A Metric Learning Approach to Eye Movement Biometrics

Dillon Lohr, Henry Griffith, Samantha Aziz, and Oleg Komogortsev
Texas State University
San Marcos, Texas, USA
djl70@txstate.edu, hkgriffith1@gmail.com, sda69@txstate.edu, ok11@txstate.edu

Abstract

Metric learning is a valuable technique for enabling the ongoing enrollment of new users within biometric systems. While this approach has been heavily employed for other biometric modalities such as facial recognition, applications to eye movements have only recently been explored. This manuscript further investigates the application of metric learning to eye movement biometrics. A set of three multilayer perceptron networks are trained for embedding feature vectors describing three classes of eye movements: fixations, saccades, and post-saccadic oscillations. The network is validated on a dataset containing eye movement traces of 269 subjects recorded during a reading task. The proposed algorithm is benchmarked against a previously introduced statistical biometric approach. While mean equal error rate (EER) was increased versus the benchmark method, the proposed technique demonstrated lower dispersion in EER across the four test folds considered herein.

1. Introduction

Eye movement biometrics have received considerable attention in the literature over the past two decades [16]. This focus is motivated by the specificity and persistence of human eye movements [2]. Eye movement biometric systems offer notable advantages over alternative modalities, including the ability to support liveness detection [17], along with spoof-resistant continuous authentication [5]. Eye movements are also well suited for integration within multimodal biometric systems [14].

While considerable studies demonstrating the general efficacy of eye movement biometrics have been conducted, existing literature is characterized by some notable limitations. Namely, prior studies have largely been conducted on a limited pool of subjects (e.g., 22 subjects in [4], 58 subjects in [18]), thereby reducing the reliability of the reported results [3]. Moreover, the majority of the existing machine learning based literature is formulated as a closed-set classification problem (e.g., [30], [18]). This formulation lacks practical feasibility, as it requires retraining the entire network upon enrollment of a new user. Additionally, the identification of imposters is complicated by this closed-set formulation.

To address this limitation, recent studies have begun considering metric learning for eye movement biometrics [1]. Initially demonstrated for facial recognition applications [29], this method directly addresses the aforementioned limitation of classification-based biometric systems. This advantage is especially pertinent, given the pending large-scale integration of eye tracking technology within emerging wearable computing platforms, including virtual and augmented reality head mounted devices.

The research described herein further explores the application of metric learning to eye movement biometrics. Namely, we train a set of eye movement type-specific multilayer perceptron (MLP) embedding networks using a triplet loss formulation introduced in [29]. Input vectors consist of a discrete feature array describing various classes of eye movements, including fixations, saccades, and PSOs. Similarity scores are produced as a weighted sum of the output of each embedding network.

The proposed technique is verified on a large-scale dataset consisting of 269 participants recorded during a reading task. To the best of our knowledge, this manuscript is the first to employ metric learning in the eye movement biometrics domain using a biometric template consisting of discrete eye movement features. We hypothesize that this approach may yield superior results versus end-to-end applications by mandating the inclusion of domain-specific knowledge through the feature extraction process. Moreover, by removing the requirement of deep convolutional layers to support direct processing of the eye position signals, the resulting architecture is well-suited for implementation on lightweight embedded architectures.

978-1-7281-9186-7/20/$31.00 ©2020 IEEE

2. Prior work

Since their introduction in 2004, significant research has focused on improving the viability of eye movements as biometrics [15]. A collective review of related work published prior to 2015 may be found in [8]. Moreover, comparative results for studies analyzing common data sets are provided in [27], which summarizes the results of the most recent BioEye competition. As noted within these reviews, the majority of prior work uses a common processing pipeline, with the recordings initially partitioned into specific eye movement events using a classification algorithm, followed by the formation of the biometric template as a vector of discrete features from each event.

The winners of the BioEye 2015 competition, George & Routray [9], used a radial basis function (RBF) network for computing similarities between probe and gallery vectors. Recordings were segmented into fixations and saccades using velocity thresholding (I-VT [28]). Features describing the position, velocity, and acceleration for each eye movement type were extracted from the segmented signal. After applying backwards feature selection, fixation and saccade feature vectors were fed into separate RBF networks, with similarity scores computed as a weighted combination of the output of each network. The algorithm was validated using a dataset of 153 individuals recorded twice during a reading task, achieving an equal error rate (EER) of 2.59%. As the proposed method requires retraining the network upon the enrollment of each new user, it is not feasible for large-scale practical deployment.

In addition to eye movement-specific features, other representations of eye movement recordings have also been explored in the literature. For example, Li et al. [18] used a multi-channel Gabor wavelet transform (GWT) to extract texture features from eye movement trajectories during a visual search task. Support vector machine (SVM) classifiers were used for biometric identification and verification. Results were verified using a data set consisting of 58 subjects recorded across several trials, with a minimum EER of 0.89% reported. Texture-based eye movement features were recently reinvestigated in [10], where downsampling of the filtered images was proposed for the feature extraction step in order to preserve spatial structure. In addition to the aforementioned restriction regarding new user enrollment, both of these studies utilized small subject pools.

Jia et al. [12] introduced deep learning techniques for eye movement biometrics. A recurrent neural network (RNN) was built using long short-term memory (LSTM) cells. The output layer used softmax to produce class probabilities. Their approach was validated using a dataset of 32 subjects recorded across several trials of a high-cognitive-load task, with a minimum EER of 0.85% reported. While valuable for its introduction of deep learning techniques to the domain, this approach is characterized by the same limitations as the aforementioned study.

Friedman et al. [6] employed a statistical approach for eye movement biometrics. The proposed identification workflow is promising for large-scale application, as it does not suffer from the new subject enrollment problem. A novel event classification algorithm, the MNH [7], was used to classify fixations, saccades, post-saccadic oscillations (PSOs), and several types of noise. A set of over 1000 features [26] was extracted from each recording. Features were transformed and selected according to various criteria, including normality, redundancy, and persistence as quantified using the intraclass correlation coefficient (ICC). This approach was validated using a dataset of 298 subjects recorded twice each during a reading task. A best-case EER of 2.01% was reported for features with ICCs thresholded at 0.75. One limitation of this approach is that the use of ICCs for feature selection before splitting into training and testing sets may bias the results, though the magnitude of this bias is unclear.

Jäger et al. [13] utilized involuntary micro eye movements for biometric identification. Raw eye movement signals were initially transformed to isolate desired micro eye movements according to their characteristic velocities, with the resulting scaled values fed into a deep convolutional neural network with two separate subnets. The approach was validated using two datasets (75 subjects during a reading task recorded at 1000 Hz [21], and a newly recorded dataset consisting of 10 users). The network was demonstrated to improve both accuracy and execution speed versus the technique described in [21]. As this approach was validated for using a multi-class classification framework to promote comparability with prior work, it also suffers from the new user enrollment problem.

Abdelwahab and Landwehr [1] introduced metric learning to the eye movement biometrics literature using deep distributional embeddings. Namely, sequences of six-dimensional vectors (binocular gaze and pupil data) at 30 Hz were fed to a deep neural network which produced distributional embeddings using a Wasserstein distance metric. The approach was validated on the Dynamic Images and Eye Movements (DIEM) dataset [22], which contains eye movement data of 210 subjects viewing various video clips (sports, movie trailers, etc.). The technique described herein is differentiated from this work, both by the underlying network structure and by the use of vector versus distributional embeddings, along with the utilization of discrete eye movement feature vectors as input.

3. Methodology

3.1. Dataset

We used the SBA-ST dataset from [6], which consists of eye movements recorded from 322 participants. Eye movements were captured using the EyeLink 1000, which monocularly tracked the left eye at 1000 Hz. Each subject visit lasted no more than one hour and was split into two sessions with a 5-minute break between sessions. Sessions consisted of a series of tasks, only one of which (reading) is used in the present study. During the reading task, several stanzas from Lewis Carroll's nonsensical poem, "The Hunting of the Snark," were displayed on the screen for 60 seconds. During this time, each subject was instructed to silently read through the poem. The two reading tasks (one per session) were separated by an average of 20 minutes. More detailed information about the experimental setup can be found in [6].

We excluded 53 subjects that had at least one recording session with 20% or more noise classified by the MNH [7], resulting in valid data for 269 subjects. The remaining subjects for the current study were split into four random folds for cross-validation. There were two recordings per subject, and each recording had hundreds of classified events. The same 4 folds were also used for the statistical method that we benchmarked our models against.

3.2. Event classification and feature extraction

Each signal was classified into fixations, saccades, and PSOs using the MNH [7]. We then extracted a subset of the features originally introduced in [26] and [9]. Like [9], our features were extracted from each individual event. It is important to note that metric learning could be applied on the raw eye movement signals without the need for event classification, perhaps using CNNs or LSTMs. However, the current analysis framework was chosen for two reasons: 1) As the MNH algorithm [7] was developed specifically for the utilized dataset, we were confident that it would produce meaningful event classification, thereby avoiding errors associated with misclassification and fully leveraging domain-specific knowledge, and 2) a priori segmentation of the raw signals into discrete feature vectors allows for the evaluation of more simplistic and lightweight network architectures, versus the multilayer convolutional architectures traditionally employed in end-to-end workflows, which may be preferable in emerging embedded architectures with limited computing and power resources.

Features were extracted on the horizontal (H), vertical (V), and combined (C) channels. Further details regarding the computation of each feature may be found in the motivating study (i.e., [26] and [9]).

3.2.1 Fixation features

For fixations, we used the following features derived from [26]: duration; drift displacement, distance, and mean velocity (H/V/C); slope and R2 of linear fit of drift (H/V); R2 of quadratic fit of drift (H/V); position centroid (H/V); percent of samples above and crossing the 90th percentile velocity threshold of the MNH; mean, median, standard deviation (SD), skewness, and kurtosis of the velocity and acceleration profiles (H/V/C). We also used the following features derived from [9]: angle with previous fixation; distance from previous fixation; SD, skewness, and kurtosis of the position trace (H/V); dispersion. In total, we extracted 59 features for each fixation event.

3.2.2 Saccade features

For saccades, we used the following features derived from [26]: duration; amplitude (H/V/C); peak velocity, acceleration, and deceleration (H/V/C); travel distance; efficiency; tail efficiency; percentage tail inconsistency; initial deviation; initial average deviation; maximum raw deviation and the respective point; area-based curvature; quadratic-fit curvature; cubic-fit-extreme-1 and cubic-fit-extreme-2 and their respective points; cubic-fit-curvature-maximum and the respective point; amplitude-duration ratio (H/V/C); peak velocity–amplitude ratio (H/V/C); peak velocity–duration ratio (H/V/C); peak velocity–local noise ratio; acceleration-deceleration duration ratio; peak acceleration–peak deceleration ratio (H/V/C); mean, median, SD, skewness, and kurtosis of the velocity and acceleration profiles (H/V/C). We also used the following features derived from [9]: angle; angle with previous saccade; distance from previous saccade; SD of the position trace (H/V); dispersion. In total, we extracted 79 features for each saccade event.

3.2.3 PSO features

For PSOs, we used the following features derived from [26]: duration; time since previous PSO; number of peaks (H/V/C); ratio of PSO duration to the preceding saccade's duration; ratio of PSO duration to the preceding saccade's amplitude (H/V/C); mean, median, SD, skewness, and kurtosis of the velocity and acceleration profiles (H/V/C). Although [9] did not classify PSO events, we used the following features inspired by that study: SD of the position trace (H/V); dispersion. In total, we extracted 42 features for each PSO event.

3.3. Algorithms

Inspired by [9], we train one multilayer perceptron (MLP) for each event type (fixation, saccade, PSO). Since our goal is for each network to learn a meaningful embedding of the features from its event type, we employ the triplet loss formulated by [29]:

$$L = \sum_{i=1}^{N} \max\left(0, \|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + a\right), \quad (1)$$

where f_i^a, f_i^p, and f_i^n are the embeddings of the anchor, positive, and negative samples, respectively, and a is the margin that should separate the positive and negative samples.

Intuitively, this loss function seeks to bring the positive sample closer than the negative sample to the anchor and enforce a minimum separation (the margin) between the positive sample and the negative sample. If this is already the case, then there is no loss; we do not care to bring the positive sample even closer nor push away the negative sample even farther. As a result, the model is not able to learn from triplets that do not violate the margin. This is why—as [29], [11], and others point out—it is imperative to select good triplets that will let the model learn.

There are three classifications of triplets: easy, semi-hard, and hard. Easy negatives do not violate the margin, so our model cannot learn from them. Semi-hard negatives violate the margin, but the positive sample is (correctly) closer to the anchor than the negative sample. Hard negatives are triplets where the negative sample is closer to the anchor than the positive sample. Both semi-hard and hard triplets may be used to train the model, and it seems to be problem-dependent whether one is more useful than the other. Note that the hardest negatives across the whole dataset are likely outliers, so it is unwise (not to mention, computationally infeasible) to train on the globally hardest negatives [29].

Input vectors to the MLPs were first scaled using scikit-learn's RobustScaler [24], with scaling parameters determined on the train set only. Each MLP had the same architecture. Namely, the network consisted of 4 hidden layers, each with 64 ReLU-activated nodes. The output layer had 64 nodes, resulting in a 64-element embedding vector. We did not use dropout on any of the layers. While the literature has suggested that the normalization of the resulting embeddings for triplet loss yields no benefit (i.e., [11]), we found that L2 normalization improved our results. We used a triplet loss margin of 0.2.

At each training iteration, we randomly selected 20 subjects and 5 events per subject. The events were sampled from both recording sessions. All semi-hard and hard triplets contributed to the loss for that iteration. Each MLP was trained for 200,000 iterations and performance measures were computed after every 1000 iterations. We used the AdamW optimizer [20] with an initial learning rate of 0.001 and a weight decay of 0.1. Other parameters were left to their default values. After 100,000 iterations, the learning rate was reduced by a factor of 10.

For each eye movement type, the embedding of each recording was defined by computing the centroid of all corresponding eye movements from that recording. This approach therefore describes each recording using three centroid embeddings, one for each eye movement type. Genuine matches correspond to recording pairs for the same individual across sessions. Distance metrics were computed for each of the three eye movement embeddings individually to allow for the assessment of the value of each eye movement type within the authentication process. In addition, a model fusing information across each eye movement class was also considered. To combine information across eye movement types, distances were fused using a weighted sum. The following optimal weights were determined through grid search: 0.77 for saccades, 0.08 for fixations, and 0.15 for PSOs. This fusion of information across movement types is consistent with prior work in the literature [9].

We also implemented a modified version of the statistical method by [6] for benchmarking purposes. The original statistical approach used the ICC as part of the feature selection process before splitting into a training and testing set, leading to some information leakage. To fix this, we split the data into training and testing sets at the very beginning, did the whole procedure on the training set, then used the best number of principal components and ICC group to evaluate the testing set. All transformations to normality, feature selections, and PCAs were performed using the training set and later applied on the testing set. As we did not have access to several oculomotor plant characteristic (OPC) features that were included in the original statistical approach, these features are excluded from this comparative analysis.

Metric learning methods were developed and tested using Python 3.6 and PyTorch 1.2 on Pop!_OS 19.10 (a Linux distribution). We utilized the PyTorch Metric Learning library (v0.9.84) [23] to enable rapid development. The statistical method was written in MATLAB 2020a. Analysis was conducted on a desktop machine with an NVIDIA GTX 1080 (8 GB memory), an Intel i7-6700K @ 4 GHz, and 32 GB RAM. Each metric learning model took approximately one hour to train on this machine, and roughly 11 hours were needed to train all 3 models with 4-fold CV.

3.4. Evaluation methodology

We evaluated the performance of 5 different models: 1) the fixations-only model (FIX only), 2) the PSOs-only model (PSO only), 3) the saccades-only model (SAC only), 4) the model resulting from taking a weighted sum of the distances produced by the three prior models (fused), and 5) the benchmark statistical method. Each model was quantitatively assessed with EER and was qualitatively assessed with receiver operating characteristic (ROC) curves and genuine-vs-impostor distributions.

To maximize the use of available data, we performed 4-fold cross-validation. For each evaluation fold, we computed the Euclidean distance between each subject's session 1 centroid embeddings and all other subjects' session 2 centroid embeddings. We computed the EER with the resulting

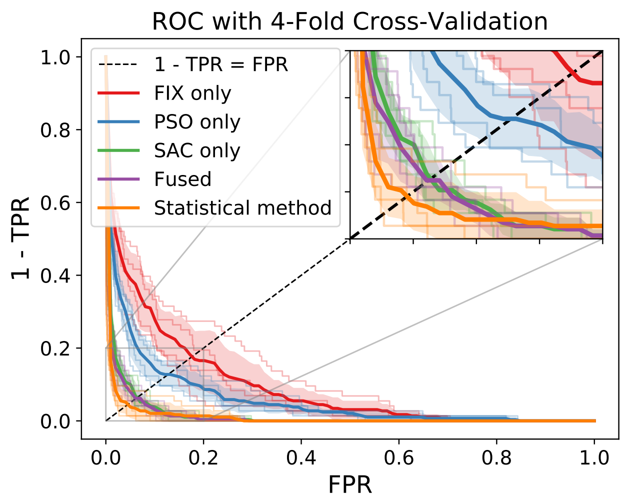

Figure 1. ROC curves for each model evaluated on each of the 4 folds. The mean ROC curve is drawn with a thicker line and is surrounded by a shaded region representing the standard deviation of the curves. To help distinguish similar curves, a zoomed-in portion of the plot is inset at the top-right. The dashed line indicates where $1 - TPR = FPR$, and the point where one of the curves intersects this line is the EER for that curve.

distances. These distances were then negated and used as similarities for computing an ROC curve with scikit-learn [24]. We also plotted the similarity score distributions for genuine and impostor matches. Since the negated distances are not bound from 0 to 1 like a similarity score, we min-max scaled the distances to be between 0 and 1 before visualizing the genuine and impostor distributions.

For the ROC curves, we plotted one curve for each model evaluated on each fold. We also computed an average curve for each model by interpolating the true-positive rate (TPR) at every percentage point of the false-positive rate (FPR) for each curve and plotting the mean TPR at each point of the FPR. The standard deviation of the TPR at each point of the FPR was also computed and plotted to show the variance of the model across folds.

4. Results

The EERs for each approach are presented in Table 1. The ROC curves of each model evaluated on each fold are presented in Figure 1. The genuine and impostor distributions for the FIX-only (A), PSO-only (B), SAC-only (C), fused (D), and statistical (E) models are displayed in Figure 2.

Fixations alone performed the worst in terms of EER and had the most variance across folds of the models tested. This indicates a lack of distinguishable information in the implemented fixation features. There is evidence that microsaccades during fixations could provide useful biometric information [25], but we did not capture microsaccades in

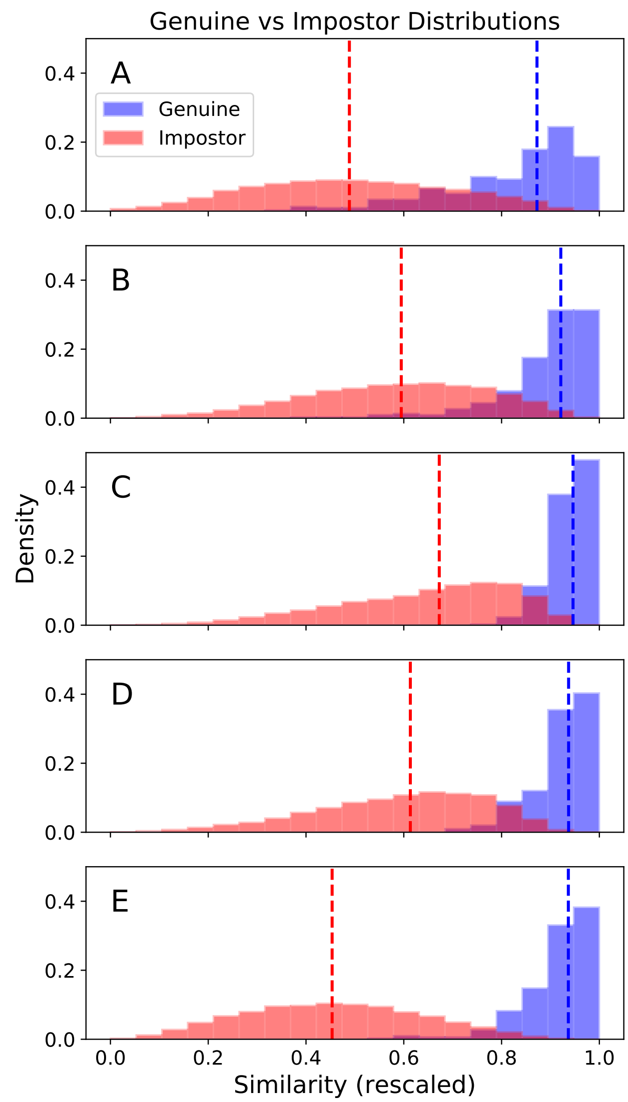

Figure 2. The genuine and impostor distributions combined across all 4 folds for each model. The dashed lines represent the median of each distribution. The distributions are plotted as densities, so the area under the curve sums up to 1. Since the models originally output distances, we negated then min-max scaled the distances within each fold to fit between 0 and 1 before plotting them here. (A) FIX-only model. (B) PSO-only model. (C) SAC-only model. (D) Fused model. (E) Statistical approach.

our feature set. Saccades alone provided most of the useful information for our metric learning models. PSOs alone performed somewhere between saccades and fixations in terms of EER. Fusing information from saccades, fixations, and PSOs led to more consistent performance, but only slightly reduced EER overall.

Our fused model performed better than the statistical method on only 1 of the 4 folds, and had a noticeably higher mean EER than the statistical method across folds. The

Table 1. Quantitative evaluation of model performance. Values are EER in %. Each row is labeled according to the fold that was used for evaluation, except the last row which contains the means and standard deviations across the 4 folds.

	FIX only	PSO only	SAC only	Fused	Statistical method
Fold 1	14.46	10.96	5.30	5.35	4.47
Fold 2	24.30	15.79	6.83	5.71	7.57
Fold 3	19.44	14.57	8.33	7.98	2.68
Fold 4	16.67	10.89	5.69	6.14	4.17
Mean (SD)	18.72 (3.67)	13.05 (2.17)	6.54 (1.18)	6.29 (1.01)	4.72 (1.78)

fused model did, however, perform more consistently across folds than the statistical method, exemplified by the significantly lower SD of the EERs produced by our model.

An interesting observation is that the largest improvement in EER moving from the SAC-only model to the fused model is on fold 2, where the FIX-only and PSO-only models individually performed the worst. This exemplifies the importance of fusing information from all of the models.

It is worth noting that our modified statistical method performed worse than the original statistical approach by [6]. This may be due to several reasons. We moved the train-test split to the start of the procedure rather than after feature selection. We also performed 4-fold CV instead of 100 random 50/50 splits which may have some effect on our results. Lastly, we did not have access to the OPC features used in the original approach.

5. Conclusion

We employed metric learning for eye movement biometrics by training MLPs using features extracted from classified saccades, fixations, and PSOs during reading. Each MLP learned an embedding function such that events from the same subject were closer to each other than events from different subjects. We compared our metric learning approach against a modified version of the statistical approach by [6] and found that, when we fuse the similarities produced by each MLP, our approach produces more consistent results but does not achieve the same level of performance as the statistical approach.

This study did not take advantage of the potential for metric learning to be used with the raw eye movement signals directly, without the need for event classification. In future studies, we would like to explore this direction of work. We would also like to see how well this approach works on other datasets. For example, the dataset described in [19] was collected using an eye tracker embedded in a virtual reality device. This dataset is generally lower quality than the one used in the present study, but it includes signals recorded from both eyes. Utilizing the information from both eyes together could prove beneficial for eye movement biometrics.

6. Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1144466. The study was also funded by 3 grants to Dr. Komogortsev: (1) National Science Foundation, CNS-1250718 and CNS-1714623, www.NSF.gov; (2) National Institute of Standards and Technology, 60NANB15D325, www.NIST.gov; (3) National Institute of Standards and Technology, 60NANB16D293. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the National Institute of Standards and Technology.

References

[1] A. Abdelwahab and N. Landwehr. Deep Distributional Sequence Embeddings Based on a Wasserstein Loss. *arXiv:1912.01933 [cs, stat]*, Dec. 2019. arXiv: 1912.01933. 1, 2

[2] G. Bargary, J. M. Bosten, P. T. Goodbourn, A. J. Lawrance-Owen, R. E. Hogg, and J. D. Mollon. Individual differences in human eye movements: An oculomotor signature? *Vision Research*, 141:157–169, Dec. 2017. 1

[3] S. C. Dass, Y. Zhu, and A. K. Jain. Validating a biometric authentication system: Sample size requirements. *IEEE Transactions on pattern analysis and machine intelligence*, 28(12):1902–1319, 2006. 1

[4] S. Eberz, G. Lovisotto, K. B. Rasmussen, V. Lenders, and I. Martinovic. 28 Blinks Later: Tackling Practical Challenges of Eye Movement Biometrics. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 1187–1199, London, United Kingdom, Nov. 2019. Association for Computing Machinery. 1

[5] S. Eberz, K. Rasmussen, V. Lenders, and I. Martinovic. Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics. 2015. 1

[6] L. Friedman, M. S. Nixon, and O. V. Komogortsev. Method to assess the temporal persistence of potential biometric features: Application to oculomotor, gait, face and brain structure databases. *PLoS ONE*, 12(6):e0178501, jun 2017. 2, 3, 4, 6

[7] L. Friedman, I. Rigas, E. Abdulin, and O. V. Komogortsev. A novel evaluation of two related and two independent algorithms for eye movement classification during reading. *Behavior Research Methods*, 50(4):1374–1397, aug 2018. 2, 3

[8] C. Galdi, M. Nappi, D. Riccio, and H. Wechsler. Eye movement analysis for human authentication: a critical survey. 84:272–283. 2

[9] A. George and A. Routray. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters*, 82:207–215, oct 2016. 2, 3, 4

[10] H. K. Griffith and O. V. Komogortsev. Texture feature extraction from free-viewing scan paths using gabor filters with downsampling. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Adjunct, pages 1–3. Association for Computing Machinery. 2

[11] A. Hermans, L. Beyer, and B. Leibe. In Defense of the Triplet Loss for Person Re-Identification. mar 2017. 4

[12] S. Jia, D. H. Koh, A. Seccia, P. Antonenko, R. Lamb, A. Keil, M. Schneps, and M. Pomplun. Biometric recognition through eye movements using a recurrent neural network. In *Proceedings - 9th IEEE International Conference on Big Knowledge, ICBK 2018*, pages 57–64. Institute of Electrical and Electronics Engineers Inc., dec 2018. 2

[13] L. A. Jäger, S. Makowski, P. Prasse, S. Liehr, M. Seidler, and T. Scheffer. Deep eyedentification: Biometric identification using micro-movements of the eye. 2

[14] P. Kasprowski and K. Harezlak. Fusion of eye movement and mouse dynamics for reliable behavioral biometrics. *Pattern Analysis and Applications*, 21(1):91–103, Feb. 2018. 1

[15] P. Kasprowski and J. Ober. Eye movements in biometrics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3087:248–258, 2004. 2

[16] C. Katsini, Y. Abdrabou, G. E. Raptis, M. Khamis, and F. Alt. The Role of Eye Gaze in Security and Privacy Applications: Survey and Future HCI Research Directions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–21, Honolulu, HI, USA, Apr. 2020. Association for Computing Machinery. 1

[17] O. V. Komogortsev and A. Karpov. Liveness detection via oculomotor plant characteristics: Attack of mechanical replicas. In *2013 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2013. 1

[18] C. Li, J. Xue, C. Quan, J. Yue, and C. Zhang. Biometric recognition via texture features of eye movement trajectories in a visual searching task. *PLoS ONE*, 13(4):e0194475, apr 2018. 1, 2

[19] D. J. Lohr, S. Aziz, and O. Komogortsev. Eye movement biometrics using a new dataset collected in virtual reality. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Adjunct, New York, NY, USA, 2020. Association for Computing Machinery. 6

[20] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR, nov 2019. 4

[21] S. Makowski, L. A. Jäger, A. Abdelwahab, N. Landwehr, and T. Scheffer. A discriminative model for identifying readers and assessing text comprehension from eye movements. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 209–225. Springer. 2

[22] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. 3(1):5–24. 2

[23] K. Musgrave, S.-N. Lim, and S. Belongie. PyTorch Metric Learning. `https://github.com/KevinMusgrave/pytorch-metric-learning/`, 2019. 4

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4, 5

[25] W. Poynter, M. Barber, J. Inman, and C. Wiggins. Individuals exhibit idiosyncratic eye-movement behavior profiles across tasks. *Vision Research*, 89:32–38, aug 2013. 5

[26] I. Rigas, L. Friedman, and O. Komogortsev. Study of an extensive set of eye movement features: Extraction methods and statistical analysis. *Journal of Eye Movement Research*, 11(1), 2018. 2, 3

[27] I. Rigas and O. V. Komogortsev. Current research in eye movement biometrics: An analysis based on BioEye 2015 competition. 58:129–141. 2

[28] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Eye Tracking Research and Applications Symposium 2000*, pages 71–78, New York, New York, USA, 2000. ACM Press. 2

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 3, 4

[30] Y. Zhang and M. Juhola. On Biometrics With Eye Movements. *IEEE journal of biomedical and health informatics*, 21(5):1360–1366, 2017. 1