Active Authentication Using an Autoencoder Regularized CNN-based One-Class Classifier

Poojan Oza and Vishal M. Patel

Department of Electrical and Computer Engineering,

Johns Hopkins University, 3400 N. Charles St, Baltimore, MD 21218, USA

email: {poza2, vpatel36}@jhu.edu

Abstract— **Active authentication refers to the process in which users are unobtrusively monitored and authenticated continuously throughout their interactions with mobile devices. Generally, an active authentication problem is modelled as a one class classification problem due to the unavailability of data from the impostor users. Normally, the enrolled user is considered as the target class (genuine) and the unauthorized users are considered as unknown classes (impostor). We propose a convolutional neural network (CNN) based approach for one class classification in which a zero centered Gaussian noise and an autoencoder are used to model the pseudo-negative class and to regularize the network to learn meaningful feature representations for one class data, respectively. The overall network is trained using a combination of the cross-entropy and the reconstruction error losses. A key feature of the proposed approach is that any pre-trained CNN can be used as the base network for one class classification. Effectiveness of the proposed framework is demonstrated using three publically available face-based active authentication datasets and it is shown that the proposed method achieves superior performance compared to the traditional one class classification methods. The source code is available at : github.com/otkupjnoz/oc-acnn.**

Fig. 1: An overview of a typical AA system. (a) Data corresponding to the enrolled user are used to train an AA system. (b) During testing, data corresponding to the enrolled user as well as unknown user may be presented to the system. The AA system then grants access to the enrolled user and blocks access to unknown users.

I. INTRODUCTION

Biometric information based on physical characteristics such as face, iris, fingerprints etc., have been an integral part of security (e.g., airport security, law enforcement) and authentication systems (e.g., banking transactions). Physiological biometrics are never lost or forgotten and are difficult to forge, which gives such systems an edge over the traditional passcode-based approaches [1]. Moreover, these methods are convenient, user-friendly and secure compared to the traditional explicit authentication mechanisms. Due to these reasons, biometric-based mobile authentication systems are receiving increased attention in recent years.

Since the introduction of the first fingerprint authentication-based mobile device in 2013, the smartphone industry has introduced a variety of mobile devices that make use of biometric authentication. According to a recently published study, by the year 2020 almost 100% of the smartphones will use biometric authentication as a standard feature [2]. Furthermore, recent years have witnessed a major shift, where smartphone makers have started moving away from password-based or fingerprint-based systems to face-based authentication systems. Considering the advances

in face recognition, it won't be too long before face-based authentication becomes a norm for mobile devices.

However, face-based verification systems are explicit in nature and are fundamentally limited. For instance, as long as the mobile phone remains active, typical devices incorporate no mechanisms to verify that the user originally authenticated is still the user in control of the mobile device. Thus, unauthorized individuals may improperly obtain access to personal information of the user if the system is spoofed or if the user does not exercise adequate vigilance after initial authentication. To deal with this issue, Active Authentication (AA) systems were introduced in which users are continuously monitored after the initial access to the mobile device [3]. Various methods have been proposed for AA including touch gesture-based [4], [5], [6], [7], gait pattern-based [8] and face-based [9], [10], [11], [12], [13] systems. In particular, face-based AA systems have gained a lot of attraction in recent years.

A naive approach for face-based AA would be to use face images correspoinding to all users and train a system to classify each user in a multi-class fashion. However, such an approach becomes counter-intuitive for AA since it requires the storage of all face images at a centralized location, raising data privacy issues [14]. Hence, one must

consider only the data collected from the enrolled user to develop an AA system. In other words, we need to explore possibilities of implementing AA systems using only the user's enrolled data. This motivates us to view AA as a one class classification problem [15]. Fig. 1 shows a typical face-based AA system, modelled as a one class classification problem. For the rest of this paper, we term the enrolled user and unauthorized user data as the target class and unknown class data, respectively.

Learning a one class classifier based on only the target class data has been one of the most challenging problems in machine learning. Some of the earlier works have used statistical methods to tackle this problem. Such statistical methods usually seek separating hyperplane/hypersphere in the feature space to enclose the target class data [16], [17], [18], [19]. These methods rely on the quality of the representations used for the target class data. Earlier approaches were based on the hand-crafted features. In recent years features based on Deep Convolutional Neural Networks (CNNs) have shown to produce better results than hand-crafted features. Several one class approaches have also been proposed in the literature that instead of learning a decision boundary, try to leverage CNNs to learn representations for the target class data. Most of these approaches are based on generative methods such as Auto-Encoders and Generative Adversarial Networks (GANs) [20]. These generative approaches either use reconstruction errors or discriminator scores to identify the target class data [21], [22], [23], [24], [25]. However, features learned using these generative approaches are not discriminative enough compared to the features learned in traditional discriminative fashion. Attempts have been made to combine deep features with the statistical classifiers for learning better representations for one class classification [26], [27]. Though, utilizing these powerful feature representations help in learning good decision boundaries, feature representations and classifiers are learned separately. In such a disjoint approach, classification module doesn't influence CNNs to modify the feature representation for a given target class data. Several recent works have explored joint learning of both features and classifiers [28], [29] for one class classification. These methods demonstrated that representation learning together with classifier training results in improved performance. Based on this motivation, an end-to-end learning approach is proposed in this paper which jointly learns feature representations and a classifier for one class classification. Furthermore, the learned representations are constrained by a decoder network which regularizes the learned representations by enforcing them to reconstruct the original data. In summary, this paper makes the following contributions:

- A new method is proposed for jointly learning representations and the decision boundary for one class classification.
- A key feature of the proposed approach is that any pre-trained CNN architecture (i.e. AlexNet, VGG16 and VGGFace) can be used as the base network.

- Extensive experiments are conducted on three face-based AA datasets and it is shown that the proposed approach can outperform many other statistical and deep learning-based one class classification approaches.

II. RELATED WORK

Many statistical approaches have been explored over the years to address the one class classification problem. A graphical illustration of popular statistical one-class classification methods is given in Fig. 2. One class support vector machines (OC-SVM), introduced by Scholkopf et al. [16], learns a classification boundary in the feature space by maximizing the margin of separating hyperplane against the origin. Fig. 2(b) shows a typical example of a hyperplane learned by the OC-SVM. Another popular class of approaches for one class classification are based on Support Vector Data Description (SVDD), proposed by Tax et al. [17]. As shown in the Fig. 2(c), SVDD learns a separating hypersphere such that it encloses the majority of target class data. Another popular one class classification method is Single-MiniMax Probability Machines (SMPM) [18] which is essentailly based on Minimax Probability Machines (MPMs) [30]. SMPM, similar to OC-SVM learns a hyperplane by maximizing margin against the origin. SMPM also considers the second order statistics of the data into its formulation which provides additional information on the structure of the data. This helps SMPM to find the best direction for separating the hyperplane as shown in Fig. 2(d). Many variations of these methods have also been explored in the literature. We refer readers to [19] for a survey of different one-class classification methods.

As disucssed earlier, most of these classifiers typically use feature representations extracted from the data for classification. Hence, the quality of these feature representations become a crucial aspect of classifier learning and can affect the classification performance significantly. As a result, representation learning-based methods for one class classification have also been proposed in the literature. These methods leverage CNNs to learn representations for a given target class data. These methods often make use of the generative networks such as auto-encoders or GANs in which reconstruction errors or discriminator scores are utilized to identify the target class data [21], [22], [23], [24], [25]. However, features learned using these generative approaches are not as powerful as those learned in a discriminative fashion. However, training methodology used for discriminative feature learning requires data from multiple classes, limiting its application to multi-class classification. Several works have utilized features from pre-trained CNNs together with one of the statistical approaches such as OC-SVM, SVDD, or SMPM to learn the decision boundary for one class classification [26], [27]. Though, utilizing these powerful feature representations help in learning good decision boundaries, representations and decision boundaries are learned seperately. In such approaches, classification module doesn't influence CNNs for improving the feature representations. Classifier and representations, if jointly learned, have shown
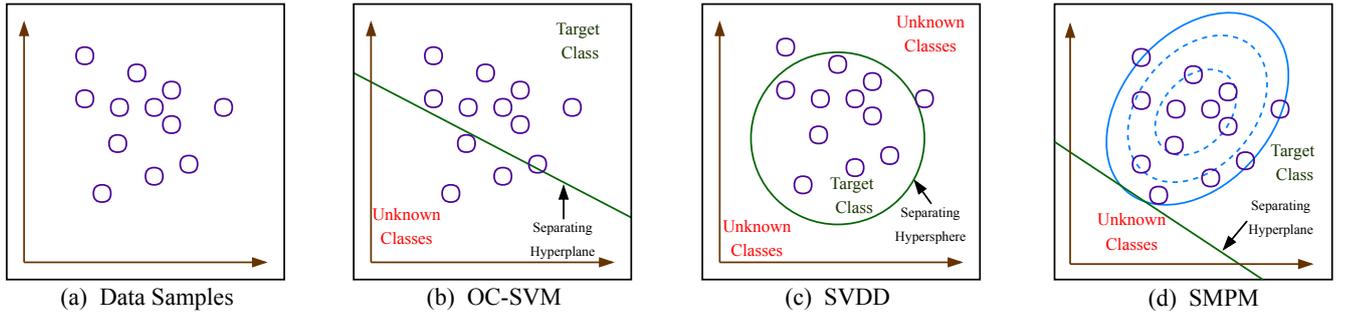
Fig. 2: A graphical illustration of popular statistical one-class classification methods. (a) Sample data corresponding to a target class. (b) One-Class Support Vector Machines (OC-SVM), learn a hyperplane separating the data by maximizing the margin against the origin. (c) Support Vector Data Description (SVDD), learns a hypersphere in the feature space that encloses the given one-class data. (d) Single Minimax Probability Machines (SMPM), learn a hyperplane that minimizes the misclassification probability.

to improve the performance of the system as a whole [28], [29], [31].

Perera et al. [28] proposed a one class algorithm to learn both features and classifier together. However, their approach requires an external dataset and can work only when multiple class data are available, making it not useful for AA systems. Chalpahty et al. [29] introduced a neural network-based one class approach called One Class classification with Neural Networks (OC-NN) which jointly learns classifier and feature representation without using any external data. OC-NN introduced a novel loss function, designed for one class classification. This loss function enables classifier to influence the representation learning process. Similar to OC-NN, the proposed approach aims to jointly learn classifier and feature representation. However, the proposed approach is completely different from that proposed in OC-NN. The feature extractor and the classifier are trained by introducing a Gaussian vector in the feature space which acts as a pseudo-negative class. Oza and Patel [31] showed that providing pseudo negative information helps learning better representations which often results in improved classification performance. In this paper, we introduce a decoder network in the pipeline which essentially constraints the feature extractor to generate feature representation that can reconstruct the original data. These self-representation constraint acts as a regularizer and helps improving the learned representation. We train the entire networks in an end-to-end fashion using only the enrolled user data. In what follows, we present details of the proposed one class classification method based on autoencoder regularized CNN (OC-ACNN).

III. PROPOSED APPROACH

An overview of the proposed OC-ACNN network architecture is shown in Fig. 3. It consists of three major modules namely, feature extractor network, classification network and decoder network. The feature extractor network generates latent space representations for a given target class data. These latent representations are then fed to a classifier and a decoder network. Before feeding them to the classifier

network, they are concatenated with a vector sampled from a zero centered Gaussian $\mathcal{N}(\mu, \sigma \cdot I)$, where σ and μ are the parameters of the Gaussian and I is the identity matrix. This Gaussian vector acts as a pseudo-negative class for the classifier. The classifier network is tasked with discriminating the target class representation from the pseudo-negative Gaussian vector. The decoder network takes in the same latent representation to reconstruct the original input. This enforces the latent representation generated by the feature extractor network to be self-representative i.e., representations are required to generate back the original input images. The classification network and the decoder network are trained end-to-end using a combination of binary cross entropy loss and $L1$ loss, respectively.

A. Feature Extractor

The feature extractor network (E) can be any state of the art network architecture. In this paper, pre-trained AlexNet [32], VGG16 [33] and VGGFace [34] are considered. Before using these architectures as feature extractor, the final layer (i.e. softmax regression layer) is removed. While training, we update weights of only the fully connected layers and freeze the weights of convolutional layers. AlexNet and VGG16 utilized here are initialized with the ImageNet pre-trained weights and VGGFace is initialized with the VGGFace dataset pre-trained weights.

B. Classification Network

Assuming that the extracted features are D-dimensional, the features are appended with the pseudo-negative data generated from a Gaussian, $\mathcal{N}(\mu, \sigma \cdot I)$, similar to [31]. Following We use a simple one layer fully connected classifier network (C) with sigmoid activation at the end, as shown in Fig. 3. The number of hidden units are the same as the length of the feature vector representation. Because of the Gaussian vector concatenation at the input, the network C observes twice the batch size (N) as of the feature extractor.
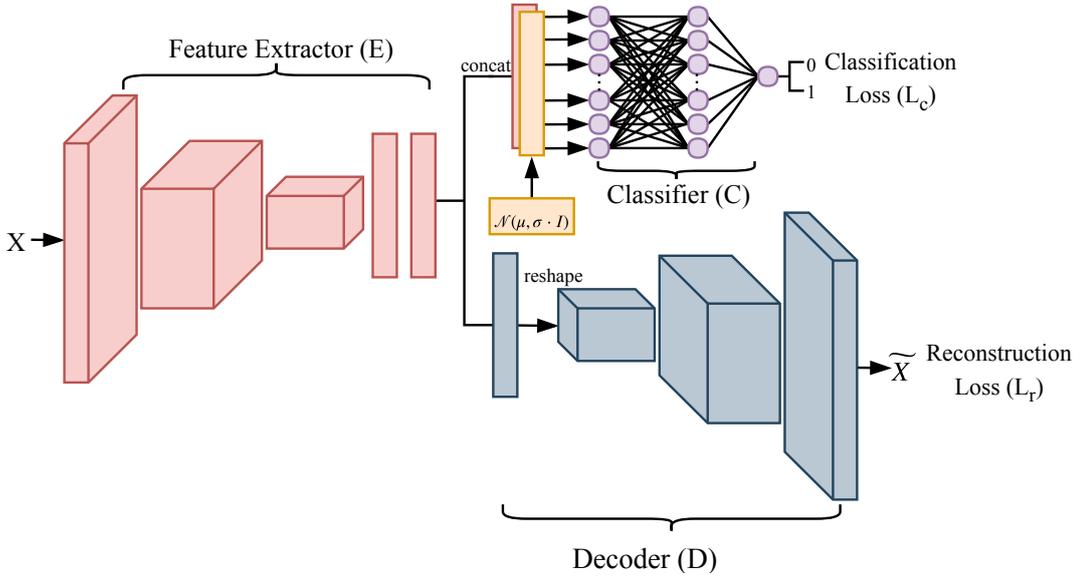
Fig. 3: An overview of the proposed OC-ACNN method. Here, X is the input. The feature extractor module (E) can be any pre-trained CNN architecture. In this paper AlexNet, VGG16 and VGGFace networks are explored. The decoder module (D) is a simple four layer fully convolutional network. The decoder network essentially reconstructs the input image X, as $\tilde{X}$. The classification network (C) is a fully connected neural network trained to distinguish between feature vectors coming from E and the Gaussian vectors sampled from $\mathcal{N}(\mu, \sigma \cdot I)$. The entire network is trained using a combination of classification loss ($\mathcal{L}_c$) and reconstruction loss ($\mathcal{L}_r$).

C. Decoder Network

The decoder network (D) architecture is a simple four layer fully convolutional neural network. This network takes feature representation learned by the network E and tries to reconstruct the original input. This in effect constraints E to generate representation which have self-representation property. It can be seen as a form of regularization on the feature representation. This regularization can be controlled with parameter λ_r given in Eq. 3 of total loss function. Since feature extractor outputs a flattened feature vector, we reshape it to an appropriate size before feeding it to the decoder network. Note that E along with D can be viewed as an auto-encoder network.

D. Loss Functions

The entire network is trained using a combination of two loss functions - classification loss ($\mathcal{L}_c$) and reconstruction loss ($\mathcal{L}_r$). The classification loss is defined as follows

$$\mathcal{L}_c = -\frac{1}{2N} \sum_{j=1}^{2N} [\, y \cdot \log_2(p) + (1-y) \cdot \log_2(1-p) \,], \quad (1)$$

where $y \in \{0, 1\}$ indicates whether classifier input corresponds to feature extractor (i.e., $y = 1$), or sampled from $\mathcal{N}(\mu, \sigma \cdot I)$, (i.e., $y = 0$). Here, p is the probability of $y = 1$. The classification network C observes twice the input batch size because we append Gaussian vector in batch dimension with extracted features, in Eq. 1, the summation is over $2N$.

The $L1$ reconstruction loss is defined as follows

$$\mathcal{L}_r = \frac{1}{N} \sum_{j=1}^{N} \|X - \tilde{X}\|_1, \quad (2)$$

where X and $\tilde{X}$ are the original input image and the reconstructed image, respectively.

Finally, the overall loss function is the sum of $\mathcal{L}_r$ and $\mathcal{L}_c$ defined as follows

$$\mathcal{L}_t = \mathcal{L}_c + \lambda_r \mathcal{L}_r, \quad (3)$$

where λ_r is a regularization parameter. Furthermore, note that $\tilde{X} = D(E(X))$ and $p = C(E(X))$.

The network is optimized using the Adam optimizer [35] with the learning rate of 10^{-4} and batch size (i.e. N) of 64. For all the experiments, μ, σ and λ_r are set equal to 0.0, 0.01 and 1.0, respectively. The decoder architecture is as follows ConvTran(1024, 256) - ConvTran(256, 64) - ConvTran(64, 16) - ConvTran(16, 3), where, ConvTran(*in*, *out*) denotes the transposed convolutions with *in* and *out* as number of input and output feature channels, respectively. All transposed convolutions are used with kernels of size 4×4. ReLU activation is used after every transposed convolution layer except the fourth, where Tanh activation is used. Instance normalization [36] is used before the classifier network and at the end of every transposed convolution layer.

IV. EXPERIMENTAL RESULTS

We evaluate the performance of the proposed approach on three publically available face-based AA datasets – MOBIO

(a) MOBIO	(b) UMDAA-01 Face	(c) UMDAA-02 Face
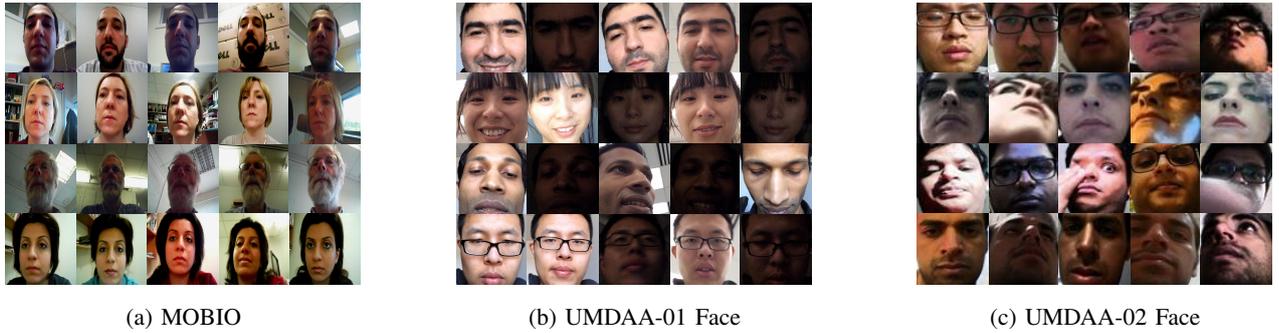
Fig. 4: Sample Images from (a) Mobio, (b) UMDAA-01 Face, (c) UMDAA-02 Face datasets. Each column represents different user and each row shows multiple images from the same user.

[37], UMDAA-01 [9] and UMDAA-02 [38]. The proposed approach is compared with the following one-class classification methods:

- **OC-SVM:** One class SVM as formulated in [16] is used. OCSVM is trained on features extracted from AlexNet, VGG16 and VGGFace.
- **SMPM:** SMPM is used as formulated in [30]. In SMPM formulation, to utilize the second order statistics, covariance matrix computation is required. Hence, before applying SMPM, we reduce the dimensionality of the features extracted from AlexNet, VGG16 and VGGFace using Principle Component Analysis (PCA).
- **SVDD:** Support Vector Data Description is used as formulated in [17], trained on the AlexNet,VGG and VGGFace features.
- **OC-NN:** One-class neural network (OC-NN) is used as formulated in [29]. The encoder network described in [29] is replaced with a pretrained CNNs, i.e. AlexNet, VGG16 and VGGFace to have a fair comparison between the methods. Apart from this change, the training procedure is exactly the same as given in [29].

The following ablation baselines are also considered to show the contribution of each module in the proposed approach:

- **Auto-Encoder baseline (only $\mathcal{L}_r$):** This is one of the ablation baselines, where we utilize the feature extractor and the decoder networks, and train with only $\mathcal{L}_r$ loss function given in Eq. 2. It can also be seen as a generative approach baseline. The reconstruction error is used for classification. In other words, a pre-determined threshold is compared against the reconstruction error and the input is rejected if the error is greater than the threshold. Otherwise, the input is declared as corresponding to the one-class data.
- **Classifier baseline (only $\mathcal{L}_c$):** Another ablation baseline includes using classifier and feature extractor networks trained with only $\mathcal{L}_c$ loss function given in Eq. 1. The classification network is not regularized by the decoder network. This baseline is equivalent to the method proposed in [31]. This ablation study will clearly show the significance of using an auto-encoder as a regularizer for one-class classification.

- **Proposed approach OC-ACNN (both $\mathcal{L}_r$ and $\mathcal{L}_r$):** OC-ACNN is the method proposed in this paper.

For OC-SVM, SMPM and SVDD distance scores from the hyperplane/hypersphere are used for performance evaluation. For OC-NN, classifier baseline and the proposed approach, scores from the classifier are used for performance evaluation. As mentioned before, the reconstruction error is used for evaluating the performance of the auto-encoder baseline.

A. Datasets

MOBIO. The MOBIO dataset is a bi-modal AA dataset containing face images and voice recordings of 150 users. In this paper, we only consider face images for conducting the experiemnts. Sample images from this dataset are shown in Fig. 4a. For each user the recordings are taken in six sessions at different locations. We combine images from all six sessions. MOBIO contains less variations in pose, illumination etc., as compared to the other datasets used in this paper. For experiments, first 48 users are considered as target users and the rest are used as unknown users. Target users' data is split into train and test set with 85/15 ratio. For each target user, the training set is used to train the networks. During evaluation, we utilize the test set of the target user along with the data from all unknown users. This process is repeated for all 48 users and average performance is reported.

UMDAA-01 Face. The UMDAA-01 dataset contains face and touch gestures recorded from a mobile device. In total 750 video sequences of 50 users are collected in three different sessions with varying illumination conditions. Sample images from this dataset are shown in Fig. 4b. Data from different sessions are combined for each user and split into train and test sets with 80/20 ratio. Considering one user as the target and the remaining 49 users as unknown, networks are trained using target train set and tested with test set consisting of all 50 users' data (i.e. 1 target and 49 unknown). This experimental protocol is followed for all 50 users and average performance is reported.

UMDAA-02 Face. Unlike the above two datasets, the UMDAA-02 dataset has multiple modalities for 44 users

(a) Pre-trained AlexNet.

(b) AlexNet trained using the propsoed method.

(c) Pre-trained VGG16.

(d) VGG16 trained using the propsoed method.

(e) Pre-trained VGGFace.

(f) VGGFace trained using the propsoed method.
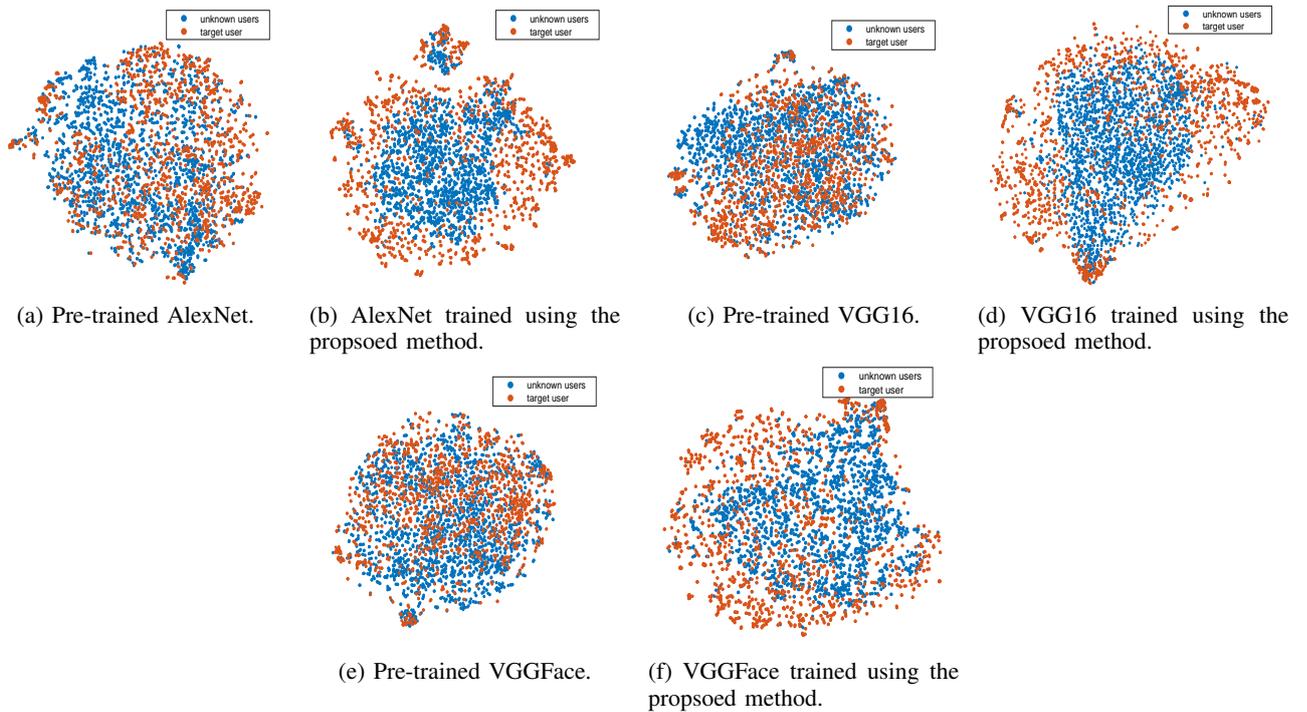
Fig. 5: t-SNE visualizations of feature representations from the feature Extractor (E) corresponding to a user from the UMDAA-02 Face dataset.

e.g. face, gyroscope, swipe patterns, key strokes etc. all recorded from 18 sensors of a Nexus mobile device. Since the dataset was collected over a period of two months, it is an extremely challenging dataset with large variations in pose, illumination, occlusions and other environmental conditions. Fig. 4c shows some sample images from this dataset. The number of sessions for each user ranges from 25 to 750 providing large number of frontal face images, i.e. more than 10k images on average per user. For each user, train and test splits are created with 80/20 ratio. We follow similar protocol as that of UMDAA01-Face for all 44 users and report the average performance.

Area Under the ROC curve (AUROC) is used to measure the performance. This is one of the most commonly used metric in the literature for evaluating the performance of the one-class classification methods.

B. Qualitative Evaluation

In this section, we present qualitative evaluation of the proposed approach by comparing the visualizations of feature representations learned by our method with those corresponding to respective pre-trained networks. Fig. 5 shows t-SNE [39] visualizations of the feature representations corresponding to AlexNet, VGG16 and VGGFace, respectively. These t-SNE plots are obtained from a single user of the UMDAA-02 Face dataset. Fig. 5a, 5c and 5e show the visualizations corresponding to pre-trained AlexNet, VGG16 and VGGFace networks, respectively. Fig. 5b, 5d and 5f show the visualizations corresponding to their counterpart networks trained using the proposed approach. As can be

seen from these figures, the pre-trained networks generate features that highly overlap between the target and unknown users. This makes sense since these networks are trained using a cross-entropy loss for multi-class classification. As a results, the features from these networks overlap significantly and it makes it difficult for a one-class classifier to correctly identify the separating decision boundary. On the other hand, the feature representations of the same networks trained using the proposed approach are quite distinctive. The learned feature representations corresponding to the target and unknown users are very well separated. These features become extremely useful while identifying the target user against the unknown users, thereby resulting in improved classification performance.

Table I shows the AUROC results corresponding to each plot for the same user computed using OC-SVM. As can be seen from this table, for all the networks the features learned using the proposed OC-ACNN provides better performance compared to the pre-trained features.

Feature Extractor (E)	Pre-Trained	OC-ACNN
AlexNet	0.5319	0.6780
VGG16	0.5698	0.8194
VGGFace	0.5428	0.8808

TABLE I: AUROC results corresponding to the study conducted in Fig. 5.

C. Quantitative Evaluation

Tables II, III and IV show the performance on all three datasets based on AlexNet, VGG16 and VGGFace as feature

Dataset	OC-SVM	SMPM	SVDD	OC-NN	Auto-encoder (only $\mathcal{L}_r$)	Classifier (only $\mathcal{L}_c$)	OC-ACNN ($\mathcal{L}_c+\mathcal{L}_r$)
Mobio	0.6578 ± 0.1132	0.7721 ± 0.1185	0.7851 ± 0.1270	0.7504 ± 0.1512	0.7526 ± 0.1075	0.8191 ± 0.1286	$\mathbf{0.8633 \pm 0.1136}$
UMDAA-01	0.6584 ± 0.1255	0.7576 ± 0.1149	0.8909 ± 0.0755	0.8684 ± 0.0913	0.6560 ± 0.1066	0.9196 ± 0.0482	$\mathbf{0.9276 \pm 0.0465}$
UMDAA-02	0.5746 ± 0.0595	0.5418 ± 0.0382	0.6448 ± 0.0725	0.6542 ± 0.0593	0.5952 ± 0.0869	0.7017 ± 0.1007	$\mathbf{0.7398 \pm 0.0787}$

TABLE II: Comparison between the proposed approach and other one-class methods with **AlexNet** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

Dataset	OC-SVM	SMPM	SVDD	OC-NN	Auto-encoder (only $\mathcal{L}_r$)	Classifier (only $\mathcal{L}_c$)	OC-ACNN ($\mathcal{L}_c+\mathcal{L}_r$)
Mobio	0.6607 ± 0.1066	0.7266 ± 0.1046	0.8212 ± 0.1130	0.7822 ± 0.1153	0.7457 ± 0.1072	0.8177 ± 0.1132	$\mathbf{0.8705 \pm 0.1278}$
UMDAA-01	0.6777 ± 0.0946	0.8664 ± 0.0765	0.9011 ± 0.0592	0.8802 ± 0.0976	0.8494 ± 0.0844	0.9348 ± 0.0384	$\mathbf{0.9486 \pm 0.0336}$
UMDAA-02	0.5828 ± 0.0757	0.5473 ± 0.0447	0.6424 ± 0.0677	0.6199 ± 0.0693	0.6042 ± 0.0939	0.7349 ± 0.0845	$\mathbf{0.8457 \pm 0.0581}$

TABLE III: Comparison between the proposed approach and other one-class methods with **VGG16** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

Dataset	OC-SVM	SMPM	SVDD	OC-NN	Auto-encoder (only $\mathcal{L}_r$)	Classifier (only $\mathcal{L}_c$)	OC-ACNN ($\mathcal{L}_c+\mathcal{L}_r$)
Mobio	0.6702 ± 0.1268	0.6619 ± 0.1068	0.7975 ± 0.1250	0.7673 ± 0.1380	0.7339 ± 0.1095	0.8347 ± 0.1324	$\mathbf{0.8859 \pm 0.1042}$
UMDAA-01	0.6763 ± 0.1237	0.7334 ± 0.1241	0.8745 ± 0.0794	0.8257 ± 0.1381	0.8237 ± 0.0923	0.9432 ± 0.0654	$\mathbf{0.9772 \pm 0.0213}$
UMDAA-02	0.5712 ± 0.0644	0.5671 ± 0.0597	0.5898 ± 0.0647	0.5987 ± 0.0652	0.6343 ± 0.0723	0.6393 ± 0.0946	$\mathbf{0.8946 \pm 0.0535}$

TABLE IV: Comparison between the proposed approach and other one-class methods with **VGGFace** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

extractors, respectively. The performance of other methods is inconsistent across the experiments. SMPM was found to perform better than OCSVM, while SVDD achieves better performance in many cases beating OC-NN. This may be due to the evaluation protocol difference between this paper and the one proposed in OC-NN [29]. In OC-NN evaluation protocol, the number of unknown classes used during evaluation are much less than the number of unknown classes used for evaluation in this paper (i.e., MOBIO(96), UMDAA-01 Face(49) and UMDAA-02 Face(43)). This can be a reason for the poor performance from OC-NN as compared to SVDD. OC-NN however, manages to perform better than SMPM and OCSVM, and in couple of cases SVDD. Meanwhile the proposed approach achieves superior performance across all the datasets and for different feature extractor models.

Comparing the performance across models, VGGFace outperforms both VGG16 and AlexNet models. This makes sense since face images (i.e. VGGFace dataset) were used to train the original VGGFace model and the corresponding weights are better suited for face-based AA application considered in this paper. In contrast, the VGG16 and AlexNet networks were trained using general object dataset (i.e. ImageNet dataset) for object recognition task. The highest performance for all these networks is achieved for UMDAA-01 Face, since this dataset only contains illumination variations. Though MOBIO contains least variations in image samples, it has large number of unknown classes to compare against. While UMDAA-02 is the most difficult dataset since it contains very unconstrained face images. As a result, the performance on this dataset is lower than the other two datasets. In summary, the proposed approach observes improvement of $\sim$6%, $\sim$9% and $\sim$16% on average across all datasets corresponding to AlexNet, VGG16 and VGGFace,

respectively.

Comparing ablation baselines, the auto-encoder baseline using only the reconstruction loss performs the poorest, while only the classifier baseline performs reasonably well. Auto-encoder and classifier baselines can be categorized as generative and discriminative approach, respectively. Since the discriminative approach can learn better representation, it helps the classifier baseline to improve its performance. However, when the decoder is added to the classification pipeline to regularize the learned representations, it improves the overall performance by $\sim$6%. This clearly shows the significance of enforcing the self-representation constraints to regularize the learned feature representations for one-class classification.

V. CONCLUSIONS

We proposed a new approach for single user AA based on auto-encoder regularized CNNs. Feature representations are jointly learned with classifier influencing the generated representations. A pseudo-negative Gaussian vector was utilized to train the fully connected classification network. Decoder was introduced to regularize the generated feature representation by enforcing it to be self-representative. Experiments were conducted using the AlexNet, VGG16 and VGGFace networks, which showed the adaptability of the proposed method to work with different types of network architectures. Ablation study was conducted to show the importance of both classification loss and feature regularization. Moreover, visualizations of the learned representations showed the ability of the proposed approach to learn distinctive features for one-class classification. Furthermore, the consistent performance improvements over all the datasets related to AA showed the significance of the proposed one-class classification method.

REFERENCES

[1] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE security & privacy*, no. 2, pp. 33–42, 2003.

[2] C. M. Most, "The global biometrics and mobility report: The convergence of commerce and privacy," *Acuity Market Intelligence Report Received from : www.acuity-mi.com*, 2016.

[3] R. P. Guidorizzi, "Security: active authentication," *IT Professional*, vol. 15, no. 4, pp. 4–7, 2013.

[4] A. Pozo, J. Fierrez, M. Martinez-Diaz, J. Galbally, and A. Morales, "Exploring a statistical method for touchscreen swipe biometrics," in *Security Technology (ICCST), 2017 International Carnahan Conference on*. IEEE, 2017, pp. 1–4.

[5] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch gesture-based active user authentication using dictionaries," in *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2015, pp. 207–214.

[6] P. Perera and V. M. Patel, "Extreme value analysis for mobile active user authentication," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 346–353.

[7] ——, "Towards multiple user active authentication in mobile devices," in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. IEEE, 2017, pp. 354–361.

[8] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*. IEEE, 2010, pp. 306–311.

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

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

[11] P. Samangouei and R. Chellappa, "Convolutional neural networks for attribute-based active authentication on mobile devices," in *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 2016, pp. 1–8.

[12] P. Perera and V. M. Patel, "Efficient and low latency detection of intruders in mobile active authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1392–1405, 2018.

[13] ——, "Quickest intrusion detection in mobile active user authentication," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2016, pp. 1–8.

[14] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[15] M. Antal and L. Z. Szabó, "An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices," in *Control Systems and Computer Science (CSCS), 2015 20th International Conference on*. IEEE, 2015, pp. 343–350.

[16] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[17] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[18] L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet, "Robust novelty detection with single-class mpm," in *Advances in neural information processing systems*, 2003, pp. 929–936.

[19] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[21] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388.

[22] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, 2018.

[23] W. Lawson, E. Bekele, and K. Sullivan, "Finding anomalies with generative adversarial networks for a patrolbot," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 12–13.

[24] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1577–1581.

[25] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 665–674.

[26] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[27] J. T. Andrews, "Transfer representation-learning for anomaly detection."

[28] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *arXiv preprint arXiv:1801.05365*, 2018.

[29] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.

[30] G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan, "Minimax probability machine," in *Advances in neural information processing systems*, 2002, pp. 801–807.

[31] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 277–281, 2019.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[34] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *Proc. of ICLR*, 2017.

[37] P. Tresadern, C. McCool, N. Poh, P. Matejka, A. Hadid, C. Levy, T. Cootes, and S. Marcel, "Mobile biometrics (mobio): Joint face and voice verification for a mobile platform," *IEEE pervasive computing*, 2012.

[38] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active user authentication for smartphones: A challenge data set and benchmark results," in *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 2016, pp. 1–8.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.