sensors

MDPI

Article

Integrating Gaze Tracking and Head-Motion Prediction for Mobile Device Authentication: A Proof of Concept

Zhuo Ma [1,2,*,†], Xinglong Wang [1,*,†], Ruijie Ma [1], Zhuzhu Wang [3] and Jianfeng Ma [1,2]

1 School of Cyber Engineering, Xidian University, Xi'an 710071, China; JMaris4733865@gmail.com (R.M.); jfma@mail.xidian.edu.cn (J.M.)
2 Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China
3 ZTE Corporation, Xi'an 710114, China; wang.zhuzhu@zte.com.cn
* Correspondence: mazhuo@mail.xidian.edu.cn (Z.M.); wxingl1994@163.com (X.W.); Tel.: +86-133-7926-8899 (Z.M.)
† These authors contributed equally to this work.

Received: 24 June 2018; Accepted: 28 August 2018; Published: 31 August 2018

Abstract: We introduce a two-stream model to use reflexive eye movements for smart mobile device authentication. Our model is based on two pre-trained neural networks, *iTracker* and *PredNet*, targeting two independent tasks: (i) gaze tracking and (ii) future frame prediction. We design a procedure to randomly generate the visual stimulus on the screen of mobile device, and the frontal camera will simultaneously capture head motions of the user as one watches it. Then, *iTracker* calculates the gaze-coordinates error which is treated as a *static feature*. To solve the imprecise gaze-coordinates caused by the low resolution of the frontal camera, we further take advantage of *PredNet* to extract the *dynamic features* between consecutive frames. In order to resist traditional attacks (shoulder surfing and impersonation attacks) during the procedure of mobile device authentication, we innovatively combine *static features* and *dynamic features* to train a 2-class support vector machine (SVM) classifier. The experiment results show that the classifier achieves accuracy of 98.6% to authenticate the user identity of mobile devices.

Keywords: smart mobile devices; gaze tracking; head motions; authentication; neural networks

1. Introduction

In the era of the Mobile Internet, a large amount of private information is stored in smart mobile devices [1–6], which make the authentication of users a vital precondition of the secure access to the sensitive data. A traditional password authentication method has some negative characteristics such as shoulder-surfing [7]. In later studies, several biometric techniques have been applied to user authentication. For example, Sin et al. proposed fingerprint recognition systems based on template updating [8]; Lin et al. utilized the robust thin plate spline (RTPS) to achieve the user identification [9]; Parkhi et al. [10] and Ning et al. [11] developed face recognition methods using Convolutional Neural Network (CNN) architectures and biomimetic uncorrelated locality discriminant projection (BULDP), respectively. Although fingerprint and face recognition methods can defend shoulder-surfing, impersonation attacks still exist [12].

Recently, eye tracking has been used in some emerging fields such as human–computer interaction [13–16] and computer vision [17,18] as an important technique across many domains with a series of decent research results. Among those results, an authentication method exploiting gaze-based information is easy to implement relying on the high precision of dedicated devices. Meanwhile, some researchers [19] used stereoscopic views with multiple devices or light sources to

achieve eye tracking, whereas, it is difficult to apply these equipment-based eye tracking methods to mobile devices.

In view of the above questions, this paper proposes a novel mobile device authentication method which integrates the gaze tracking and head-moving prediction. The whole procedure of our method is shown in Figure 1. First, the smart mobile device randomly shows an interactive visual stimulus on the screen, which is referred to as *gaze–challenge*. Simultaneously, the front camera captures head motions of the user as he/she watches the screen. Then, this system adopts two kinds of deep neural networks to extract gaze features that can be divided into two categories, one using a convolutional neural network to extract *static features* that rely on patterns where the user is looking, and another using a recurrent convolutional neural network to extract *dynamic features* that depend on patterns of how the user's head is moving. Finally, the smart mobile device combines *static features* and *dynamic features* that are used to train a dedicated two-class Support Vector Machine (SVM) classifier and make a decision about the user's identity which is referred to as *gaze–response*.

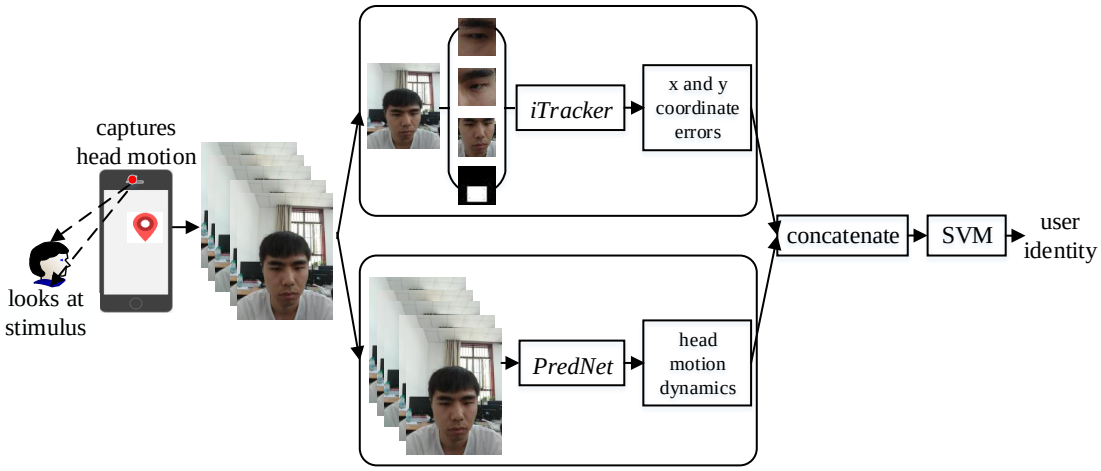


Figure 1. Authentication workflow.

In the process of feature extraction, we consider using the *iTracker* [20] to calculate the error between the prediction and the ground truth of gaze coordinates cast on the screen. It can provide a tiny circle whose radius is the value of error, and which is surrounded around the visual stimulus to determine the movement of gaze. Obviously, the random stimulus trajectory ensures that the gaze maintains freshness to avoid the impersonation attack. In other words, if there are some predicted coordinates beyond the tiny area, the gaze might be invalid or even fake. Therefore, we refer to the two errors of abscissa and ordinate, respectively, as *static features*.

Static features give a fresh gaze trajectory but is weak at providing the information of identification. To enhance the features of identification when the user's head moves along with the visual stimulus, we adapt *PredNet* [21] to extract dynamic representation that is used to determine the user's identity. *PredNet* [21] applies the existing video frame sequence to predicting the future frame. The *Representation* module extracts the *dynamic features* from consecutive frames, so we utilize *dynamic features* to identify valid users. In conclusion, integrating *static features* and *dynamic features* not only defends the impersonation attacks but also provides an efficient identity recognition.

2. Related Work

Human visual system is reflexive and fast, and different people possess different gaze information, which has attracted great attention in the biometric authentication field. Over the past two decades, plenty of gaze tracking methods have emerged and been used in the medical field, but their utilization in attack detection and authentication field has rarely been found until recent years. Two attack

scenarios were considered by Komogortsev et al. [22] of whether the imposter had access to the biometric database. Their results suggested that eye movement biometrics were highly resistant to circumvention by artificial recordings. A novel method for face liveness detection by tracking the gaze of the user with an ordinary webcam was proposed by Ali et al. [23] to resist spoofing attacks on biometric systems. A summary of works on the authentication methods and systems previous to 2010 were given by Zhang et al. [24], while the methods and results of gaze tracking authentication systems provided in works of recent years were compared carefully in Saeed [25]. Zhang et al. [26–28] presented a person-independent eye gaze interface that immediately supported spontaneous interaction with displays, without any prior user calibration or training. In their work, localisation of inner eye corners and eye centres was used to realize calibration-free interaction and gaze tracking. Moreover, the existing works can be classified into two types, of which one takes gaze tracking as a human–computer interaction interface and utilizes gaze patterns as Personal Identification Numbers (PINs) to access authentication systems, and the other exploits the classification results concluded by universal gaze features that are extracted from several gaze patterns of individual people to differentiate one user from another.

The first type of authentication method needs users to stare at the screen so that the secret information can be covertly inputted in a natural way, where the secret information is usually in the form of passwords [29–31] or other distinct ways [32–34]. The work of Chen et al. [35] belongs to the first method, taking gaze tracking as security primitives. It endows authentication systems with some additional advantages including but not limited to the protection from shoulder-surfing and smudge attacks. However, this authentication method is still in need of some memorization of secret information and as a result suffers from some latent attacks such as reply attacks.

The work of Sluganovic et al. [36] belongs to the second method, making use of the different biometric gaze features of different people to distinguish different users. An authentication system equipped with this method is more friendly to use since the users no need to remember any secret information, and can also defend impersonation attacks due to the unique gaze characteristics. Nowadays, the existing works using this method always have a high cost because of the use of expensive invasive gaze tracking devices. Consequently, we aim to make the gaze tracking authentication approach a more pervasive and available technology by only using the front camera in smart phones rather than high cost gaze tracking devices. In the rest of this paper, we will introduce our authentication procedure in detail.

3. Authentication Procedure

The authentication approach is summarized in Figure 1 inspired by [37–40]. We can divide the work into five subsections: (i) generating interactive visual stimulus; (ii) preprocessing head-moving frames; (iii) extracting *static features*; (iv) extracting *dynamic features* and (v) classifying user's identity. The individual pieces are described in turn in the following subsections.

3.1. Generating Interactive Visual Stimulus

The smart mobile phone shows a randomly interactive visual stimulus on the screen. The stimulus' motions should conform to the moving habits of human eyes. Therefore, we start with a short background of the human visual system. Even when one's gaze is firmly fixated on a single stimulus, human eyes are never completely still. They are constantly making hundreds of micro movements per second, which are interlaced with more than 100,000 larger movements during the course of one day [41]. During visual tasks, such as search or scene perception, our eyes alternate between *fixations* and *saccades*. *Fixations* are used to maintain the visual focus on a single stimulus, while *saccades* reorient the eye to focus the gaze on the next desired position [36].

Inspired by the above description, we design a visual stimulus icon that can rotate around the vertical central axis in Figure 2. Instead of showing a still icon, we show a rotated icon when it stays at a fixed position on the screen, which directs the fixation of the gaze to lie in the middle of the icon.

The white central circle of icon is different from the red surroundings, which can also attract the user's attention on the center point of the icon.

Figure 2. Visual stimulus icon.

The stimulus movements are represented in Figure 3. First of all, to avoid distraction from notifications, we ensure that the user applies *Airplane Mode* with no network connection throughout the task, until the task is complete. The icon rotates at a random position in the beginning and then moves to the next position with a random orientation, each of the motions takes 2 s and alternates. We start the recording after 1 s; in this way, the mussy gaze scanning in the first second can be wiped off. Moreover, the recording frequency is 25 fps. Last but not least, the user needs to ensure that his/her face is visible in the front camera. This is critical as we do not hope to track where someone is looking without a picture of the face. For portions of Android layouts, please refer to Figure A1 in Appendix A in detail.

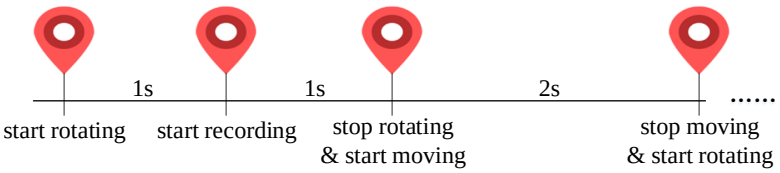
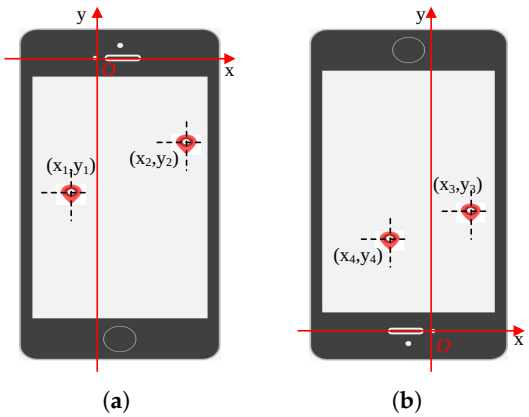


Figure 3. Temporal movements of the stimulus.

When eyes fixate on the rotated stimulus at a fixed position on the screen, we predict the gaze coordinates relative to the front camera taken as the ordinate origin that are shown in Figure 4. In Figure 4a, (x_1, y_1) is located in the third quadrant and (x_2, y_2) is located in the second quadrant, so $x_1, y_1, y_2 < 0$ and $x_2 > 0$. An inverted screen is represented in Figure 4b, (x_3, y_3) is located in the first quadrant and (x_4, y_4) is located in the fourth quadrant, so $x_3, y_3, y_4 > 0$ and $x_4 < 0$. It is also beneficial to keep data variability in the pictures when users change the orientation of their mobile devices to be inverted.



(**a**) (**b**)

Figure 4. The coordinate axis takes the front camera as the ordinate origin. (**a**) Normal orientation of smart phone; (**b**) Orientation of smart phone is inverted.

3.2. Preprocessing Head-Moving Frames

During the preprocessing duration, we crop the image of eyes and faces to satisfy the input demand of the neural network in Section 3.3 by adopting the interface provided by OpenCV. After preprocessing, we can obtain the following images shown in Figure 5. We need to emphasize face grid that is a binary mask used to indicate the location and size of the head within the original image while the face, left eye and right eye are simply detected and cropped from the original image.

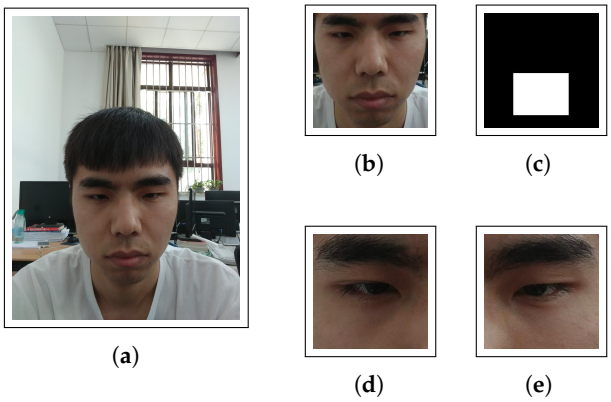


Figure 5. Image data after preprocessing to satisfy the input demand of the neural network in the next subsection; (**a**) original image; (**b**) face; (**c**) face grid; (**d**) right eye; (**e**) left eye.

3.3. Extracting Static Features

iTacker [20] is an end-to-end CNN for robust eye tracking shown in Figure 6. Inputs include right eye, left eye and face images of size 224×224 and face grid of size 25×25. In addition, the distance between the user and the smart mobile device can be measured and calculated by *iTracker* because the area of the face grid will decrease when the subject is far away from the screen. Otherwise, the area will increase. The relationship between them is shown in Table 1. It can be seen that there is a linear relationship between the distance from head to screen and the area of the face grid. Therefore, we can infer the position of the head based on the size of the face grid. On this basis, we can perform a calibration of distance using the face grid information. In addition, it can enhance the variety of the dataset when the subject observes the stimulus from different distances. Thus, we adopt the various distance samples to *iTracker* to fine-tune a relatively advantageous calibration effect.

Table 1. Relationship between distances from head to screen and the black area (area of the face grid is always 25×25 pixels).

Distance from head to screen (cm)	10	20	30	40
Area of black	$\approx 16 \times 16$	$\approx 12 \times 12$	$\approx 8 \times 8$	$\approx 4 \times 4$

Parameters of each convolutional layer are shown in the Table 2 and the number of neurons of each full-connected layer are shown in Table 3. The original output is the Euclidean distance, in centimeters, from the front camera. However, we adapt the output to distances between the predicted gaze coordinate and the ground truth along the x-axis and y-axis respectively, which act as *static features*.

Table 2. Parameters of each convolutional layer.

Name	Filter Size	Number of Kernel
CONV-E1	11×11	96
CONV-F1	11×11	96
CONV-E2	5×5	256
CONV-F2	5×5	256
CONV-E3	3×3	384
CONV-F3	3×3	384
CONV-E4	1×1	64
CONV-F4	1×1	64

Table 3. Number of neurons of each full-connected layer.

Name	Number of Neurons
FC-E1	128
FC-F1	128
FC-F2	64
FC-FG1	256
FC-FG2	128
FC1	128
FC2	2

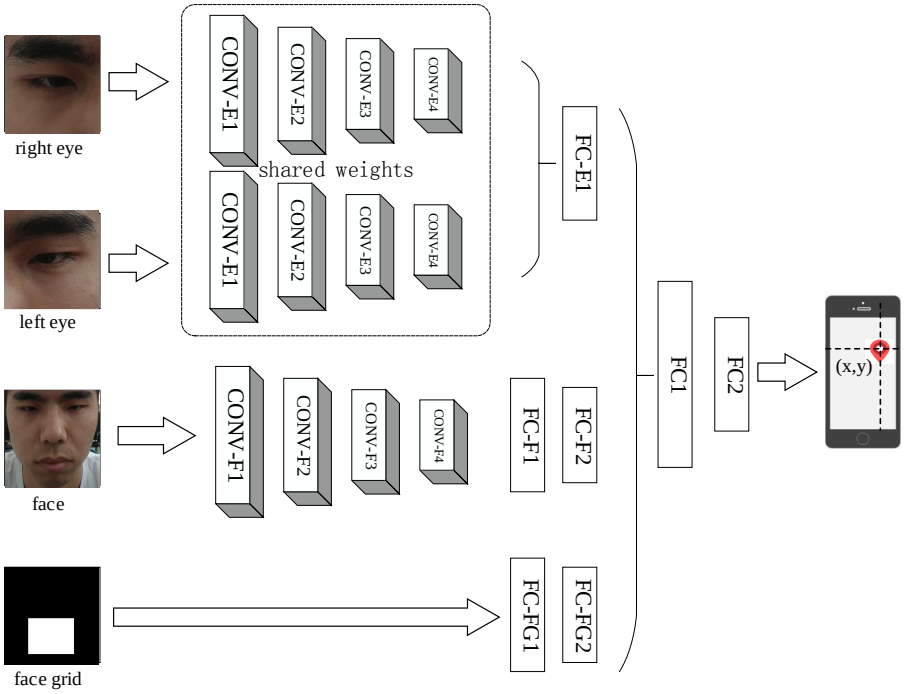


Figure 6. Convolutional neural network architecture named *iTracker* [20].

3.4. Extracting Dynamic Features

PredNet [21] can predict the future frames in a video sequence by learning about the structure of the visual world shown in Figure 7. We review the relations between each modules of *PredNet* [21] in the following formulas (x_t denotes a sequence of images, t denotes the time, and l denotes the layer):

$$A_l^t = \begin{cases} x_t, & \text{if} \quad l = 0, \\ MaxPool\left(ReLU\left(Conv\left(E_{l-1}^t\right)\right)\right), & \text{if} \quad l > 0, \end{cases} \tag{1}$$

$$\hat{A}_l^t = \begin{cases} SatLU\left(ReLU\left(Conv\left(R_l^t\right)\right)\right), & \text{if} \quad l = 0, \\ ReLU\left(Conv\left(R_l^t\right)\right), & \text{if} \quad l > 0, \end{cases} \tag{2}$$

$$E_l^t = \left[ReLU\left(A_l^t - \hat{A}_l^t\right); ReLU\left(\hat{A}_l^t - A_l^t\right)\right], \tag{3}$$

$$R_l^t = ConvLSTM\left(E_l^{t-1}, R_l^{t-1}, UpSample\left(R_{l+1}^t\right)\right). \tag{4}$$

We adopt a 4-layer *PredNet* [21] model to the consecutive head-moving frames and illustrate five comparisons in Figure 8 and then calculate the difference between ground truth and prediction shown in Figure 9. It can be inferred from Figure 9 that the prediction is more approximate to the ground truth as time goes by. As for the representation module in *PredNet* [21], whose tensor field was proved to generalize well to other classification tasks, we can utilize this dynamics tensor coding as *dynamic features* to identify the valid user and complete the authentication task. Our representation module of *PredNet* learns to predict future frames in *dynamic features* of a video sequence. Each layer in the module makes local predictions and only forwards the differences between predictions and ground truth from those predictions to subsequent network layers. The module learns internal representations that are useful for decoding latent object parameters (e.g., head motion) that support object recognition with fewer training views. To possess more knowledge of the representation module, we refer to [42–46] to learn that the representation module acts as an upsampling method to reconstruct the scale of feature maps and identifies with a deconvolutional network to revivify the original images. Therefore, we visualize a second layer of a representation module of Group 1 data in Figure 10, we can see: the head turns right a little bit from the a channel to b channel, the squint of head motion appears in the c channel, and an approximate intensity simulation exists in the d channel.

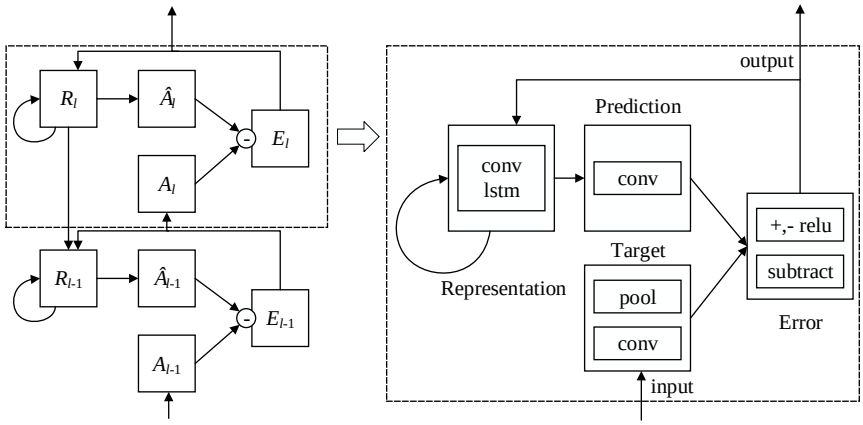


Figure 7. Recurrent convolutional neural network architecture named *PredNet* [21].

Figure 8. Comparisons between ground truth and prediction.

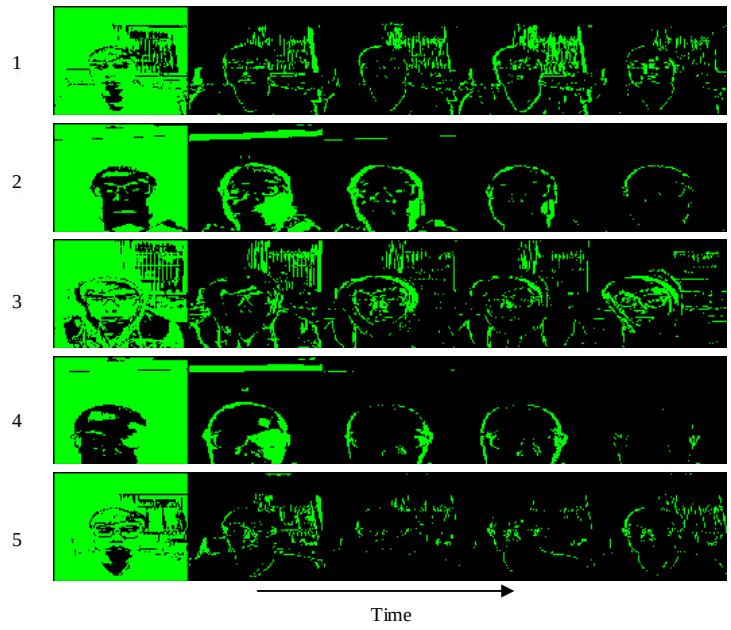


Figure 9. Difference between ground truth and prediction. The green color denotes that the absolute value of subtraction is greater than the threshold and the black region corresponds to the absolute value of subtraction being less than the threshold. The threshold is set as 0.1.

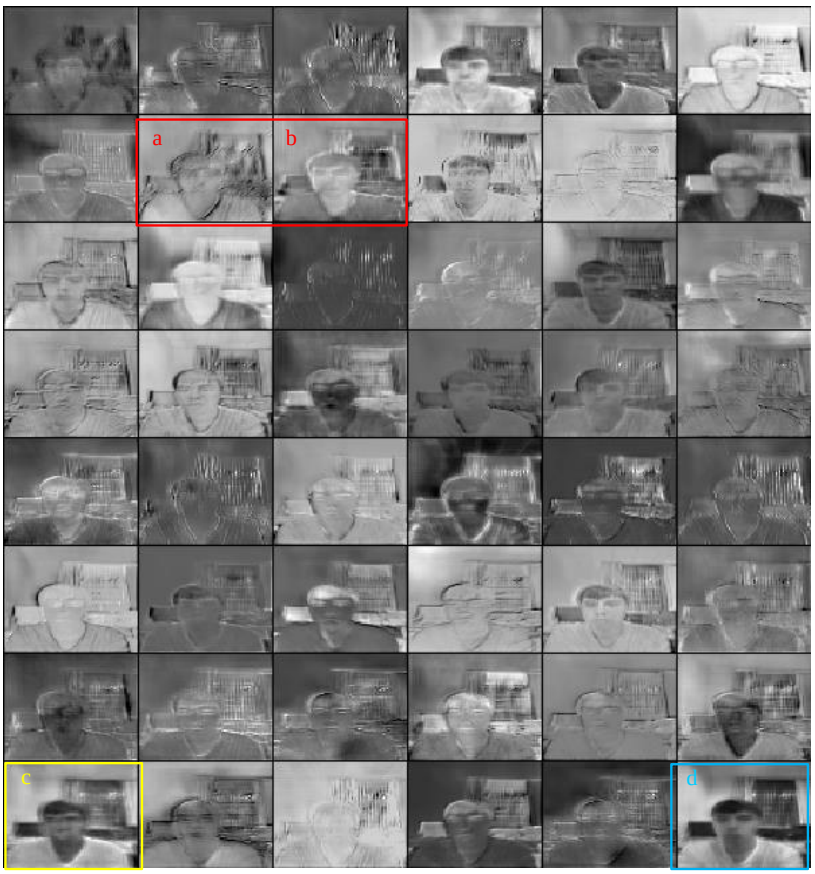


Figure 10. Visualization of the different channels of a representation module.

3.5. Classifying User's Identity

Static features and *dynamic features* are concatenated and then labelled in line with the user's identity. Finally, features and labels are fed into an SVM [47] algorithm to train a dedicated classifier to identify the valid user. For head-moving samples, please refer to Figure A2 in Appendix A. We adopt the interface provided by scikit-learn [48,49] to model a two-class classifier by cross-validation and fit the test dataset. It is critical for the valid user to participate more in data collection to solve the data-imbalanced problem. The detailed experiment results are shown in Section 4.

4. Results

There are 37 distinct participants (26 males and 11 females, 14 teachers and 23 students) aged between 22–38 years (mean = 26.5, std = 3.4) that are involved in our experiments. Among them, 36 participants belong to invalid users and one participant is the valid user because each phone device generally has one owner. The facial videos are recorded with 25 fps using the mobile phone. Each invalid user repeats the experiment four times and each experiment produces a sequence of 500 frames. That is, the length of the collection time for each sequence lasts 20 s. In order to solve the imbalance problem of the dataset, the valid user repeats the experiment 144 times. Therefore, the dataset consists of 288 groups of the frame sequence. A sequence is labelled 1.0 if it belongs to the valid user; otherwise, it is labelled -1.0.

In order to ensure the reliability of the experimental results, we adopted the three-fold cross-validation. Specifically, we first randomly split the whole dataset into four equally sized segments, i.e., the training set is 216 and the test set is 72. Three segments are used to train a model and the rest is employed to test it. In the procedure of cross-validation, the training and testing datasets must be crossed over in successive rounds such that each frame sequence trial has a chance of being invalidated. It is reasonable to adopt a grid search to find the best hyper-parameters of SVM [47] shown in Table 4. It takes 3.5 s to finish feature processing, and, in some significant authentication systems, which care more about attacks by imposters such as shoulder surfing and impersonation attacks, the time consumption is absolutely tolerable.

Our experiments show that hyper-parameters of the best classifier are *kernel* = 'linear' and $C = 0.001$. Therefore, we apply this classifier to the test set and obtain the following confusion matrix shown in Table 5.

Table 4. Hyper-parameters of SVM.

Hyper-Parameter	Explanation	List of Values
kernel	kernel function	['rbf', 'linear']
C	cost parameter	[1 $\times 10^{-07}$, ..., 1, 10, 100]
gamma	hyper-parameter of RBF (only if kernel = 'rbf')	[1 $\times 10^{-05}$, ..., 1, 10, 100]

Table 5. Confusion matrix.

ground truth \ Prediction	**1.0**	**−1.0**
1.0	37	0
−1.0	1	34

According to the confusion matrix, we can infer that the *TP* (true positive) is 37, *FP* (false positive) is 1, *FN* (false negative) is 0 and *TN* (true negative) is 34. Finally, we calculate the *accuracy*, *precision*, *recall*, *f1-score*, *AUC* and plot the *ROC* curve to report the classification performance:

1. *Accuracy* is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Based on Equation (5), *accuracy* of our classifier is 0.986:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5}$$

2. *Precision* is the ratio of correctly predicted positive observations to the total predicted positive observations. Based on Equation (6), *precision* of our classifier is 0.97:

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

Recall is the ratio of correctly predicted positive observations to the all observations in true class. Based on Equation (7), *recall* of our classifier is 1.0:

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

F1-score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Based on Equations (8) and (9), *f1-score* of our classifier is 0.99:

$$\frac{2}{F1} = \frac{1}{P} + \frac{1}{R'} \tag{8}$$

$$\Rightarrow F1 = \frac{2TP}{2TP + FP + FN}. \tag{9}$$

A heat map is plotted in Figure 11 to represent the above measures clearly.

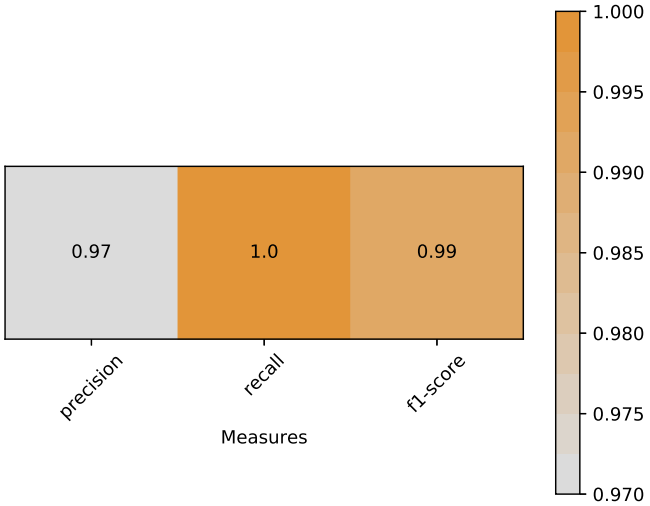


Figure 11. Heat map of measures include *precision*, *recall* and *f1-score*.

3. *ROC* curves typically feature a true positive rate (Equation (10)) on the *y*-axis, and the false positive rate (Equation (11)) on the *x*-axis. This means that the top left corner of the plot is the ideal point where a false positive rate of zero and a true positive rate of one. It does mean that a larger area

under the curve (*AUC*) is usually better. We plot the *ROC* curve of our classifier in Figure 12. The area of *AUC* is labelled in the bottom right corner:

$$TPR = \frac{TP}{TP + FN}, \tag{10}$$

$$FPR = \frac{FP}{FP + TN}. \tag{11}$$

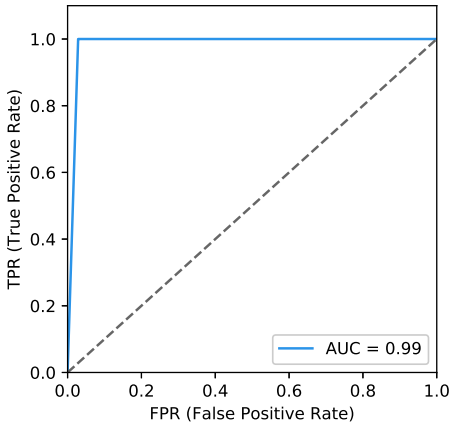
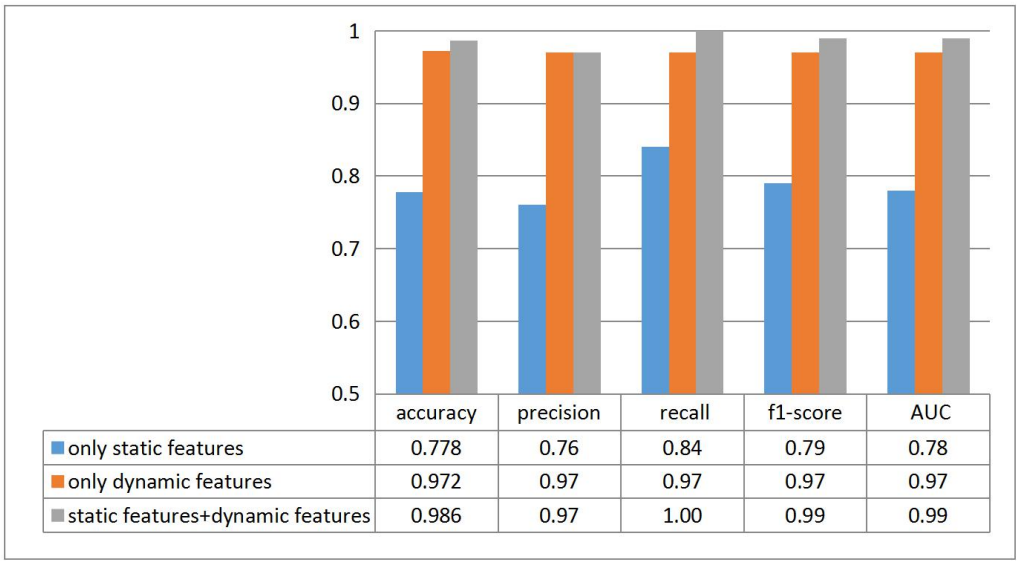


Figure 12. *ROC* curve.

To observe the contribution of different features, we experiment on both single and combined features, the results are shown in Figure 13. It is discriminative that *dynamic features* are more helpful than *static features* and the combined features cover the benefits of both.



	accuracy	precision	recall	f1-score	AUC
only static features	0.778	0.76	0.84	0.79	0.78
only dynamic features	0.972	0.97	0.97	0.97	0.97
static features+dynamic features	0.986	0.97	1.00	0.99	0.99

Figure 13. Contributions of different types of features.

To validate that the SVM [47] algorithm we choose in the work is ideal, we also check other classification algorithms and represent the results in Figure 14. It is obvious that SVM possesses the optimal performance while the Random Forest and AdaBoost provide the same statistics on the test set.

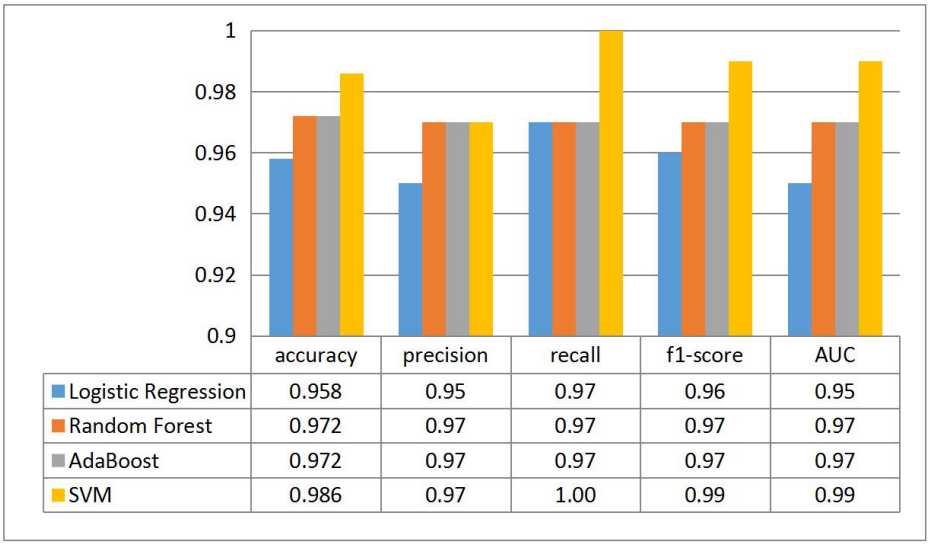


Figure 14. The effects of different classification algorithms.

Finally, we compare our method with other pervasive authentication methods in Table 6. Of course, it is injudicious to use False Positive Rate (FPR) and False Negative Rate (FNR) to compare our method with a password and increasingly accurate fingerprints and face recognition. As we all know, we can enter the authentication system effortlessly if we remember the login password, so both FPR and FNR of passwords are 0.00. In particular, we adopt our dataset to state-of-the-art fingerprints [8] and the face recognition [10] algorithm. It is noticeable that our method is able to resist shoulder surfing and impersonation attacks, and achieve an acceptable result in terms of accuracy at the same time.

We have a simple test that image and video data of 16 students are trained respectively by the deep face recognition [10] and our authentication method. As can be seen in Table 7, 16 videos of the corresponding participants recorded in advance are used to find out whether the impersonation can be detected by these two methods. The error rate of 68.75% indicates that general face recognition methods have difficulty dealing with impersonation attacks. The random gaze–challenge and the corresponding gaze–response of our method guarantee authentic human behaviors. Thus, the videos recorded in advance can be detected. As a result, the novel authentication method we proposed combines gaze tracking and head-moving prediction to determine user identity. We provide higher security since the traditional biometric authentication methods are easily cheated by impersonation attacks.

Table 6. Comparison between different authentication methods.

Authentication Method	Accuracy	FPR	FNR	Resist Shoulder Surfing?	Resist Impersonation Attacks?
Password	100.0%	0.00	0.00	No	No
Fingerprint [8]	98.6%	0.00	0.01	Yes	No
Face recognition [10]	100.0%	0.00	0.00	Yes	No
Our method	98.6%	0.01	0.00	Yes	Yes

Table 7. A test against the impersonation attack.

Authentication Method	Accepted Impersonation	Rejected Impersonation	Error Rate
Deep face recognition [10]	11	5	68.75%
Our method	0	16	0.00%

5. Conclusions

In this paper, we proposed a novel method to use reflexive eye movements for smart mobile device authentication. Inspired by a two-stream neural network that has become a pervasive domain in recent years, we utilize *iTracker* [20] to extract the location of gaze and *PredNet* [21] to extract the dynamics of head motions when users are tracking the randomly interactive visual stimulus. Due to the fact that human eyes are fast, reflexive, responsive, and carry information unique to other individuals, attempting to apply gaze patterns to authentication is particularly attractive. Moreover, dynamics of head motions facilitate the authentication system to become more robust.

In the experiment, 288 groups of frame sequence data were collected from all the students and teachers in our lab. Indeed, we know that the insufficient sample data were not conducive to the accuracy of the model, but the main contributions of this paper are to guarantee the authentic human behaviors to resist impersonation attacks, and elucidate a proof of concept prototype implementing the fundamental concepts of a new authentication method by integrating gaze tracking and head-motion prediction. Using *static features* and *dynamic features* improves the recognition rate in spite of the small sample data set. The accuracy of our method is 98.6% by the complementary features provided by *iTracker* [20] and *PredNet* [21]. Furthermore, simplifying the input format and giving a deeper understanding of features extracted by the architecture are important future directions.

Author Contributions: Conceptualization, Z.M. and X.W.; Formal analysis, X.W.; Methodology, X.W.; Project administration, Z.M.; Validation, R.M. and Z.W.; Writing—Original draft, Z.M. and X.W.; Writing—Review and editing, Z.M., X.W. and J.M.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. U1764263, U1405255, 61872283), the Natural Science Basic Research Plan in the Shaanxi Province of China (Grant No. 2016JM6074), and the Shaanxi Science and Technology Coordination and Innovation Project (Grant No. 2016TZC-G-6-3).

Acknowledgments: The authors would like to thank the editor and the anonymous referees for their constructive comments.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support Vector Machine
CNN	Convolutional Neural Network
PINs	Personal Identification Numbers
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
AUC	Area Under Curve
ROC	Receiver Operating Characteristic Curve
TPR	True Positive Rate
FPR	False Positive Rate
FNR	False Negative Rate

Appendix A

Figure A1 represents portions of Android layouts.

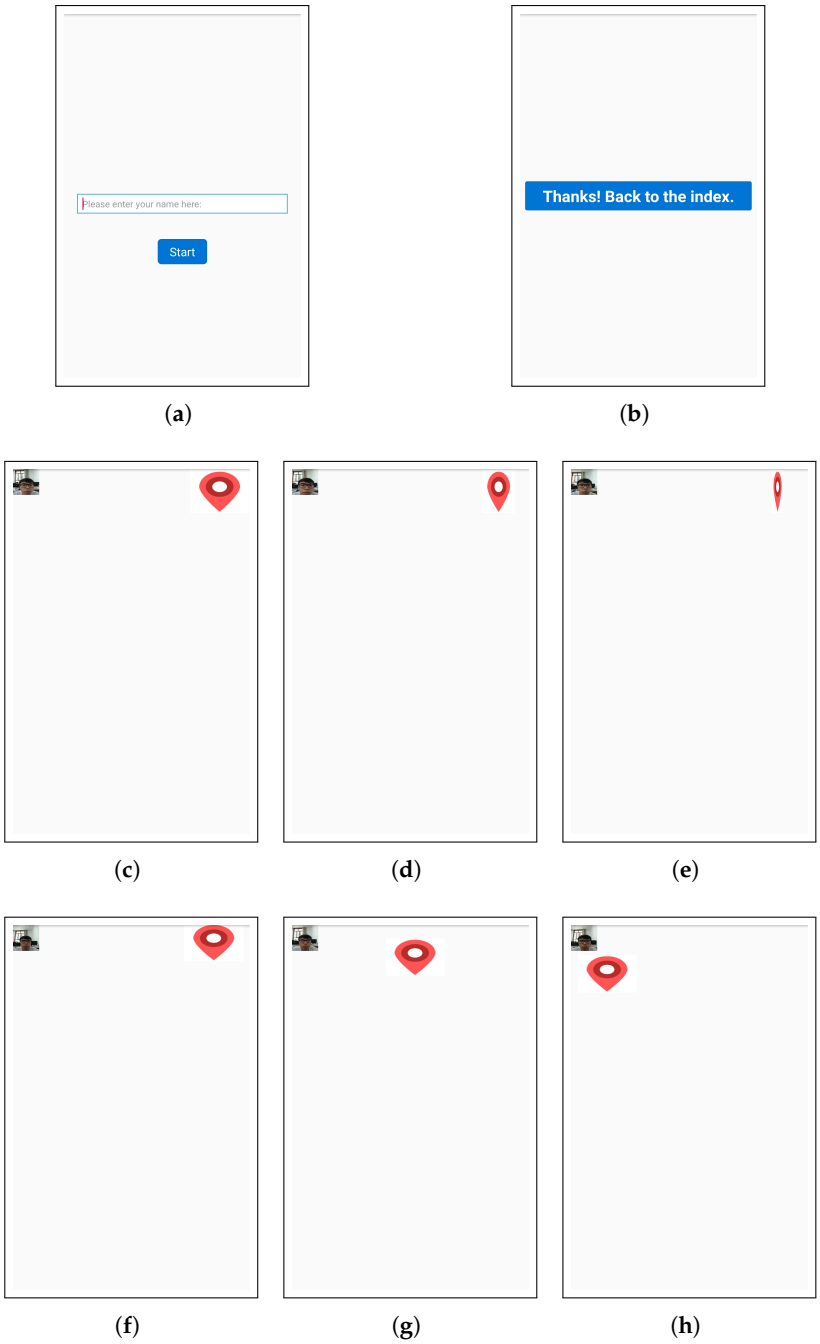


Figure A1. Portions of Android layouts. (**a**) instructs the user to input his/her name and start paying attention and fixating on the stimulus icon. (**b**) shows the end of the procedure. (**c**–**e**) shows the rotation procedure at three discrete time points. (**f**–**h**) shows the movement procedure in three discrete time points. Moreover, in order to help the user ensure that his/her face can be captured by the front camera, we put a *SurfaceTexture* component in the top left corner of the screen.

Figure A2 shows a part of the consecutive head-moving sequence samples.

Figure A2. Sample frames from the whole dataset.

References

1. Wu, D.; Yan, J.; Wang, H.; Wu, D.; Wang, R. Social Attribute aware Incentive Mechanisms for Video Distribution in Device-to-Device Communications. *IEEE Trans. Multimedia* **2017**, *8*, 1908–1920.
2. Wu, D.; Liu, Q.; Wang, H.; Wu, D.; Wang, R. Socially Aware Energy Efficient Mobile Edge Collaboration for Video Distribution. *IEEE Trans. Multimedia* **2017**, *10*, 2197–2209.
3. Wu, D.; Si, S.; Wu, S.; Wang, R. Dynamic Trust Relationships Aware Data Privacy Protection in Mobile Crowd-Sensing. *IEEE Internet Things J.* **2017**, *5*, 2958–2970.
4. Jiang, Q.; Chen, Z.; Li, B.; Shen, J.; Yang, L.; Ma, J. Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. *J. Am. Intell. Humaniz. Comput.* **2017**, *9*, 1–13.
5. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392.
6. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S.A. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195.
7. Raza, M.; Iqbal, M.; Sharif, M.; Haider, W. A survey of password attacks and comparative analysis on methods for secure authentication. *World Appl. Sci. J.* **2012**, *19*, 439–444.
8. Sin, S.W.; Zhou, R.; Li, D.; Isshiki, T.; Kunieda, H. Narrow fingerprint sensor verification with template updating technique. *IEICE Trans. Fundam.* **2012**, *95*, 346–353.
9. Lin, C.; Kumar, A. Matching Contactless and Contact-Based Conventional Fingerprint Images for Biometrics Identification. *IEEE Trans. Image Process.* **2018**, *4*, 2008–2021.
10. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep Face Recognition. In Proceedings of the British Machine Vision Conference, Swansea, UK, 10 September 2015; pp. 41.1–41.12.

11. Ning, X.; Li, W.; Tang, B.; He, H. BULDP: Biomimetic Uncorrelated Locality Discriminant Projection for Feature Extraction in Face Recognition. *IEEE Trans. Image Process.* **2018**, *27*, 2575–2586
12. Delac, K.; Grgic, M. A survey of biometric recognition methods. In Proceedings of the Elmar 2004. 46th International Symposium Electronics in Marine, Zadar, Croatia, Croatia, 18–18 June 2004; pp. 184–193.
13. Jacob, R.J.; Karn, K.S. Eye tracking in human–computer interaction and usability research: Ready to deliver the promises. *Mind Eye* **2003**, 573–605, doi:10.1016/B978-044451020-4/50031-1.
14. Majaranta, P.; Bulling, A. Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing*; Springer: London, UK, 2014; pp. 39–65.
15. Morimoto, C.H.; Mimica, M.R. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* **2005**, *98*, 4–24.
16. Lv, Z.; Zhang, C.; Zhou, B.; Gao, X.; Wu, X. Design and implementation of an eye gesture perception system based on electrooculography. *Expert Syst. Appl.* **2018**, *91*, 310–321.
17. Karthikeyan, S.; Jagadeesh, V.; Shenoy, R.; Ecksteinz, M.; Manjunath, B.S. From where and how to what we see. In Proceedings of the 2013 IEEE International Conference on Computer Vision , Karlsruhe, Germany, 6–10 May 2013; Volume 10, pp. 625–632.
18. Borji, A.; Itti, L. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 185–207.
19. Hansen, D.; Ji, Q. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *3*, 478–500.
20. Krafka, K.; Khosla, A.; Kellnhofer, P.; Kannan, H.; Bhandarkar, S.; Matusik, W.; Torralba, A. Eye tracking for everyone. *arXiv* **2016**, arXiv:1606.05814.
21. Lotter, W.; Kreiman, G.; Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv* **2016**, arXiv:1605.08104.
22. Komogortsev, O.V.; Karpov, A.; Holland, C.D. Attack of mechanical replicas: Liveness detection with eye movements. *IEEE Trans. Inf. Forensics Secur.* **2015**, *4*, 716–725.
23. Ali, A.; Deravi, F.; Hoque, S. Spoofing attempt detection using gaze colocation. In Proceedings of the 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG), Darmstadt, Germany, 5–6 September 2013; pp. 1–12.
24. Zhang, Y.; Chi, Z.; Feng, D. An Analysis of Eye Movement Based Authentication Systems. In Proceedings of the International Conference on Mechanical Engineering and Technology, London, UK, 24–25 November 2011.
25. Saeed, U. Eye movements during scene understanding for biometric identification. *Pattern Recognit. Lett.* **2016**, *82*, 190–195.
26. Zhang, Y.; Bulling, A.; Gellersen, H. Towards pervasive eye tracking using low-level image features. In Proceedings of the Symposium on Eye Tracking Research and Applications, Santa Barbara, CA, USA, 28–30 March 2012; pp. 261–264.
27. Zhang, Y.; Bulling, A.; Gellersen, H. SideWays: A gaze interface for spontaneous interaction with situated displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013; pp. 851–860.
28. Zhang, Y.; Bulling, A.; Gellersen, H. Pupil-canthi-ratio: A calibration-free method for tracking horizontal gaze direction. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, Como, Italy, 27–29 May 2014; pp. 129–132.
29. Kumar, M.; Garfinkel, T.; Boneh, D.; Winograd, T. Reducing shoulder-surfing by using gaze-based password entry. In Proceedings of the 3rd symposium on Usable privacy and security, Pittsburgh, PA, USA, 18–20 July 2007; pp. 13–19.
30. Weaver, J.; Mock, K.; Hoanca, B. Gaze-based password authentication through automatic clustering of gaze points. In Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 2749–2754.
31. Bulling, A.; Alt, F.; Schmidt, A. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 3011–3020.
32. Boehm, A.; Chen, D.; Frank, M.; Huang, L.; Kuo, C.; Lolic, T.; Martinovic, I.; Song, D. Safe: Secure authentication with face and eyes. In Proceedings of the 2013 International Conference on Privacy and Security in Mobile Systems, Atlantic City, NJ, USA, 24–27 June 2013; pp. 1–8.

33. De Luca, A.; Denzel, M.; Hussmann, H. Look into my eyes!: Can you guess my password? In Proceedings of the 5th Symposium on Usable Privacy and Security, Mountain View, CA, USA, 15–17 July 2009; p. 7.
34. Kocejko, T.; Wtorek, J. Gaze pattern lock for elders and disabled. In *Information Technologies in Biomedicine*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 589–602.
35. Chen, Y.; Li, T.; Zhang, R.; Zhang, Y.; Hedgpeth, T. EyeTell: Video-Assisted Touchscreen Keystroke Inference from Eye Movements. In Proceedings of the EyeTell: Video-Assisted Touchscreen Keystroke Inference from Eye Movements, San Francisco, CA, USA, 20–24 May 2018.
36. Sluganovic, I.; Roeschlin, M.; Rasmussen, K.B.; Martinovic, I. Using reflexive eye movements for fast challenge-response authentication. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; Volume 10, pp. 1056–1067.
37. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 568–576.
38. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Las Vegas, NV, USA, 26 June–1 July 2016; Springer: Cham, Switzerland, 2016; Volume 10, pp. 20–36.
39. Ma, C.Y.; Chen, M.H.; Kira, Z.; AlRegib, G. TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition. *arXiv* **2017**, arXiv:1703.10667.
40. Tesfaldet, M.; Brubaker, M.A.; Derpanis, K.G. Two-stream convolutional networks for dynamic texture synthesis. *arXiv* **2017**, arXiv:1706.06982.
41. Abrams, R.A.; Meyer, D.E.; Kornblum, S. Speed and accuracy of saccadic eye movements: Characteristics of impulse variability in the oculomotor system. *J. Exp. Psychol. Hum. Percept. Perform.* **1989**, *15*, 529.
42. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; Volume 6, pp. 2528–2535.
43. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the 2011 IEEE International Conference on Computer Vision , Barcelona, Spain, 6–13 November 2011; Volume 11, pp. 2018–2025.
44. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–7 and 12 September 2014; Volume 9, pp. 818–833.
45. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
46. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
47. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
48. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
49. Lars, B.; Gilles, L.; Mathieu, B.; Fabian, P.; Andreas, M.; Olivier, G.; Vlad, N.; Peter, P.; Alexandre, G.; Jaques, G.; et al. API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop Lang. Data Min. Mach. Learn.* **2013**, 108–122, arXiv:1309.0238.

Sample Availability: All the experiments data and source code are available from the authors.