Journal of the Institute of Engineering
Volume 16, No. 1 Published: April 2021
ISSN: 1810-3383

Printed in Nepal

Signature Verification using Convolutional Neural Network and Autoencoder

Prakash Ratna Prajapati [a], Samiksha Poudel [b], Madan Baduwal [c], Subritt Burlakoti [d], Sanjeeb Prasad Panday [e]

[a, b, c, d] *Dept. of Computer and Electronics Engineering, Kantipur Engineering College*
[e] *Dept. of Computer and Electronics Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal*
Corresponding Author: [e] sanjeeb@ioe.edu.np

Received: 2020-08-15 Revised: 2021-02-06 Accepted: 2021-02-25

Abstract:
Signature has been one of the widely used verification biometrics out there. Handwritten signatures are used in cheques, forms, letters, applications, minutes, etc. The Signature of every individual is unique in nature, that is why it is essential that a person's handwritten signature be uniquely identified. Signature Verification is a widely used method for authenticating any individual during absence. Human verification is prone to inaccuracy and sometimes indecisiveness. This paper presents an investigation of using Convolutional Neural Network (CNN) for Writer-Dependent models in signature verification. Random distortions were generated in genuine images using an autoencoder to get forged signatures, which were passed to the classifier during training. The paper details all the pre-processing steps carried out on the image and shows various test results for changing the number of training sets of images. The average test accuracy for Persian dataset is 83% when the system was trained with 22 genuine images. There was a decrease of 9.4% in accuracy when the model was trained with 9 genuine images.

Keywords: Offline Signature Verification, WD (Writer Dependent), CNN (Convolutional Neural Network), FAR (False Acceptance Ratio), FRR (False Rejection Ratio), Autoencoder

1. Introduction

Handwritten signature verification is the technique of confirming the user's identity by using the handwritten signature of the user as a form of behavioral biometrics. Many studies have been done in automatic handwritten signature verification using different artificial models. Some of the former researches are reviewed in the Related Works section of this research paper. One of the advantages that signature verification has over many other forms of biometric technologies, for example, voice authentication, retina scan, fingerprint, etc is that handwritten signature is already one of the most widely accepted methods of identification in our society for hundreds of years as well as it doesn't require complex technologies. Still now, banks and other institutions accept signature-based verification methods for user authentication. Furthermore, digital techniques of PIN and password are also the most used technique for validating a user in a system. But the issue with these verification methods is that these details can be forgotten or can get stolen by others.

Signature verification can be divided into two main areas on the basis of which data acquisition method is used, these are: offline and online signature verification. In offline signature verification, the signature is presented in a document which is scanned to capture its digital image. Online signature verification uses electronic hardware such as electronic signature capture pads, which records the gestures made by pen during signing. Each of these techniques are applicable in their own respective fields. Online signature verification is utilized in authorization and authentication of personal computer users for accessing censorious and highly sensitive information, e.g. credit card details. Offline signature verification is used to verify signatures in documents and bank cheques [1].

Similar to other biometric verification systems, images

of genuine signatures of the user needs to be collected. Later, when users provide a signature which they claim to be of a particular individual, the system trains a writer dependent classifier [2] using those images which will be used to classify signatures for that particular user. A certain threshold value can be assumed on the basis of confidence required for signature verification. If the dissimilarity exceeds beyond that threshold, the signature is rejected (i.e. marked as a forgery).

1.1 Convolutional Neural Network (CNN)

In 1995 A.D., Yann LeCun and Yoshua Bengio introduced the concept of CNNs. CNN is a feed-forward neural network which has the ability of extracting topological features from the input image. It extracts features from the image and those extracted features are inputted to a classifier which categorizes the image. CNNs are generally invariant to distortions and simple geometric transformations like translation, scaling, rotation and squeezing.

CNNs combine 3 architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and spatial or temporal sub-sampling [3]. CNNs are usually trained like a standard ANN using back propagation.

CNN layers are alternation of convolutional layers and max-pooling layers. A convolutional layer is used to extract features from local respective fields. It is organized in planes of neurons called feature maps. In a network with a 5x5 convolution kernel each unit has 25 inputs connected to a 5x5 area in the previous layer, which is the local receptive field. Each connection is assigned with a trainable weight which is shared by all units of one feature map. This feature allows reducing the number of trainable parameters called weight sharing technique and is applied in all CNN layers.

Convolution Layer:
During convolution operation, Filter (kernels) which is a nxn matrix of value, is convolved around each pixel of the image along with the surrounding region of that pixel. The operation includes element wise multiplication of the filter matrix with the same dimension region in the image matrix. The resulting sum is then output into a result image at the same pixel coordinate on which the present convolution kernel was originally centered on.

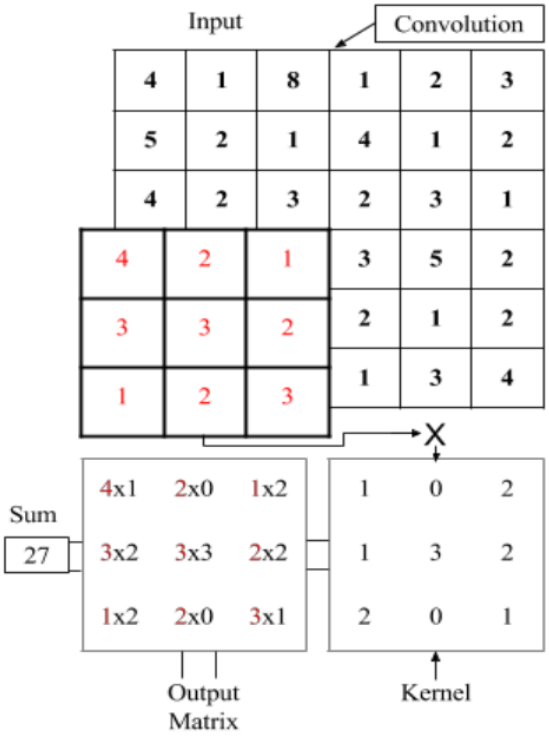


Figure 1: Convolution Operation

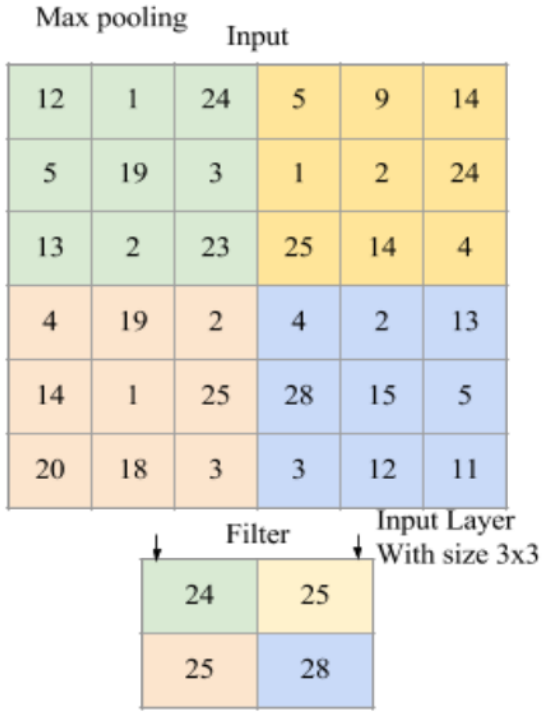
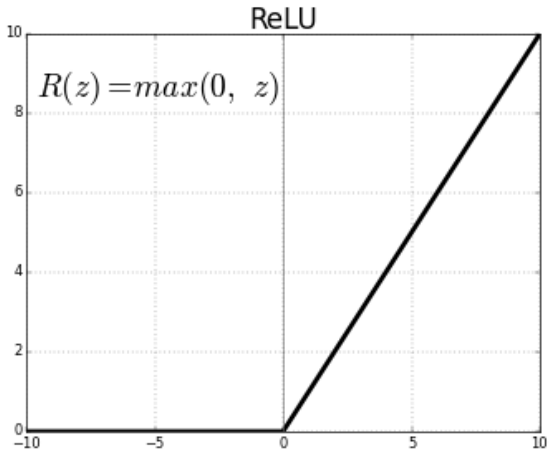
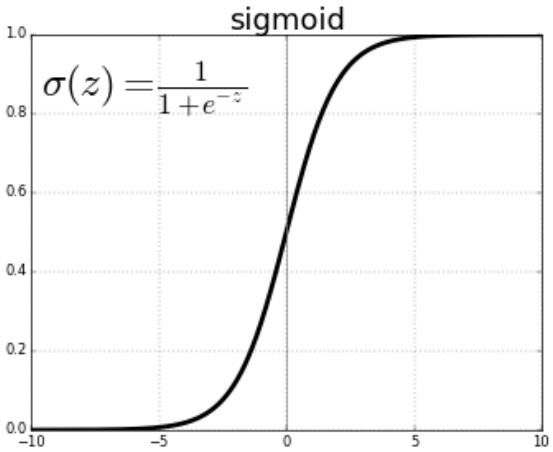


Figure 2: Max-Pooling Operation

Max-Pooling Layer:
The main objective of max-pooling layer is to down-sample an image. Max-Pooling operation divides



(a) ReLU activation



(b) Sigmoid activation

Figure 3: Activation Functions

the image matrix into multiple regions of pixels, separated both vertically and horizontally by a stride of N. Maximum value of each divided matrix is outputted to a smaller output image.

Rectification Linear Unit (ReLU): ReLU is an activation function on the outputs of the convolution layer. For each node, the mathematical expression for ReLU activation is:

$$f(x) = \max(0, x)$$

This sets all negative pixel values from the convolution to zero, while leaving each positive value unchanged.

Sigmoid Activation:
A sigmoid function is a mathematical function having a characteristic ”S”-shaped curve or sigmoid curve. The equation is given by:

$$\text{sigma}(x) = \frac{1}{1+e^{(}-x)}$$

The output of sigmoid function ranges between 0 and 1.

1.2 Autoencoder

Autoencoder consists of an encoder, a decoder, and a loss function. It is a popular approach to unsupervised learning of complicated distributions. The main objective of an autoencoder is to compress the data by encoding it using mathematical equations or other mechanisms and decompress the down-scaled data to original size using reverse process. During this process some data gets lost causing random variations in the original data. More the amount of compression, the more information gets lost by it.

2. Related Works

In general, there are two approaches used for offline signature verification, these are: writer-dependent (WD) and writer-independent (WI). In the WD scenario, a classifier is trained for each writer which is responsible for authenticating that writer's signatures. In the WI context a single classifier is trained for all writers which is responsible for categorizing the unknown entry signatures to one or more class of signatures in a dissimilarity space [4].

Kumar [5] presented an offline signature verification method using global and texture features of signature. The scheme is based on a technique that applies pre-processing on the signature to get a binary image and then calculate the global and texture feature points from it and maintain a feature vector. All calculations were done based on those feature points. Artificial Neural Network (ANN) was used as a classifier for training and verification of signatures.

Batista et al. [6] presented a survey which defined the most important techniques for feature extraction and verification of offline signature. They further presented strategies for facing the problems of minimum number of data.

Khalajzadeh et al. [7] performed signature verification implementing Convolutional Neural Network (CNN) reaching 95 percent of accuracy in Persian dataset. Miah et al. [8] in 2015, developed a model using the ANN in order to analyze signatures and balance the amount

on cheque. But, his proposed system didn't consider signatures which were signed on physical cheques.

Guerbai et al. [9] submitted One-Class SVM (OC-SVM) based writer-independent Human Signature Verification system which aims to lower the problems of multiple number of signee in the system. The proposed model approaches only single class i.e. genuine images of signatures, which is a good characteristic. The classifier used only genuine signatures for training. But the low number of genuine images was a difficult challenge for the system.

Brittany et al. [10] got 83 percentage accuracy using CNN to verify signature. They used SIGCOMP 2011 dataset and implemented CNN to evaluate any abnormalities in the signature image. Unique features from signatures could be collected and processed using multiple layers of receptive fields and hidden layers. Their system was primarily focused on the banking industry to detect and minimize fraud committed.

Rezaei et al. [11] performed signature verification on Persian dataset using fully convolutional networks (FNN). Their method proposed a FNN with image input size of 270 x 360 pixels. The proposed method does not include any pre-processing methods. The system could predict signatures with an accuracy of 76.71

Our paper proposed a new CNN classifier for training purposes. In real-life scenarios, the number of signature images provided by the user is less in quantity, so the paper proposes using an autoencoder to generate fake images using the genuine ones. These forgery images were used while training so the system could determine minor distortions in genuine images.

3. Proposed Method

In the field of offline handwritten signature verification, the authors seek for a classifier that's good enough to verify a user's signature with minimal dataset during training (approximately 9 genuine images and random image as forged for binary classification). CNNs have proven to be successful in image processing based machine learning tasks. CNNs have a very good performance in feature selection from an image by finding the appropriate value for filters, which convolve around the 2D matrix of the image. Before feeding the images for training, these must be pre-processed and following steps are carried on.

3.1 Image Pre-Processing

1. **Image Acquisition and Resizing:** Image of any shape, normally 1024x768, was collected from a digital camera. The image was resized to shape with maximum pixel length 600, maintaining the initial aspect ratio, for faster operations.

2. **Median Blur:** Median blur in a window size of 7x7 was carried on each RGB channel in order to eliminate any grainy noise in the captured image.

3. **Gray Scaling:** The 3 channel RGB image was then converted to single channel grayscale image by obtaining the mean of channels in each pixel.

4. **Fast Non-Local Means Denoising:** OpenCV tool fastNlMeansDenoising [12] was used to remove fine noise with Template window size = 10 pixels and Search window size = 21 pixels.

5. **Image Segmentation:** Global thresholding was performed on the image, with thresholding value equal to average value of image pixels. Threshold Value = Avg(image).

6. **Image Localization:** Cropped a rectangular portion of image that contains only the signature.

   Algorithm 1: Image Localization

   Generate Horizontal & Vertical Spatial Histogram
   Initialize:
   $left = 0$
   $right = widthOfImage$
   $top = 0$
   $bottom = heightOfImage$
   Assign: $count = 0$
   for i in range(length of side) **do**
   if $count \leq 0$ **then**
   $count = 0$
   if $SpatialHistogram[i] \neq 0$ **then**
   $left = i$
   count += 1
   end if
   else
   if $SpatialHistogram[i] \neq 0$ **then**
   $count+ = 1$
   else
   $count- = lengthOfSide/100$
   end if
   if $count > lengthOfSide/30$ then

18: **break**
19: **end if**
20: **end if**
21: **end for**
22: Repeat Step 3 to 21 for top
23: Repeat Step 3 to 21 in reversed range(length of side) for right and bottom Crop image in co-ordinates (left, top) and (right, bottom) to segment image

7. **Padding:** Padded the image on the necessary side in order to convert the image into square without distorting the signature image.

   Algorithm 2: Image Padding

   1: Initialize:
      $leftPad = 0$,
      $rightPad = 0$,
      $topPad = 0$,
      $bottomPad = 0$
   2: **if** $ImageWidth < ImageHeight$ **then**
   3: $leftPad = int((ImageHeight - ImageWidth)/2)$
   4: $rightPad = leftPad$
   5: **else**
   6: $topPad = int((ImageWidth - ImageHeight)/2)$
   7: $bottomPad = topPad$
   8: **end if**
   9: Pad the image with values $leftPad$, $rightPad$, $topPad$, $bottomPad$ on 4 sides with $value = 255$

8. **Image Resize:** The image was resized using Nearest neighbourhood to 300x300, as per the input shape of CNN model.

9. **Image Negative:** Generated negative of the image by: Negative = 255 - Image

3.2 Proposed CNN Architecture

The architecture of the proposed model consists of 5 convolutional layers each followed by a max-pooling layer. The output was then flattened and followed by ANN for binary classification. Convolutional layers were used to extract features from signatures while ANN was used to categorize the image. The architecture is depicted in figure 4.

The input image of size 300x300 was fed to the network. In the first convolution operation, 16 filters of dimension 9x9 were used to convolve in the image followed by max-pooling operation in the region of 2x2. In the second convolution operation, 32 feature maps were sampled from previous 16 layers using a filter of dimension 5x5 followed by max-pooling in a region of 2x2. In the third convolution operation, 32 feature maps were sampled using a filter of dimension 3x3 followed by max-pooling in a region of 3x3. In the fourth convolution operation, 16 feature maps were sampled from the previous 32 layers using a filter of dimension 2x2, followed by max-pooling in a region of 2x2. In the fifth convolution operation, 8 feature maps were sampled from the previous 16 layers using a filter of dimension 3x3, followed by max-pooling in a region of 2x2. Rectifier Linear Unit (ReLU) was used as activation function in all five convolutional layers.

The tensor was then flattened into a single dimension having 1936 neurons. Another fully connected hidden layer of 256 neurons was connected using 20% dropout in connection. Dropout was added in order to make the model less overfit. Similarly, another fully connected hidden layer of 128 neurons with 40% dropout was connected. Both of the hidden layers implemented ReLU activation functions. Finally, the output layer of a single neural network using Sigmoid activation function was used to determine output between range 0-1. The final output of neuron defined the degree of signature match with the genuine signatures. 1 being the accurate match while 0 being no match.

3.3 Proposed Autoencoder Architecture

Autoencoder was used in the system in order to generate forgery images from the genuine images by convolving and downsampling the images, and rescaling it back to the original dimension. When encoding was performed, the image data gets compressed which on further upsampling produces a noisy image as shown in figure 4. The encoder part consisted of 5 convolution operations, each followed by max pooling. Similarly, the decoder part consisted of 5 convolution operations, each followed by up sampling operation. The loss function implemented was Binary Cross-entropy with adadelta optimization. Adadelta optimization seems to work better on fewer training epochs. The proposed autoencoder architecture along with kernel dimensions taken on each layer is depicted in figure 5.

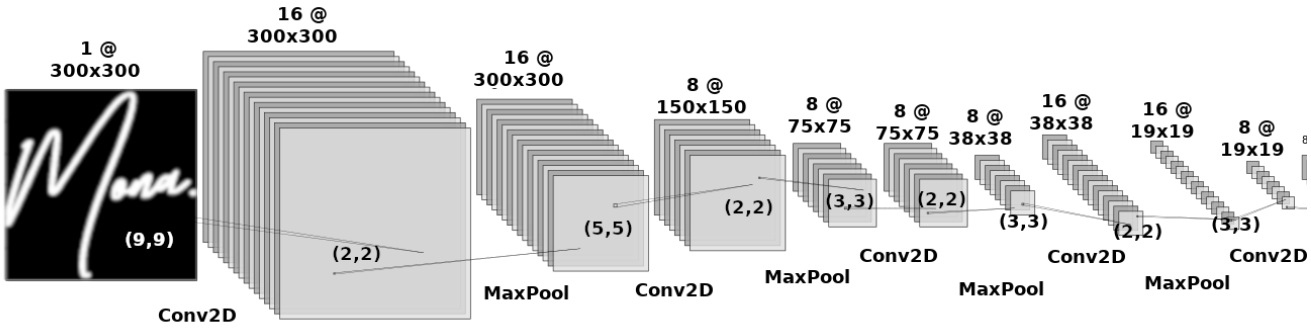
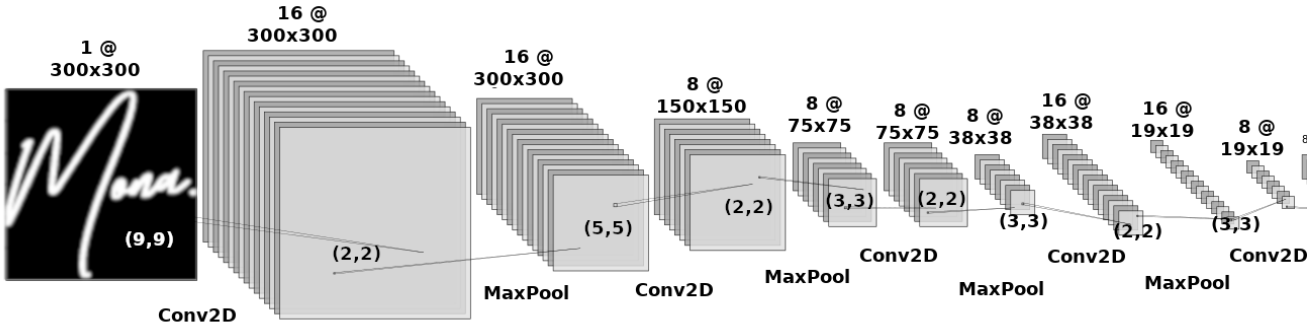
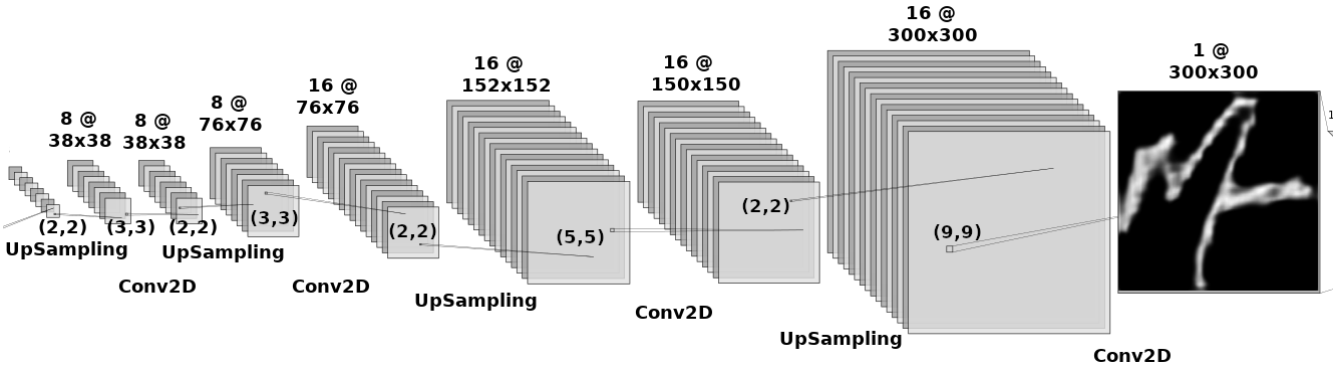


Figure 4: Proposed CNN Model Architecture



(a) Encoder



(b) Decoder

Figure 5: Proposed Autoencoder

4. Results and Discussion

4.1 Datasets

In this research, Persian Signature of 30 different people was used [13]. Each Individual dataset consisted of 27 genuine images, and 3 categories of forgery: Opposite Hand (3 images each), Simple (66 images each), Skilled (6 images each). Since, we were training for binary classifiers, a set of 115 images collected from 115 different people's simple forgery (1 from each person) were set as random images i.e. set false during training.

4.2 Training

In training Case I, for each WD classifier, 9 genuine images and 9 forgeries generated using autoencoder were used for training along with 115 random images. A total of 133 images were trained for 20 epochs. 5 genuine images and 3 Opposite Hand forgeries were used as testing data.

In training Case II, 22 genuine images and 9 forgeries generated using autoencoder were used for training along with 115 random images, a total of 146 images were trained for 20 epochs. Similarly, 5 genuine and 3 Opposite Hand forgeries were used as testing data.

The images were trained using Adam (Adaptive momentum) [14] optimization with learning rate of 0.001. Binary Cross-entropy loss function was used since it is a binary classification. The training graph of 20 epochs is shown in figure 6.

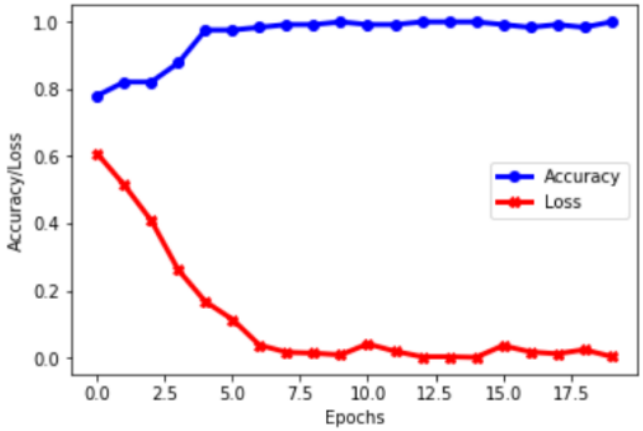


Figure 6: Training Graph

4.3 Results

For general context, signatures are accepted (considered genuine) if the output of the model is greater than 0.5 i.e. 50%. For specific use cases, the acceptance criterias could be set. As an instance, institutions requiring higher accuracy can set requirements for high results and vice versa. The result of 30 classifiers trained in case I and II are shown in table 1.

The False Acceptance Ratio (FAR) of the Signature is given by equation (1):

$$FAR = \frac{\text{Number of Forgery Signature Accepted}}{\text{Total Number of Forgery Images}} \quad (1)$$

For case I (9 Genuine Images), the average FAR of the system is 0.1555 i.e. 15.55%. For case II (22 Genuine Images), the average FAR of the system is 0.0777 i.e 7.78%.

The False Rejection Ratio (FRR) of the Signature is given by equation (2):

$$FRR = \frac{\text{Number of Genuine signature Rejected}}{\text{Total Number of Genuine Signature}} \quad (2)$$

Table 1: Accuracy, FAR, FRR

SN	Image	Case I (9 Images)			Case II (22 Images)		
		FAR	FRR	Acc	FAR	FRR	Acc
1.		0%	40%	79%	0%	20%	87%
2.		0%	20%	78%	33%	0%	87%
3.		0%	20%	86%	0%	0%	95%
4.		0%	0%	94%	0%	0%	99%
5.		0%	40%	72%	0%	40%	69%
6.		0%	20%	87%	0%	20%	87%
7.		0%	0%	98%	0%	0%	99%
8.		0%	60%	72%	0%	0%	94%
9.		0%	40%	67%	0%	0%	99%
10.		0%	20%	85%	0%	0%	99%
11.		66%	100%	24%	66%	20%	62%
12.		0%	60%	60%	66%	40%	53%
13.		0%	0%	85%	0%	20%	84%
14.		0%	60%	59%	0%	60%	63%
15.		66%	20%	59%	100%	0%	62%
16.		0%	20%	86%	66%	20%	71%
17.		0%	0%	99%	0%	0%	99%
18.		0%	0%	98%	0%	0%	99%
19.		0%	40%	74%	0%	40%	80%
20.		0%	20%	62%	0%	40%	77%
21.		0%	60%	62%	0%	0%	99%
22.		66%	20%	66%	66%	0%	65%
23.		33%	40%	66%	66%	20%	68%
24.		0%	80%	49%	0%	60%	66%
25.		0%	0%	87%	0%	0%	99%
26.		0%	20%	87%	0%	20%	87%
27.		0%	60%	61%	0%	20%	85%
28.		0%	0%	99%	0%	0%	99%
29.		0%	80%	54%	0%	60%	69%
30.		0%	20%	87%	0%	20%	92%

For case I (9 Genuine Images), the average FRR of the system is 0.32 i.e. 32%. For case II (22 Genuine Images), the average FRR of the system is 0.1733 i.e. 17.34%.

The Accuracy of the system is calculated using equation (3):

$$Accuracy = \frac{\sum_{i=1}^{N} \text{Genuine}_i + \sum_{j=1}^{M} 1 - \text{Forgery}_j}{\text{Total Number of Images}} \quad (3)$$

For case I (9 Genuine Images), the average Accuracy of the system is 74.13%. For case II (22 Genuine Images), the average accuracy of the system is 83.73%.

5. Conclusion

Humans also do make errors while verifying signatures. The proposed system has proven to demonstrate considerable results for signature verification in Persian signature dataset. The accuracy of the system upgraded by 9.6% when 22 genuine images were passed instead of 9. There was a noticeable decrease of 7.78% in False Acceptance Ratio when number of genuine images were increased while training the model. Similarly, there was a significant reduction of 14.66% in False Rejection Ratio on increasing genuine image. Thus the model was getting more versatile and less overfit in increasing the number of genuine images while training. We support Rezai et al.'s studies, as our system is primarily based on their architecture. Our proposed method added image processing for removing any distorted noises present in the input images. Further, our proposed system implements Autoencoder to generate forged images using the genuine images. This method supports real life situations when the user does not provide their forged images when training. Due to optimized features, our proposed system was slightly better than theirs with an increase in average accuracy and decrease in FAR and FRR ratios.

Despite new features, our proposed system still lags many techniques that can optimize our system e.g: data augmentations, regularizations, residual network, etc. Besides, due to computational constraints, our input sizes were limited and this caused the CNN to over fit on smaller datasets, leading to FAR issues.

References

[1] S. Garhawal, N. Shukla et al., "A study on handwritten signature verification approaches," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 8, pp. 2497—2503, 2013.

[2] K. Manjunatha, M. Shantharamu, D. Guru, and M. T. Somashekara, "Online Signature Verification based on Writer Dependent Features and Classifiers," *Pattern Recognition Letters*, 2016.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient -based learning applied to document recognition," in *Proc. IEEE* vol. 86, no. 11, pp. 2278—2324, 1998.

[4] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163–176, 2017.

[5] M. Kumar, "Signature Verification Using Neural Network," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 4, no. 09, Sep 2012.

[6] L. Batista, E. Granger, and R. Sabourin, "Dynamic selection of generative–discriminative ensembles for off-line signature verification," *Pattern Recognition*, vol. 45, no. 4, pp. 1326—1340, 2012.

[7] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Persian signature verification using convolutional neural networks," *International Journal of Engineering Research and Technology*, vol. 1, no. 2, pp. 7–12, 2016.

[8] M. B. A. Miah, M. A. Yousuf, M. S. Mia, M. P. Miya, "Handwritten Courtesy Amount and Signature Recognition on BankCheque using Neural Network," *International Journal of Computer Applications*, vol. 118, no. 5, 2015.

[9] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognition*, vol. 48, no. 1, pp. 103–113, 2015.

[10] B. Cozzens, R. Huang, M. Jay, K. Khembunjong, S. Paliskara, F. Zhan, M. Zhang, and S. Tayeb, "Signature Verification Using a Convolutional Neural Network," *BigData*, pp. 2644–2651, 2017.

[11] M. Rezaei and N. Naderi, "Persian Signature Verification using Fully Convolutional Networks", 2019.

[12] J. Darbon et al., "Fast nonlocal filtering applied to electron cryomicroscopy," in *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Proceedings, ISBI, IEEE*, pp. 1331–1334, 2008.

[13] A. S. Bajestani, K. Fouladi, Kazim, and B. Araabi, "UTSig: A Persian offline signature dataset," *Preprint, submitted to IET biometrics*, 2016.

[14] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization." University of Toronto, 2015.