Plain convolutional neural networks (CNNs) have been used to achieve state-of-the-art performance in various domains in the past years, including biometric authentication via eye movements. There have been many relatively recent improvements to plain CNNs, including residual networks (ResNets) and densely connected convolutional networks (DenseNets). Although these networks primarily target image processing domains, they can be easily modified to work with time series data. We employ a DenseNet architecture for end-to-end biometric authentication via eye movements. We compare our model against the most relevant prior works including the current state-of-the-art. We find that our model achieves state-of-the-art performance for all considered training conditions and data sets.

**Keywords** Eye movements, biometric authentication, metric learning, template aging

1 Introduction

Eye movement biometrics have received considerable attention in the literature over the past two decades [1]. This focus is motivated by the specificity and permanence of human eye movements [2]. Eye movement biometrics systems offer notable advantages over alternative modalities, including the ability to support liveness detection [3, 4] and spoof-resistant continuous authentication [5]. Eye movements are also well suited for integration within multimodal biometrics systems [6].

Most biometric modalities can be separated into two categories: behavioral and physical. Behavioral biometrics reflect a person’s patterns of behavior, including eye movements, gait, signature, and voice. Physical biometrics reflect the physical traits of a person, including face, fingerprint, iris, and retina. Physical biometrics tend to be more discriminative, while behavioral biometrics tend to be more spoof-resistant. It is generally understood that behavioral biometrics are not sufficient to be used alone for authentication and/or identification. Instead, a multimodal biometrics system comprising both behavioral and physical biometrics should be used in practice to take advantage of both the increased security of physical biometrics and the increased spoof-resistance of behavioral biometrics. That said, the present study is interested in assessing eye movement biometrics in isolation with the knowledge that superior performance could later be achieved with a multimodal system (e.g., combining eye movement analysis with iris and/or retina scans).

Biometrics can be used for authentication and identification. In the authentication scenario, a prospective user presents their biometric template as if they were a specific, previously enrolled user. A judgment must then be made if the prospective user’s biometric template is similar enough to that specific enrolled user’s enrolled biometric template. In the identification scenario, a prospective user presents their biometric template as if they were any previously enrolled user. A judgment must then be made if the prospective user’s biometric template is similar enough to any enrolled user’s enrolled biometric template.
The difficulty of biometric authentication is largely independent of gallery size, while biometric identification becomes more difficult with increasing gallery sizes [7]. Therefore, the present study focuses on the authentication scenario to facilitate more fair comparisons with other works with various dataset sizes.

The main contributions of the present study are:

- State-of-the-art biometric authentication performance.
- A new training/evaluation paradigm that is the first to result in better-than-chance FRR @ FAR $10^{-4}$.
- The first application of a more modern convolutional architecture for eye movement biometrics.

The rest of the manuscript is structured as follows. First, we discuss relevant prior works in Section 2. Next, we introduce our proposed DenseNet-based architecture in Section 3. Then, we present a comparison against the current state-of-the-art, DeepEyedentificationLive, in Section 4 and a smaller-yet-competitive model, Eye Know You, in Section 5. Afterward, we propose a new methodology for training and evaluating eye movement biometrics models in Section 6. Finally, we close with a discussion of our results and ideas for future work in Section 7.

2 Prior Work

2.1 Convolutional neural networks (CNNs)

Since the seminal works of AlexNet [8] and VGGNet [9], CNNs have quickly become some of the most popular types of neural networks for image processing tasks. They also started seeing use in time series domains like audio synthesis and time series classification. In these domains, various flavors of recurrent neural networks (RNNs) [10, 11] were once the most common, but CNNs have empirically shown to be capable of similar-or-better performance while also being much faster to train [12].

Several pivotal architectural improvements have been made to CNNs since their infancy. We focus on two such improvements: residual networks (ResNets) [13] and densely connected convolutional networks (DenseNets) [14]. ResNets [13] introduce so-called “skip connections.” These skip connections combine the output of each convolutional block with its input via summation. ResNets consequently enable the training of significantly deeper networks than was previously possible. DenseNets [14] include similar skip connections between each convolutional block and all subsequent blocks, using channel-wise concatenation instead of summation to facilitate even better information flow than ResNets. One study visualizing loss landscapes [15] showed that DenseNets have much smoother loss landscapes than ResNets which may lead to increased ease of convergence during training.

We acknowledge that there are more recent architectures than DenseNets that claim better performance on image processing tasks (e.g., ResNeXT [16], DSNet [17], EfficientNet [18], EfficientNetV2 [19]). Rather than using one of these more recent architectures, we chose to base our architecture on DenseNet because of its relative simplicity and better established history.

2.2 Eye movement biometrics

Eye movement biometrics has been studied extensively since the modality’s introduction in 2004 [20]. Most earlier works (e.g., [21]) require explicit classification of eye movement signals into physiologically-grounded events, from which hand-crafted features are extracted and fed into statistical and/or machine learning models. Since the recent introduction of deep learning to the field of eye movement biometrics [22, 23], end-to-end biometric authentication has become more common. The current state-of-the-art model is DeepEyedentificationLive (DEL) [24] which utilizes subnets that separately focus on “fast” (e.g., saccadic) and “slow” (e.g., fixational) eye movements. Another recent model, Eye Know You (EKY) [25], achieves biometric authentication accuracies that are competitive with DEL despite EKY having roughly 400-times fewer learnable parameters than DEL (~500K vs ~200M). This was achieved through the use of exponentially dilated convolutions to produce an exponential growth in receptive field at only a linear cost in parameter count. The present study expands upon the work of [25] by switching to a DenseNet-based architecture to simultaneously increase expressive power and further reduce parameter count.

3 Network Architecture

The proposed network architecture (see Figure 1) comprises a single dense block with 8 pre-activated convolution layers, followed by a pre-activated global average pooling layer, and ending with a fully-connected layer that outputs a 128-dimensional embedding. An exponentially increasing dilation rate $d = 2^\ell \mod 7$ is used in the convolution
Figure 1: The proposed DenseNet-based network architecture. We use a growth rate of 32, meaning each convolution layer outputs 32 feature maps that are then concatenated with the previous feature maps. Each convolution layer uses kernel size $k = 3$, dilation $d$ (varies by layer), stride $s = 1$, and (zero) padding $p = d$ (to preserve the length along the feature dimension). All but the first convolution layer are preceded by BN and ReLU. The final feature maps are passed into a global average pooling layer that reduces each feature map to a single value. The result is fed into a fully-connected layer that outputs a 128-dimensional embedding of the input. When classification is required (e.g., for cross-entropy loss), we include an additional fully-connected layer that outputs class logits. This network has $\sim 125K$ learnable parameters without the final classification layer, and $\sim 150K$ with it.

layers $\ell = 0, \ldots, 7$ to produce an exponential growth of the receptive field. As a result, the final convolution layer has a maximum receptive field of $2^8 + 1 = 257$ time steps from the input. When classification is required (e.g., for cross-entropy loss), an additional pre-activated fully-connected layer is appended to the end of the network that outputs class logits.

We use a pre-activation architecture, meaning we follow the order of BN-ReLU-Conv instead of Conv-BN-ReLU, where BN is batch normalization and ReLU is the rectified linear unit activation function. Pre-activation DenseNet (and ResNet) architectures generally produce lower errors than their post-activation counterparts [26]. In preliminary experiments, we found no significant difference between pre- and post-activation, nor between BN-ReLU and ReLU-BN. We also tried using global max pooling instead of global average pooling, but we found that the latter generally performed better.

4 Baseline 1: DeepEyedentificationLive (DEL)

Our first baseline is the current state-of-the-art model, DEL [24]. We reimplemented the original training and evaluation routines as closely as possible and then swapped out the network architecture with our own while keeping everything else the same.

4.1 Dataset

For this baseline, we use the JuDo1000 dataset [4] comprising 150 subjects each recorded over 4 sessions with at least 1 week between each session. During each recording session, a dot stimulus jumped to random positions on the screen. There were 9 trial configurations varying the screen area and the duration of each stimulus presentation, and there were 5 stimulus presentations during each trial. Each trial configuration was repeated 12 times throughout the session, totaling 108 trials. Both eyes were recorded binocularly.

The validation set is built by randomly sampling 1 trial per configuration from each recording. The remaining data is split by subject, where 125 subjects form the training set and the other 25 subjects form the test set. This train-test split is repeated 10 times to get an average measure of performance.

4.2 Data preprocessing

Starting with binocular position signals, we compute velocity using the two-point central difference method:

$$\delta_x^{(t)} = \frac{x^{(t+1)} - x^{(t-1)}}{2r},$$

where $r = 1000$ is the sampling rate in Hz. We then perform the “fast” and “slow” transformations proposed in [27]:

$$\left(\delta_{x,\text{fast}}^{(t)}, \delta_{y,\text{fast}}^{(t)}\right) = \begin{cases} (\delta_x^{(t)}, \delta_y^{(t)}), & \sqrt{\delta_x^{(t)} + \delta_y^{(t)}^2} > v \\ (0, 0), & \text{otherwise} \end{cases}$$

3
\[ \delta^{(t)}_{x,\text{slow}} = \arctan \left( c \delta^{(t)}_x \right) , \]  

where \( z \) is the z-score transformation, \( v = 40 \) °/s is the velocity threshold for the fast transformation, and \( c = 0.02 \) is the scale factor for the slow transformation. We also compute vergence between the left and right eyes as

\[ \delta^{(t)}_{x,\text{vrg}} = \delta^{(t)}_{x,\text{right}} - \delta^{(t)}_{x,\text{left}} . \]  

Analogous transformations are applied for horizontal and vertical velocities for the left and right eyes. Any NaN values are replaced with zeros after all these transformations.

Lastly, the stimulus positions are converted to degrees of visual angle. We then compute the offsets between consecutive stimulus positions, \( \delta_{x,\text{stim}} \) and \( \delta_{y,\text{stim}} \).

In total, we have 12 channels of information: \( \delta_{x,\text{left},\text{fast}}, \delta_{x,\text{left},\text{slow}}, \delta_{x,\text{right},\text{fast}}, \delta_{x,\text{right},\text{slow}}, \delta_{y,\text{left},\text{fast}}, \delta_{y,\text{left},\text{slow}}, \delta_{y,\text{right},\text{fast}}, \delta_{y,\text{right},\text{slow}}, \delta_{x,\text{vrg}}, \delta_{y,\text{vrg}}, \delta_{x,\text{stim}}, \) and \( \delta_{y,\text{stim}} \).

### 4.3 Training

DEL contains 2 subnets that each process different subsets of the input channels. The slow subnet processes 6 channels: \( \delta_{x,\text{left},\text{slow}}, \delta_{x,\text{right},\text{slow}}, \delta_{y,\text{left},\text{slow}}, \delta_{y,\text{right},\text{slow}}, \delta_{x,\text{vrg}}, \) and \( \delta_{y,\text{vrg}} \). The fast subnet processes 8 channels: \( \delta_{x,\text{left},\text{fast}}, \delta_{x,\text{right},\text{fast}}, \delta_{y,\text{left},\text{fast}}, \delta_{y,\text{right},\text{fast}}, \delta_{x,\text{vrg}}, \delta_{y,\text{vrg}}, \delta_{x,\text{stim}}, \) and \( \delta_{y,\text{stim}} \).

Each subnet is trained separately. Input samples consist of 1000 contiguous time steps. The slow subnet is optimized through cross-entropy loss. The fast subnet is optimized through the sum of cross-entropy loss (for subject classification) and binary cross-entropy loss (for liveness detection). After training both subnets, their weights are frozen and their embeddings are concatenated and fed into a final set of fully-connected layers that are optimized through cross-entropy loss. The subnets are trained with a learning rate of 0.001, while the final layers are trained with a learning rate of 0.0001.

For our proposed DenseNet-based model, we feed all 12 channels into our model at once. Our model is optimized through the sum of cross-entropy loss (for subject classification) and binary cross-entropy loss (for liveness detection). Our model is trained with a learning rate of 0.001 to match the subnets.

The Adam optimizer is used along with a minibatch size of 64. Early stopping monitoring validation loss with a patience of 10 is used, and the best weights are kept.

### 4.4 Evaluation

We evaluate the models using equal error rate (EER) which is the point where false acceptance rate (FAR) is equal to false rejection rate (FRR). This requires a set of data used for enrollment and a separate set of data for authentication/application/verification. The enrollment set is formed using 1 trial per configuration sampled from the first 3 sessions for each of the 25 test subjects. The authentication set is formed using 10 randomly selected 1-second windows of data from the 4th session for each of the 25 test subjects.

Similarity between an enrolled trial and an authentication window is the maximum cosine similarity between the embedding of the authentication window and all embeddings of the enrolled trial windows. Similarity between an enrolled user and an authentication user is the maximum of the similarities across all enrolled trials for the enrolled user and all authentication windows for the authentication user.

We compute such similarities between each enrolled user and each authentication user \((25 \times 25 = 625 \text{ total similarity scores})\). These similarity scores are then fed into a receiver operating characteristic (ROC) curve that is used to compute EER—the point where false acceptance rate (FAR) is equal to false rejection rate (FRR).

### 4.5 Results

Results comparing our proposed model against the DEL baseline are presented in Table 1. Both models performed uncharacteristically poorly on split 8, but our model consistently performed at least as well as DEL.
Table 1: Comparison of DEL baseline and our proposed DenseNet-based model. The best result in each row is in bold.

| Split | DEL | Ours |
|-------|-----|------|
| 0     | 7.67| 4.67 |
| 1     | 12.00| 9.50 |
| 2     | 9.50| 6.83 |
| 3     | 12.00| 6.50 |
| 4     | 12.00| 4.00 |
| 5     | 12.00| 8.00 |
| 6     | 8.00| 8.00 |
| 7     | 10.00| 7.33 |
| 8     | 20.00| 18.17 |
| 9     | 6.00| 4.00 |

|       | Mean  | SD    |
|-------|-------|-------|
| EER   | 10.92| 3.85  |
|       | 7.70  | 4.11  |

|       | Median | IQR   |
|-------|--------|-------|
| EER   | 11.00  | 3.63  |
|       | 7.08   | 2.87  |

5 Baseline 2: Eye Know You (EKY)

Our second baseline is the previous iteration of our work, EKY [25]. We reimplemented the original training and evaluation routines as closely as possible and then swapped out the network architecture with our new one while keeping everything else the same.

5.1 Dataset

For this baseline, we use the GazeBase dataset [28] comprising 322 subjects each recorded up to 18 times over a 3-year period. During each recording session, subjects performed a battery of tasks including following a jumping dot, reading text, watching a video, and playing an interactive game. The left eye was recorded monocularly.

We only consider the reading task (TEX). We first construct a test set using all 59 subjects present during Round 6. This test set comprises nearly 50% of all recordings in the dataset. We split the remaining data into subject-disjoint sets using 4-fold cross-validation in a way that balances number of subjects and recordings between folds as well as possible.

5.2 Data preprocessing

Starting with monocular (left eye) position signals, we compute velocity using the one-sample backward difference method:

\[
\delta_x^{(t)} = (x^{(t)} - x^{(t-1)}) r, \tag{5}
\]

where \( r = 1000 \) is the sampling rate in Hz. We then clamp velocities between ±1000 to reduce the influence of noise. Clamped velocities are then transformed using the fast and slow transformations from Equation (2) and Equation (3), respectively. Any NaN values are replaced with zeros after all these transformations.

This produces 4 channels of information: \( \delta_x, \delta_y, \delta_x, \delta_y \).

5.3 Training

Input samples consist of 1024 contiguous time steps. Models are optimized through multi-similarity loss [29].

Each minibatch consists of 80 samples constructed in the following manner. We select one subject that appears in all of Rounds 1–5. We then select, for each of Rounds 1–5, a different subject than the one already chosen. For each round, for the two subjects chosen for that round, we select 4 samples from session 1 and 4 samples from session 2. With 5 rounds, 2 subjects per round, and 8 samples per subject, we end up with a total of 80 samples per minibatch.

The AdamW optimizer is used with a learning rate of \( 9.365 \times 10^{-5} \) and a weight decay of \( 4.758 \times 10^{-3} \). For multi-similarity loss, we use hyperparameters \( \alpha = 2.336, \beta = 48.98, \lambda = 0.1782, \) and \( \varepsilon = 0.2413 \). These hyperparameter
values were chosen based on the best results from running 1 fixed configuration followed by 5 iterations of random search followed by 25 iterations of Bayesian optimization.

Training lasts for a maximum of 100,000 iterations but may stop early if the mean EER across Rounds 1–5 does not improve after 20,000 iterations.

5.4 Evaluation

We evaluate the models using EER and FRR @ FAR. These measures require a set of data used for enrollment and a separate set of data for authentication/application/verification. The enrollment set is formed using the first 10 windows of session 1 from Round 1 for each subject in the test set. The authentication set is formed using the first 10 windows of session 2 from Round \( R \) for each subject in the test set. A separate authentication set is made for each of \( R = 1, \ldots, 9 \).

Similarity is computed as the mean cosine similarity across temporally-aligned window embeddings. The empirical genuine and impostor similarity score distributions are resampled by fitting and sampling from the Pearson family of distributions to generate 20,000 resampled genuine and 20,000 resampled impostor similarity scores. These resampled similarity scores are fed into a ROC curve to estimate EER and FRR @ FAR (for fixed \( \text{FAR} \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\} \)).

5.5 Results

Results are shown in Table 2 for the EKY baseline and in Table 3 for our model. Our model consistently outperforms the EKY baseline in every metric and for every Round.

We also plot genuine and impostor similarity score distributions in Figure 2. ROC curves are plotted in Figure 3.

### Table 2: Results for the EKY baseline trained with 4-fold cross-validation. Values are given as mean (SD).

| Round | EER        | FRR @ FAR          |
|-------|------------|--------------------|
|       |            | \( 10^{-1} \) | \( 10^{-2} \) | \( 10^{-3} \) | \( 10^{-4} \) |
| 1     | 0.1319 (0.0118) | 0.1812 (0.0443) | 0.5828 (0.0822) | 0.8417 (0.0411) | 0.9408 (0.0198) |
| 2     | 0.1977 (0.0275) | 0.3574 (0.0692) | 0.7783 (0.0435) | 0.9485 (0.0140) | 0.9855 (0.0048) |
| 3     | 0.2055 (0.0144) | 0.3746 (0.0416) | 0.7614 (0.0383) | 0.9276 (0.0179) | 0.9826 (0.0082) |
| 4     | 0.2009 (0.0168) | 0.3654 (0.0514) | 0.7531 (0.0435) | 0.9334 (0.0183) | 0.9827 (0.0115) |
| 5     | 0.1925 (0.0252) | 0.3457 (0.0784) | 0.7148 (0.0447) | 0.8848 (0.0267) | 0.9604 (0.0137) |
| 6     | 0.2098 (0.0281) | 0.3766 (0.0650) | 0.7994 (0.0218) | 0.9536 (0.0103) | 0.9873 (0.0067) |
| 7     | 0.2360 (0.0166) | 0.4229 (0.0421) | 0.7628 (0.0463) | 0.9186 (0.0311) | 0.9699 (0.0112) |
| 8     | 0.2192 (0.0189) | 0.4092 (0.0506) | 0.8211 (0.0338) | 0.9671 (0.0150) | 0.9953 (0.0048) |
| 9     | 0.2087 (0.0306) | 0.3945 (0.0868) | 0.8466 (0.0926) | 0.9676 (0.0248) | 0.9936 (0.0110) |

### Table 3: Results for our proposed model trained with 4-fold cross-validation. Values are given as mean (SD).

| Round | EER        | FRR @ FAR          |
|-------|------------|--------------------|
|       |            | \( 10^{-1} \) | \( 10^{-2} \) | \( 10^{-3} \) | \( 10^{-4} \) |
| 1     | 0.1138 (0.0085) | 0.1273 (0.0163) | 0.4664 (0.0250) | 0.7514 (0.0558) | 0.9114 (0.0691) |
| 2     | 0.1643 (0.0122) | 0.2485 (0.0263) | 0.6734 (0.0235) | 0.9071 (0.0298) | 0.9673 (0.0336) |
| 3     | 0.1826 (0.0139) | 0.2927 (0.0264) | 0.6588 (0.0050) | 0.8644 (0.0207) | 0.9249 (0.0231) |
| 4     | 0.1659 (0.0164) | 0.2470 (0.0341) | 0.6181 (0.0167) | 0.8430 (0.0246) | 0.9366 (0.0255) |
| 5     | 0.1432 (0.0123) | 0.2106 (0.0270) | 0.6277 (0.0269) | 0.8507 (0.0099) | 0.9312 (0.0197) |
| 6     | 0.1833 (0.0083) | 0.2946 (0.0219) | 0.6902 (0.0290) | 0.8981 (0.0402) | 0.9822 (0.0245) |
| 7     | 0.1927 (0.0213) | 0.3129 (0.0388) | 0.6569 (0.0165) | 0.8414 (0.0290) | 0.9284 (0.0199) |
| 8     | 0.1691 (0.0141) | 0.2890 (0.0358) | 0.7345 (0.0303) | 0.9177 (0.0549) | 0.9607 (0.0396) |
| 9     | 0.1793 (0.0079) | 0.2976 (0.0095) | 0.7391 (0.0343) | 0.9287 (0.0604) | 0.9638 (0.0524) |
6 Proposed Methodology

So far, we have showcased our model’s superiority over the previous state-of-the-art under each baselines’ original training and evaluation paradigms. Next, we introduce a new methodology that allows our model to achieve better-than-chance FRR @ FAR $10^{-4}$.

6.1 Dataset

We use the GazeBase dataset [28] comprising 322 subjects each recorded up to 18 times over a 3-year period. During each recording session, subjects performed a battery of 7 tasks: random saccades (RAN), horizontal saccades (HSS), fixation (FXS), a ball-popping game (BLG), reading (TEX), and two video-viewing tasks (VD1 and VD2). More details about each task can be found in the dataset’s paper [28]. The left eye was recorded monocularly.

We exclude BLG from our analysis due to its large variability in task duration. We construct a test set using all 59 subjects present during Round 6. This test set comprises nearly 50% of all recordings in the dataset. We split the remaining data into subject-disjoint sets using 4-fold cross-validation in a way that balances number of subjects and recordings between folds as well as possible.

6.2 Data preprocessing

Starting with monocular (left eye) position signals, we estimate the first derivative (i.e., velocity) using a Savitzky-Golay filter with order 2 and window size 7. Each recording is split into non-overlapping windows of 1 second (1000 time
steps). Any windows with one or more NaN values and/or one or more radial velocities over 1000 °/s are excluded from the analysis. Excess time steps at the end of a recording are also discarded. Velocities are clamped between ±1000°/s and then zscore transformed.

6.3 Training

Input samples consist of windows of 1000 time steps. The model is optimized through a weighted sum of multi-similarity loss (using the output of the embedding layer) and cross-entropy loss (using the output of the classification layer). Multi-similarity loss is given a weight of 1.0, while cross-entropy loss is given a weight of 0.1. The hyperparameters for multi-similarity loss are left at their defaults: $\alpha = 2.0$, $\beta = 50.0$, $\lambda = 0.5$, and $\varepsilon = 0.1$.

Each minibatch consists of 256 samples constructed in the following manner. First, 16 subjects are selected at random from the train set. Next, 16 windows were randomly selected without replacement for each of the selected subjects. These windows could be selected from any of the rounds and sessions that each subject was present for in the train set. In total, 256 windows are sampled in this way to form each minibatch. For each “epoch”, we sample as many windows as there are in the train set, though not every window may be included in any given epoch due to the sampling method.

The Adam optimizer is used with a one-cycle cosine annealing learning rate scheduler [30]. The learning rate starts at $10^{-4}$, gradually increases to a maximum of $10^{-2}$ after 30 epochs, and then gradually decreases to a minimum of $10^{-7}$ after 100 epochs. Training lasts for 100 epochs, and the final weights of the model are saved.

6.4 Evaluation

We evaluate the model using EER and FRR @ FAR in almost the same way as the EKY baseline (see Section 5.4). The only difference is that we do not use temporally-aligned window embeddings. Instead, we compute the centroid of the 10 window embeddings for each subject in the enrollment and authentication sets. Cosine similarity is then computed using these centroid embeddings.

Note that since we discard windows with one or more NaN values and/or with one or more radial velocities over 1000 °/s, some subjects have fewer than 10 valid windows at enrollment or authentication time. Indeed, some subjects have no valid windows. Subjects with no valid windows are effectively excluded from the enrollment and/or authentication sets. Therefore, we have introduced positive failure to acquire (FTA) and failure to enroll (FTE) rates using this new methodology, whereas these rates were both 0% for the two baseline methodologies we examined previously. We note that other biometric modalities like fingerprint scans and facial recognition also often do not accept low-fidelity samples in practice, so it is normal to have positive FTA and FTE rates.

6.5 Results

Thanks in large part to the exclusion of low-fidelity samples, we are able to achieve better performance under strict FAR requirements using this new methodology. Preliminary results show that we can achieve 28.62% FRR @ FAR $10^{-4}$ when using R1 as the authentication set. Compared to 91.14% from Table 3, this is significantly better and finally performs better than chance (50% FRR @ FAR $10^{-4}$). However, this is still a far cry from the FIDO Biometrics Requirements’ [31] recommendation of at least 5% FRR @ FAR $10^{-4}$ for a biometrics system.

7 Conclusion

We presented a novel DenseNet-based architecture for biometric authentication via eye movements. We compared our proposed architecture against two baselines, DEL [24] and EKY [25]. Keeping everything constant except for the architectural differences, our model consistently outperforms both baselines.

We also proposed a new training and evaluation methodology. Using this methodology in a preliminary experiment, we achieved as low as 28.62% FRR @ FAR $10^{-2}$. The FTE and FTA rates should be calculated to determine the difficulty of using this method in practice.

Acknowledgment

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,
References

[1] Christina Katsini, Yasmeen Abdrabou, George E. Raptis, Mohamed Khamis, and Florian 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, April 2020. Association for Computing Machinery.

[2] Gary Bargary, Jenny M. Bosten, Patrick T. Goodbourn, Adam J. Lawrence-Owen, Ruth E. Hogg, and J. D. Mollon. Individual differences in human eye movements: An oculomotor signature? Vision Research, 141:157–169, December 2017.

[3] O. V. Komogortsev, A. Karpov, and C. D. Holland. Attack of mechanical replicas: Liveness detection with eye movements. IEEE Transactions on Information Forensics and Security, 10(4):716–725, 2015.

[4] S. Makowski, L. A. Jäger, P. Prasse, and T. Scheffer. Biometric identification and presentation-attack detection using micro- and macro-movements of the eyes. In 2020 IEEE International Joint Conference on Biometrics (IJCB), pages 1–10, 2020.

[5] Simon Eberz, K. Rasmussen, Vincent Lenders, and Ivan Martinovic. Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics. 2015.

[6] Paweł Kasprowski and Katarzyna Harezlak. Fusion of eye movement and mouse dynamics for reliable behavioral biometrics. Pattern Analysis and Applications, 21(1):91–103, February 2018.

[7] Lee Friedman, Hal S Stern, Vladyslav Prokopenko, Shagen Djanian, Henry K. Griffith, and Oleg V. Komogortsev. Biometric performance as a function of gallery size, 2020.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 11 1997.

[11] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[12] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[14] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[15] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2018.

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017.

[17] Chaoning Zhang, Philipp Benz, Dawit Mureja Argaw, Seokju Lee, Junsik Kim, Francois Rameau, Jean-Charles Bazin, and In So Kweon. Resnet or densenet? introducing dense shortcuts to resnet, 2020.

[18] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[19] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training, 2021.

[20] Paweł Kasprowski and Józef 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.

[21] Lee Friedman, Mark S. Nixon, and Oleg 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):1–42, 06 2017.
[22] Shaohua Jia, Do Hyong Koh, Amanda Seccia, Pasha Antonenko, Richard Lamb, Andreas Keil, Matthew Schneps, and Marc 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.

[23] Ahmed Abdelwahab and Niels Landwehr. Deep Distributional Sequence Embeddings Based on a Wasserstein Loss. *arXiv:1912.01933 [cs, stat]*, December 2019. arXiv: 1912.01933.

[24] Silvia Makowski, Paul Prasse, David R. Reich, Daniel Krakowczyk, Lena A. Jäger, and Tobias Scheffer. Deep-eyedentificationlive: Oculomotoric biometric identification and presentation-attack detection using deep neural networks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(4):506–518, 2021.

[25] Dillon Lohr, Henry Griffith, and Oleg V Komogortsev. Eye know you: Metric learning for end-to-end biometric authentication using eye movements from a longitudinal dataset, 2021.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.

[27] Lena A. Jäger, Silvia Makowski, Paul Prasse, Sascha Liehr, Maximilian Seidler, and Tobias Scheffer. Deep eyedentification: Biometric identification using micro-movements of the eye. In Ulf Brefeld, Elisa Fromont, Andreas Hotho, Arno Knobbe, Marloes Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 299–314, Cham, 2020. Springer International Publishing.

[28] Henry Griffith, Dillon Lohr, Evgeny Abdulin, and Oleg Komogortsev. Gazebase, a large-scale, multi-stimulus, longitudinal eye movement dataset. *Scientific Data*, 8(1):184, Jul 2021.

[29] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5017–5025, 2019.

[30] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

[31] Stephanie Schuckers, Greg Cannon, and Nils Tekampe. FIDO biometrics requirements. [https://fidoalliance.org/specs/biometric/requirements/](https://fidoalliance.org/specs/biometric/requirements/). Accessed: 2021-04-04.