Array 7 (2020) 100029

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal

Using deep learning techniques and genetic-based feature extraction for presentation attack mitigation

John Jenkins [a], Kaushik Roy [a], Joseph Shelton [b,*]

[a] Department of Computer Science North Carolina A&T SU, Greensboro, NC, USA
[b] Department of Engineering and Computer Science Virginia State University, Petersburg, VA, USA

ARTICLE INFO

Keywords:
Spoofing mitigation
Biometric recognition
Deep learning

ABSTRACT —

Biometric authentication systems are becoming more prevalent for commercial use with computers and smart devices. Biometric systems also have several vulnerable points that can be exploited by a hacker to gain unauthorized access to a system. Replay attacks focus on capturing feature extractors (FEs) during transmission, decrypting, and replaying for illegal access. The Genetic and Evolutionary Feature Extraction (GEFE) technique, developed at North Carolina A&T State University, recently showed promising results in mitigating replay attacks in combination with a feature selection algorithm. Biometric-based presentation attacks, the focus of this work, is another biometric system vulnerability primarily focused on presenting a biometric sample of quality to illegally gain access to secured data. Recently, deep learning techniques to mitigate presentation attacks have shown promising results. However, the accuracy of deep learning-based biometric presentation attack detection (PAD) methods are limited by the quality of the samples provided. In absence of large sets of original biometric sample data, data augmentation has been shown to be successful in generating synthetic biometric image data and improving the performance of deep learning techniques applied. The novelty of this paper lies in the following two aspects: First, a data augmentation technique with Generative Adversarial Networks (GANs) is used to generate comparative synthetic (spoofing) dataset. With the proliferation of deep fakes in media, this technique should provide insight on the GAN technique often used. Once properly trained, the synthetic images are used to create spoofing datasets. Second, the GEFE technique is used in combination with the GANs to generate improved anti-spoofing feature extractors optimized to mitigate presentation attacks. The combination of GEFE and GANs is used to identify those discriminative biometric features used to mitigate synthetic presentation attacks. The GEFE + GAN technique outperforms the LBP and GEFE techniques alone in overall identification and verification results on spoofing datasets.

1. Introduction

Biometric authentication systems are becoming more prevalent for commercial use with computers and smart devices [1]. Biometric authentication has inherent advantages over other authentication methods such as token-based and knowledge based methods [2–5]. Token-based authentication systems use some form of token, such as a driver's license, or an ID card [5,6]. This form of authentication could be considered the most vulnerable to compromise, due to the fact that tokens can be lost or stolen. Knowledge-based authentication systems have a slight advantage from a security standpoint, in that authentication is based on what a user knows, i.e. password or pin. A password cannot be easily stolen, but if a user writes it down, or is seen entering it, then it can be taken and used maliciously [6,7]. A biometric-based authentication system has an advantage over the two previously mentioned systems in that biometric modalities are difficult to replicate and are unique to individuals [6–8]. The focus of this research is on evolving unique feature extractors for cyber security.

Biometric systems have several vulnerable points that can be exploited by a hacker to gain unauthorized access to a system. One vulnerable point, and the focus of this work, is the acceptance of presentation attacks. Biometric-based presentation attacks are primarily focused on gaining access to a counterfeit biometric sample and implementing the same biometric to illegally gain access to secured data. Although biometric authentication strengthens security protocols through unique feature extraction, presenting a biometric sample of

* Corresponding author.
E-mail addresses: jmjenki1@aggies.ncat.edu (J. Jenkins), kroy@ncat.edu (K. Roy), jshelton@vsu.edu, jshelton@vsu.edu (J. Shelton).

https://doi.org/10.1016/j.array.2020.100029
Received 12 June 2019; Received in revised form 21 April 2020; Accepted 26 April 2020
Available online 30 April 2020

quality to illegally gain access to a biometric system is feasible. It is known that any 2D face biometric system with no anti-spoofing measures can be easily spoofed by presentation attacks [22].

Local Binary Patterns (LBP) is one of the more popular textural techniques for facial recognition [26,27]. The LBP technique has also been used for spoofing detection. In Chingovska et al. [23], anti-spoofing efforts went towards creating a face-spoofing database. The database followed guidelines for developing anti-spoofing algorithms. Despite its simplicity, LBP has proven to be very powerful in texture classification. In Gragnaniello et al. [17], LBP is used in combination with a liveness detection algorithm on multimodal biometrics. Recently, deep learning techniques to mitigate presentation attacks have shown promising results [18–20]. The popularity of Convolutional Neural Networks (CNN) arose after "AlexNet" [25] won the Large Scale Visual Recognition Challenge (LSVRC) of 2012. AlexNet recorded a rank-5 error rate of 15.3% when sorting over a million images into 1000 different classes. In Menotti et al. [20], the focus in building an anti-spoofing system with convolutional networks lead to in outstanding classification results. Results strongly indicate that deep learning techniques can be used for robust spoofing detection [24].

This proposes a few questions. How does training deep learning based biometric system using deep learning generated images effect biometric recognition? What are the distinct (discriminative) biometric features used to mitigate presentation attacks? In this work, we contribute a more complex and novel data augmentation technique to generate a synthetic (spoofing) datasets. Using the deep learning technique of Generative Adversarial Networks (GANs) [28], the generated images will be of quality to spoof the popular texture-based feature extraction method, LBP. Referred to by some as the most interesting idea in machine learning in the last 10 years, GANs have the ability to generate realistic photos of objects, scenes, and people that do not exist, yet are undetectable by humans [35]. GANs are simple yet powerful. The generated spoofing datasets will then be used to improve Genetic and Evolutionary Feature Extraction (GEFE) feature extraction accuracies. The GEFE feature extractors will optimize the LBP feature extractors' dimensions and displacement. The proposed GEFE technique will use a fitness function optimized for mitigation synthetic (generated) images. Through these steps, a novel feature extraction technique to mitigate the effects of presentation attacks is contributed. With the proliferation of deepfakes [36,37] in media, this technqiue should provide insight on the GAN technique often used. In combination with GEFE, the generated feature extractors will also provide insight on which biometric features on the samples has the higher discriminative capabilities with its optimize dimensions and displacement. The remainder of this work follows with a literature review of popular techniques for mitigatigating presentation attacks. This is followed by our methodology, experiments, and results. Lastly, we have our discussion and future direction.

2. Literature review

Presentation attacks are the presentation of impersonated human characteristics to the biometric capture subsystem. This is done to spoof biometric recognition to illegally gain access to the biometric system. Print photo attacks, 3-D masks, and image regeneration have all been used to successfully gain access to biometric systems [17]. A 2D face system with no anti-spoofing measures can be easily spoofed by presentation attacks. Multimodal systems are suggested in scenarios with high probabilities of spoof attacks. Unfortunately, a biometric system with several modes may not need all modalities to be spoofed to compromise the entire multimodal system [22]. Multimodal biometrics has shown to improve the performance of biometric systems, though they are often expensive, and they can also increase the number of vulnerabilities that can be exploited by an intruder [22]. The next section of this chapter will be an overview of some texture-based presentation attack mitigation. LBP is texture-based and one of the more notable techniques for biometric classification. In Section II·B, an overview of recent deep learning based mitigation techniques is given. With promising results from deep learning techniques in image classification, similar findings in spoofing detection and mitigation are favorable. The scope of this review explores mitigation techniques for facial biometrics, excluding techniques designed generally for image classification.

2.1. Texture-based mitigation techniques

Identify applicable funding agency here. If none, delete this text box.

The LBP feature extraction technique forms texture patterns from the pixel intensity values of biometric images [26,27]. This method uses the texture patterns to create Feature Vectors (FVs) associated with the images. The LBP method can be applied to any uniquely textured data, such as facial and iris recognition. For biometric recognition, the first step is to split an image into even sized regions. A histogram is associated with each region, where the frequencies of texture patterns are stored. A FV is formed from the concatenated histograms resulting from each region.

Texture patterns are measured by comparing pixel intensity values to one another within a region. More specifically, each pixel that is inclusively within a region will be compared with its nearest neighboring pixels. A pixel on the border of a region cannot be considered as a center pixel since it does not have pixels within its region surrounding it entirely. Each texture pattern is represented as a binary string. Equations (1) and (2) show how a binary string can be extracted from a region on an image and converted into a decimal value denoting a histogram bin. The term LBP(Ni,c), denotes the decimal value of a texture pattern for a neighborhood of pixels. The term c represents the pixel intensity value of the center pixel, N represents the set of neighboring pixel intensity values for c, and i represents the ith neighboring pixel of c. Equation (2) signifies the difference being taken between each neighboring pixel and the center pixel.

$$LBP(N_i, c) = \sum_{i=0}^{i-1} s(N_i, c)2^i \tag{1}$$

$$s(N_i, c) = \begin{cases} 0, if\ N_i - c \leq 0 \\ 1, if\ N_i - c > 0 \end{cases} \tag{2}$$

Some of the earlier techniques for biometric spoofing detection used textural methods classification. In Chingovska et al. [23], anti-spoofing efforts went towards creating a face-spoofing database. The database followed guidelines for developing anti-spoofing algorithms. The database should provide attacks capable of penetrating simple face recognition systems. Each database provided an evaluation of the scores that a baseline face recognition system would be vulnerable for spoof attacks [23]. LBP based face spoofing counter-measure, variants of LBP were gauged for its efficiency against a variety of attacks. The LBP (3x3) technique resulted in a 14.84% Half Total Error Rate (HTER) on the training set and a 15.17% HTER on the test set. Results also showed that the traditional LBP technique had the best performance/complexity tradeoff [23]. Despite its simplicity, LBP has proven to be very powerful in texture classification.

In Gragnaniello et al. [17], biometric spoofing detection explored using the LBP technique in combination with face liveness detection methods on multimodal biometrics. The multimodal system includes samples from the fingerprint, the iris, and face biometrics. For the experiment, protocol required a cropped small region (64x64 pixel to 80x80 pixel). The novel Histogram of Invariant Gradients (HIG), a variant of Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) are all textural methods tested with the intent of preserving robustness. A linear Support Vector Machine (SVM) classifier was used to avoid feature selection. K-means clustering with Euclidean distance was also used for the joint quantization of features. Although some techniques reduce the average error by as much as 75%, the overall analysis could not clearly distinguish a descriptor performing uniformly better than others [17].

Presentation Attack Detection (PAD) algorithms have also been proposed exploring micro-texture variation using Binarized Statistical Image Features (BSIF) and micro-frequency variations using 2D Cepstrum [24]. The 2D Ceptrum feature extraction is widely used in the domain of speech and image processing. The BSIF method denotes each pixel as a binary string obtained by computing its response to a filter. The filters are trained utilizing the statistical properties of the natural images. The BSIF and 2D Ceptrum feature vectors are concatenated to form a single vector before obtaining a decision using linear SVM classifier. Experimental results revealed that, the PAD algorithm's best scheme was an Average Classification Error Rate (ACER) of 10.21% on face and an ACER of 0% on the iris modality [24].

2.2. Deep learning based mitigation techniques

The popularity of Convolutional Neural Networks (CNN) arose after "AlexNet" [25] won the Large Scale Visual Recognition Challenge (LSVRC) of 2012. AlexNet recorded a rank-5 error rate of 15.3% when sorting over a million images into 1000 different classes. Deep learning techniques have shown promising results in mitigating presentation attacks [18–20].

In Bharati et al. [18], a novel algorithm for facial image retouching detection using deep learning techniques. Digitally altered, "photoshopped", images are common practice in the online/social media communities. To detect digital retouching in facial images, the Supervised Restricted Boltzmann Machine (SRBM) based algorithm is proposed [18]. The detection algorithm uses four local facial patches extracted from a full facial image; the left and right periocular, mouth, and nose regions. The size of each extracted facial patch is 64×64. Each patch is trained on a three layer SDBM. The size of learned representation for each SDBM is 256. Once the features are trained and concatenated, a SVM classifier is trained for two-class classification. The proposed algorithm showed significant advances in retouching detection. Experiments yield a high of 55.7% classification accuracy for existing makeup detection algorithm, with the proposed algorithm achieving nearly 87% classification accuracy on the same dataset. Additionally, experiments showed that the improvements in classification accuracy were attributed to the supervised DBM and SVM classification [18].

In Yang et al. [19], spoofing mitigation relied on the deep Convolutional Neural Network (CNN) to learn features of high discriminative ability in a supervised manner. Combined with some data pre-processing, the face anti-spoofing performance improves drastically [19]. First with data preparation, spatial and temporal augmentation is performed on the face, periocular and iris modalities [19]. After face localization, the temporal augmentation extracts spatiotemporal texture features from multi-frames in the video dataset. Temporal augmented data sets generally contain more information about the images. The spatial augmentation approach will employ a bottleneck approach to extract more information from the background region [19]. A canonical CNN structure is used for feature learning. The proposed network uses five convolutional layers, followed by three fully-connected layers. Response-normalization layers are used for the outputs of the first and second convolutional layers. The max-pooling layers are plugged to process the outputs of the first, second and fifth convolutional layers. The ReLU (Rectified Linear Unit) non-linearity is applied to the output of all convolutional and fully connected layers. To avoid over-fitting, two dropout layers and a soft max function follows the first two fully connected layers and the output layer, respectively. The LibSVM (Library for Support Vector Machines) toolkit was used as the classifier for face anti-spoofing [19]. Results displayed a better performance as the spatial scale increased. The proposed method achieved HTERs (Half Total Error Rates) lower than 5% on two datasets. These results prove the power of CNN in anti-spoofing efforts. These results also indicate the positive effect of background region on face anti-spoofing task. The anti-spoofing system achieves nearly perfect performance on the max scale [19]. Such results show the promise in deep learning techniques in discriminating features.

In Menotti et al. [20], the focus in building an anti-spoofing system with convolutional networks resided on a combination of two approaches. The proposed architecture, spoofnet, entailed detecting spoofing in different biometric modalities. The first approach focused on learning the appropriate CNN architecture for each biometric modality. The second approach consisted of learning filter weights via back-propagation. Architecture optimization (AO) is based upon the CNN architectures with stacked feedforward convolutional operations by means of hyperparameter optimization. Filter optimization (FO) consists of learning filter weights via the well-known back-propagation algorithm [20]. The AO is used to adapt the architecture to the biometric samples, as the FO is used to mold important discriminative features for real and fake biometric classification. The AO and FO techniques are first evaluated separately, then in combination. The results from this research strongly indicate that convolutional networks can be used for robust spoofing detection. The AO/FO approach resulted in accuracies of 98.93% and 99.38% on separate datasets with multiple biometric modalities [20]. The outstanding classification results emphasized the interplay between the architecture and filter optimization approaches for the spoofing problem [20].

With this review we identified a few similarities in the mitigation techniques. The textured based methods are often used in combination in with deep learning techniques. Deep learning based techniques showed significant improvement from baseline in classification accuracies for spoofing attacks. We also found similarities in the preprocessing of biometric samples before the use of the techniques. Although we can identify the periocular region as a strength in biometric recognition, gaps exist in finding the distinct features in the periocular region that mitigate spoofing attacks.

3. Methodology

In this section, we provide an overview of our proposed methodology including a novel deep learning approach to generate a synthetic (spoofing) dataset. We will then address the feature extraction techniques for biometric recognition. This research proposes to contribute empirical findings in improving biometric recognition against presentation attacks using deep learning techniques.

3.1. Generating spoofing datasets

In our experiment, we will use *Generative Adversarial Networks (GANs)* to generate a synthetic (spoofing) dataset. GANs to generate a synthetic (spoofing) dataset. GANs are a novel and promising approach to various problems that involve generating photorealistic images [28]. The basic approach involves two competing neural networks. Fig. 1 shows an

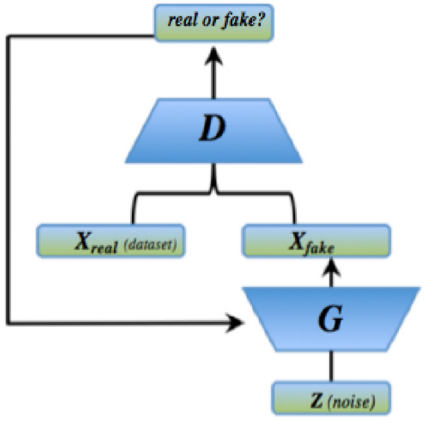

Fig. 1. Illustration of competing neural networks, discriminator D and generator G, in Generative Adversarial Networks (GAN) technique.

illustration of the GAN approach.

One is a discriminator, D, that attempts to determine whether a presented image is authentic or synthetic. The other is a generator, G, that tries to generate images that can successfully deceive the discriminator. The generator begins with a random noise (Z) input and continues generating samples with information from the discriminator. The two networks are trained in tandem, competing against the other, and each has information about the other's successes or failures. As the generator improves, the images become increasingly realistic to deceive the discriminator. In unison, the discriminator continually improves its discriminative capability, requiring the generated (spoofing) images to become more detailed in order to remain undetected. Training is often unstable with simple GANs, generating senseless noise for output [29, 30].

Advancements in GANs stability introduced a class of CNNs called Deep Convolutional Generative Adversarial Networks (DCGANs) [30,36, 37]. The proposed architecture is proven to be more stable in training GANs for image generation [30]. The architectural guidelines for stable DCGANs replace all pooling layers, in a typical CNN, with strided convolutions (discriminator) and fractionally-strided convolutions (generator). Batch normalization is used in both the generator and the discriminator to normalize the inputs to nonlinearities. All fully connected hidden layers are also removed. The use of ReLU and LeakyReLU activations are also used in all layer of the generator and discriminator, excluding the output. Fig. 2 shows a graphical model of the DCGAN architecture [30] (see Fig. 3).

3.2. Genetic and evolutionary feature extraction (GEFE)

GEFE, a technique proposed in Shelton et al. [13], is a hybrid of a

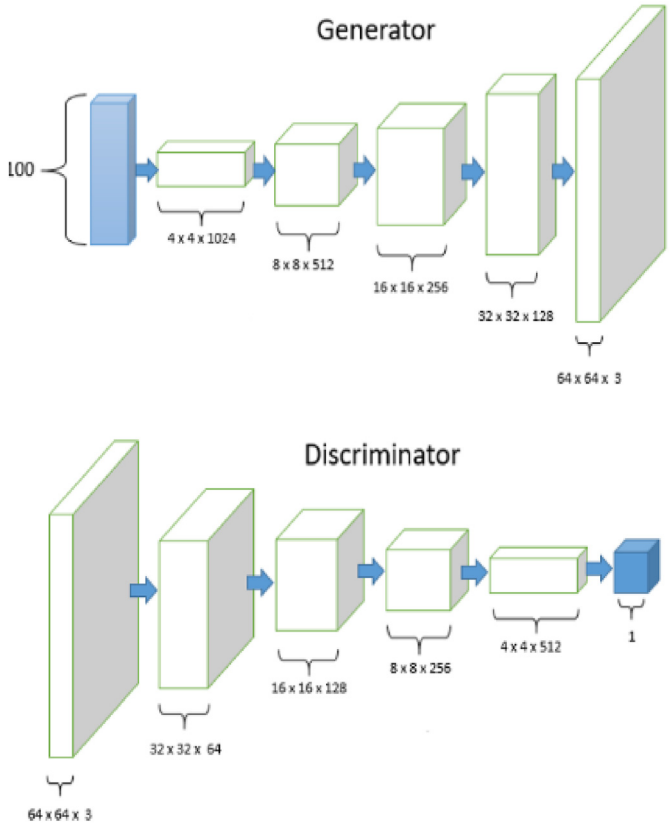

Fig. 2. DCGANs architecture uses a series of four fractionally-strided (5x5) convolutions. The first layer of the GAN takes a 100 dimensional uniform noise distribution as the input. For the discriminator, the last convolution layer is flattened and then fed into a single sigmoid output [30].

Fig. 3. Image samples from the same subject on iPhone5 inside and outside datasets were focused and cropped around the periocular regions. Resulting an image size of 64x64 pixel.

GEFE+GAN

```
x = 25//Number of subjects
y = 4//Number of Instances per subject
FV = {fv0,0, fv0,1, …, fv0,y, fv1,0, …, fvx,y}
numErrors = 0
closestSubj null, closestDist = MaxDist
for(i = 0; i < (x*y); i++){
 for(j = 0; j < (x*y); j++){
  if(i != j){
   if(Dist(FV[i], FV[j]) < closestDist){
        closestSubj = FV[j].id;
        closestDist = Dist(FV[i], FV[j]);
   }}}
  if(FV[i].id != closestSubj){
   if(FV[i].id == DCGAN sample){
        numErrors+=2
   }else numErrors++
   }
  closestSubj=null, closestDist = MaxDist
}
```

Fig. 4. GEFE + GAN technique presented in psuedocode; The fitness doubles the error value if the closest subject is a DCGAN (spoof) sample.

Genetic and Evolutionary Computations (GEC) to optimize the LBP feature extractors. A GEC is a general problem solving technique based on simulated evolution. A fitness function is used to compute the wellness of a solution and the best solutions procreate to create better solutions. The GEFE feature extractors (FE) optimize the texture-based feature extractors' dimensions and displacement on the image sample [13–16].

The ith candidate feature extractor, fe_i, consists of uniform patches. The candidate is a six-tuple, $<X_i, Y_i, W_i, H_i, M_i, f_i>$, where $X_i = \{x_{i,0}, x_{i,1}, \ldots, x_{i,n-1}\}$ represents the x-coordinates of the center of the n possible patches and $Y_i = \{y_{i,0}, y_{i,1}, \ldots, y_{i,n-1}\}$ represents the y-coordinates of center of the n possible patches. The widths and heights of the n patches are represented by $W_i = \{w_{i,0}, w_{i,1}, \ldots, w_{i,n-1}\}$ and $H_i = \{h_{i,0}, h_{i,1}, \ldots, h_{i,n-1}\}$. Because the patches are uniform, $W_k = \{w_{k,0}, w_{k,1}, \ldots, w_{k,n-1}\}$ is equivalent to $w_{k,0} = w_{k,1}, \ldots, w_{k,n-2}, w_{k,n-1}$ and $H_k = \{h_{k,0}, h_{k,1}, \ldots, h_{k,n-1}\}$ is equivalent to $h_{k,0} = h_{k,1}, \ldots, h_{k,n-2}, h_{k,n-1}$, meaning that the widths and heights of every patch

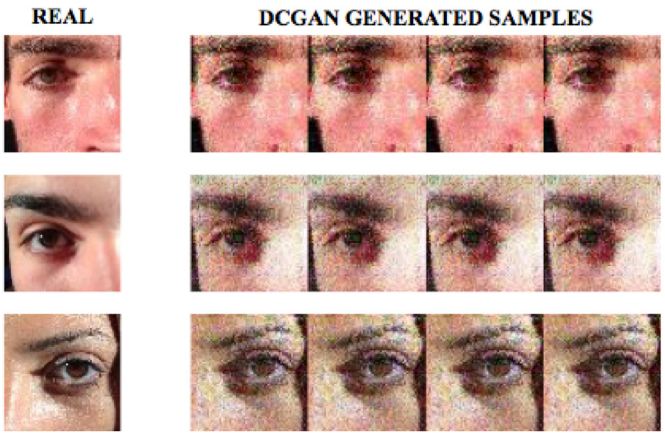

Fig. 5. Image samples from original Apple iPhone-inside (real) dataset and the DCGAN generated (counterpart) datasets for spoofing.

are the same. Uniform sized patches are used because uniform sized patches outperformed non-uniform sized patches [13]. $M_i = \{m_{i,0}, m_{i,1}, \ldots, m_{i,n-1}\}$ represents the masking values for each patch and f_i represents the fitness of fe_i. The masking value determines whether a patch is activated or deactivated, and by extension, whether certain pixels will be processed or not. If a patch is deactivated, by setting $m_i = 0$, then the sub-histogram will not be considered in the distance measure, and the number of features to be used is reduced. Otherwise, the patch is activated, with $m_i = 1$ [13–16].

The fitness, fi, is determined by applying fei towards a dataset of subject's biometric images. A subject has a number of images that vary, and these images are separated into a probe set and a gallery set (G). The fe_i is applied on these images to create FVs, and the FVs in the probe set are compared to all of the FVs in the gallery set using the Manhattan distance measure. The two FVs that have the least Manhattan distance are considered to be matches. If a probe FV is incorrectly matched with a gallery FV, then fe_i is said to cause an error. The fe_i also considers how much surface area of the image it covers. The resulting fe_i is the number of errors (ε) added to the percent of patches not masked out (the summation of all masking values over the total number of patches, (n)). Both the error and percentage of patches must be reduced so the fitness is being minimized.

Traditionally, GEFE is applied on a simulation of a biometric identification system. To simulate this, K biometric images are collected from N different individuals. One image per individual will represent a biometric sample being identified and the K-1 images of all N individuals represents the database of known individuals. A GEFE FE is applied on every image and feature vectors (FVs) are created for each. Each FV that is to be identified is compared to the database of FVs and the most similar match is considered the identity of the unknown FV. If there is a mismatch, the fitness of the FE is penalized. With this scheme, there are $N*(N*(K-1))$ comparisons made.

The fitness is based on the number of correct matches between instances, so the best fitness occurs when discriminating features are extracted from each instance. The more discriminating the features, the more likely that the resulting FVs from the same individuals will have a better similarity score than FVs from different individuals. The $GEFE_{many}$ scheme, proposed in Jenkins et al. [15], compares every instance in a dataset into every other instance. The instances involved do not change from the dataset used by the traditional GEFE method, but the number of comparisons is now $(N*K)*((N*K)-1)$. The number of comparisons is now greater, thus allowing for a more complex environment to evolve FEs in. Though the number of instances has not changed, the increase of instance comparisons results in FEs that have a superior identification performance.

The proposed methodology for this work is to use the DCGAN architecture as a data augmentation technique to generate a spoofing dataset. The generated spoofing dataset will be tested for its quality and used in the GEFE technique to generate improved feature extractors that can mitigate presentation attacks. Similar to the combination of BSIF and 2D Ceptrum [24], the GEFE technique has been used previously to optimize LBP FEs to mitigate replay attacks with high biometric recognition accuracy. The proposed GEFE technique will be optimized to mitigate synthetic images. The GEFE FEs also will provide insight on which biometric features have the higher discriminative capabilities as GEFE optimizes the dimensions and displacement of FEs on the biometric sample.

4. Experiment

This research applies the DCGAN and GEFE techniques to periocular images from the BIPLab MICHE-I dataset [32]. The periocular images from the BIPLab MICHE-I datasets are taken from the Apple iPhone5 and Samsung GalaxyS4 smartphones. From this dataset, we form two subsets: the iPhone5 dataset and the Galaxy dataset. For the iPhone5 datasets, images were taken from the FaceTime HD (front) camera of 1.2 megapixels (MP). For the GalaxyS4 datasets, images were taken from the CMOS (front) camera of 2.0 MP. For the BIPLab dataset, 25 subjects were used with 8 periocular samples per subject. Image samples consisted of 4 indoors using artificial light and 4 taken outdoors using natural light. All images from the BIPLab dataset were, taken 10 cm away from the device. Comparisons among image data were done using the approach described in the Methodology section where there are K images collected from N different individuals.

Originally, images from iPhone5 & GalaxyS4 datasets had a resolution of 480x640 pixel or greater. For ease of image generation, classification, and recognition, all images were resized and cropped to 64x64 pixel; similar to protocols in previous research [17,18]. The images were zoomed in to ensure coverage of the periocular region. In cropping the image, superfluous noise, such as the shirt textures, hair textures, and other background noises are removed. Cropping also allows flexibility when images are not taken exactly as directed. The 64x64 pixel size also works best for this experiment as the DCGAN technique generates four 64x64 pixel sized images as the output. All four samples for each subject are used for input comparison (discriminator) in the DCGAN technique. This DCGAN is trained for 2500 epochs. The DCGAN techniques will use a learning rate of 0.0005. This process generates four unique synthetic samples for each subject. These generate samples are placed in the gallery set of the original datasets to test the technique's accuracy.

First we test the quality of the images generated by the DCGAN technique. The spoofing samples will be added to the gallery set of a simple LBP-3x3 identification system. An LBP 3x3 system means that images are split into three rows and three columns of even sized regions where features are extracted from each region and concatenated to form the feature vector for an image. The identification and verification accuracies will be recorded and compared, that with the original dataset and the spoofing datasets as well. These results will show us the quality of DCGAN generated samples and the power of Generative Adversarial Networks. Next, we will use the spoofing datasets for training with the GEFE technique to generate improved feature extractors. During this process, the GEFE technique will not only be looking for distinct features for periocular recognition, but also features to mitigate presentation attacks. GEFE uses the Estimation of Distribution Algorithm (EDA) [31] as the GEC with 250 function evaluations with a population size of 20 candidates FEs.

All subject samples are used for input in DCGAN technique to generate corresponding spoofing datasets. This DCGAN technique trains on a GPU for 2500 epochs. The DCGAN techniques will use a learning rate of 0.0005. This DCGAN technique generates equal unique synthetic samples for each subject. These generate samples are then placed in the gallery set of the original datasets to test the technique's accuracy; doubling the original dataset in size.

For the GEFE experiments, we will use a 60/40 training/testing set split. $GEFE_{many}$ is applied on the original datasets for a baseline accuracy; referred as GEFE in our experiments. The GEFE technique is then applied on the spoofing datasets with the DCGAN generated samples added to the gallery set; referred to as GEFE + GAN in our experiments. GEFE + GAN is optimized for DCGAN generated presentation attacks. The original GEFE ($GEFE_{many}$) technique applies an equal value of error for all non-matching subjects. In the GEFE + GAN2 technique, the error value is double for matching a subject with a DCGAN generated sample during the evolutionary process. By doubling the error value for matching subjects with a DCGAN generated samples, the evolved feature extractors are less likely to match a presentation attack over real subjects. This is proposed to optimize the anti-spoofing fitness of the generated feature extractors. This is also shown in the pseudocode below.

The identification performance will be presented using Cumulative Match Characteristic (CMC) curves. As each probe sample is compared against all gallery samples, the resulting scores are sorted and ranked. This determines the True Positive Identification Rate (TPIR); the rank at which a true match occurs. CMC curves plot the TPIR against ranks. The verification performance will be presented using Receiver Operator Characteristic (ROC) curves. As each probe sample is compared against all gallery samples, the True Accept Rate (TAR) and False Accept Rate (FAR) of subjects are calculated at multiple thresholds. TAR is the measure of likelihood that the biometric security system will correctly verifies a true claim of identity. The FAR is the measure of likelihood a system incorrectly accepts an access attempt by an unauthorized user. The TAR and FAR are stated as the ratio of the number of true and false acceptances divided by the number of identification attempts.

5. Results and discussion

This research applied the LBP feature extraction technique on datasets with samples generated from the DCGAN (see Fig. 4). For the LBP identification system, the first sample of each subject is placed into a probe set. All other samples are placed in a gallery set. The system compares each subject's probe sample to the gallery set, containing the other samples of the current subject as well as the samples of other subjects in the system. For calculating the identification fitness for the LBP histograms, the Manhattan distance metric was used.

Figs. 6–13, below, shows the Cumulative Match Characteristic (CMC) curves and the Receiver Operator Characteristic (ROC) curves on the iPhone5 datasets (see Fig. 5). The CMC curve plots the identification accuracies of each method, while the ROC curve plots the verification accuracies. The CMC curve plots the rank at which a true match occurs The ROC curve plots the True Accept Rate (TAR) and False Accept Rate (FAR) of subjects calculated at multiple thresholds.

Fig. 6 shows the CMC results of GEFE generated feature extractors on the iPhone5-inside dataset. GEFE + GAN is the first technique to reach 100% at Rank 4, followed by GEFE at Rank 5. LBP reaches 100% at Rank 14. Fig. 7 shows the ROC results (Loged Scaled) of the GEFE generated feature extractors on the iPhone5-inside dataset. LBP has a TAR of 0.8 at a FAR of 0.04. GEFE has a TAR of 0.811 at a FAR of 0.0402. GEFE + GAN has a TAR of 0.757 at a FAR of 0.04.

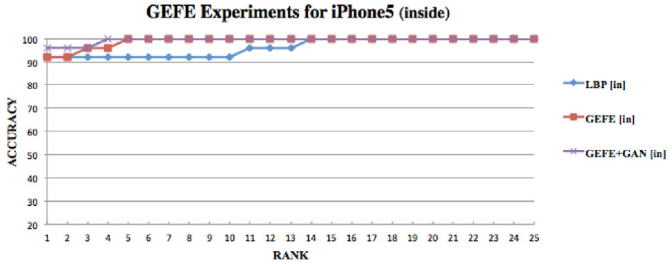

Fig. 6. CMC results for LBP and GEFE techniques on the iPhone5-inside dataset.

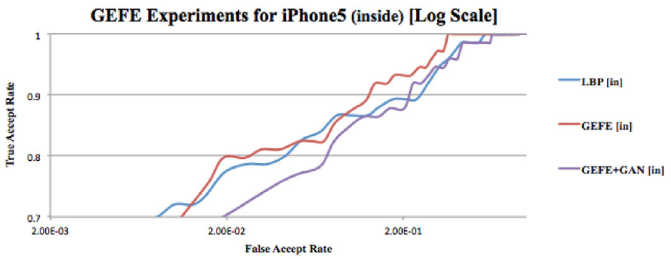

Fig. 7. ROC results for LBP and GEFE techniques on the iPhone5-inside dataset.

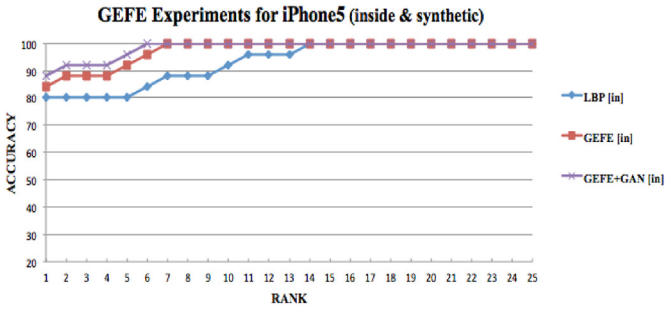

Fig. 8. CMC results for LBP and GEFE techniques on the iPhone5-inside dataset with DCGAN samples added.

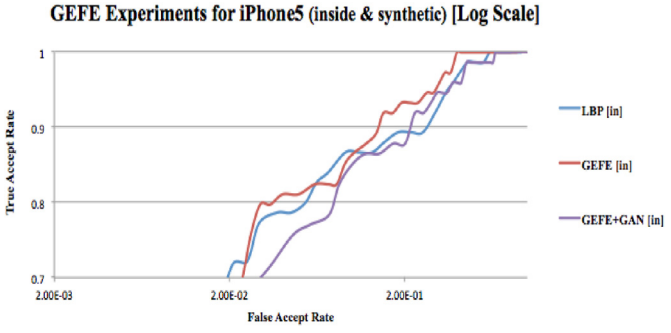

Fig. 9. ROC results for LBP and GEFE techniques on the iPhone5-inside dataset with DCGAN samples added.

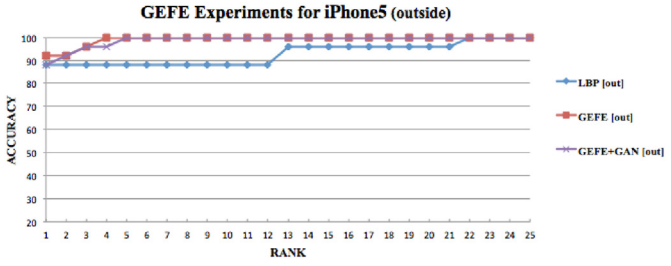

Fig. 10. CMC results for LBP and GEFE techniques on the iPhone5-outside dataset.

Fig. 8 shows the CMC results of GEFE generated feature extractors on the iPhone5-inside dataset with DCGAN samples added. GEFE + GAN is the first technique to reach 100% at Rank 6, followed by GEFE at Rank 7. LBP reaches 100% at Rank 14. Fig. 9 shows the ROC results (Log Scaled) of the GEFE generated feature extractors on the iPhone5-inside dataset with DCGAN samples added. LBP has a TAR of 0.787 at a FAR of 0.0452. GEFE has a TAR of 0.811 at a FAR of 0.0402. GEFE + GAN has a TAR of 0.757 at a FAR of 0.0461. Soon after, GEFE has a TAR of 0.892 at a FAR of

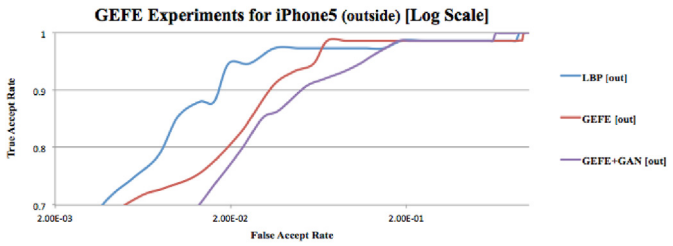

Fig. 11. ROC results for LBP and GEFE techniques on the iPhone5-outside dataset.

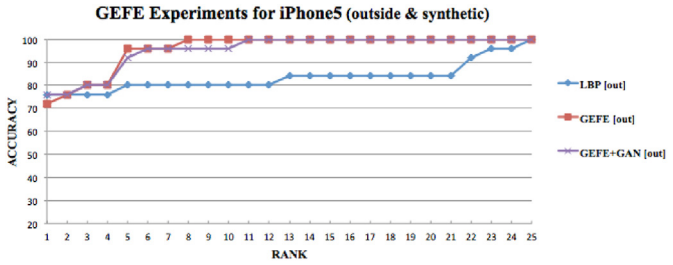

Fig. 12. CMC results for LBP and GEFE techniques on the iPhone5-outside dataset with DCGAN samples added.

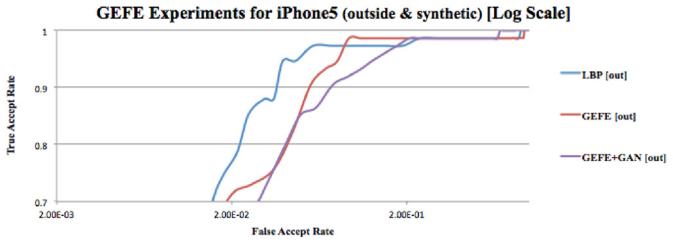

Fig. 13. ROC results for LBP and GEFE techniques on the iPhone5-outside dataset with DCGAN samples added.

0.138 while GEFE + GAN has a TAR of 0.865 at a FAR of 0.143.

Fig. 10 shows the CMC results of GEFE generated feature extractors on the iPhone5-outside dataset. GEFE is the first technique to reach 100% at Rank 4, followed by GEFE + GAN at Rank 5. LBP doesn't reaches 100% until Rank 22. Fig. 11 shows the ROC results (Log Scaled) of the GEFE generated feature extractors on the iPhone5-outside dataset. LBP has a TAR of 0.947 at a FAR of 0.0194. GEFE has a TAR of 0.824 at a FAR of 0.0225. GEFE + GAN has a TAR of 0.797 at a FAR of 0.0231.

Fig. 12 shows the CMC results of GEFE generated feature extractors on the iPhone5-outside dataset with DCGAN samples added. At Rank 5, GEFE reach 96% as GEFE + GAN reaches 92%. GEFE is the first technique to reach 100% at Rank 8, followed by GEFE + GAN at Rank 11. LBP doesn't reaches 100% until Rank 25. Fig. 14 shows the ROC results (Log Scaled) of the GEFE generated feature extractors on the iPhone5-outside dataset with DCGAN samples added. LBP has a TAR of 0.950 at a FAR of 0.0392. GEFE has a TAR of 0.824 at a FAR of 0.0450. GEFE + GAN has a TAR of 0.851 at a FAR of 0.0493.

Figs. 14–20, below, shows the Cumulative Match Characteristic (CMC) curves and the Receiver Operator Characteristic (ROC) curves on the GalaxyS4 datasets. The CMC curve plots the identification accuracies of each method, while the ROC curve plots the verification accuracies. The CMC curve plots the rank at which a true match occurs The ROC curve plots the True Accept Rate (TAR) and False Accept Rate (FAR) of subjects calculated at multiple thresholds.

Fig. 14 shows the CMC results of the GEFE generated feature extractors on the GalaxyS4-inside dataset. GEFE + GAN is the first technique to reach 100% at Rank 8. All other techniques are 96% at Rank 8. The GEFE technique reaches 100% at Rank 14, followed by LBP at Rank 15. Fig. 15 shows the ROC results (Log Scaled) of the GEFE generated feature extractors on the GalaxyS4-inside dataset. LBP has a TAR of 0.7733 at a FAR of 0.08611. GEFE has a TAR of 0.8378 at FAR of 0.0895. GEFE + GAN has a TAR of 0.8108 at a FAR of 0.0957. The ROC curves begin to alter as GEFE + GAN technique surpasses the GEFE techniques. GEFE + GAN technique has a TAR of 0.8649 at a FAR of 0.128. The GEFE technique has a TAR of 0.8514 at a FAR of 0.1402.

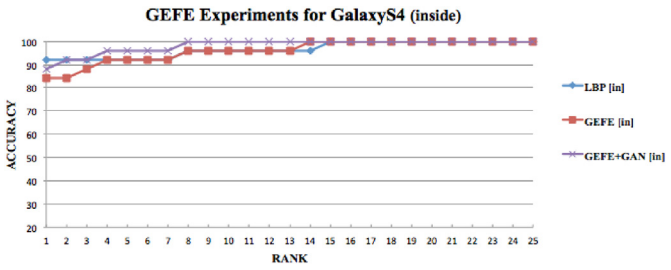

Fig. 14. CMC results for LBP and GEFE techniques on the GalaxyS4-inside dataset.

Fig. 16 shows the CMC results of the GEFE generated feature extractors on the GalaxyS4-inside dataset with DCGAN (synthetic) samples added. GEFE + GAN is the first technique to reach 100% at Rank 17. This is followed by the GEFE techniques reaching 100% at Rank 20 and LBP at Rank 25. The GEFE technique is the first reach 92% at Rank 5, followed by GEFE + GAN at Rank 11. Fig. 17 shows the ROC (Log Scaled) of the GEFE generated feature extractors on the GalaxyS4-inside dataset with DCGAN (synthetic) samples added. LBP has a TAR of 0.82667 at a FAR of 0.1333. GEFE has a TAR of 0.8514 at a FAR of 0.1235. GEFE + GAN has a TAR of 0.7838 at a FAR of 0.130.

Fig. 18 shows the CMC results of the GEFE generated feature extractors on the GalaxyS4-outside dataset. GEFE + GAN is the first technique to reach 100% at Rank 2, followed by GEFE at Rank 3. The LBP technique reaches 100% at Rank 19. Fig. 19 shows the ROC results (Loged Scaled) of the GEFE generated feature extractors on the GalaxyS4-outside dataset. LBP has a TAR of 0.8133 at a FAR of 0.03944. GEFE has a TAR of 0.8514 at a FAR of 0.0376. GEFE + GAN has a TAR of 0.8649 at a FAR of 0.0355.

Fig. 20 shows the CMC results of the GEFE generated feature extractors on the GalaxyS4-outside dataset with DCGAN (synthetic) samples added. Both GEFE and GEFE + GAN reach 100% at Rank 5. LBP reach 96% at Rank 13, but doesn't reach 100% until Rank 23. Fig. 21 shows the ROC (Log Scaled) of the GEFE generated feature extractors on the GalaxyS4-outside dataset with DCGAN (synthetic) samples added. LBP has a TAR of 0.8 at a FAR of 0.0373. GEFE has a TAR of 0.8243 at a FAR of 0.0355. GEFE + GAN has a TAR of 0.7973 at a FAR of 0.0337 (see Fig. 22).

The GEFE ($GEFE_{many}$) and GEFE + GAN techniques are applied to a total of eight datasets (real and spoofing). Both GEFE techniques used an EDA as the GEC, with 250 function evaluations, and a population size of 20 candidates FEs. The identification performances are presented using Cumulative Match Characteristic (CMC) curves. CMC curves plot the True Positive Identification Rate (TPIR), the rank at which a true match occurs. Table 1 shows a summary of the identification accuracies for GEFE experiments on iPhone5 datasets. For the iPhone5-inside datasets, GEFE reaches 100% at Rank 5. For the iPhone5-inside spoofing datasets, 100% is reached at Rank 7. Also for the iPhone5-inside dataset, GEFE + GAN reaches 100% at Rank 4, and Rank 6 on spoofing dataset. For the iPhone5-outside datasets, GEFE reaches 100% at Rank 4, and Rank 8 on the spoofing dataset. Also for the iPhone5-outside dataset, GEFE + GAN reaches a 100% at Rank 5. GEFE + GAN reaches 100% on iPhone5-outside spoofing dataset at Rank 11.

Table 2 shows a summary of the identification accuracies for GEFE

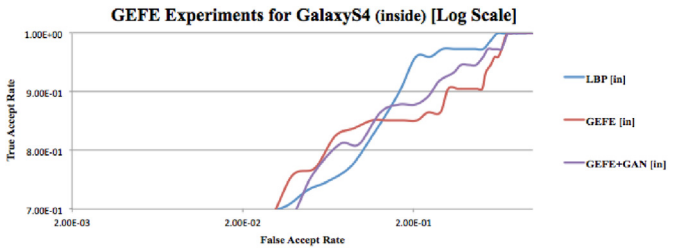

Fig. 15. ROC results for LBP and GEFE techniques on the GalaxyS4-inside dataset.

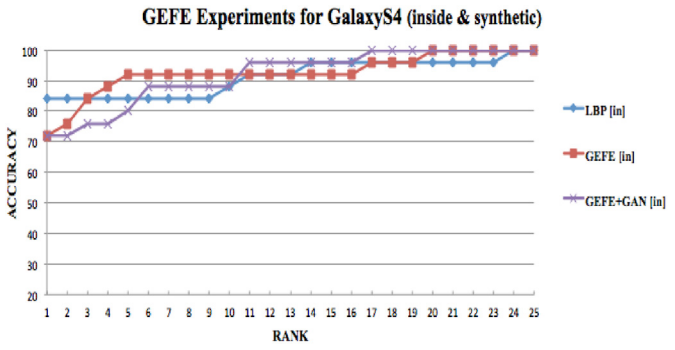

Fig. 16. CMC results for LBP and GEFE techniques on the GalaxyS4-inside dataset with DCGAN samples added.

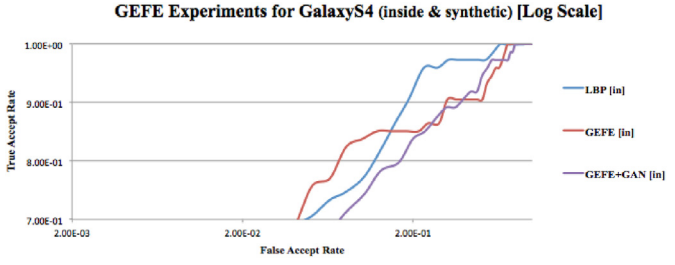

Fig. 17. ROC results for LBP and GEFE techniques on the GalaxyS4-inside dataset with DCGAN samples added.

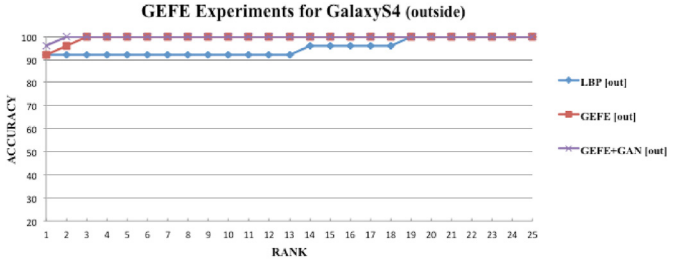

Fig. 18. CMC results for LBP and GEFE techniques on the GalaxyS4-outside dataset.

experiments on GalaxyS4 datasets. For the GalaxyS4-inside datasets, GEFE reaches 100% at Rank 14 (96% at Rank 8). For the iPhone5-inside spoofing datasets, 100% is reached at Rank 20 (92% at Rank 5). Also for the GalaxyS4-inside dataset, GEFE + GAN reaches 100% at Rank 8, and Rank 17 on spoofing dataset. For the GalaxyS4-outside datasets, GEFE reaches 100% at Rank 3, and Rank 5 on the spoofing dataset. Also for the GalaxyS4-outside dataset, GEFE + GAN reaches a 100% at Rank 2. GEFE + GAN reaches 100% on GalaxyS4-outside spoofing dataset at Rank 5. The GEFE + GAN technique showed promising results with higher

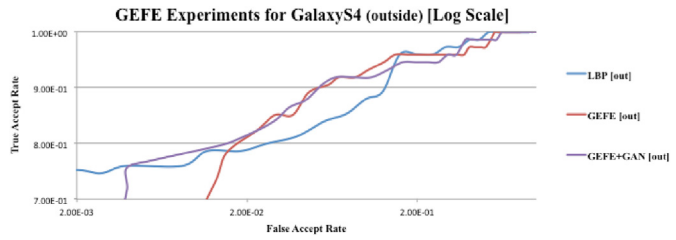

Fig. 19. ROC results for LBP and GEFE techniques on the GalaxyS4-outside dataset.

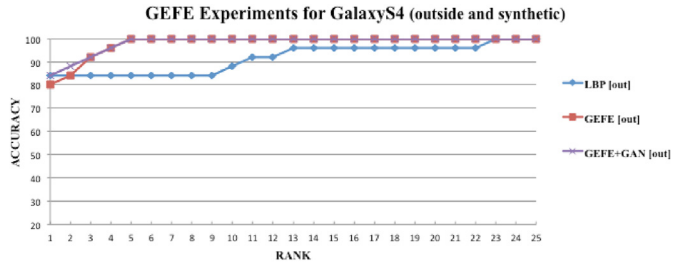

Fig. 20. CMC results for LBP and GEFE techniques on the GalaxyS4-outside dataset with DCGAN samples added.

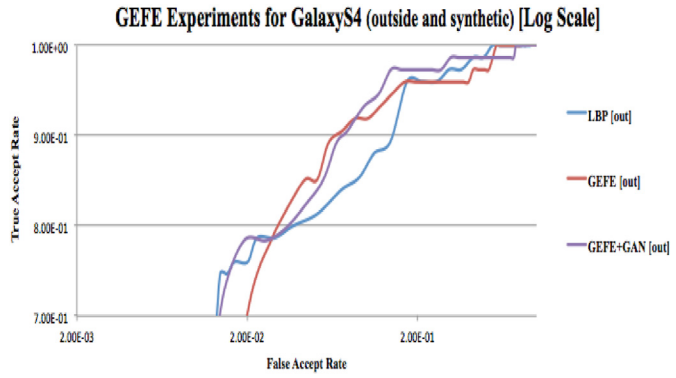

Fig. 21. ROC results for LBP and GEFE techniques on the GalaxyS4-outside dataset.

Fig. 22. GEFE (left) and GEFE + GAN (right) generated feature extractors (FEs) illustrated on biometric samples.

identification accuracies than the GEFE technique on six out of eight

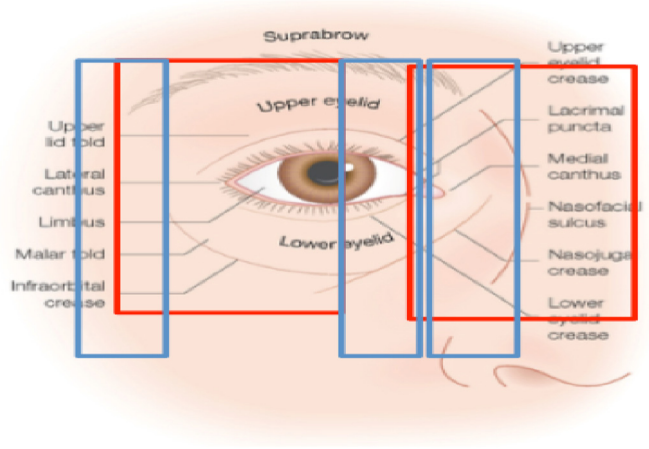

Fig. 23. GEFE (red) and GEFE + GAN (blue) generated feature extractors (FEs) transposed a detailed illustration of the periocular region. The illustration details the surface anatomy [34].

Table 1
Summary of Identification Accuracies for. GEFE Experiments on iPhone5 datasets.

Dataset	Technique	CMC Rank at 100%	
		Real	Spoof
iPhone5-inside	LBP	14	14
	GEFE	5	7
	GEFE + GAN	4	6
iPhone5-outside	LBP	22	25
	GEFE	4	8
	GEFE + GAN	5	11

Table 2
Summary of Identification Accuracies for. GEFE Experiments on GalaxyS4 datasets.

Dataset	Technique	CMC Rank at 100%	
		Real	Spoof
GalaxyS4-inside	LBP	15	25
	GEFE	14	20
	GEFE + GAN	8	17
GalaxyS4-outside	LBP	19	23
	GEFE	3	5
	GEFE + GAN	2	5

datasets. Verification results are also comparable on all datasets. The results shows that the GEFE + GAN technique optimizes the anti-spoofing fitness of the generated feature extractors. (see Fig. 23).

For the GEFE FEs, the experimental results show that the majority of the eyes and eyelids are chosen for extraction. The FEs tends to favor the upper and lower lid folds. The suprabrow and upper eyelid are covered by the FEs. The FEs also covers the most of the nasal area. For the GEFE + GAN FEs, one could extrapolate from the displacement that the canthi (eye corners) are keys to identification and presentation attack mitigation. The generated FEs favor the lateral and medial canthus over using the entire eye area. The malar fold is the defined groove extending from the lateral canthus. The nasojungal crease is also a defined groove in the lower lid fold, between the medial canthus and the nose [34]. The nasofacial sulcus is covered by both GEFE FEs, but more significantly on the outside dataset; instead of the entire nose. It seems the natural light in the iPhone5-outside datasets exposes the periocular features better for discriminative capabilities. These areas could all be outlining features used to mitigate presentation attacks with synthetic images.

6. Conclusion and future reasearch

In this paper, we apply a data augmentation technique with GANs to generate comparative synthetic (spoofing) dataset. We trained the synthetic images using GAN and created spoofing datasets. The GEFE technique is then used in combination with the GANs to generate improved anti-spoofing feature extractors optimized in an attempt to mitigate presentation attacks. The combination of GEFE and GANs is used to identify those discriminative biometric features used to mitigate synthetic presentation attacks. The GEFE + GAN technique outperforms the LBP and GEFE techniques alone in overall identification and verification results on spoofing datasets.

To improve upon the GEFE + GAN technique, we would continue by applying experiments on images taken from the UBI periocular recognition datasets [38]. The UBI dataset uses images taken from a professional camera, the Canon EOS 5D (12.8 MP). The UBI dataset also has 15 periocular samples per subject. With an increase of samples, the DCGAN generated samples should be more accurate. With higher quality synthetic samples, the GEFE experiments should increase the quality of discriminative feature extraction.

CRediT authorship contribution statement

John Jenkins: Writing - original draft, Writing - review & editing, Software, Investigation. **Kaushik Roy:** Supervision, Writing - original draft, Conceptualization, Funding acquisition. **Joseph Shelton:** Writing - original draft, Writing - review & editing, Conceptualization.

Acknowledgment

This research is based upon work supported by the Army Research Office (Contract No. W911NF-15-1-0524) and National Science Foundation (Award Number:1900187). 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 Army Research Office and National Science Foundation.

References

[1] EyeLock debuts first laptop with embedded EyeLock ID Iris authentication technology - eyeLock. Retrieved April 1, 2018.
[2] Ratha N, Connell J, Bolle R. Enhancing security and privacy in biometrics-based authentication systems. IBM Syst. J. IBM Systems Journal 2001:614–34.
[3] Hassan A, Bhram A, Saleh A. Applying a new functional model to improve the security of biometric systems. In: 3rd international conference on communication software and networks; 2011.
[4] Jain A k, Ross A, Prabhakar S. An introduction to biometric recognition. IEEE Transactions on Circuits and for Video Technology 2004:4–20.
[5] Hao Zhuo, Yu Nenghai. A security enhanced remote password authentication scheme using smart card. In: Second international symposium on data, privacy, and E-commerce; 2010. p. 56–60.
[6] Lamport L. Password authentication with insecure communication. Commun ACM 1981;24(11):770–2.
[7] Robert W Frischholz, Dieckmann Ulrich. BioID: a multimodal biometric identification system,". Computer Feb. 2000;33(2):64–8. https://doi.org/10.1109/2.820041.
[8] O'Gorman L. "Comparing passwords, tokens, and biometrics for user authentication,". Proc IEEE Dec. 2003;91(12):2019–40.
[13] Shelton, J., Dozier, G. V., Bryant, K. S., Small, L., Adams, J., Popplewell, K.,... & Ricanek, K. (2011, April). Comparison of genetic-based feature extraction methods for facial recognition. In MAICS (pp. 216–220).
[14] Alford A, Steed C, Jeffrey M, Sweet D, Shelton J, Small L, Kelly JC. Genetic & Evolutionary Biometrics: hybrid feature selection&weighting for a multi-modal biometric system. IEEE; 2012. p. 1–8.
[15] Jenkins, J., Shelton, J., & Roy, K. (2016, October). A comparison of genetic based extraction methods for periocular recognition. In Information Communication and Management (ICICM), International Conference on (pp. 309–313). IEEE.
[16] Shelton J, Jenkins J, Roy K. Extending disposable feature templates for mitigating replay attacks. Int J Inf Priv Secur Integr 2017;3(2):96–116.
[17] Gragnaniello D, Poggi G, Sansone C, Verdoliva L. An investigation of local descriptors for biometric spoofing detection. IEEE Trans Inf Forensics Secur 2015;10(4):849–63.
[18] Bharati A, Singh R, Vatsa M, Bowyer KW. Detecting facial retouching using supervised deep learning. IEEE Trans Inf Forensics Secur 2016;11(9):1903–13.

[19] Yang J, Lei Z, Li S. Learn convolutional neural network for face anti-spoofing. 2014. arXiv preprint arXiv:1408.5601.
[20] Menotti D, Chiachia G, Pinto A, Schwartz WR, Pedrini H, Falcão AX, Rocha A. Deep representations for iris, face, and fingerprint spoofing detection. IEEE Trans Inf Forensics Secur 2015;10(4):864–79.
[22] Rodrigues RN, Kamat N, Govindaraju V. Evaluation of biometric spoofing in a multimodal system. 2010, September. In: Biometrics: theory applications and systems (BTAS); 2010. Fourth IEEE International Conference on (pp. 1–5). IEEE.
[23] Chingovska I, Anjos A, Marcel S. On the effectiveness of local binary patterns in face anti-spoofing. 2012, September. In: Biometrics special interest group (BIOSIG); 2012. BIOSIG-Proceedings of the International Conference of the (pp. 1–7). IEEE.
[24] Raghavendra R, Busch C. Presentation attack detection algorithm for face and iris biometrics. 2014, September. In: Signal processing conference (EUSIPCO); 2014. Proceedings of the 22nd European (pp. 1387–1391). IEEE.
[25] Raghavendra R, Venkatesh S, Raja KB, Busch C. Transferable deep convolutional neural network features for fingervein presentation attack detection. 2017, April. In: Biometrics and forensics (IWBF); 2017. 5th International Workshop on (pp. 1–5). IEEE.
[26] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 2002;24(7):971–87.
[27] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 2006;28(12): 2037–41.
[28] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y. Generative adversarial nets. Adv Neural Inf Process Syst 2014:2672–80.
[29] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Shi W. Photo-realistic single image super-resolution using a generative adversarial network. 2016. arXiv preprint arXiv:1609.04802.
[30] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. arXiv preprint arXiv: 1511.06434.
[31] Larranaga P. A review on estimation of distribution algorithms. In: Estimation of distribution algorithms. Boston: Springer; 2002. p. 57–100.
[32] De Marsico M, Nappi M, Riccio D, Wechsler H. "Mobile Iris Challenge Evaluation (MICHE), biometric iris dataset protocols,". Pattern Recogn Lett 2015:17–23.
[34] Bowman PH, Fosko SW, Hartstein ME. Periocular reconstruction. Semin Cutan Med Surg 2003;22(4):263–72.
[35] Miller Arthur I. 10 ian goodfellow's generative adversarial networks: AI learns to imagine,". The Artist in the Machine: The World of AI-Powered Creativity, MITP 2019:87–98.
[36] Chesney B, Citron D. Deep fakes: a looming challenge for privacy, democracy, and national security. Calif Law Rev 2019;107:1753.
[37] Korshunov P, Marcel S. Deepfakes: a new threat to face recognition? assessment and detection. 2018. arXiv preprint arXiv:1812.08685.
[38] Padole C, Proença H. Periocular recognition: analysis of performance degradation factors, in proceedings of the fifth IAPR/IEEE international conference on biometrics – ICB 2012. 2012. New Delhi, India, March 30.