Offline Handwritten Signature Recognition Using Deep Convolution Neural Network

Article *in* European Journal of Engineering and Technology Research · August 2022

DOI: 10.24018/ejeng.2022.7.4.2851

2 authors:

Quazi Saad
Islamic University of Technology

Mousumi hasan Mukti
Bangladesh Army International University of Science and Technology

Offline Handwritten Signature Recognition Using Deep Convolution Neural Network

Quazi Saad-ul Mosaher and Mousumi Hasan

Abstract — **In the modern age, technological advancement reached a new limit where authentication plays a vital role in security management. Biometric-based authentication is the most referenced procedure for authentication where signature verification is a significant part of it for authentication of a person. To prevent the falsification of signatures on important documents & legal transactions it is necessary to recognize a person's signature accurately. This paper focused on recognizing offline handwritten original & forged signatures using a deep convolution neural network. We use a completely new dataset & also downloaded datasets to train the system & verify a random signature as genuine or forgery. All testing samples are collected from several individuals after several steps of preprocessing the model is fed with the resultant image to our system, the experimental results give us an accuracy of 95.5% from the dataset.**

Keywords — **Convolution Neural Network (CNN), Forged signatures, Deep-Learning, Handwritten signatures.**

I. Introduction

Biological characteristics differ from person to person which allows for verifying the identity of individuals. Biometric authentication is a process by which individuals can be identified through their unique characteristics. Among the several biometric tasks, one of the age-old processes is signature verification. A signature is a distinctive handwritten depiction of an individual which could be an individual's name, nickname, symbols, etc. to identify the specific person's consent on the document, letter, or financial transactions. Signer sign signatures have been a significant innovation for identifying individuals for thousands of years. Generally, biometric systems are of two sections:

– Verification
– Identification

Verification & identification of individuals is two different things where verification verifies the authenticity of a person whether the biometric is really of that individual or not whereas identification recognizes the individual's biometric from the available batch. This paper reflects the verification of a person through means of signatures. The verification process includes two labels for signatures which are original and forge.

There are three forms of forgeries:

– Random forgery
– Simple forgery
– Skilled forgery

Random forgeries are the ones where the forgery signer has no idea or information about the user or the signature. In the case of simple forgeries signer has a vague idea about the name but is fully unaware of the signature style of the user. Skilled forgeries are possible when the signer has information and also the signature sample of the user which is to be forged. This signature is very much similar to the original one so it's tough to distinguish whether it is original or forged.

Signature verification is categorized into two forms: Dynamic (online) & Static (offline). Several parameters are taken into account while verification in these two different forms. Parameters for dynamic verification are pretty advanced as if locations, Pressure, acceleration, and time for the signature of the signer all are recorded.

But for static verification only starts after the signature is completed in the paper, calculating the edges, vertices, and shape of the signatures having maximum similarity with the original user is concluded as original & dissimilarity results as the forged signature.

Signature verification plays a significant role in different sectors which are as follows:

– Commercial
– Legal documents
– Financial transactions
– Acknowledgment & Acceptance

This paper focuses on static signature verification using a deep convolution neural network in pattern recognition.

II. Related Works

In reference papers [1]–[7] various writer-dependent methodologies are studied which are based on deep learning. Convolution neural network shows a tremendous impact in the field of imagery. Image preprocessing is common in the very article as all the models deal with images of signatures.

CNN was implemented based on VGG16 architecture which is a CNN architecture containing 16 layers along with learning features [2]. The model was experimented with by two types of data set: unseen identifier and writer dependent. In unseen identifiers, the network was tested by the signature of persons who were not in the training set. Writer-dependent data set is the opposite of it that is pair of (genuine, genuine), (genuine, forgery) was previously set in the data set. Inspired by Google Net an architecture called Inception SVG Net framework is used to implement CNN in [3].

In [4] Explainable Deep Learning Method was used to implement DCNN. Verification of a signature can be tested

Submitted on June 27, 2022.
Published on August 29, 2022.
Q. S. Mosaher, Islamic University of Technology, Bangladesh.
(e-mail: saad.quazi44@gmail.com)

M. Hasan, Bangladesh Army International University of Science & Technology, Bangladesh.
(e-mail: mousumi1612@gmail.com)

in this article whether it is original or forgery. Every time the system was tested with a new sample of signatures that were not present in the training time of the model. A unique method was introduced for signature verification and forgery detection using CNN and Harris corner detection algorithms. CNN was used for signature verification and the Harris surf-based model was used for forgery detection of any signature [5].

CNN was also implemented on two levels in the file management system: genuine and forgery. A GUI was used to test whether the signature in question is genuine or not [6].

A method was experienced to extract features from a signature image using DCNN, Deep Convolutional Neural Network [7]. A classifier architecture was formed to graphically analyze the error rate of the model and the error rate in feature extraction of the algorithm.

III. Proposed Methodology

This paper focuses on extracting features from the training dataset to train the model and evaluate the test dataset based on its learning from the features. A convolution Neural Network is used for extracting features and training the model. Convolution Neural Network proves the fact that original & forged signatures differ at some point.

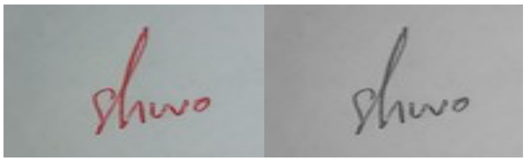
Here two classes are introduced which are original & forged signatures which can also be acknowledged as two class labels. CNN plays a significant role in extracting necessary features that differ between the two class labels. A person who is used to signing signatures, while signing the nerve impulse is controlled by the brain without paying any attention to the detail. On the other hand, the person forging another person's signature pays attention to every detail of the original signature which differs nerve impulses resulting in some trembling in the signature. This difference leads to a change in features of both the signatures which are identified and extracted by CNN.

A. Pre-Processing

Data preprocessing facilitates the training & testing process by transforming the data into a more understandable form of an image. Pre-processing removes the outliers and scales the features into a viable range. Several steps are followed to complete the preprocessing so that the model gets a clearer, enhanced, noise-free training dataset to learn for better accuracy in the verification of the signature.

B. Grayscale

Signature verification does not depend on the color of the pixels rather it focuses on the luminance of the pixels of an image. Grayscale is a monochromatic shade from black to white. Removes all the information about the color. The luminance of the pixel remains as it is. OpenCV python library is used to solve computer vision problems following paper used this library to read/load an image and to write an image to a given destination. Moreover, there are 150 color space conversion methods available in OpenCV. In Fig. 1, two images were taken, among them, the colored image is the original image which was converted into a grayscale image. Both the images are labeled as Fig. 1a and 1b.

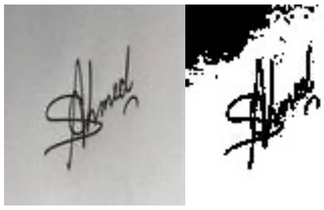


a. b.

Fig. 1. a. Original Image and b. Grayscale Image

C. Binary Images

Images can have two possible intensity values which are either 0 for black and either 1 or 255 for white are named binary images. In Fig. 2, a gray-scaled image has been transformed into a binary image. Binary images allow the separation of an object from its background. The thresholding technique aims to extract the foreground of an image from its background & grayscale images are taken as input for binary thresholding to obtain binary images for desired output.

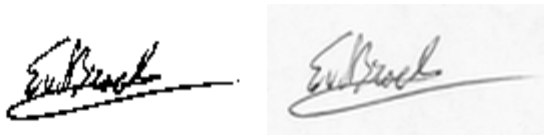


a. b.

Fig. 2. a. Grayscale Image and b. Binary Image

D. Resized Images

The desired output of the binary conversion images is fed as input to resize the images. In Fig. 3, the image labeled as Fig. 3a is the binary image that is to be resized. The expected size for the resized images is 140*110 pixels. Resized images aim to aid in better learning for CNN. The entire dataset is resized into the given pixel size.

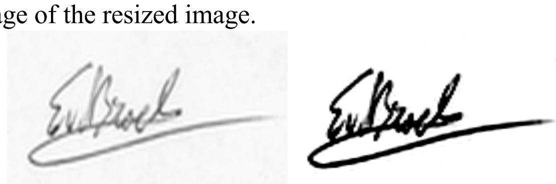


a. b.

Fig. 3. a. Binary Image and b. Resized Image

E. Image Enhancement

Finally, after completing all the preprocessing steps in a stepwise manner all the resized images in the dataset are enhanced. Several factors assist the image enhancement process among them contrast plays a vital role in deepening the object. By availing of the image enhancement technique, every contrast of the images in the dataset is enhanced by a factor of 5.0. Images in the dataset are saved in the PNG format in the destination folder. Fig. 4b depicts the enhanced image of the resized image.



a. b.

Fig. 4. a. Resized Image and b. Enhanced Image

F. File Management

Every Preprocessing step results in a different directory folder with the name associated with the preprocessing step. Such as the directory folder of this work named Grayscale-images is the output of grayscale images from original images in the following way binary, resized and enhanced image outputs are saved in a distinct folder with the name associated with the pre-processing steps.

a. b.
Fig. 5. a. Original Image and b. Pre-processed Image

Following all the pre-processing steps a final preprocessed image is obtained which is fed to the CNN model as an original image so that it can learn the features from it. Different images have been taken as input in this paper in different preprocessing steps to give a proper indication of the system that works on them. In Fig. 5a, the original image is shown which has gone through all the preprocessing steps and finally gave a clear view of the preprocessed image labeled as 5b.
In the same procedure forged signatures are pre-processed & kept in a distinct folder. Both the preprocessed original and forged images are fed into the CNN model.

IV. Experimental Protocol

Signatures are stored in a specific directory that can be accessed by the Keras Python library. CNN is being implemented using a python programming language with Keras & TensorFlow backend so that the model can learn other patterns associated with the images and verify how the model fits the data loss metrics and accuracy used. Finally, the model is being tested accordingly.

A. Dataset

The Dataset which is being used in this research work is a collection of 130 images from 130 individuals. The training dataset consists of two folders namely original and forged & the test dataset also consists of the same two original and forged datasets. Later on, we specifically provide an original or forged signature selected from those folders and fed to the model. The model will extract the features of the given image and will compare the features to learned features and finally will conclude the result in a two-class label which is either original or forged.

B. Experimental Setup

Convolution Neural Network requires a specific set of computational setups for their working procedure. Due to its different layers with a complex calculation based on the equation, it requires a minimum configuration of hardware and specific software to run. The experiment carried out in this paper is run on a device that satisfies the environment and hardware specifications for the program to run which took a significant amount of time to complete the steps and epochs showing the desired model output for loss and accuracy in each epoch also the summary of the model.

Following computational setups are required for the experiment to run:

a. Laptop Configuration: Dell Vostro 14 3000, Intel Core i5 10th Gen, Ram: 8GB, HDD: 1Terabyte, SSD: 128GB, Clock speed: 2.11GHz.
b. Operating System: Windows 10 64-Bit.
c. Software Package: Anaconda3, Tensorflow, Keras, Numpy, OpenCV, Glob.

TABLE I: Summary of the Model

Type of Layer	Shape of Output	No. of Parameters
Conv2d_1(Conv2D)	(None,98,98,32)	896
Activation_1(Activation)	(None,98,98,32)	0
Max_pooling_2d_1(Max-pooling 2D)	(None,49,49,32)	0
Conv2d_2(Conv2D)	(None,47,47,32)	9248
Activation_2(Activation)	(None,47,47,32)	0
Max_pooling_2d_2(Max-pooling 2D)	(None,23,23,32)	0
Conv2d_3(Conv2D)	(None,21,21,64)	18496
Activation_3(Activation)	(None,21,21,64)	0
Max_pooling_2d_3(Max-pooling 2D)	(None,10,10,64)	0
Flatten_1 (Flatten)	(None, 6400)	0
Dense_1 (Dense)	(None, 64)	409664
Activation_4 (Activation)	(None, 64)	0
Dropout_1 (Dropout)	(None, 64)	0
Dense_2 (Dense)	(None, 1)	65
Activation_5 (Activation)	(None, 1)	0
Total Parameters	-	438369
Trainable parameters	-	438369

C. Experiments

Experiments were performed in two possible means to test whether the model gives proper output or whether the model is fit to the dataset. So first a set of 20 images is taken in both the training and validation set, where the batch size is 4 and the number of epochs is 5. This experiment gives an outstanding result obtaining 100% accuracy from the model. Later on, the model is tested for 130 samples of signatures where the batch size was 10 and the number of epochs was 13. From the later experiment, we obtained a maximum accuracy of 95.5% in epoch 10 so we later kept the number of epochs equal to 10 for a better result.

TABLE II: List of Formulas for the Operation of CNN

Operation	Formula
Convolution	$Z^1 = H^{l-1} \times W^1$
Max pooling	$h_{xy}^1 = max_{i=0..s, j=0..s} h^{l-1}(x+i)(y+j)$
Fully-Connected Layer	$Z_l = W_l \times h_{l-1}$
ReLu(Rectifier)	$Relu(Z_i) = max(0, Z_i)$
Softmax	$Softmax(Z_i) = e^{zi} / \sum_j e \hat{} zj$

Among the list of formulas, we used the first two formulas in our model for the convolution layer and max-pooling layer to calculate the values of the output shape.

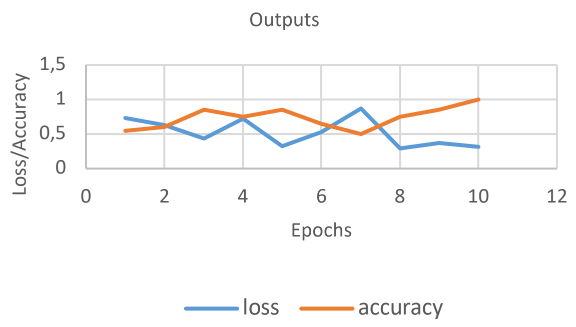


Fig. 6. Model output for loss and accuracy in each epoch.

In Fig. 6, the line graph depicts the accuracy and loss of the CNN model created based on every consecutive epoch. The graph shows a brief idea of where maximum accuracy and minimum loss are obtained in epoch 10. Since epoch 10 lead to maximum accuracy so further training of the model and testing might occur an overfitting problem.

V. Conclusion and Future Work

Throughout several years much research work has been performed on signature verification which led to less forgery and identifying fraudulent documents. Convolution Neural Network proved to be the most efficient model for the verification of signatures along with patterns associated with it. In this research work, CNN successfully classified given input images as either original or forged which were the two class labels for the outcome. Thus the accuracy of 100% is obtained from the model testing with 20 images and 130 images giving an accuracy of 95.5%.

Pattern matching will proceed to further better results based on feature sets, parameters, and classifiers using a Deep neural network and can give better outcomes holding the accuracy rate from 98% to 100% for any number of samples of any datasets.

Acknowledgment

Thanks to all the people who helped us to make the dataset used for this research with their signatures.

References

[1] Poddar J, Parikh V, Varti SK. Offline Signature Recognition & Forgery Detection using Deep Learning. *The 3rd International Conference on Emerging Data and Industry 4.0* (EDI40), Warsaw, Poland, April 6 - 9, 2020.

[2] Hazra TK, Sarkar R, Kumar Handwritten English Character Recognition Using Logistic Regression and Neural Network. *International Journal of Science and Research (IJSR)*, 2016; *5*(6): 750–754. https://doi.org/10.21275/v5i6.nov164228.

[3] Gideon J, Kandulna A, Kujur AA, Diana A, Raimond K. Handwritten Signature Forgery Detection using Convolution Neural Networks. *8th International Conference on Advances in Computing and Communication (ICACC-2018)*.

[4] Alginahi Y. Preprocessing Techniques in Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTech, Available from: http://www.intechopen.com/books/characterrecognition/preprocessing-techniques-in-character-recognition

[5] Alajrami E, Ashqar BAM, Abu-Nasser BS, Khalil AJ, Musleh MM, Barhoom AM, Abu-Naser SS. Handwritten Signature Verification using Deep Learning. *International Journal of Academic Multidisciplinary Research (IJAMR),* 2019; 3(12): 39-44

[6] Mohapatra RK, Shaswat K, Kedia S. Offline Handwritten Signature Verification using CNN inspired by Inception V1 Architecture. *IEEE Conference on Computer Vision & Pattern Recognition*.

[7] Kao HH, Wen CY. An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach. *Applied Sciences*, 2020; 10(11): NA. https://link.gale.com/apps/doc/A638408557/AONE?u=anon~bb7c1f31&sid=googleScholar&xid=afeacccd

[8] Hafemann LG, Sabourin R, Oliviera LS. Analyzing features learned for Offline Signature Verification using Deep CNNs. *Conference Paper for ICPR 2016*.

[9] Alvarez G, Sheffer B, Bryant M. Offline Signature Verification with Convolution Neural Network.

[10] Upadhye GD, Shirsat SM, Shinde SR, Sonawane MA, Pandit KD. Pattern Recognition using Convolution Neural Network for Handwritten Gujarati Numerals. *International Journal of Recent Technology and Engineering (IJRTE).* 2020; 8(5). ISSN: 2277-3878.

Quazi Saad-ul Mosaher was born and raised in Khulna, near the largest mangrove forest in Bangladesh. He completed his under-graduation program at Bangladesh Army International University of Science & Technology in the field of Computer Science & Engineering in the year 2019. He is been studying at the Islamic University of Technology to complete a post-graduation Master's program in Computer Science & Engineering. His major field of interest is image processing and pattern recognition.

He was working as a senior executive in the department of IT in an Insurance Company later on he shifted to an agency in their MIS department. At present, he is a full-time postgraduate Master's program student. His previous publication was "Bengali Longhand Character Recognition using Fourier Transform and Euclidean Distance Metric" European Journal of Engineering Research and Science 3(7):67-72, DOI:10.24018/ejers.2018.3.7.831. His previous research is on image processing and currently working on both image processing and pattern recognition.

Mousumi Hasan was born in Tangail, Bangladesh. She completed her graduation and post-graduation from Jahangirnagar University, Savar, Dhaka. Now she is working as Assistant Professor at Bangladesh Army International University of Science and Technology, Cumilla Cantonment, Cumilla, Bangladesh.