An Offline Writer-independent Signature Verification System using AutoEmbedder

Conference Paper · December 2021

DOI: 10.1109/ICCIT54785.2021.9689780

5 authors, including:

Zabir Mohammad
Bangladesh University of Business and Technology
4 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Israt Jahan
East Delta University
2 PUBLICATIONS **103** CITATIONS
SEE PROFILE

Md Mohsin Kabir
Mälardalen University
64 PUBLICATIONS **1,071** CITATIONS
SEE PROFILE

M. Ameer Ali
Bangladesh University of Business and Technology
27 PUBLICATIONS **234** CITATIONS
SEE PROFILE

2021 24th International Conference on Computer and Information Technology (ICCIT), 18-20 December, 2021.

An Offline Writer-independent Signature Verification System using AutoEmbedder

Zabir Mohammad, Israt Jahan, Md. Mohsin Kabir, M. Ameer Ali, M.F. Mridha
Department of Computer Science & Engineering, Bangladesh University of Business & Technology, Dhaka, Bangladesh
zabirmohammad02@gmail.com, israt0jahan7@gmail.com, mdmkabi@gmail.com, dmaa730@gmail.com, firoz@bubt.edu.bd

2021 24th International Conference on Computer and Information Technology (ICCIT) | 978-1-6654-9435-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCIT54785.2021.9689780

Abstract—Handwritten Signature is considered one of the most effective behavioral biometrics. It plays an important role in identifying and verifying persons for banking access control, criminal investigation, legal support, etc. Since the handwritten signature is used in such a high prominence, its misuse can be dangerous. Deep learning-based verification approaches are becoming extremely popular to reduce the risk of signatures misuse. Signature verification depends on pairwise constraints to verify if the person is genuine that he/she claims to be or forged. This paper proposes an Autoembedded system that uses Deep Neural Network (DNN) with the pairwise loss for signature verification. The model either generates embedding vectors closer to zero if the input pair is in the same class or generates a value greater or equal to α (a hyperparameter) that indicates a different class. The proposed approach uses a Siamese network that computes the pairwise distance in feature learning phase. The performance has been evaluated based on CEDAR dataset in a writer-independent (WI) context, and the experimental result shows clear distance between the genuine and forged signatures and verifies genuine ones.

Index Terms—Handwritten Signature Verification, AutoEmbedder, Twin Network, Deep Learning

I. Introduction

In this era, people are used to handwritten signatures in their daily life for its non-invasive standard. Though electronic signatures are being acquainted with the digitization of countries, handwritten signatures have been prevalent since the fourth century [1]. For security controls, handwritten signatures are used widespread in biometric systems like iris, fingerprints which can be deployed for identification and verification. Signature identification systems match and compare with every possible identity of the related database. On the other side, verification is needed to check if the person is genuine that he/she claims to be. It's basically used to distinguish between authentic and forged signatures.

For the expansion of documentations, Handwritten Signature Verification (HSV) is becoming a challenging task for the experts. Therefore to reduce the risk of signatures misuse, online and offline both come across [2], [3]. Online HSV uses electronic input devices for a dynamic signing process. In offline, the manual signatures are digitized by scanning and verified based on the strokes positions of handwriting. Offline HSV has two different kinds: Writer-independent (WI) and Writer-dependent (WD). The writer-dependent module needs to retrain reference signatures if any new signer arrives, which holds a drawback indeed. In contrast, the writer-independent approach does not require any modification for the existing models when adding a new verified signer. It executes well even for a few signatures per writer and consumes new writers without generating or retraining the disturbance of new models.

Several researchers have already proposed Artificial Neural Network (ANN) [4], Hidden Markov Models (HMM) [5], Support Vector Machine (SVM) classifier [4], and other techniques as well. For the massive progress of deep learning, clustering has been popular where it is determined to congregate similar data points together from the dataset. It can lead to a great impact on the implementation of novel approaches to signature verification. Hence, we are using the embedding system AutoEmbedder, based on Siamese Neural Network (SNN) architecture. In this paper, we tend to deal with the WI handwritten signature verification with skilled forgery where the forger knows the signer's signature accurately and is able to duplicate the genuine one.

The main contribution can be included as follows,

- We have reviewed and analyzed the difficulties of text-dependent offline signature verification systems for the deep learning domain.
- This paper utilizes the framework by using single signature reference pattern per writer for skilled forgery verifying problems.
- The proposed model generates embeddings in AutoEmbedder, deploying robust performance for signature verification.
- Finally, the paper introduces a modified Xception architecture with a lesser number of parameters.

The rest of the paper is constructed as follows: The related research is described in section 2. The overall proposed architecture of deep CNN is explained in section 3. Section 4 includes results and comparisons. Finally, Section 5 draws the conclusion and future scope.

II. Realted Work

Past few years, researchers bring forward many methods and feature learning techniques of handwritten signature verification [6], [7], [8]. As a result, it's been fairly difficult to make a comparison with the existing system, which is

978-1-6654-9435-9/21/$31.00 ©2021 IEEE

already developed even with some of their own databases. Here, we approach a discussion of some recent ongoing works that address deep learning and other approaches with forgery problems.

Rateria et al. [9] proposed offline signature verification using a fully connected 3 layer Deep Convolutional Neural Network (DCNN). The author used 3 databases: CEDAR, GPDS Synthetic & BHSig260 signature corpus, and gained accuracy of 80.26%, 75.06% for Writer-Independent(WI) verification in CEDAR, BHsig260 (Bengali) respectively. Shaikh et al. [10] proposed soft-attention and cross-attention combined methods to represent the highly coordinated and noticeable points of the inputs in feature space. The methods were implied based on Inception-Resnet-v2 (IRv2) and 92.37% accuracy achieved on CEDAR cursive, "AND" dataset for detecting WI cases. Poddar et al. [11] presented a CNN and Crest-Trough method for signature recognition and Corner detection algorithm, SURF algorithm for forgery detection. The system attained almost an average accuracy of 85%-89%.

Over time, researchers found that SNN is performing satisfying results on signature verification. Xing et al. [12], Mshir and Kaya [13] proposed Convolutional SNN for offline signature verification. Unlike the existing works, a unified framework of feature extraction and metric learning combination, evaluation through triple loss of scheme methodology terminated better results. Amruta B. Jagtap et al. [14] proposed offline signature verification using SNN with signature embeddings. Experimental results on GPDS Synthetic and MCYT-75 benchmark signature databases achieved 84.58% and 85.38%, respectively. Chen et al. [15] came with a process for offline HSV based on graph matching, word shape descriptors, and thin-plate spline mapping. The method was evaluated on a testbed of signatures from 55 volunteers & achieved an accuracy of over 91%. Kumar et al. [16] extracted features known as surroundedness from binary signature images for WI offline HSV. Multilayer perceptron and SVM were implemented to examine the efficacy where achieved 86.24% & 91.67% on GPDS300 corpus and CEDAR, respectively.

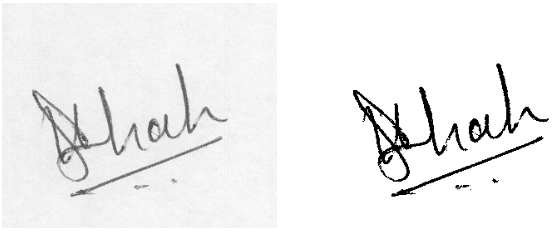
Although there's an impressive improvement in Deep Learning techniques, DCNN and other approaches can still be modified by using embedding inputs in feature space. This can convert high dimensional data to low dimensional meaningful data using dimension reduction techniques which could be a revolutionary step for verifying the forgeries. In this paper, we have discussed how AutoEmbedder solves the issue of offline handwritten signature verification.

III. Methodology

In this section, at first, the preprocessing performed on signature images is explained. Deep CNN and Siamese Network are discussed then, followed by a detailed description of the proposed model. Finally, the last section contains the key architecture of AutoEmbedder.

A. Data Preprocessing

Our proposed signature verification model is based on an offline platform, inputs of offline signature verification scanned handwritten signature images. However, some issues like lighting and background noise have been observed in those scanned images. We've applied a supplementary python script based on OpenCV script [17] to eliminate these issues. Figure 1 represents (a) the original version and (b) indicates the refined version of the signature images. Afterward, we've inverted the images and converted the RGB channel to Grayscale. The signature images we've considered have inconsistent size ranges of 153 x 258 to 819 x 1137. We have rescaled the height of the image to 80 pixels while maintaining the particular aspect ratio. Then a random 80 x 113 image patches are cropped from the resized images. Furthermore, we normalize the images, which makes convergence faster while training Deep CNN Architecture.



(a) Original version

(b) Improved version of signature image

Fig. 1: The figure present (a) input sample of CEDAR dataset and (b) improve sample after applying python script based on OpenCV

B. Deep Convolution Neural Network and Twin Network

Deep Convolutional Neural Network (DCNN) consists of various convolutional layers with several kernel shapes. Pooling layers are placed between convolutions to encapsulate and downsamples the output before proceeding to the next convolutional layer. Various activation functions are used to introduce nonlinearity in the network. Batch-Normalization is used for a faster and more stable network through normalization of the layers' inputs by re-centering and re-scaling. Generally, during training DCNN models, a differentiable loss function is chosen to apply the gradient descent to the network. By using gradient descent, the loss is backpropagated through the network's weights. As a result, the network's weights can be optimized. The backpropagation can not be applied to each training instance when the training sample is large, which is time-consuming. Batch optimization gives a fair alternative to optimize the Deep CNN.

Siamese neural network (SNN) architecture has two identical subnetworks having the same architecture with the same number of parameters. The weights of the CNN can be shareable, which reduce the parameters, and during backpropagation, updating parameter is mirrored for the subnetworks. This framework can be used effectively for dimensionality reduction in weakly supervised metric learning [18]. It aims to learn a neural network that can discriminate between the image pairs, which is the standard verification task for image recognition. A loss distance is joining these subnets between feature extraction of the network's each side.

C. Proposed Method

In this work, we use AutoEmbedder [19], which relates to the dimension reduction technique using SNN architecture. Basically, a classifier network classifies input depending on the activation value placed on the output layer. Any DNN architecture for classification, output layer refers to the softmax layer. Then, the previous layer of this is known as 'Feature space,' which works the output of Autoembedder embedding. The dimension of the generating embedding points to the particular input refers to feature space layer's nodes. Unlike autoencoder, Autoembedder not only works better in downscaling data dimension higher to lower feature space further used in clustering analysis.

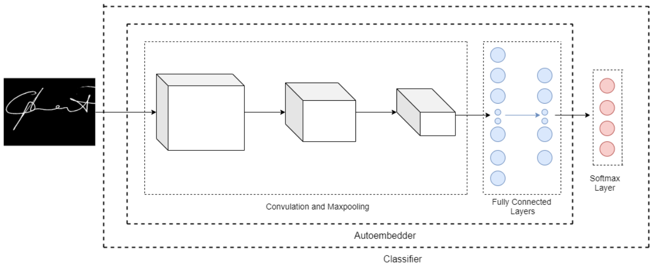


Fig. 2: This figure illustrates Autoembedder extraction from traditional Convolutional Neural Network Classifier.The previous layer of the softmax layer refers the output of the Autoembedder known as 'feature space'.

The AutoEmbedder architecture learns based on the similarity and dissimilarity of the particular input pair. Each branch of AutoEmbedder can be used as a function to generate embedding the input image into the feature space. In this process, an AutoEmbedder subnetwork transforms dimensional data higher into lower meaningful embedding points. The embedding point can distinguish between signature images using Euclidean distance. where, $p = [p_1, p_2, p_3, ..., p_n]$ is the embedding points of first identical subnetwork and $q = [q_1, q_2, q_3, ..., q_n]$ is the embedding points of second identical subnetwork. This can be calculated as,

$$dist(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

The distance is further passed through a Rectified Linear Unit (ReLU) activation function with an upper bound value α.

$$Relu(x) = \begin{cases} x & ; \quad \text{if } 0 \leqslant x < \alpha \\ \alpha & ; \quad \quad \text{if } x \geqslant \alpha \end{cases} \tag{2}$$

Furthermore, the distance calculation layer is pass to a sigmoid activation function. Sigmoid functions have the property that they map the output between 0 and 1.

$$S(x) = \frac{1}{1 + e^x} \tag{3}$$

AutoEmbedder aims to bring output feature vectors closer for similar input pairs, and dissimilar input pair feature vectors are forced to be separated away. Due to the loss function, our proposed method, feature space will maintain a property that images in the same class or genuine-genuine signature pair will be close to each other. And images of different label or genuine-skilled forgery pairs will be separated away with a hyper-parameter α value.

To train the AutoEmbedder with a precise target value, a hyperparameter α is to be decided that indicates the distance threshold of genuine and forgery pairs. For each input pair xi and xj, the pairwise constraint is α if there exists a genuine-forgery pair. Otherwise, the distance value is estimated to be 0. The target values can be mathematically expressed as,

$$Label(x_i, x_j) = \begin{cases} 0 & ; \quad \text{if } x_i \text{ and } x_j \text{ genuine-genuine pair} \\ \alpha & ; \quad \text{if } x_i \text{ and } x_j \text{ genuine-forgery pair} \end{cases} \tag{4}$$

If the input pair refer to the same class, the AutoEmbedder pair is trained to generate embedding points closer to zero. Otherwise, it is instructed to generate embedding points greater or equal to α.

Most of the SNN based architecture is implemented on contrastive loss function. Contrastive loss function evaluates the siamese network's distinguished performance of image pairs. Because our purpose is to not only classify the image pairs but also make a clear difference between image pairs. The Contrastive loss for each iteration batch is calculated as,

$$Loss(Y, \hat{Y}) = Y * \hat{Y} + (1 - Y) * max(margin - \hat{Y}, 0) \tag{5}$$

where, Y be the ground truth vector, and $\hat{Y}$ be the predicted pairwise vector distances. The margin defines a radius around the embedding space of a sample so that dissimilar pairs of samples only contribute to the contrastive loss function if the distance is within the margin.

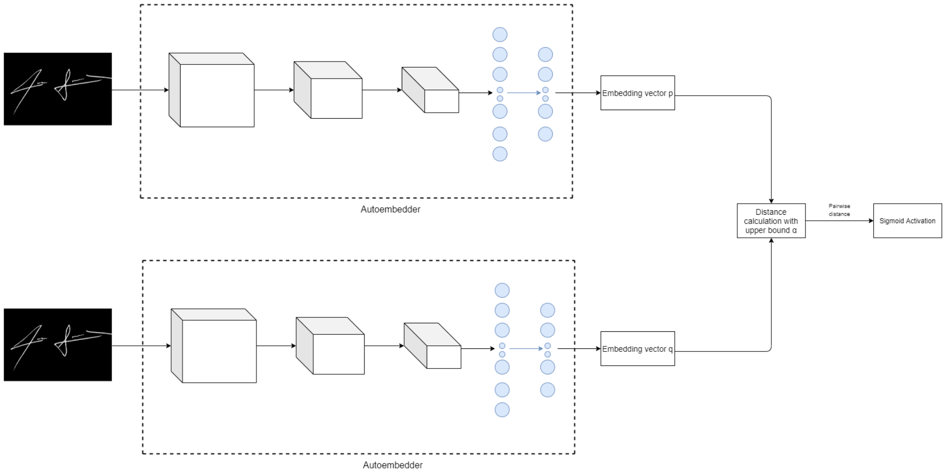


Fig. 3: This figure illustrates the overall SNN training architecture of AutoEmbedder

D. Autoembedder Architecture

The architecture of the Autoembedder is inspired by ‘Xception’ [20] architecture for image recognition problems with a competitively smaller number of parameters. The combined form of VGG and Inception architecture is called Xception. Furthermore, traditional CNN models use general convolutions, but Xception uses spatial convolutions which work on multiple filters side by side and tend to understand texture features well, which leads to achieving great convergence speed, slight accuracy improvement, and a meaningful reduction in model parameters. The architecture and complete list of the Autoembedder layer parameters in SNN are presented in figure 4. Our proposed model for constructing Autoembedder consists of 14 convolutional blocks and 3 Maxpooling layers. The 14 convolutional layers are formed into seven modules having linear residual connections without the first and last ones.

In this study, a convolutional block contains an activation layer, a convolutional layer, and a batch Normalization layer. Each of the Convolutional layer’s kernel sizes is 3x3, and stride is 1 except the first convolutional layer, which has a stride of 2. Each max-pooling layer pool size is 3, and stride is 2. Unlike Xception, our proposed architecture is divided into three parts: entry flow, middle flow, and exit flow, and each of them consists of 6,6,4 modules respectively.
We trained the model using Adam for 100 epochs and a mini-batch size equal to 32. We started initially with a learning rate (LR) equal to 0.0001 and a Weight decay of 0.00001.

IV. Evaluation

In this section, we’re defining the evaluation metrics first and in order to evaluate our proposed system, we’ve gathered samples from the CEDAR Database. Then, the experimental setup is described, and finally, we present the experimental results with a detailed explanation.

A. Evaluation Metrics

We use accuracy, false acceptance rate (FAR), and false rejection rate(FRR) depending on the confusion matrix. In all these evaluations, a chosen threshold is used for measuring the distance pairwise to check whether the signature belongs to a similar cluster or not. However, the accuracy formula can be defined as,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

where, TP = true positive , TN = true negative, FP = false positive, FN = false negative
The completion of the system is measured by two individual error rates to verify a signature:

- The percentage of false marked genuine signatures that are rejected by the system is called False Rejection Rate (FRR).
- The percentage of genuine marked forgery signatures that are accepted by the system is called False Acceptance Rate (FAR).

B. Datasets

The CEDAR database known as the Center of Excellence for Document Analysis and Recognition has been applied in this paper. Consisting of 55 signers that have 24 genuine and the other 24 forgery for each, CEDAR is much more compressed, unlike the other existing databases. The image comprehends a signature that comes with a black pen. Therefore, all of the included images are scanned in grayscale with 300DPI/8bit.

C. Experimental Setup

Python is used in the model which usually strives for a less-shuffled, simpler language while giving developers a way out by the programming methodology. Using Keras, the deep learning architecture has been implemented. Numpy is used to compute mathematical operations and

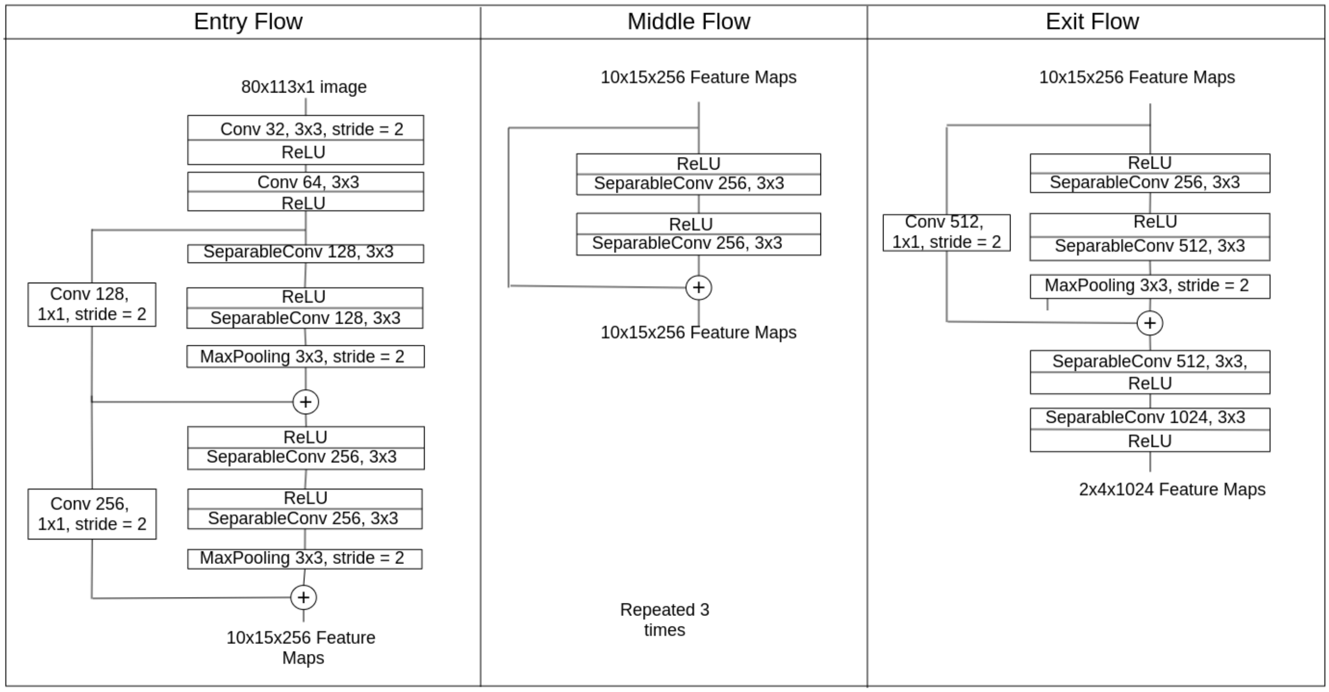


Fig. 4: The figure illustrates the entry flow, middle flow and exit flow of the autoembedder architecture. The entry flow network recieves input image, and the processed data is passed to the middle flow network. It further forwards the processed data to exit flow. Each of the convolutions is followed by a batch normalization layer, not illustrated in the image.

TensorFlow helps to execute the GPU compilation of neural networks.
Moreover, we have divided the dataset for executing our designed method. Since the CEDAR contains 55 signature writers only, we've selected 10 fixed individuals as test data and used the remaining signatures of 45 individuals as train data. Here we trained pairwise genuine-genuine and genuine-forged with skilled forgery.

D. Results and Comparison

The proposed SNN architecture with identical subnetwork Autoembedder for signature verification is evaluated by accuracy, FAR, FRR measurements based on CEDAR dataset. Table I illustrates the performance of the proposed approach and compares the predicted result with the combined SNN and autoencoder model [21]. The identified result shows that using autoencoder as a Siamese identical subnetwork does not perform well in CEDAR dataset. Autoencoder siamese architecture gives only 0.86 accuracy while AutoEmbedder siamese approach gives an accuracy of 0.92. It is noted that the performance measurement by FAR and FRR are also higher for our proposed approach compared to the autoencoder architecture.

TABLE I: The table represents the accuracy, FAR, and FRR of siamese neural network with autoencoder and the proposed AutoEmbedder Architecture

Model	Accuracy	FAR	FRR
Prajapati et al [21]	86.73	7.78	13.34
Ours	92.76	7.34	6.94

TABLE II: This table illustrates the accuracy, FAR, and FRR of various famous CNN architecture models evaluated on the validation dataset.

Model	Accuracy	FAR	FRR
VGG16	88.98	10.14	12.69
MobileNet	92.01	8.47	8.29
Inception	91.25	8.72	8.03
Xception	92.47	7.32	7.53
our	92.76	7.34	6.94

Besides, Table II demonstrates the proposed architecture evaluation in different models for the validated dataset. Almost all of the models generate better accuracy but VGG16 slightly shows high acceptance of forgery signature as genuine alongside the rejection of genuine signature. On the contrary, Xception reaches the highest FAR score of 7.32 among all of the models. The proposed model gives the best accuracy of 92.76% with 6.94% false rejection rate which indicates a robust model.

The benchmarks of the various models on CEDAR dataset are provided in Table III. The comparison shows