Handwritten Signature Recognition : A Convolutional Neural Network Approach

Krishnaditya Kancharla
Dept. Of Electronics and Telecommunication Engineering
Sardar Patel Institute of Technology
Mumbai, India 400058
krishnaditya.kancharla@spit.ac.in

Varun Kamble
Dept. Of Electronics and Telecommunication Engineering
Sardar Patel Institute of Technology
Mumbai, India 400058
varun.kamble@spit.ac.in

Mohit Kapoor
Dept. Of Electronics and Telecommunication Engineering
Sardar Patel Institute of Technology
Mumbai, India 400058
mohit.kapoor@spit.ac.in

Abstract—**Handwritten Signature Recognition is an important behavioral biometric which is used for numerous identification and authentication applications. There are two fundamental methods of signature recognition, on-line or off-line. On-line recognition is a dynamic form, which uses parameters like writing pace, change in stylus direction and number of pen ups and pen downs during the writing of the signature. Off-line signature recognition is a static form where a signature is handled as an image and the author of the signature is predicted based on the features of the signature. The current method of Off-line Signature Recognition predominantly employs template matching, where a test image is compared with multiple specimen images to speculate the author of the signature. This takes up a lot of memory and has a higher time complexity. This paper proposes a method of off-line signature recognition using Convolution Neural Network. The purpose of this paper is to obtain high accuracy multi-class classification with a few training signature samples. Images are preprocessed to isolate the signature pixels from the background/noise pixels using a series of Image processing techniques. Initially, the system is trained with 27 genuine signatures of 10 different authors each. A Convolution Neural Network is used to predict a test signature belongs to which of the 10 given authors. Different public datasets are used to demonstrate effectiveness of the proposed solution.**

I. INTRODUCTION

Signature recognition is fundamental to forensic analysis of documents. Signatures are a major tool for document verification. Every individual has a unique way of doing his/her signature and this signature, on forgery, loses its key features. Thus, signature verification becomes a very important security aspect. Handwritten signature recognition can be operated in two ways:

Online Recognition : In this method, signature is acquired from the user on a digital device in real time, usually, operated by a stylus. Dynamic information about the signature, like, spatial coordinates, pressure, azimuth, inclination and number of pen-up/down are recorded and analyzed to predict the author of the signature. This is also known as Dynamic Signature Recognition.

Off-line Recognition : In this method, signature is written on paper, digitized through a scanner or a camera and the biometric system recognizes the signature based on unique features of the signatures. This is also known as Static Signature Recognition.

These methods are widely used for authentication purposes in the following sectors:

Finance : Financial Institutions store specimen signatures of their customers to verify the authenticity of the documents submitted to them by comparing the signature on the document with the specimen signatures.

Health : The "National Health Service" , United Kingdom has started capturing signatures of doctors who fill patient records for authentication purpose.

Real Estate: In the USA, there are options of paperless contracting by signing on Tablet PCs or PDA with help of a stylus [1].

II. RELATED WORK

- In 2009, an off-line signature identification and verification system based on image registration and Discrete Wavelet Transform (for Persian signatures) was proposed by Ghandali and Moghaddam. It is a language dependent method which uses DWT for feature extraction and Euclidean distance for comparing the extracted features [2].
- General methods use feature extraction including geometric, graph metric, directional, wavelet, shadow, and texture features.
- Ferrier et al. calculates geometric features of a signature in fixed-point arithmetic for offline verification. The proposed features are then checked with different classifiers, such as Hidden Markov Models, Support Vector Machines, etc [3].
- Two methods were proposed by Fang et. al. for detection of signature forgeries. These were based on template matching. The first one was based on optimal matching of the one dimensional projection profiles of the signature patterns and the other was based on elastic matching of the strokes in the two dimensional signature patterns [4].

978-1-5386-5367-8/18/$31.00 ©2018 IEEE

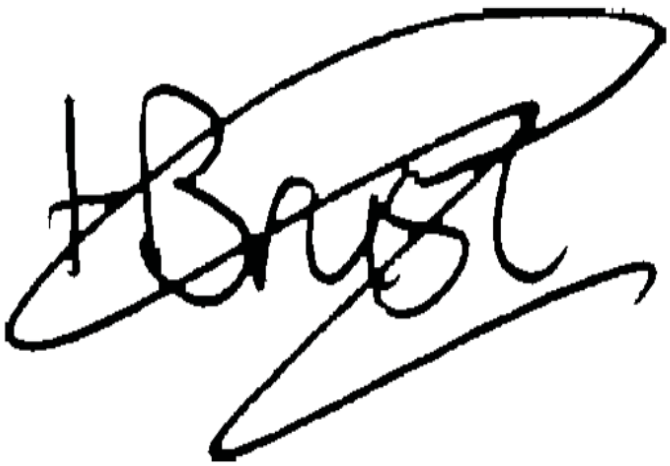

Fig. 2. Pre-Processed Image

- A fuzzy approach was proposed by M. Nassir and A. Javaheri for verification using Takagi-Sugeno Fuzzy Inference System (FIS) for off-line signatures [5].

III. PREPROCESSING

The images present in the dataset have inherent noise present in the form of small, dark clusters of pixels randomly distributed around the signature. This noise may affect our results by generating errors when the neural network is identifying unique features of a signature class. Hence, to ensure efficient feature extraction, the image first undergoes a series of preprocessing algorithms for noise removal:

A. Grayscale Conversion & Finalization

All images are converted to grayscale format. Furthermore, to emphasize only on the signature pixels and suppress any noise which may have been introduced during the data acquisition, the images are converted to binary images. A threshold is chosen such that pixels above the threshold are the signature pixels, having pixel value as 255. Accordingly, all pixels below the threshold are background pixels, having pixel value as 0.

B. Morphological Transformation

- Erosion: Since, our background is made of white pixels and the foreground object (signature) is made of darker pixels, the thickness of the foreground object increases when erosion is applied and small white noises are removed.
- Dilation: This is the reverse process of erosion and helps in thinning of foreground object (signature) and completely removes any dark dots or specs present in the background. Thus the image now contains only signature pixels with a white background [6].

C. Image Resizing

The images in the dataset are of varying width and height which would not be suitable for training the CNN. For the CNN to learn which features need to be extracted, all images must have a uniform size and aspect ratio. So, the images are resized to have dimensions of width 160 pixels and height 120 pixels.

Fig. 1. Original Image

IV. DATASET

Two datasets are used. Both datasets contain genuine as well as forged signatures of various authors. For signature recognition purpose, only the genuine signature samples would be utilized for training and validation of the model.

SigComp 2011 dataset [7]: This dataset was presented for the ICDAR 2011 Signature Verification Competition (SigComp2011). It consists of Dutch and Chinese signature samples. The images are in Portable Network Graphics (PNG) format scanned at 400 dpi having RGB color channels. The Dutch and Chinese training sets both contain about 24 genuine signature samples each from 10 reference writers.

UTsig Persian dataset [8]: This is a public Persian off-line signature dataset that consists of 8280 signature images from 115 classes. Each author has 27 genuine signatures, 3 opposite-hand signatures, and 42 skilled forgeries made by 6 forgers. All images are stored in TIFF format but have non-uniform width and height. For uniformity, 24 genuine signatures each from 10 authors are used from this dataset.

In sum, the combined dataset for training and model tuning contains 720 signatures of 30 classes. The model is trained on the Dutch, Chinese and Persian signatures separately and independently to validate its versatility.

V. CNN AND ITS COMPONENTS

A Convolutional Neural Network (CNN) model or Conv-Net for short has become the gold standard for image recognition algorithms [9]. A CNN model applies convolution operations on the image which are accompanied by some pooling operations to generate features which are fed into the fully connected layers.

A. Convolutional Layer

The Convolutional layer is a fundamental part of any Convolutional Neural Network and does the majority of the computational operations.

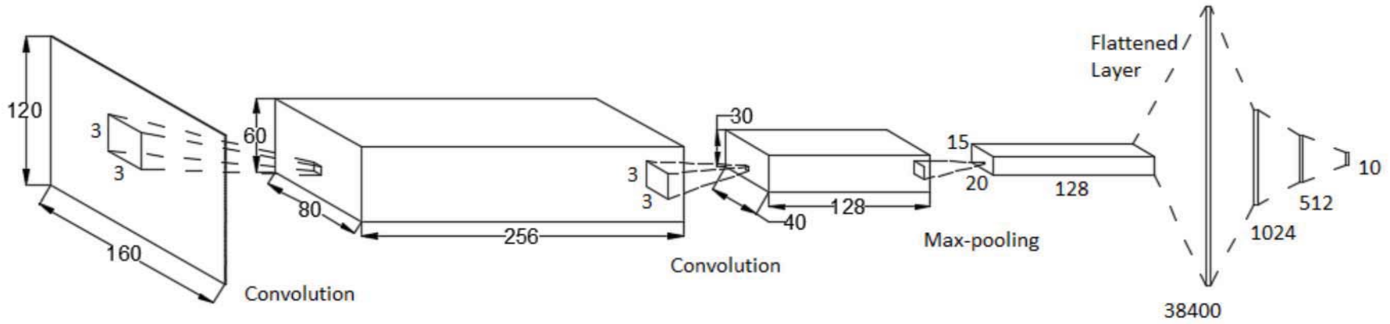

Fig. 3. CNN Architecture

The convolution operation works as a scanning process with defined filter dimensions. Here, we also set the stride for the filter which is the amount of pixels it moves after completing a scan of a part of an image. When it reaches the end of the image on the boundaries we can define what should be done if the filter is hanging from the image. There are 2 ways to handle this, either skip some values at the boundaries which do not fit the filter dimensions or pad zeros after the image boundary to make up the space in the filter. The first way is called 'valid' padding and second way is called 'same' padding. Hence, the width and height of the image does change for 'same' padding and reduces little for 'valid' padding.

B. Pooling Layer

The pooling layer is an important block of a convolutional model whose function is to progressively reduce the spatial size of model. This reduces the amount of trainable parameters and thereby reduces the computation time. It is usually added after a convolutional layer. There are different types of pooling methods.

The most widely used is the max-pooling method. In Max-pooling method, the image is partitioned into smaller rectangles and only the maximum value from each rectangle is returned as output [10]. The pooling operation effectively eliminates values, hence there is a loss of information as well. This is called 'down-sampling'. But there are some benefits to this, it works against over-fitting and the decrease in size results in a reduced computational load for the succeeding layers of the network.

C. Fully Connected Layer

The Fully Connected layer consists of densely packed neurons which are connected to all the activations in the preceding layer. It is also sometimes referred to as the dense layer. The output of the preceding layer is fed to a fully connected layer which generates an N dimensional output vector, where N represents the number of neurons in the particular fully connected layer.

D. Softmax Layer

The final layer which produces the output of the neural network is also a fully connected layer but with a special activation function called softmax activation function [10]. The main goal of this function is to squish the

output neuron's value between the values of 0 to 1. Also, the sum total of the output value equals to one. Hence, we get the output in terms of probabilities.

$$P(y=i|x,W_1,...,W_N,b_1,...,b_N)=\frac{e^{W_ix+b_i}}{\sum_{j=1}^{N}e^{W_jx+b_j}} \quad (1)$$

Here, x is the output of the softmax layer, W_i and b_i are the weights and biases of i^{th} neuron of this layer, and N is the number of neurons. The neuron which has the maximum probabilistic value is taken as the output [11]. Equation below calculates this neuron position.

$$\widehat{y}=arg\max_i P(y=i|x,W_1,...,W_N,b_1,...,b_N) \quad (2)$$

Here, y denotes the predicted class.
Conventional sigmoid activation functions have the problem of gradient vanishing due to their mechanism of decreasing the gradient as the input to the sigmoid function increases. Hence, for minimizing the likelihood of gradient vanishing, ReLUs [12] are used in their place.

E. Training

The network was separately trained using two adaptive learning rate methods of Adam [13], which stands for Adaptive Moment Estimation that computes adaptive learning rates for each parameter \& RMSprop for 100 epochs each with a starting learning rate of $1x10^{-4}$, $1x10^{-4}$ & $5x10^{-5}$ for Chinese, Dutch & Persian datasets respectively. A uniform batch size of 32 was selected for all datasets

978-1-5386-5367-8/18/$31.00 ©2018 IEEE

TABLE I
RESULT TABLE

Signature Type	Optimizer	Avg. Classwise Precision	Avg. Classwise Recall	Avg. Classwise F-measure	Accuracy (%)
Chinese	Adam	0.905	0.8217	0.8328	**85.11**
Chinese	RMSprop	0.8799	0.7917	0.8146	82.98
Dutch	Adam	1.0	1.0	1.0	**100**
Dutch	RMSprop	0.9833	0.9667	0.9709	97.91
Persian	Adam	0.9722	0.9764	0.9725	**96.29**
Persian	RMSprop	0.9722	0.9764	0.9725	**96.29**

VI. RESULTS

A. *Evaluation Metrics*

The dataset was split into training set and a validation set containing 80\% and 20\% of the data respectively. Accuracy of the model has been reported on the validation sets of the Dutch, Chinese and Persian signatures. The performance of the model is evaluated using the following metrics [14]:

1) Precision (P) :
This gives a measure of the proportion of positive identifications that were actually correct for each of the 10 classes.

2) $$P = \frac{True\_Positives}{True\_Positives + False\_Positives} \quad (3)$$

call (R) :
This is used for measuring the fraction of positive identifications that are correctly classified for each of the 10 classes.

3) $$R = \frac{True\_Positives}{True\_Positives + True\_Negatives} \quad (4)$$

easure (FM) :

This is the harmonic mean between the Precision and Recall values.
$$FM = \frac{2 \times P \times R}{P + R} \quad (5)$$

B. *Learning Curve and Loss Function*

The loss function and accuracy graph of the final model for the 3 different datasets is shown in figures 4, 5 & 6.

From Fig. 4, it can be observed that when the model was trained on the Chinese dataset using Adam, it achieved its optimum value of validation performance after 38 epochs. The same model when trained using RMSprop, required around 70 epochs to reach its maximum validation performance.

From Fig. 5, it is evident that RMSprop required fewer epochs to achieve its peak validation performance on the Dutch signatures. However, it was outperformed by Adam after 80 epochs. The flat response of Adam between 20 and 60 epochs can be attributed to the convergence of the Gradient Descent Algorithm to a local minimum. From Fig. 6, it can be inferred that both models reached the same level of validation performance for the Persian dataset. RMSprop, however, required fewer epochs to attain optimum performance than Adam.

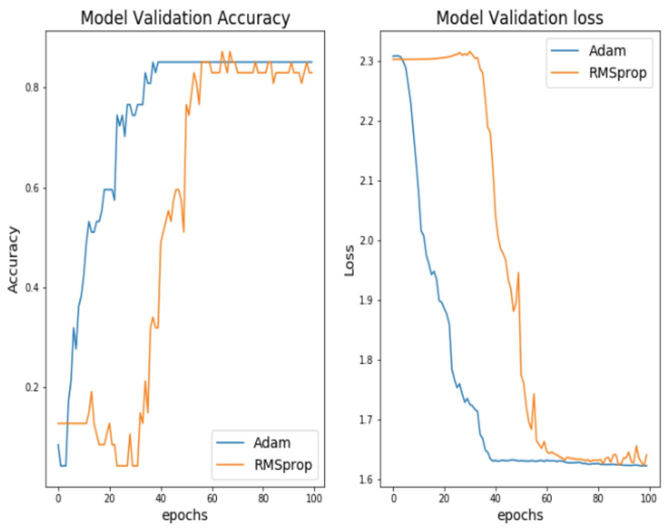

Fig. 4. Learning & Curve for Chinese Signatures

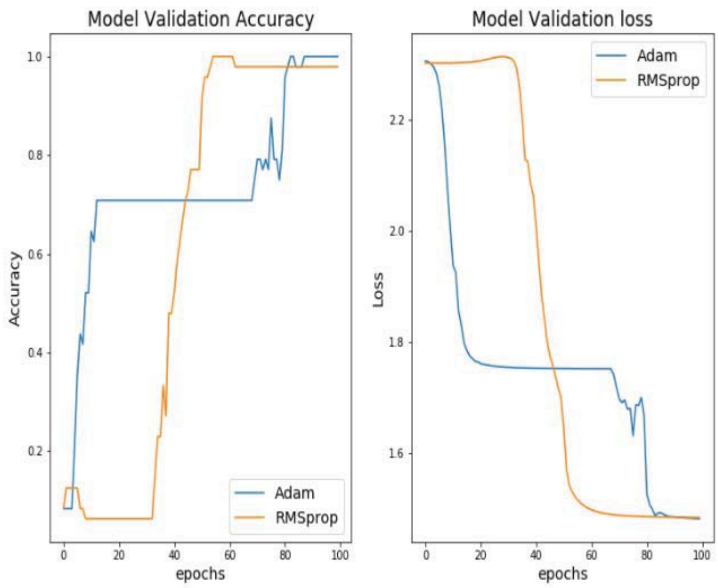

Fig. 5. Learning & Curve for Dutch Signatures

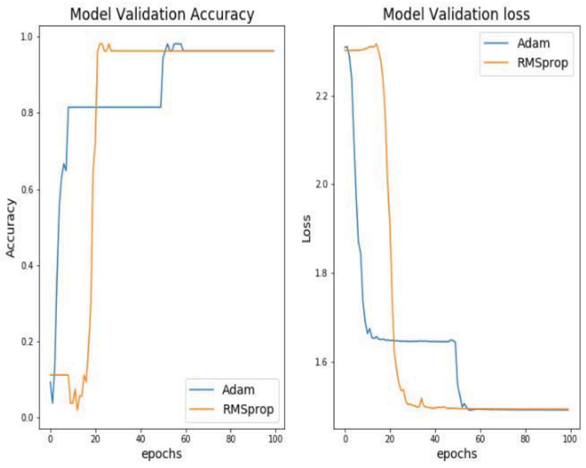

Fig. 6. Learning & Loss Curve for Persian Signatures

VII. CONCLUSION AND FUTURE SCOPE

We have proposed a convolutional neural network architecture for off-line handwritten signature recognition. We experimented the model over 3 different datasets using 2 optimizers and evaluated its performance. It can be concluded that training the model using Adam enhanced its versatility because it provided better validation accuracy over all 3 types of signatures.

Future work may include using the proposed model along with pre-trained models like Inception-V3 for signature verification where the model would classify whether a signature is genuine or forged, with a high accuracy using few training samples.

REFERENCES

[1] Zaidi, Syed Faraz Ali, and Shahzaan Mohammed, "Biometric Handwritten Signature Recognition".TDDD17: Information Security CourseLinköpings universitet, Sweden.

[2] S. Ghandali and M. Ebrahimi Moghaddam, Off-Line Persian Signature Identification and Verification based on Image Registration and Fusion, In: Journal of Multimedia, volume 4, 2009, pages: 137-144.

[3] Miguel A.Ferrer, Jesu's B. Alonso, and Carlos M. Travieso, Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic IEEE transactions On Pattern Analysis And Machine Intelligence, vol. 27, No. 6, June 2005.

[4] B. Fang, C.H Leung, Y.Y. Tang, K.W. Tse, P.C.K. Kwok, and Y.K.Wong, Off-line signature verification by the tracking of feature and stroke positions, Pattern Recognition, vol. 36, pp. 91-101, 2003.

[5] M. Nasiri and A. Javaheri, "A Fuzzy Approach for the Automatic Off-Line Persian Signature Verification Problem," 2011 7th Iranian Conference on Machine Vision and Image Processing, Tehran, 2011, pp. 1-5.

[6] N. Jawas and N. Suciati, "Image Inpainting using Erosion and Dilation Operation", International Journal of Advanced Science and Technology, vol. 51, pp. 127-134, 2013.

[7] Liwicki, M., Malik, M., van den Heuvel, C., Chen, X.,Berger, C., Stoel, R., Blumenstein, M., Found, B.: Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). in 2011 International Conference on Document Analysis and Recognition.

[8] A. Soleimani, K. Fouladi and B. N. Araabi, "UTSig: A Persian offline signature dataset," in IET Biometrics, vol. 6, no. 1, pp. 1-8, 1 2017.

[9] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1097–1105,2012.

[10] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6.

[11] Duan K., Keerthi S.S., Chu W., Shevade S.K., Poo A.N. (2003) Multi-category Classification by Soft-Max Combination of Binary Classifiers. In: Windeatt T., Roli F. (eds) Multiple Classifier Systems. MCS 2003. Lecture Notes in Computer Science, vol 2709. Springer, Berlin, Heidelberg.

[12] Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network. ICML Deep Learning Workshop , Lille, France, 06-11 July 2015.

[13] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization", arXiv:1412.6980v9 [cs.LG], Jan. 2017

[14] Hossin, Mohammad & M.N, Sulaiman (2015). A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process. 5. 01-11.10.5121/ijdkp.2015.5201.

[15]