Temporal Face Embedding as Biometric Tokenization for Decentralized IoT

Seyed Ali Miraftabzadeh
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, Texas, USA 78249
Email: Ali.Miraftab@utsa.edu

Paul Rad
Department of Information Systems and Cyber Security
University of Texas at San Antonio
San Antonio, Texas, USA 78249
Email: Paul.Rad@utsa.edu

Mo Jamshidi
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, Texas, USA 78249
Email: Mo.Jamshidi@utsa

***Abstract*—With IoT proliferation, a decentralized biometric access control system is crucial to ensure the identity consistency in cyber physical space. In this paper, a biometric secure modality is proposed based on temporal domain that is often available in IoT systems. A Novel neural network architecture, named LSTM-ResNet, is designed to learn the temporal dynamics of human biometrics such as face for authentication. The model is designed to implement as a device-embedded temporal face embedding as biometrics tokenization to keep data securely decentralized on IoT devices for real-time on-line and off-line identification. The combined novel Temporal Face Embedding, LSTM-ResNet and decentralized embedded IoT approaches prevent tampering and spoofing unauthorized access. The preliminary results are promising for decentralized biometrics IoT and motivate future work in this area.**

I. Introduction

From a system-level perspective, the IoT authentication can be looked at as a highly dynamic and heterogeneously distributed networked system, composed of a very large number of smart objects producing and consuming information. For instance, mobile communications security, smart vehicles, door locks, connected homes, medical devices and critical infrastructure all part of a growing world of IoT devices. For these devices, an automated recognition of authorized individuals poses a serious technical challenge which needs to be addressed. Such a system should be convenient and guarantees the specific level of accuracy as well as keeping the individuals' data safe from hackers. These requirements necessitate changes in conventional security modalities methods, and in implementing authentication systems and algorithms.

Security modalities are classified into three categories: (1) possession, what we have as an authentication tools: e.g. key, RFID or ID card. (2) cryptographic, knowledge base authentication: e.g. PIN, password, challenge-response answers like as parents maiden name. (3) biometrics: e.g. fingerprint, face, iris, etc. Obviously, possession base methods are in contrast with the easygoing usage of IoT devices and are not always applicable. IoT authentication requires automated recognition of individuals in many smart connected devices continuously which causes cryptographic methods not enforceable as primary secure modality. However, knowledge base methods can be incorporated in two-factor authentication for some cases

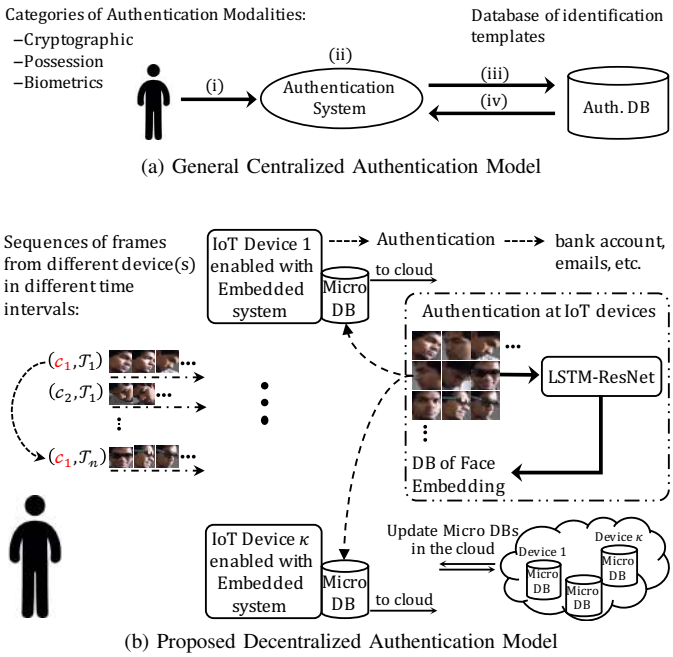


Fig. 1. Centralized authentication and overview of proposed decentralized authentication.

through the secure sockets layer protocol. Due to the IoT requirements, biometric is a preliminary candidate to address the issues.

Biometrics focus on evaluation, comparison, and statistical analysis of people characteristics. It falls into three main classes: physiological which are related to the shape and size of the body, examples include face. Behavioral which are related to the change in human behavior over time, for instance gait (the way one walks), rhythm of typing keys. Combination of both which includes both traits, where the traits are depending upon physical as well as behavioral changes, for example, voice recognition. It depends on health, size, and shape of vocal cord, nasal cavities, mouth cavity, shape of lips, etc. and the emotional status, age, illness (behavior) of a person. A secure and dynamic biometric modality is proposed based on physical and behavioral changes of facial images over

time. The proposed method is different from common face recognition models designed for identification and verification tasks in main viewpoints.

The face recognition task can be done in a number of various ways, such as by capturing a facial image using an optical camera (in the visible spectrum) or by processing the facial heat emission captured by an infrared camera. In the visible spectrum, face recognition tasks focus on modeling features from the central portion of the facial images that do not change over time while avoiding superficial features such as facial expressions or hair. Face recognition models are generally designed to increase identification and verification accuracy in large datasets. However, proposed model relies on dynamic physical and behavioral characteristics of a person facial images over time. It process the central portion of the person facial images in a sequence of frames coming from one or multiple cameras. It considers the face variations in different expressions in an uncontrolled environment, and focuses on an application for a typical security coverage area with a near perfect accuracy rate. Notwithstanding, the proposed model can be extended for smart city as a distributed model.

A centralized biometric system is essentially a pattern recognition process in which (i) a raw data from an individual is acquired, (ii) a notable feature set from the raw data are extracted, (iii) and then new extracted feature set is evaluated against the feature sets stored in the database, (iv) finally a command to execute an action according to the result of the comparison is transmitted by a center located not at the IoT device. The similar process is applied for other secure modalities as well, see figure 1a. To maintain privacy and eliminate the inconvenience of storing sensitive biometric data, the proposed model is implemented as a device-embedded biometric authentication system, see figure 1b.

Device-embedded biometric authentication keeps data securely decentralized on local IoT devices and safe from hackers. In addition, it improves the security of endpoint IoT devices; all the process, feature extraction and comparison software built into the IoT device itself. Since the device essentially belongs to a person or specific group of people, the typical usage is a one-to-one comparison of a probe biometric sample against the group biometric reference stored on the device, see figure refdecentralized.

In the visible spectrum, several approaches to modeling facial images are Local Feature Analysis, Elastic Graph Theory, Principal Component Analysis, Multi-Resolution Analysis, and Neural Networks. The authentication problem can be translated to a binary classification problem, giving input faces the answer is yes or no for each face. A new neural network architecture is proposed to learn temporal features captured from a sequence of frames to classify faces' dynamic structure. The network learns a global interpretation of a face' temporal evolution over time from a camera or a groups of cameras. The network, consist two sub-networks: RNN-LSTM and ResNet.

To learn hidden patterns in time sequence, the proposed architecture drives a benefit from a recurrent neural network (RNN) with Long Short Term Memory (LSTM) units. RNN has a drawback of vanishing and exploding gradients called Vanishing Gradient Problem, which makes it inefficient for discovering long-term temporal relationships from the input sequences. Thus, the RNN is implemented by leveraging enabled memory LSTM units to store the information for long-term temporal learning. To extract dynamic facial features, residual neural network, ResNet, is exploited as a discriminative learning neural network which also addresses the vanishing gradient problem in the optimization phase. For each coming frame, one LSTM unit stack above ResNet as a top layer before the final softmax layer. Therefore, the architecture leverages the local and dense property from residual operation and learn long-term temporal structure by supplying information in each LSTM units. Our contributions can be summarized as follows:

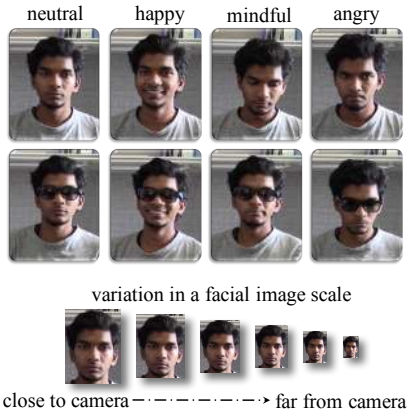
1) A biometric secure modality is proposed based on person dynamic facial features changes over time instead of face recognition task for authentication.
2) Novel LSTM-ResNet architecture is leveraged to learn the long-term temporal structure from a sequence of frames captured by a camera(s), and then classifying faces' dynamic structure for authentication task.
3) The model is designed to implement as a device-embedded biometric authentication to keep data securely decentralized on local devices and safe from hackers. In addition, preserve users privacy and eliminate the inconvenience of storing sensitive biometric data.

II. Related Works

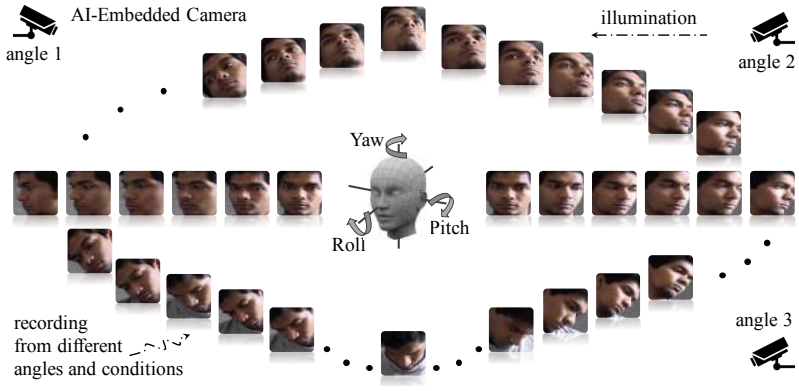
Face identification is turning into a vital research subject, because of its extensive variety of conceivable applications, similar to security get to control, demonstrate based video coding, content-based video ordering, or propelled human and PC collaboration. It is additionally a required preparatory stride to face acknowledgment and appearance examination.

Various methodologies for face identification have been proposed in the most recent decade, a significant number of them depicted and analyzed in two intriguing late reviews by Learned-Miller [1], and Zhang et al. [2]. Most face identification techniques depend on neighbourhood facial element recognition and order utilizing factual and geometric models of the human face. Low level investigation initially manages the division of visual components utilizing picture properties, for example, edges, force, shading, movement, or summed up measures. Different methodologies depend on format coordinating where a few relationship layouts are utilized to identify nearby sub features, considered as inflexible in appearance (eigen features) or deformable. At that point, visual elements are composed into a more worldwide idea of face through facial component and heavenly body investigation utilizing face geometry imperatives.

The principle downside of highlight based methodologies is that either minimal worldwide requirements are connected on the face layout or removed components are essentially impacted by impediments, and changes in face look and perspective. With a specific end goal to deal with troublesome



(a) Sample of facial images of an individual in different expressions and scales



(b) Captures images of the same person in different views. The illumination direction is from right to left

Fig. 2. Possibility variation of an individual's facial images for the authentication task

situations where various appearances of changed sizes and postures must be recognized in vigorously jumbled foundations, some propelled picture based example acknowledgment methods have been produced. They keep away from the particular and conceivably off base face displaying by picking up fundamental tenets contained in very factor confront designs from vast preparing sets of face illustrations. They have ended up being exceptionally tolerant to commotion and twists influencing the face designs.

In [3], Erkin et al.used the standard Eigen faces recognition algorithm in order to create the first privacy-preserving biometric face recognition protocol. The protocol is a secure two-party computation protocol which allows only the inquiring side (client) to see the matching value. Hence, only a client is interested in determining whether it has any image in common with server's database, without worrying about the leakage of unnecessary information. The Euclidean distance between the face image feature vectors from the client and server's face image database is computed. The facial image data record that has the smallest distance to the client's input biometric data is returned. Anadditive homomorphic encryption (AH-Enc) scheme is used on the data exchanged. Only the data exchanged during execution of the protocol are encrypted. However, the data on the server is not encrypted, and so, the server is aware of its biometric database.

A privacy-preserving biometric identification protocol was developed by Blanton and Gasti in 2011 [4]. This protocol used iris codes based on Hamming distance. AH-Enc and garbled circuits were also used with this protocol. The iris reading of the client is compared with an iris image in the database managed by the server. The client is made aware of the comparison results and every record in the database of server, but the contents of the server are not disclosed to the client.

Later, in 2012, a secure outsourcing approach developed by Blanton and Aliasgari [5] outsourced the computations of iris biometric comparison with data records. Two protocols were put forward; single-server setting and multiple-server setting. The first protocol utilizes a predicated-encryption scheme, where the server performs non-interactive computations. The homomorphic encryption is more secure than the predicate encryption. This is why the protocol presents a more secure biometric system, with greater privacy protection. In their work, the protocol is dependent on at least three independent servers. Another drawback is that, at the end of the protocol execution, the server detects which encrypted biometric data record matches the client's input. This access pattern leakage can breach the security guarantee of the underlying encryption scheme. our scheme does not disclose any information; therefore, it offers the same security protection as the encryption scheme used to encrypt the outsourced biometric image.

III. Method

To include all the facial image variations, see figure 2, a sequence of video frames $(c_1, c_2, ..., c_n)$ is considered as an input and the output of the network is a binary number y authorizing the user in multiple devices connected to the IoT, see figure 3. We propose a residual network with a LSTM layer on top of that to extract intra-class similarity and inter-class discriminatory of captured facial images from different video frames; in other words, the conditional probability of the output, $p(y|(c_1, c_2, ..., c_n))$.

A. Extract Temporal Features as an Embedding Vector

Temporal feature of a facial image in a frame is presented as an embedding vector. The embedding vector per identity is constructed through the residual network architecture consisting residual blocks. The general form of each block can be formulated as:

$$\begin{aligned} y_l &= h(x_l) + F(x_l, (W_r, b_r)_l) \\ x_{(l+1)} &= f(y_l) \end{aligned} \tag{1}$$

where x_l and x_{l+1} are input and output of the lth unit, h is a forward function of the plain unit, F is a residual function and r stands for the number of repeated convolution layer

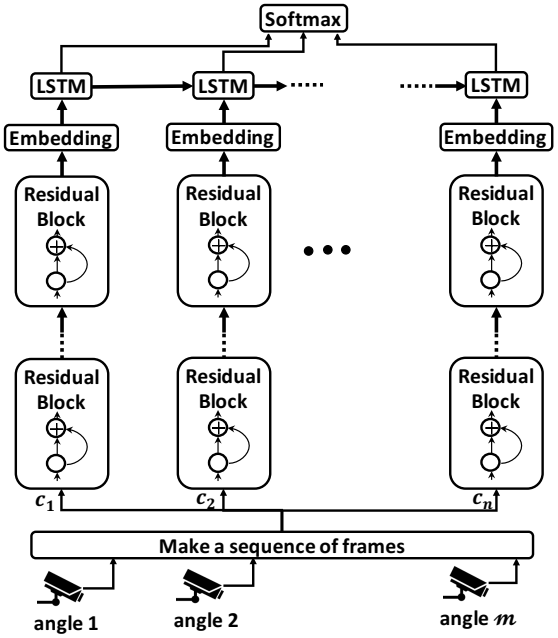


Fig. 3. LSTM-ResNet architecture unrolled in time for m cameras and n frames, $n > m$.

in the residual function, and f is a differentiable threshold function. Figure 5 (b) demonstrates an example of the general form of the residual unit. The initial idea of ResNet is to achieve additive residual function F with respect to $h(x_l)$ and facilitates minimizing the loss function. In this regards, [6], [7] emphasize on the importance of the identity mapping, $h(x_l) = x_l$, so in the general formula we denote on r to represent the repetition times of the convolutional layers in residual branch and we follow the identity mapping for the plain branch. In residual block, the other noteworthy nob is differentiable threshold function. If f is also considered identify mapping, for any deeper unit L and shallower unit l:

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, (W_r, b_r)_i) \tag{2}$$

This assumption turns the matrix-vector products, say:

$$x_L = \prod_{i=0}^{L-1} W_i x_0, b_i = 0 \tag{3}$$

to the summation of the outputs of all preceding residual functions (plus x_0) [6], and consequently clean backpropagation formula:

$$\begin{aligned} \frac{\partial E}{\partial x_l} &= \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} \\ &= \frac{\partial E}{\partial x_L}\left[1 + \frac{\partial \sum_{i=l}^{L-1} F(x_i, (W_r, b_r)_i)}{\partial x_L}\right] \end{aligned} \tag{4}$$

one of the most interesting properties of this architecture is reducing the probability for the gradient to be canceled out. Refer back to the general form of the residual units,

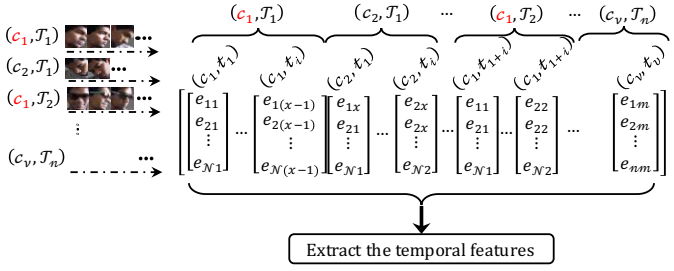


Fig. 4. Illustration of mapping facial images captured from different cameras to the embedding vectors

there are other residual units with the properties of increasing dimensions and reducing feature map sizes [7], [8] by using the conventional activation function, Rectified Linear Unit (ReLU), as the differentiable threshold function:

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left[\frac{\partial x_L}{\partial h}\frac{\partial h}{\partial x_l} + \frac{\partial \sum_{i=l}^{L-1} F(x_i, (W_r, b_r)_i)}{\partial x_L}\right] \tag{5}$$

The last residual block maps a facial image into the embedding vector. Figure 4 illustrates such a mapping for different facial images captured from a camera(s) in different angles and time windows.

B. Exploring Temporal Relation in a Sequence of Embeddings

The output of the embedding vector is feed to the LSTM unit which is the modified version described in [9]. LSTM units [10] have the ability to learn long range dependency from the input sequences. At the time step t, the behavior between input(x_t), output(h_t), and internal state is controlled through three gates. For each unit c_t stores the internal state and three gates are input gate(i_t), output gate(o_t), and forget gate(f_t). where W and b are model parameters, σ is sigmoid function and g_t is the non-linear transformation of inputs, see Figure 6. To capture the temporal relation from the video frames sequence which is importance for identity authentication, outputs and cell memories from last time step are connected to the three gates through defined dot products in:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{6}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{7}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \tag{8}$$

$$g_t = \text{PReLU}(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \tag{9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{10}$$

$$h_t = o_t \odot \text{PReLU}(c_t) \tag{11}$$

inputs of the three gates consist of the current time step of the input and last time step of the output and internal memory. The cell memory is updates as a result of the combination of input gate(i_t) and forget gate(f_t), Equation 10. The influence of the input in the internal state is controlled by input gate and forget gate takes the control over the contribution of the last internal state to the current internal state.

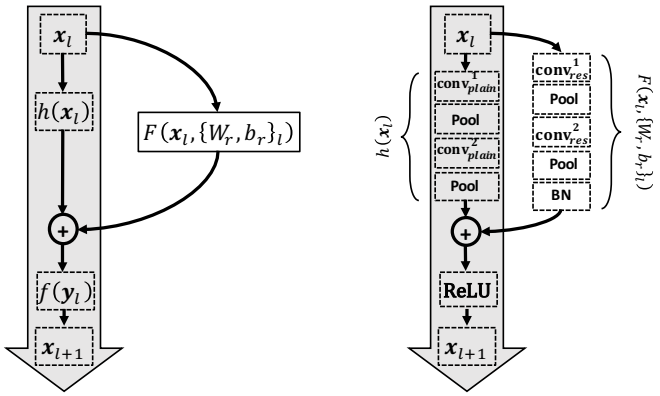


Fig. 5. Left: (a) General form description of the residual unit. Right: (b) illustration of the residual unit including: residual branch consists of two convolution layers each following by pooling and Batch Normalization at the end, and plain branch with two convolution layers follow by the pooling.

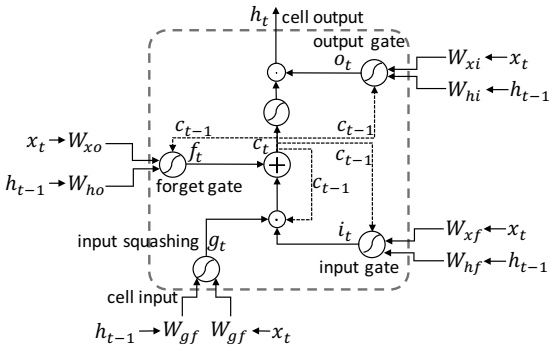


Fig. 6. LSTM unit illustration. Each circle with a curve means a non-linear transformation. Circle with a dot is element-wise product operation. Square with a plus sign is element-wise plus operation. The dashed arrow means connection from the last time step.

C. Architecture

In general, the experiment results show combination of residual architecture and embedding address the over-fitting and degradation problems. The augmented LSTM units with PReLU can learn the fine distinction between sequences of nonlinear periodic embeddings patterns without external resets or teacher forcing.

The implementation for residual network follows the practice in [6], [7]. To address the different scales of the facial images, a pyramid scale of each sample is crated the face images with a resolution in $\{250 \times 250, 120 \times 128, 60 \times 64, 30 \times 32\}$ are used and he maximum mini-batch sizes are $\{32, 64, 128, 128, 324\}$ respectively.

IV. Training Methodology

The LSTM_ResNet model is implemented and optimized by using TensorFlow, the open-source software library for machine intelligence [11]. The residual network and LSTM units are trained separately on advanced NVIDIA Tesla P100 GPUs with 5.3 TeraFLOPS built for data center. The test is carried on embedded system, NVIDIA Jetson TX2.

Different residual network architectures are trained from scratch while sharing almost the same optimization strategy and parameters. In the training phase, precise investigation of the effects of different stochastic gradient descent (SGD) algorithms are conduced: Adaptive Subgradient [12], RMSProp [13], Adam [14], and Momentum [15]. It is observed that Adam optimization algorithm outperforms the other methods because of the bias-correction characteristic of the algorithm. The best performance were achieved with $\beta_1 = 0.96$, $\beta_2 = 0.\overline{9}$, and $\epsilon = 91$. For several epoch iterations the learning rate is set to 0.001 and it is reduced by a factor of 10 to stabilize the optimization. $L2-$norm with a weight decay of 0.1 is applied as an artificial constraint to implicitly reduce the number of free parameters and not to make the network difficult to optimize. Since the network is trained for the specific group of authorized users, it is avoid to exploit dropout as the regularization method.

LSTM units are trained following the practice in [9] using stochastic gradient decent (SGD) and a momentum of 0.9 with the truncated BPTT. To capable LSTM to learn bridging minimal time lags, the gradient is truncated where the performance degradation is observed.

V. Experiment

The residual network is trained using FaceScrub dataset [16] which is one of the most accurate face datasets in terms of duplicated, mislabeled, and morphed faces. It comprises a total of 106,863 facial images of 530 celebrities (male and female), with about 200 images per person. As such, it is one of the largest public face databases, with an average of 2.15 images per person. Images are collected from Internet and are taken in uncontrolled conditions (real-world situations). To maximize the usage of the dataset, it is splitted into seven folds; five folds for training, one fold for validation, and two folds for testing. Note that there is no common subjects in the folds. To prevent the unbalance problem, it is tried to make an even ratio of image per person in each fold.

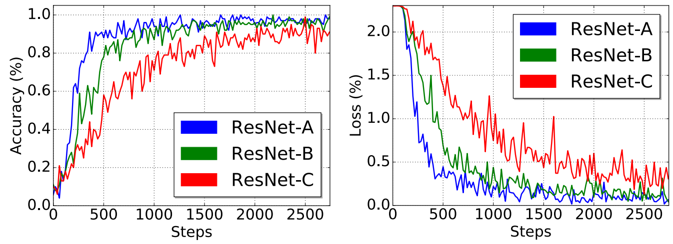
The LSTM units of the LSTM-ResNet model are trained using YTF dataset [17], with no people overlapping with FaceScrub. YTF dataset consists a large variations in pose, expression and illuminations for 1,595 recorded different identities in 3,425 videos (in average 2.15 videos per person). The dataset is made up of divers video durations, 48 frames to 6,070 (in average 181.3 frames per video).

The facial data in the images and the videos varies in size and length, and contains a lot of background noises. To make them ready for LSTM-ResNet model to train, face detection and alignment are the primary steps. Multitask convolutional neural network proposed in [18] is applied on both datasets for the experiment. Cases in which the algorithm fail to detect face locations are eliminated from the experiment process. For data augmentation in the training phase, after cropping face locations images are randomly horizontally flipped, rotated

TABLE I

DIFFERENT ARCHITECTURE FOR APPLIED RESIDUAL NETWORK FOR LSTM-RESNET. BUILDING BLOCKS ARE SHOWN IN BRACKETS, WITH THE NUMBERS OF BLOCKS STACKED.

Layer name	ResNet-A	ResNet-B	ResNet-C
conv_0	$7 \times 7, 64$ stride 2		
conv_1	$7 \times 7, 128$ stride 1		
block_0	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$
block_1	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$
block_2	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$ $\begin{bmatrix} 1 \times 1, 32 \end{bmatrix} \times 1$	–
block_3	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$	–	–
Embedding	Average pool, $128 - d$ fc, softmax		



(a) Training accuracy (b) Learning rate

Fig. 7. Classification results for training different ResNet architectures on FaceScrub. The deeper network has better training accuracy, and thus training error.

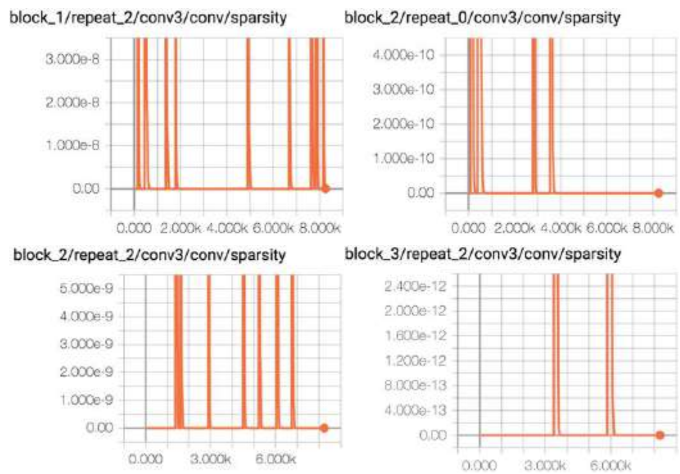


Fig. 8. Sparsity illustrations of the activation function at the end of each residual block for ResNet-A architecture.

$\{3°, 5°, 7°, \text{or } 10°\}$, and brightness and contrast of the images are manipulated.

A. Model Details and Results

Three types of residual network architecture are explored, see Table 1. Their practical differences lie in the variation in the number of residual blocks and consequently FLOPS. Depending on the application, the best model may be different. A model running at the embedded system as the edge device can have many parameters and handle more computation per second, where as the model running on a mobile phone has restrictions in a term of memory capacity and device battery life. Suggested approach in [19] is followed and feature extraction is performed by adding $1 \times 1 \times d$ convolutional layers rather than playing with different filter sizes. Dimension matching in connections is performed by the sole convolutional layer at the end of each block except the last one connected to the embedding.

On the FaceScrub dataset, the residual networks are trained to learn a feature mapping from facial images to a feature vectors in a compact euclidean space in which $L2-$norm distances present the similarity of faces [20]. To address hard samples problem [20], the residual networks supervised by center loss are trained and optimized by standard stochastic gradient descent instead of the original triplet loss function for such an embedding system. All residual architectures are trained with previously described folds, the average loss of a validation fold in multiple experiments is used to tune the models parameters. The best parameter is exploited to retrain models on the whole training folds and then the final result is reported on the fold test.

Figure 7a shows the classification accuracy obtained from the training on different permutations of the five training folds. The deeper residual architecture, ResNet-A, was training much faster, see figure 7b. The experiment illustrates the deeper network has better training accuracy, and thus training error. The proposed network is explored and examined with multiple probes to monitor the bias-variance trade-off of the weight vectors, in the training phase. To increase the algorithm performance, the hyper-parameters are fined and tuned to achieve a balance between bias and variance. In turn, the algorithm utilize all variables to learn fair assumptions about the form of the target function while suggesting small changes to the estimate of the target function, see figure 8 showing sparsity of the activation layer at the end of each residual block. Table II presents some comparisons between various versions of residual networks for top-1 error and top-5 error on validation fold and test fold. These major observations affirm ResNet-A reduces the top-1 error by 2.68%, which is a result from the successfully reduced learning error in the training phase, see figure 7b. This comparison verifies the effectiveness of residual learning on extremely deep systems. Figure 9 illustrates a classification result of the $128 - d$ embedding for ResNet-A architecture after applying $2D$ principal component analysis.

TABLE II
ERROR RATES (%) OF SINGLE-MODEL RESULTS ON THE FACESCRUB VALIDATION FOLD AND TEST FOLD.

Model	Validation Fold		Test Fold	
	top-1 error	top-5 error	top-1 error	top-5 error
ResNet-C	17.81	5.22	18.03	3.01
ResNet-B	15.37	3.12	16.36	1.88
ResNet-A	15.13	1.97	16.12	1.23

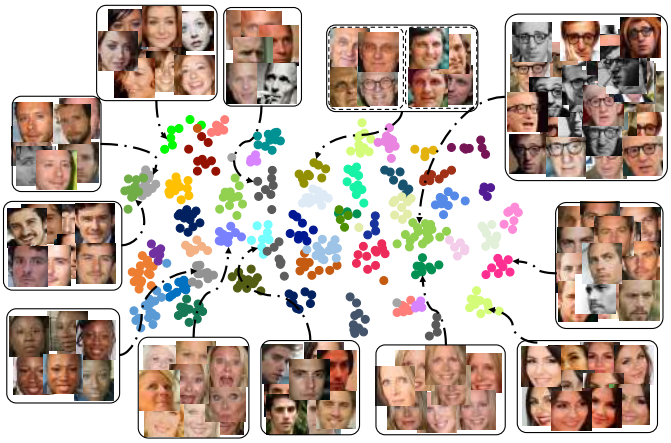


Fig. 9. Sparsity illustrations of the activation function at the end of each residual block for ResNet-A architecture.

The basic result of the LSTM-ResNet architecture on YTF dataset is summarized in table III. The result indicates that LSTM-ResNet architecture can learn a discriminant representation of the input facial images over a sequence of frames. Two protocols are introduced in table III, large protocol which indicates that the related method is trained using large datasets in sophisticated efforts, and small protocol indicating that small training dataset is used for the related method. Note that the proposed approach focuses on small protocol rather than the large one. The methods which are trained in large datasets show decent results in face recognition task on YTF. Significantly, the proposed LSTM-ResNet method exhibits analogous results. Note that face recognition methods try to recognize an individual on an image correctly, but LSTM-ResNet tries to recognize an individual on a sequence of frames recorded the person. The best obtained accuracy of ResNet-A architecture by itself is 84.4%, and 94.7% accuracy for LSTM-ResNet-A architecture, which is a notable improvement compared to the ResNet-A. By learning temporal features from input sequences, the LSTM-ResNet-A model performs superior than general residual neural network and analogously to sophisticated architectures trained on large datasets.

VI. CONCLUSION

A biometric secure modality is proposed to recognize an individual. The model performs base on the temporal extracted person facial feature changes over time. A neural network based architecture is proposed, named LSTM-ResNet, to make a precise recognition task. LSTM-ResNet model leverages cascaded residual neural network blocks to extract the temporal facial feature and map them into the embedding vector; each embedding vector presents a sole person in a frame. Then for each individual related embedding vectors are fed into LSTM units to learn the long-term temporal structure from a sequence of frames captured by a camera(s), and then perform the recognition task. The model shows decent results comparing with sophisticated models, although it is trained in small dataset. The model is designed to implement as a device-embedded biometric authentication to keep data securely decentralized on local devices and safe from hackers. In addition, the decentralized model preserves users privacy and eliminate the inconvenience of storing sensitive biometric data.

TABLE III
PERFORMANCE OF DIFFERENT METHODS ON YTF DATASETS.

Method	Protocol	Acc. on YTF
DeepFace [21]	Large	91.4%
FaceNet [20]	Large	95.1%
Deep FR [22]	Large	97.3%
ResNet-A	Small	84.4%
LSTM-ResNet-A	Small	94.7%

ACKNOWLEDGMENT

This work was supported, in part, by Open Cloud Institute at University of Texas at San Antonio, Texas, USA and by Grant number FA8750-15-2-0116 from Air Force Research Laboratory and OSD, USA. The authors gratefully acknowledge use of the services of Chameleon cloud and Jetstream cloud, funded by NSF awards 1419165 and 1445604 respectively.

REFERENCES

[1] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*. Springer, 2016, pp. 189–248.

[2] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1–24, 2015.

[3] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2009, pp. 235–253.

[4] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[5] M. Blanton and M. Aliasgari, "Secure outsourced computation of iris matching," *Journal of Computer Security*, vol. 20, no. 2-3, pp. 259–305, 2012.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[7] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. , "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[9] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[12] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[13] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

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

[15] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning." *ICML (3)*, vol. 28, pp. 1139–1147, 2013.

[16] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 343–347.

[17] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 529–534.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[19] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

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