which we iteratively use two videos from a session as the training video and the remaining one as a testing video. We use 150 facial attribute feature vectors selected evenly from all sessions of a users data from the training videos to form a profile. We can then test samples from all testing sessions to these for similarity using the verification techniques discussed previously. We use this number of profile vectors as it was found to yield good results during initial experimentation.

We show the EER results of this experiment for each dataset with average times taken to extract facial attributes from an aligned face and produce a score in Table 2. Our results show that for both datasets a uniform LBP feature type and the Chi Square verification technique yield the lowest EER. We find that for the UMDAA dataset the eyes attribute achieves the best EER and for the CALF dataset the eyes, nose and mouth combination yields the lowest EER. These results are to be expected because the eyes are the most consistently visible attribute in each face video. The improved performance on our dataset when nose and mouth attributes are included appears to be because the face is not as close to the edge of the frame in our dataset due to a wider camera angle on our mobile devices and therefore nose and mouth attributes are less likely to be partial or misrecognised. We observe our CALF dataset yields lower EERs than the UMDAA dataset. We find the time taken for each different approach differs negligibly. The majority of the time is spent detecting and warping the face; the verification for facial regions takes relatively minimal time.

We follow this experiment by exploring the accuracy and efficiency compromises for our face recognition approach when we adjust the pixel dimensions of the facial attributes extracted from the face image and the number of cells used in feature extraction. We adjust the size of the facial attributes for sizes 12×12px, 24×24px and 36×36px. We further experiment with the number of feature cells using 1×1, 2×2, 4×4 and 6×6. For each parameter variation, we record both the EER and the time taken to extract features and calculate a score. We use the facial recognition attributes and verification techniques that produced the lowest EER for each dataset that we identified in the previous experiment (e.g.: LBP, Eyes and Chi-Square for UMDAA). The results of this experiment are shown in Table 3. We find that as the image size and number of cells increases, as does the average processing time to extract and authenticate the features. An attribute size of 24×24px divided into 4×4 feature cells produces the best feature vector for classification with minimal computational overhead for both datasets. We note that our lowest EER for the UMDAA dataset of 25.46% is lower than the 30.00% achieved on the same dataset in the state-of-the-art scheme in [7].

Next, we evaluate the effectiveness of our face recognition technique when different illumination and activity sessions (from UMDAA and CALF datasets, respectively) are cross-tested. We follow the set-up of previous experiments and use the techniques that provide the lowest EERs so far. In these tests we rotate different sessions for training and testing. The results are presented in Table 4. Results show EERs are low when the training and test sessions are the same. In the CALF dataset the EERs for standing and sitting are similar and cross-comparison of the two activities does not impede the classification more than ∼4%. However, walking yields higher EERs and does not perform well in cross-comparison tests. The higher EERs appear to be due to the additional blur and movement in the video data. For the UMDAA data we observe that the indoor-light performs better than low-light and natural-light sessions. Because of the illumination differences, cross-comparison EERs are relatively high. Despite our considerations for pose and illumination in our scheme, we identify there is still need for improvement in cross-session face authentication.

4.7 Face Tracking

In this experiment, we evaluate the effect and performance of the novel tracking component of our continuous authentication scheme. We show and discuss the added robustness and security that our tracking scheme yields when compared to other state-of-the-art schemes that inefficiently re-authenticate for each frame or leave an attack window by continuously re-authenticating after an elapsed time period.

We first show how tracking an authenticated face can reduce attack windows that exist in schemes with a time delay between re-authentication (as exists in [6] and [13]). We run each test video in real-time and record the time each face was tracked to illustrate the benefits of tracking. We vary the frame sizes and padding because we posit that adapting these features may yield advantageous benefits (e.g.: padding can allow faces close to or over the frame edge to be tracked better). This experiment is performed on both UMDAA and CALF datasets. Results for this experiment are in Table 5. They show that faces can be tracked for significant periods before re-authentication is required or triggered; instead of having an insecure time delay that causes an attack window. In our scheme, when a face is lost, re-authentication can simply be immediately initiated, eliminating attack windows. We also collect the time taken track each face frame compared to the time taken to authenticate it, shown in Table 6. This allows us to evaluate the enhanced efficiency of tracking an authenticated face compared to the process of face detection, feature extraction and authentication for each frame (as appears in [17] at a rate of 4 frames per second). In all cases we observe tracking a frame uses less computational time than authenticating a frame. Our observations also reveal that in all cases additional padding allows the face to be tracked for greater time periods, verifying our padding

Method		LBP (Cosine Similarity)		LBP (Chi Square Similarity)		LBP (Euclidean Similarity)		HOG (Cosine Similarity)		HOG (Chi Square Similarity)		HOG (Euclidean Similarity)	
Facial Attributes	Dataset	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)
Whole face	UMDAA	43.36	354.68	30.23	354.68	37.43	354.68	36.31	355.16	28.68	355.16	30.42	355.16
	CALF	27.69	357.08	16.70	357.08	25.21	357.08	22.49	357.55	16.92	357.55	18.26	357.55
Eyes, Nose & Mouth	UMDAA	34.01	354.18	27.90	354.18	32.01	354.18	34.48	354.31	31.24	354.31	32.34	354.31
	CALF	23.46	356.58	12.76	356.58	21.81	356.58	17.99	356.71	16.00	356.71	16.45	356.71
Eyes & Nose	UMDAA	34.26	354.14	25.73	354.14	30.73	354.14	32.07	354.24	27.51	354.24	29.23	354.24
	CALF	25.74	356.53	13.90	356.53	23.16	356.53	18.77	356.63	16.12	356.63	16.99	356.63
Eyes & Mouth	UMDAA	35.31	354.13	27.64	354.13	33.24	354.13	35.96	354.23	32.81	354.23	34.00	354.23
	CALF	22.54	356.53	13.93	356.53	21.94	356.53	19.08	356.63	17.26	356.63	17.69	356.63
Mouth & Nose	UMDAA	37.42	354.08	33.21	354.08	35.74	354.08	37.86	354.15	34.78	354.15	35.61	354.15
	CALF	29.90	356.48	15.90	356.48	24.74	356.48	22.08	356.55	19.49	356.55	20.18	356.55
Eyes only	UMDAA	36.17	354.09	25.63	354.09	32.68	354.09	33.54	354.15	28.41	354.15	30.51	354.15
	CALF	26.87	356.49	16.43	356.49	24.53	356.49	22.54	356.55	18.92	356.55	20.29	356.55
Mouth only	UMDAA	39.03	354.03	36.23	354.03	38.21	354.03	41.86	354.07	39.44	354.07	40.44	354.07
	CALF	29.70	356.44	19.93	356.44	25.48	356.44	28.87	356.47	24.02	356.47	24.92	356.47
Nose only	UMDAA	38.71	354.04	31.74	354.04	34.12	354.04	34.77	354.07	30.13	354.07	31.37	354.07
	CALF	34.29	356.44	18.24	356.44	28.39	356.44	23.05	356.47	20.30	356.47	21.31	356.47

TABLE 2: Given a frame the table shows EERs and times taken for each different combination of facial attributes, feature types and verification techniques.

		Number of Feature Cells							
		1×1 cells		2×2 cells		4×4 cells		6×6 cells	
		EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)	EER (%)	Avg Time (ms)
UMDAA Attribute Size (px)	12×12	29.65	355.22	26.83	353.69	29.01	353.57	29.93	354.26
	24×24	29.20	356.03	25.5	354.00	25.46	354.76	28.08	355.54
	36×36	31.62	354.64	27.66	354.47	25.50	355.23	26.58	357.08
CALF Attribute Size (px)	12×12	16.38	359.06	13.05	356.16	15.16	351.21	17.67	359.05
	24×24	17.37	359.26	12.64	356.39	11.68	358.37	13.08	361.45
	36×36	19.55	357.62	14.88	356.97	12.08	360.05	12.39	362.78

TABLE 3: The EERs and average times when the number of feature cells and the number of pixels per facial attribute are varied. This experiment uses the technique for each dataset that achieved the lowest EER in the previous experiment.

Session (UMDAA)	EER (%)	Session (CALF)	EER (%)
1→1 UMDAA	15.81	1→1 CALF	8.55
2→2 UMDAA	24.13	2→2 CALF	8.26
3→3 UMDAA	22.53	3→3 CALF	13.32
1→2 UMDAA	34.92	1→2 CALF	12.37
1→3 UMDAA	28.97	1→3 CALF	23.55
2→1 UMDAA	36.32	2→1 CALF	12.65
2→3 UMDAA	32.34	2→3 CALF	22.68
3→1 UMDAA	33.70	3→1 CALF	16.43
3→2 UMDAA	37.61	3→2 CALF	15.60

TABLE 4: The EERs and average times when different illumination and activity sessions are compared. For the UMDAA dataset sessions 1, 2 and 3 correspond to indoor light, low light and natural light, respectively. For the CALF dataset sessions 1, 2 and 3 correspond to sitting, standing and walking, respectively.

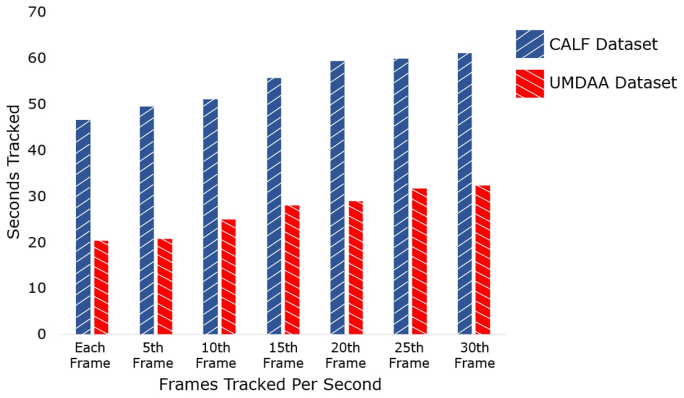

Fig. 6: We show the length of time faces are tracked when we vary the frame sampling rate. We see a lower sampling rate per second can enhance tracking time. We note that we find the UMDAA dataset yields a lesser tracking time due to the videos being shorter in length.

expectation. However, we observe that padding does require greater processing time. We further see that as the padding increases the smaller sized frames produce higher and more stable tracking times.

We perform an experiment to demonstrate how the improved processing time for tracking rather than face recognition can be realised in our scheme. We adjust our face tracking scheme to track a face in a frame after every n frames as opposed to every available frame (as we did before). We hypothesise that this tracking scheme will still be able to effectively track the face but use less processing time per second. We note that a value of n is selected with consideration to avoid creating an attack window. This scheme is compared with a continual recognition approach that samples and authenticates every frame. We use the UMDAA and CALF datasets in this experiment (set up with the best performing padding and video sizes in the last experiments). Videos from all sessions are used. We display the results for the tracking duration against the frequency of frames sampled per second is varied in Figure 6. Interestingly, we notice that as the frame frequency and processing time per second decrease, the tracking duration increases. On inspection, this appears to be due to the lower likelihood a frame contains a blur or occlusion that would stop the tracking process. We also show results for the average processing time per second to track a face for the different

		Frame Padding							
		0%		10%		20%		30%	
		Mean Time (s)	Median Time (s)	Mean Time (s)	Median Time (s)	Mean Time (s)	Median Time (s)	Mean Time (s)	Median Time (s)
UMDAA Frame Size (px)	1280×720	2.13	0.05	3.78	0.13	7.62	0.87	14.28	5.50
	960×540	1.73	0.04	2.67	0.04	5.68	0.33	11.19	2.37
	480×270	1.40	0.05	3.59	0.05	13.50	17.41	20.36	16.19
CALF Frame Size (px)	1280×720	4.41	0.18	11.18	1.77	25.48	10.04	39.69	33.48
	960×540	3.31	0.05	8.46	0.66	19.45	5.06	37.28	28.41
	480×270	2.05	0.05	12.47	0.05	38.01	28.05	46.65	47.09

TABLE 5: The mean and median times (in seconds) that faces were tracked in different sized videos from the UMDAA and CALF datasets when different variations of padding are applied.

		Frame Padding							
		0%		10%		20%		30%	
		Track Time (ms)	Auth. Time (ms)	Track Time (ms)	Auth. Time (ms)	Track Time (ms)	Auth. Time (ms)	Track Time (ms)	Auth. Time (ms)
UMDAA Frame Size (px)	1280×720	53	441	72	467	85	489	107	528
	960×540	38	374	42	392	51	394	54	408
	480×270	35	326	35	325	35	325	35	330
CALF Frame Size (px)	1280×720	54	441	75	478	79	505	96	532
	960×540	34	373	37	394	40	400	52	418
	480×270	33	335	33	332	33	329	33	332

TABLE 6: The average processing time for each available frame from live video (30FPS) when tracking and recognition are continuously applied. Results are shown for different frame sizes and different amounts of padding.

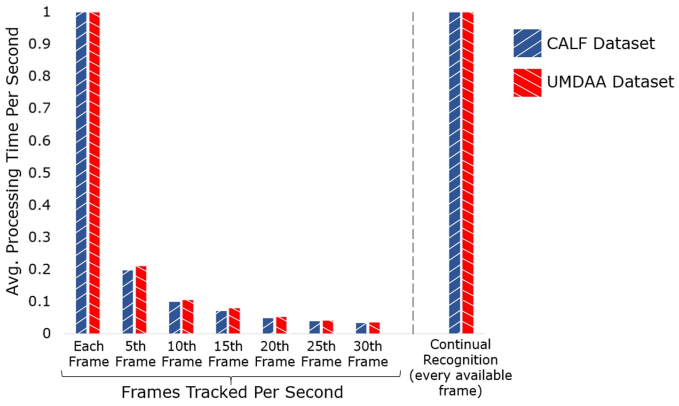

Fig. 7: The time taken to process the frames per second. The figure shows the time taken to track frames for different frame frequencies compared to a continual face recognition scheme that authenticates each frame.

number of frames per second in Figure 7. The figure shows that a face can be tracked with minimal time per second, requiring only ~33ms for tracking every 30th frame. We see the time taken for continual recognition is comparatively inefficient to tracking. From these results we conclude that tracking a face less frequently (e.g.: every 30th frame) increases the efficiency and can improve longevity of authentication.

5 Conclusion

In this paper, we have presented a novel continuous face authentication scheme for mobile devices that incorporates liveness detection to prevent spoof attacks and face tracking to prevent attack windows between re-authentication. We trained SVM classifiers on facial features extracted from genuine and spoofed images such that we can identify live faces. In our face recognition component, live faces were warped to a standardised pose and textual features extracted into a vector and scored using distance algorithms, improving on previous works. We used tracking to show that an authenticated face can be efficiently tracked, removing the need for continual re-authentication or periodic authentication. Our novel dataset was used to show differences in face recognition and tracking when the user performed three different activities.

6 Future Work

Our future work will focus on further enhancing the framework by addressing its current limitations. We will firstly investigate the prospect of enhancing the facial recognition component of our scheme by implementing state-of-the-art convolutional neural networks.

Secondly, in the future we will consider colour and texture information to expand our liveness detection analysis. We will also expand our liveness detection dataset to include spoofing attacks using masks or 3D models of the face in order to evaluate the matching performance of our scheme on unconstrained subjects. Furthermore, we will test this through adapting our 2D-camera based scheme to a 3D-camera based scheme.

We will lastly investigate protocols of dealing with an absence of a fully detectable face. We will look at developing partial face recognition approaches for situations where a face is only partially in the frame.

Acknowledgement

This research work is carried out as part of a research studentship funded by British Telecommunications, UK.

The authors would also like to thank the group members of the Information Security Group (ISG), International Institute of Cavity Research (IICR), and the Erasmus Mundus A2 team, Centre for Software Reliability (CSR), Machine Learning group, Department of Library & Information Science at City, University of London and employees of British Telecommunications at Adastral Park, Ipswich, UK for providing their help in collecting the CALF dataset.

References

[1] A.J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J.M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, Berkeley, CA, USA, 2010, WOOT'10, pp. 1–7, USENIX Association.

[2] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Jan 2013.

[3] R.V. Yampolskiy and V. Govindaraju, "Behavioural biometrics; a survey and classification," *Int. J. Biometrics*, vol. 1, no. 1, pp. 81–113, June 2008.

[4] N.L. Clarke, S. Karatzouni, and S.M. Furnell, "Transparent facial recognition for mobile devices," in *Proceedings of the 7th Security Conference, Las Vegas*, June 2008.

[5] A. Hadid, J.Y. Heikkila, O. Silven, and M. Pietikainen, "Face and eye detection for person authentication in mobile phones," in *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, Sept 2007, pp. 101–108.

[6] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain, "Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data," in *2015 International Conference on Biometrics (ICB)*, May 2015, pp. 135–142.

[7] P. Samangouei, V. M. Patel, and R. Chellappa, "Attribute-based continuous user authentication on mobile devices," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Sept 2015, pp. 1–8.

[8] A. Hadid, "Face biometrics under spoofing attacks: Vulnerabilities, countermeasures, open issues, and research directions," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 113–118.

[9] M. Smith-Creasey and M. Rajarajan, "Adaptive threshold scheme for touchscreen gesture continuous authentication using sensor trust," in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug 2017, pp. 554–561.

[10] Z. Sitov, J. ednka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, May 2016.

[11] A. Primo, V.V. Phoha, R. Kumar, and A. Serwadda, "Context-aware active authentication using smartphone accelerometer measurements," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 98–105.

[12] U. Mahbub and R. Chellappa, "Path: Person authentication using trace histories," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, Oct 2016, pp. 1–8.

[13] M.E. Fathy, V.M. Patel, and R. Chellappa, "Face-based active authentication on mobile devices," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 1687–1691.

[14] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location," *IEEE Systems Journal*, vol. 11, no. 2, pp. 513–521, June 2017.

[15] T. Ahonen, A. Hadid, and M. Pietikinen, *Face Recognition with Local Binary Patterns*, pp. 469–481, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I–511–I–518 vol.1.

[17] P. Samangouei, V.M. Patel, and R. Chellappa, "Facial attributes for active authentication on mobile devices," *Image and Vision Computing*, vol. 58, pp. 181 – 192, 2017.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] H. Zhang, V.M. Patel, and R. Chellappa, "Robust multimodal recognition via multitask multivariate low-rank representations," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. IEEE, 2015, vol. 1, pp. 1–8.

[20] M. Smith-Creasey and M. Rajarajan, "A continuous user authentication scheme for mobile devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Dec 2016, pp. 104–113.

[21] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *International conference on neural information processing*. Springer, 2007, pp. 598–607.

[22] K. Patel, H. Han, and A.K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, 2016.

[23] D. Wen, H. Han, and A.K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[24] D.W. Wagener and B. Herbst, "Face tracking : An implementation of the kanade-lucas-tomasi tracking algorithm," 2016.

[25] K. Mock, B. Hoanca, J. Weaver, and M. Milton, "Real-time continuous iris recognition for authentication using an eye tracker," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 1007–1009.

[26] J.S. Hare, S. Samangooei, and D.P. Dupplaw, "Openimaj and imageterrier: Java libraries and tools for scalable multimedia analysis and indexing of images," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 691–694.

[27] J.M. Saragih, S. Lucey, and J.F. Cohn, "Face alignment through subspace constrained mean-shifts," in *Computer Vision, 2009 IEEE 12th International Conference on*. Ieee, 2009, pp. 1034–1041.

[28] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: a comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 1, pp. 8, 2017.

[29] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Biometrics (IJCB), 2011 international joint conference on*. IEEE, 2011, pp. 1–7.

[30] C. McCool, S. Marcel, A. Hadid, M. Pietikinen, P. Matejka, J. Cernock, N. Poh, J. Kittler, A. Larcher, C. Lvy, D. Matrouf, J. F. Bonastre, P. Tresadern, and T. Cootes, "Bi-modal person recognition on a mobile phone: Using mobile phone data," in *2012 IEEE International Conference on Multimedia and Expo Workshops*, July 2012, pp. 635–640.

[31] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*. IEEE, 1994, vol. 1, pp. 582–585.

[32] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[33] J. Heo and M. Savvides, "Face recognition across pose using view based active appearance models (vbaams) on cmu multi-pie dataset," *Computer vision systems*, pp. 527–535, 2008.

[34] A. Asthana, T.K. Marks, M.J Jones, K.H. Tieu, and M.V. Rohith, "Fully automatic pose-invariant face recognition via 3d pose normalization," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 937–944.

[35] Z. Yang, M. Li, and H. Ai, "An experimental study on automatic face gender classification," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 3, pp. 1099–1102.

[36] K.S. Killourhy and R.A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*. IEEE, 2009, pp. 125–134.