An Assignment Report on:

“Handwritten signature verification using shallow Convolutional Neural Network”

Prepared by : **Amit Kumar**

Roll. No. : **P19CO013**

Class : **M. Tech. I yr. (Computer Engineering) 2nd Semester**

Year : **2019-20**

Department of Computer Engineering

Sardar Vallabhbhai National Institute of Technology,

Surat -395007 (Gujarat), India

Sardar Vallabhbhai National Institute of Technology,

Surat -395007 (Gujarat), India

CERTIFICATE

This is to certify that the Assignment report entitled "**Handwritten signature verification using shallow Convolutional Neural Network**" is prepared and presented by **Mr. Amit Kumar** bearing Roll No.: **P19CO013**, 1st Year of **M. Tech. (Computer Engineering)** and his work is satisfactory.

GUIDE **HOD**

COED

Table of Contents

List of tables

List of figures

Abstract

Handwritten signatures are an obvious and special way of showing a person's identity. It finds an essential place in the area of biometric behavior, thanks to its simplicity and uniqueness. Signatures are the most commonly recognized biometric feature for identification purposes by law enforcement agencies / personnel, in particular in financial institutions, legal transactions, etc. and thus safe authentication is essential. Numerous transactions are taking place in the digital age era, where the agencies, e.g. banks, etc., require handwritten signature verification. In such scenarios, in addition to being accurate and secure, the signature verification process should be very rapid, i.e. it can be done in real time. We have proposed in this paper a convolutionary, language-independent, neural network-based shallow architecture (sCNN (Shallow Convolutional Neural Network)) for signature verification. The architecture proposed is very basic but extremely accurate in terms of precision. A custom neural network with a shallow convolution is used to automatically learn signature features from the training data provided. Another contribution from the research work, namely the handwritten collection of signature data for 137 subjects and 467 subjects, named respectively as CVBLSig-V1 and CVBLSig-V2, has been reported in this paper. On publicly accessible datasets, i.e., MCYT-75, MCYT-100 and GPDS, as well as CVBLSig-V1 and CVBLSig-V2, the performance of the proposed architecture was assessed. The performance was also compared and shown to be improved with state-of-the-art recorded methods, thus considering the precision and equal error rate (EER) as output measurements.

Section 1

Introduction

Verification and authorization of an individual is one of the basic needs of financial institutions and is mandatory to complete a transaction. Hence making use of biometrics, builds a secure authentication system. Biometric is related to human characteristics, and biometric recognition means to recognize a person based on his/her physical or behavioral traits. Biometrics is classified into two groups, namely physiological and behavioral, where the physiological features include the face, fingerprint, iris, palm-print, etc., while the biometrics of behaviour, signature, gait, voice, etc. For this purpose, signatures were used for ages, and are the preferred biometric modality. With the technology's advances, a large number of financial transactions are taking place today, which need to be verified against the genuineness of such transactions. Even today, most organizations use traditional methods to verify the individuals' signatures. The traditional techniques are mostly manual, and for the said purposes, experienced persons are required. The manual verification takes considerable time, and it is a fully subjective method that is highly dependent on the human verifier 's expertise. Signatures are highly influenced by the subject's mental state and thus become even harder for the verifier to verify. Biometric plays a vital role in the design of the computer-based automatic state of the art method for identification and verification.

A signature verification system verifies a person based on his / her handwritten content and is used in government and private financial institutions to verify the individual's authenticity as the owner of any document, record, or account. Signatures, as biometric is popular due to their simple procurement process, requiring less effort. In fact, the mechanism is common to humans. Automatic signature verification can be done in online or offline mode. The signatures are collected electronically in an online process, i.e. on the signature pad, and also contain some other auxiliary information such as pen-up, pen-down, pressure, angle, etc. In offline mode, only pen and paper are used to collect signatures, and the auxiliary information is not available, thus less features and versatility with the verification the of offline signatures, making the method more difficult.

Offline Signature verification aims to automatically verify an individual based on one's signature and the result of a positive or negative decision. It is possible to befool the signature verification system by a forgery. In the literature, mainly two types of forgery have been reported, namely the random and skilled forgery. In random forgery, all genuine signatures are considered as the forged signatures in the dataset for all other users, while in skilled forgery, a skilled person practices signing the signature of a specific person. The rest of the paper was split into six sections. A discussion on the related state of the art methods exists in Section 2. Section 3 contains information about methodology and network architecture. In Section 4, there is a discussion about datasets used in the experiment, parameters, and experimental setup. Results are discussed in Section 5. In Section 6 conclusion and future scope are discussed.

Section 2

Literature Survey

Lot of research work has been done in the past and is published in the literature [2, 3, 6, 7, 9–11, 13, 19, 21, 23, 26, 27, 29].

Napa et al. [26] proposed a histogram-based method of using mobile devices to verify a signature. Signatures collected from a mobile device can be represented as a series of histograms. Signatures are standardized in the time domain prior to histogram extraction, and strokes of the signature are concatenated. Histogram functions can capture the required attributes and can also preserve the relationship of the attributes. Histograms can be used in many fields as identification tools, e.g. object identification, Offline mode signature verification. A characteristic vector is measured and quantified in O(n) time, where n is the signature length, depending on the length of the signature. Of both signatures, the length of the function vector is not standardized, raising the question at matching.

Manjunatha et al. proposed an online signature verification algorithm specific for writers [19]. Dedicated features and a classifier have been learnt for each writer. The EER of all the user-specific classifiers is determined at the later stage of the process, and the classifier is selected based on minimum EER for each case. The classifier has the highest frequency, considered the best classifier among the trials. In their experiment the publicly available dataset MCYT-100 was used. In their approach time complexity is higher because they have used different classifiers for each writer.

Abhishek et al. proposed an online signature verification method, using the Gaussian Mixture Model (GMM) features [27]. These characteristics are used to align signature that was derived from Dynamic Time Warping (DTW). DTW is a measure showing the similarity between reference and test signature. Score of the warping path can be obtained by taking the similarity between histograms generated from the cost matrix. Warping path score and DTW score is fused using the sum rule for validating signature authenticity. The goal of DTW fusion and warping path score is to increase distinction between true and forged signatures. In their experiment MCYT-100 dataset was used. Signatures are suffering from a major problem, called intra-class variation, so the signature verification system needs to be improved.

Songxuan et al. [17] The concept emerged of using a recurrent neural network (RNN) to validate a signature. The authors aim to minimize the variation between genuine and forged samples in the intraclass. To pre-process the signatures, the authors proposed a descriptor called Length Normalized Path Signature (LNPS). Direction of signature with this descriptor is normalized. The descriptor LNPS is invariant of size and rotation. The authors tested their approach on the SVC-2004 dataset which is publicly available. To train RNN requires a large dataset and also high computational power. Negative examples are also required to train RNN, but it cannot be possible for each dataset because there is no sample created by any dataset.

In [16], authors presented an online verification method which uses the features of curvature and torsion. These functionalities are described in [16]. Signature was viewed as the spatial curve. The curvature and torsion characteristics were determined on the basis of the peaks and valley points in the signature curve, and an eight-dimensional feature set was developed. Verification of the test sample in respect of the reference sample was performed based on the Hausdorff distance.

In offline mode, dynamic information such as pressure, angle, pen-up pen-down etc. is lacking. The lack of knowledge makes the verification process offline a difficult job in contrast with the online methods. The signature online mode has additional helping details to support the signature function vector. There is the need to focus on an offline signature mode as many organizations such as banks favor the offline signature mode. Researchers must therefore turn the literature towards verification of of the offline signature.

Hafemann et al. suggested a method for verifying of offline signature on the Convolutional Neural Network [11]. They investigated that, in the authentication process, handcrafted features often have little or even less similarity to a signature. The authors claimed that CNN offers features that are better than those handcrafted. Publicly accessible GPDS, PUC-PR datasets are used to evaluate the method 's performance. They said their approach reached the lowest EER but there is an imbalance between the False Positive Rate (FPR) and False Negative Rate (FNR). Later, the authors expanded their work [10] and studied the deeply learned features extracted in [11]. They have examined several architectures and recorded on the GPDS dataset the lowest EER in literature. Sounak et al. documented a Siamese convolutionary network for signature verification in 2017 [6] Siamese network has two identical networks with common weights, the same parameters, and configuration that take different image pairs as input. Connecting these two networks using a contrastive loss function. According to the loss function similarity score between two images was determined using Euclidean distance, parameters were modified equally in both networks at the time of back-propagation. The network was trained with the intention of reducing the gap from genuine to genuine pair and increasing the gap from genuine to imposter pairs. Authors have tested their approach on radically different datasets, e.g. BHSig260, GPDS, CEDAR. Since two networks trained concurrently, this method requires sufficient time and high computing power.

Vargas et al. reported a signature verification method of offline mode [29]. This approach operates on statistical characteristics of wavelets and textures at local and global image rates. Any of the characteristics derived in this process are based on the form of ink used when a signature is captured. The matrix of co-occurrence gray-levels (GLCM) was created based on local characteristics. Support Vector Machine (SVM) was used to provide instruction. To achieve optimum results, they performed training and testing ten times. The reported false-positive rate (FPR) was very high for the MCYT dataset, 28 per cent.

YILMAZ et al. [3] suggested a system which would use two-channel CNN to check signatures. The authors used CNN for extraction and testing of apps. The Network accepts signatures as a pair, the first channel receives a reference signature and the second channel takes the signature of the query. They retained the output while at the last locally connected layer rising the dimensionality to 200. They effectively reduced the EER on the GPDS dataset to 4.13 per cent. They later used fusion of the score point and the EER dropped further to 1.73 percent. In [9] HADJADJI proposed a writer-dependent method which used curvelet transformation (CT) and main component analysis (PCA). They used curvelet transform to create the features, and PCA is designed to reduce the size of the feature that CT generates. They have used the Choquet fuzzy integral to combine the results of multiple scores produced by the program. They performed their GPDS data set experiments and recorded 94.96 percent accuracy. Manabu et al. [21] proposed a method which combined features of the fisher vector with features of KAZE [1] fused. KAZE functions collected pictures from the foreground and background signatures. They showed that KAZE features have good results when checking signatures. They also discussed that in the spatial domain the fishery vector provides a more accurate distribution of the characteristics per writer. They also used the MCYT-75 dataset in their experiments. Dutta et al. suggested a system that would use the local characteristics and global local characteristics statistics [7]. Histogram of Oriented Gradients (HOG) based features were used in their method, and weight was added to the local characteristics based on the signature height. They used the idea that local features play a significant role in signature geometry, and one can differentiate between genuine and imposter signatures by using these features. To advertise, authors generated a local pairwise feature with the access size of the signature, and this feature was described as a collective descriptor. They tested their system on data set GPDS and CEDAR.

Authors in [23] used the most robust algorithm for extracting texture, namely local binary pattern (LBP) and uniform local binary pattern (ULBP), to extract the signature image features. Authors also used the nearest neighbor method for similarity analysis. They found that the results obtained by LBP and ULBP on the publicly accessible BHSig260, GPDS100 dataset were no remarkably different. Neural Network has one drawback all the inputs must be of the same size to the network. Authors at [13] used the idea of spatial pyramid pooling (SPP)[15] to resolve the signature size variability. SPP is used to generate a fixed vector size from the different image sizes. They have also investigated the machine efficiency improved by using fine tuning on SigNet [12].

Authors in [2] introduced a method that uses freeman chain code (FCC) to represent the data. The FCC was pulled from the longest linked route. It provides a comprehensive overview of the FCC in [2]. Later the FCC was divided into different number of parts (four, eight, sixteen). The authors have estimated six global features outlined in [2]. Signature picture checking was performed using neighbor K-nearest. Their analysis was conducted on the MCYT dataset. Signature images have major variability in the intraclass. To address the issue, the authors in [4] suggested a method based on extraction of run-length features. To measure the run-length features the signature image was transformed into a binary image. The black pixels were deemed in the binary picture. Run can be defined in a given direction as the chain of connected points; all pixels have the same intensity level. Authors used SVM of a single class for classification purposes. Authors in [20] tried to improve Automated Signature Verification efficiency. The authors used Visual Word Bag (BoVW) and Locally Aggregated Descriptors (VLAD) vector in this process. A thorough description of BoVW and VLAD was given at [20].

KAZE elements, along with the BoVW and VLAD, were extracted from the signature picture strokes. SVM was used for the classification.

Given advances in the field of signature verification, when the inter-class similarity is higher, the program fails to verify the signature images.

Offline signature authentication is commonly used by many companies, and is a challenging issue due to the lack of accessory details such as strain, pen up, pen down angle, etc. In addition, the accurate production of signature verification architecture is the need for the hour, and this has inspired us to work on biometric verification of an offline signature. The purpose of this manuscript is to develop a very simple model that outperforms the state-of-the-art precision methods. The convolutionary neural network has proved its superior process of extraction of features over most conventional approaches. That motivated us to use CNN as stated in this paper in our proposed research work. We decided to develop a simple but effective model that would satisfy our demands. Since the signature images are not the images rich in detail, and there are just shapes, curves, and edges; thus, there are no very deep features. In the proposed study, we used shallow neural network architecture to verify signatures, since for some data set and for some data set performance degradations, accuracy saturates to increase layers. They contributed to this work in two ways. Firstly, in two sessions we have built an offline signature dataset (CVBLSig). Sessions one and two have 137 and 467 participants, respectively. Detailed information concerning this dataset can be found in Section 4.1. Second, we've built sCNN (shallow Convolutional Network), which includes a smaller number of layers that have been well performed yet.

Section 3

Proposed Method

3.1. Methodology

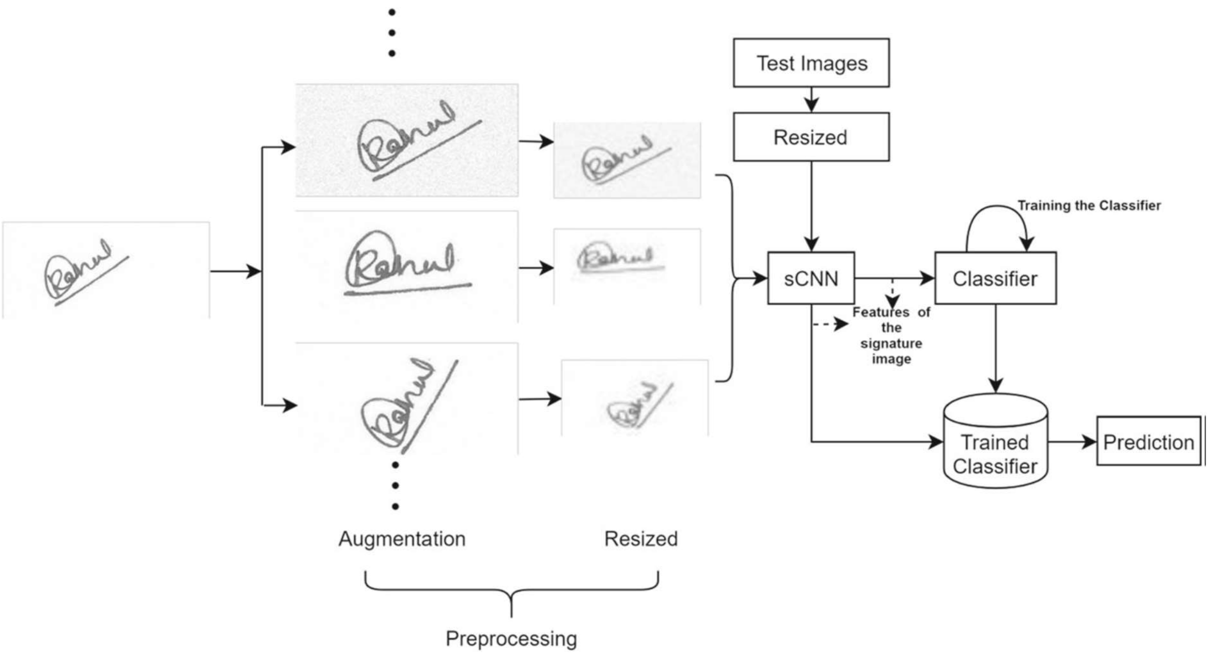

Fig. 1. Methodology of proposed work.

Figure 1 shows the block diagram of the proposed methodology. Before training, all signatures images are passed through the preprocessing stage, as described in Section 4.2. Preprocessed signature images are fed to the sCNN (Shallow Convolutional Neural Network) block, which is described in Section 3.2, and shown in Fig. 2. The sCNN is trained over the training images with the parameters described in Table 2 and Section 4.3. The sCNN shall be trained with the parameters mentioned in Table 2 and Section 4.3 over the training photos. The sCNN Extracted features are fed to a classifier with softmax. We have a trained model at the end of the training and classification process, and based on the threshold, the test signature image is checked with the help of that trained model. The detailed description of the various block, of Fig. 1 is described in Section 3.2.

3.2. Network Architecture

As discussed in Section 1, often handcrafted features are not enough to identify any person, giving us the incentive to use such a model that the related features of the signature image can be found. These features will help us correctly identify every individual.

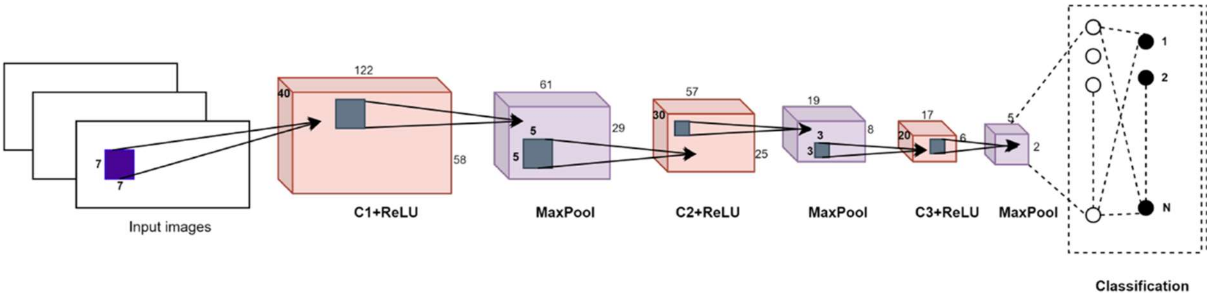

Fig. 2 Network Architecture of Shallow Convolutional Neural Network

In this proposition we used CNN as a base architecture to extract the related signature features. CNN is a layered system of several layers, such as convolutionary, pooling, and completely linked layers. In the [5] the detailed description of the CNNs was given.

The model proposed consists of three convolutionary layers and one completely connected layer. The proposed network architecture is shown in Fig. 2. Raw signature images are provided to the network's input layer which takes size 64x128 input. The initial convolution layer (C_1) is connected to the input layer. Fourty random filter kernels of size 7x7 can filter the input image when using the stride one. The convolutional layer will produce the respective function maps, depending on the number of filters and channels. We have forty feature maps after the first convolutionary sheet, each 58x122 in scale. The feature maps generated by C_1 are being fed to the activation function Rectified Linear Unit (ReLU). To resolve the problem of vanishing gradients, ReLU applies some non-linearity to the functionality. ReLU enabling function is: $R(x) = \max(0, x)$. Where R is a vector for activation of ReLU, and where x is vector values from the feature maps. This function tests the space of the feature, and if any, removes the negative value of the feature. The pooling layer ($Maxpool_1$) is placed after the nonlinearity layer. The aim of the pooling layer is to reduce the output size from the ReLU layer and to introduce the non-linearity in the space of features. In our proposed architecture, we used the max-pooling. The window size of 2x2 is used in this layer over the image with stride two and takes full value from the window. After the scale of each function map maxpool layer became

29x61. Thirty filters with a kernel size 5x5 will filter feature maps created by the Maxpool1 layer using stride 1 using the second convolution layer (C2). In the proposed architecture we have maintained symmetry, i.e., convolution layer followed by non-linearity followed by the maxpool layer. C2 created feature maps are further passed to the ReLU, and then the window size 3x3 and stride 3 Maxpool2 sheet. Maxpool2 is connected to the third C3 convolution layer which has twenty 3x3 size kernels with a stride 1. The function maps of C3 are fed to the ReLU, and the window size of Maxpool3 is 3x3 with stride 3. We will have a fully connected layer after Maxpool3 that layer will contain some specific neurons according to the number of classes in the database. There are 41370 parameters in the network to a fully-connected node. According to [31], 2n + d parameters are sufficient to express the dataset for a two-layer neural network with n samples having d dimensions. Parameters in the proposed network are appropriate to mark the dataset used in the proposed research as the number of parameters needed falls within the range of available parameters. We used a softmax classifier for classification, which has the cross-entropy loss function [18]. This loss function is represented according to equation (1).

$$Loss_i = -\log(e^{K_{z_i}} / \sum_j (e^{K_j}))) = -K_{z_i} + \log(\sum (e^{K_j})) \quad (1)$$

In kj, k means score of class, and j means the value of vector k. The classifier softmax gives the probability that every test sample belongs to all groups. Initially, filter values are assigned by a Gaussian distribution and generated randomly, and these weights are modified at the time of back-propagation. We used a stochastic momentum gradient optimizer (SGDM) 0.9 [25].

Table 1 summarizes the activation parameters, output form and number of parameters. In Table 1, the filter size is defined by F, and the phase by S. The second column in the table describes the form of an activation map for the convolutionary and pooling layer as M x W x H, where M is the number of filters, and W, H is the height and width of the activation map. The third column in the table includes the number of parameters for weights and bias.

Table 1. Network architecture of sCNN

Layer	Activation shape	Number of learnable parameters	Total count
Input Layer	$64 \times 128 \times 3$	0	0
Convolution Layer (C1) (F=7, S=1)	$40 \times 58 \times 122$	weight = $7 \times 7 \times 3 \times 40$, bias = $1 \times 1 \times 40$	5920
ReLU	$40 \times 58 \times 122$	0	0
Pool (S=2)	$40 \times 29 \times 61$	0	0
Convolution Layer (C2) (F=5, S=1)	$30 \times 25 \times 57$	weight = $5 \times 5 \times 40 \times 30$, bias = $1 \times 1 \times 30$	30030
ReLU	$30 \times 25 \times 57$	0	0
Pool (S=3)	$30 \times 8 \times 19$	0	0
Convolution Layer (C3) (F=3, S=1)	$20 \times 6 \times 17$	weight = $3 \times 3 \times 30 \times 20$, bias = $1 \times 1 \times 20$	5420
ReLU	$20 \times 6 \times 17$	0	0
Pool (S=3)	$20 \times 2 \times 5$	0	0
Fully Connected			

Section 4

Implementation

The aim of this experiment is to design a model capable of extracting relevant features from the signature image. The extracted feature set should be sufficiently robust to distinguish between the two distinct samples. The proposed architecture extracts the set of functions that can differentiate between the different samples. After the different convolutionary layers were shown in Figs. 3 and 4, respectively the one function map and one filter. It's notable from Fig. 3, that the edges of the signature image were extracted via the starting convolutional layers. There are no such specifics in the function diagram, after two convolutionary operations. The Line 3 in the case of the signature photos, supports the use of the shallow network.

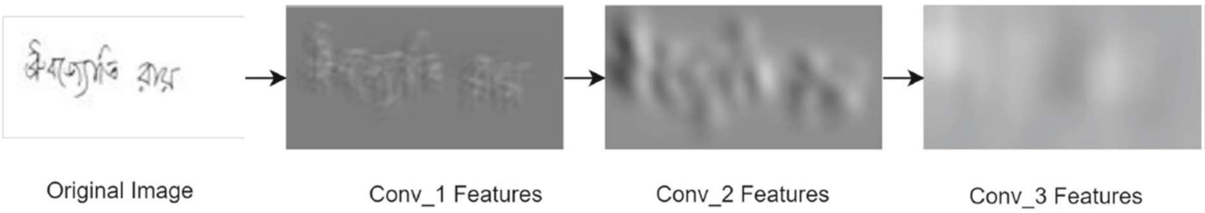

Fig. 3 Visualization of the features at different layers on Bengali dataset

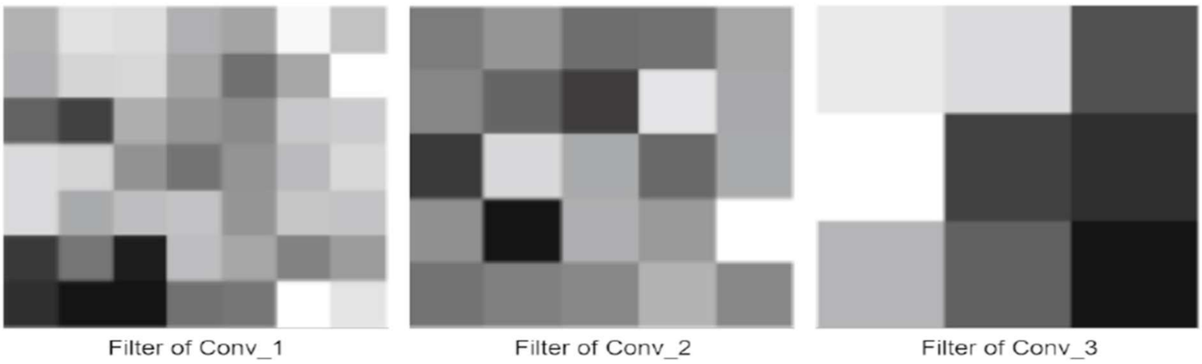

Fig. 4 Visualization of the filters at different Conv layers on Bengali dataset

We have set, with extensive experiments, that three convolutionary layers are enough to work with signature images. We also tried to vary the number of convolutionary layers, and found that there is no impact on device output by increasing the number of convolutionary layers.

4.1 Dataset

The proposed architecture has been trained on a variety of publicly accessible datasets, namely MCYT-100 [22], MCYT-75 [17, 22], and GPDS datasets [16, 33], SVC-2004 [30], and CVBLSig-V1, CVBLSig-V2 that have handwritten images where English is the basic language used. Two other datasets, namely BHSig260 Hindi [23] and BHSig260 Bengali [23], were also considered to demonstrate the broad applicability of the proposed method where the basic language used is Hindi and Bengali, respectively. Information about the various signature data sets that we find were elaborated in the following subsections.

4.1.1 MCYT-100

The MCYT-1001 was compiled on a WACOM ink tablet and consists of 25 genuine and 25 fake signatures for 100 people, so just a total of 5000 signature images [17, 22]. This dataset includes additional supporting details such as an azimuthal angle, strain, etc. But in our experiments, we did not find these pieces of information because; these are not needed for verification process of offline signatures.

4.1.2 MCYT-75

Dataset MCYT-752 is a subset of MCYT-100 which includes offline signature mode of some MCYT-100 individuals. This dataset has 75 users, with 15 genuine signatures and 15 forged signatures each [17, 22].

4.1.3 BHSig260

This dataset3 has two regional languages, Bengali and Hindi, with a signature. There were 100 signers in the Bengali dataset, and every signer has 24 genuine signatures and 30 fake signatures [23]. Total signature was 24 x 10 = 2400 Genuine and forged signature was 30 * 100 = 3000. Signatures of 160 people were collected in the Hindi dataset, and each individual has 24 genuine and 30 forged signatures.

4.1.4 GPDS

This dataset has the individual's offline signatures. The GPDS4 dataset has 4000 users. GPDS is the largest dataset of signatures present to date. GPDS Dataset comes with GPDS-160, GPDS-300 and GPDS-960 models. When we use the GPDS-300 dataset it means that we choose the first 300 of 4000 people for the experiment. Dataset also contains forged signatures which skilled people collect. Everyone has 24 genuine pictures and 30 fake pictures. Since the data set is collected in offline mode, no angle, strain, etc. information exists [15, 16].

4.1.5 SVC- 2004

SVC-20045[30] was the first international competition in the field of signature verification to support the signature verification process. This dataset was gathered at two tasks. Every task involves 100 individuals, but the authors have published the signatures of 40 individuals, as the remaining 60 requires the same person with different inks and strokes. The WACOM Intuos tablet was used for collecting signatures and stored in a separate text file. All has 20 authentic signatures and 20 forged ones. These test files include the sequence of points in the signature, and the number of signature sequence points will be in the first line of text file. Signature files at Task 1 and Task 2 have X-coordinate, Y-coordinate, time-stamp, button status. Additionally, task 2 signature file will have azimuth, altitude, and pressure.

4.1.6 CVBLSig

We have developed an offline signature dataset CVBLSig, as a contribution to the research community. This dataset was obtained from a sheet of paper. Figure 5 displays the setup after which the CVBLSig dataset signature was obtained. Every person must sign in one column of the board. One sheet of paper can be signed by max. four people. Using a digital scanner, certain sheets are scanned and signatures are cropped. Session one includes 137 (CVBLSig-V1) people, and session two has 467 (CVBLSig-V2) signers. Each signer in CVBLSig-V1 has a total of 20 images, and each signer will have 15 images in CVBLSig-V2, respectively. There are no forged signatures in this dataset but we also intend to include each individual's forged signature. Since this data set is so gathered in offline mode, there is no supporting information such as strain, angle, pen-up, pen-down.

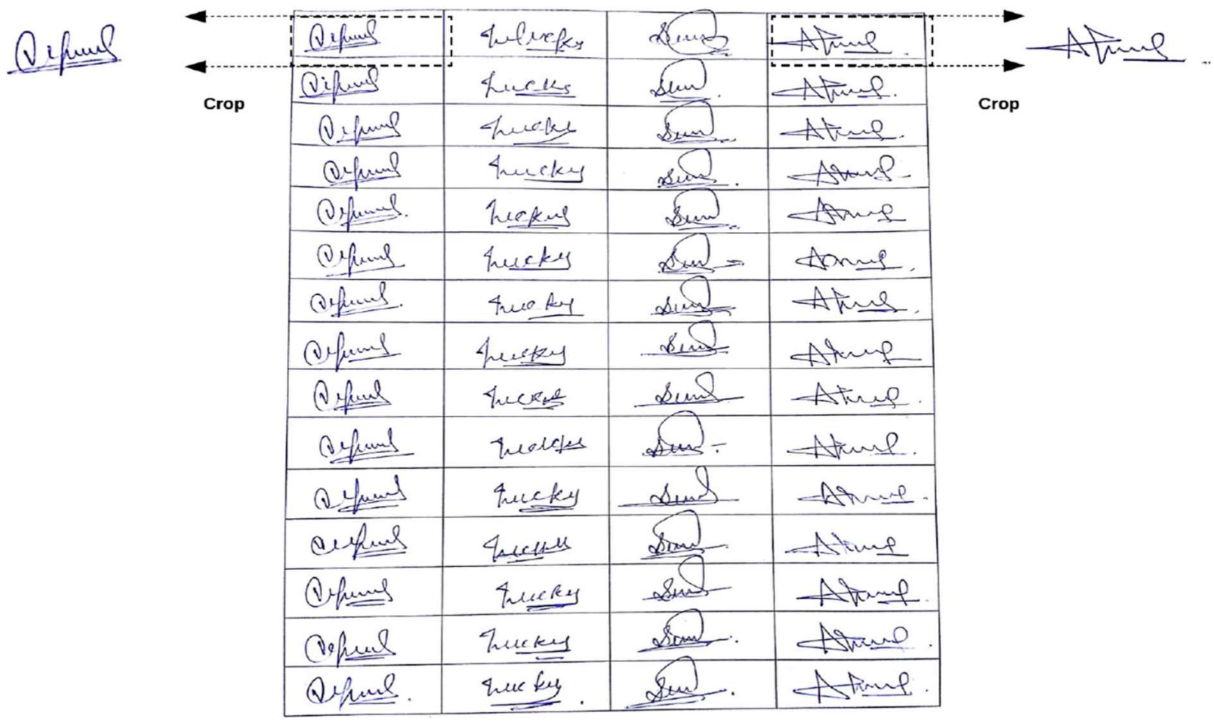

Fig. 5 Setup of Signatures form CVBLSig Dataset

Figures 6, 7, 8, 9, 10, 11 and 12 shows samples of CVBLSig, MCYT-100, GPDS, BHSig Bengali, BHSig Hindi and SVC-2004 dataset respectively.

Fig. 6 Sample Signature of CVBLSig Dataset

Fig. 7 Sample Signature of MCYT-100 Dataset

Fig. 8 Sample Signature of MCYT-75 Dataset

Fig. 9 Sample Signature of GPDS Dataset

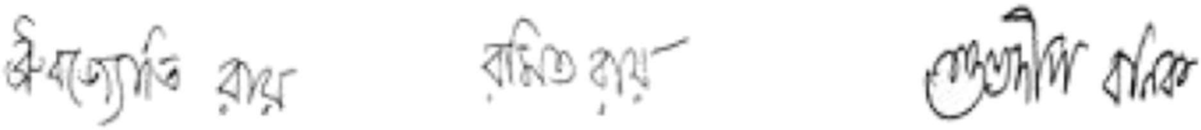

Fig. 10 Sample Signature of BHSig260 Bengali Dataset

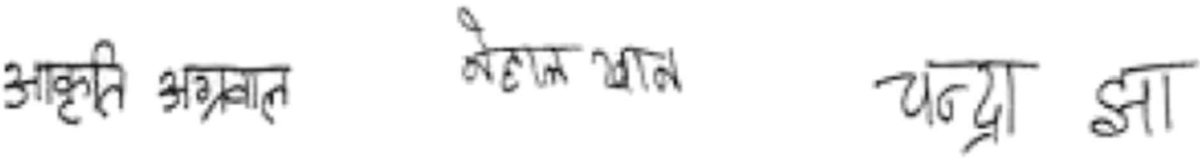

Fig. 11 Sample Signature of BHSig260 Hindi Dataset

Fig. 12 Sample Signature of SVC-2004 Dataset

4.2 Pre-processing

We know very well that the biometric signature is the behavioral trait which poses substantial variations in the intraclass. The appearance of the handwritten signatures depends on several intrinsic factors (the person's mood, mental state and cognitive behavior), and extrinsic factors (Background, Pen, Ink Color, physiological style, etc.). Due to the intrinsic and extrinsic effects, the captured signature images, which are mostly horizontally aligned, can differ in scale, angular divergence, orientation, line thickness, background paper color, etc. So, a lot of possibilities for the captured signatures, and hence it is important to have the signatures in the training dataset with all possible variations for the better generalization and accuracy. Because most of the publicly available data sets, i.e., MCYT-75, MCYT-100, and GPDS, BHSig260, and SVC-2004 do not have a large number of samples per piece, we have accomplished many data increases, i.e. 10 and 30 degree clockwise and anticlockwise rotations, factor 0.6 shears, Gaussian noise (zero mean and variance 0.01) has been applied and the aspect ratio modified by 3/5. Section 4.1 has given detailed overview of the datasets. The signatures images in the different datasets are of varying sizes, so resizing to standard 64x128 pixels was achieved. The preprocessing results were shown in Fig. 13 On the V2-CVBLSig. The proposed sCNN feeds these preprocessed images for further processing.

4.3 Parameters

Table 2 displays the parameters used for network training as shown in Fig. 2. We have included all the parameters used in the experiment to reproduce the effects of that experiment. With momentum [25], we used a stochastic gradient downflow optimizer. SGDM has as parameter momentum, learning rate, gradient threshold. We used a non-adaptive rate of learning equivalent to 10−4. Momentum is 0.9 with threshold gradient 7. Unless the low value of momentum is selected, the optimizer will fluctuate a lot between local minima and maxima for a new sample. Other parameters affect network efficiency, i.e. mini-batch size and number of epochs. We assigned batch size to 32, and completed the experiment after 150 epochs.

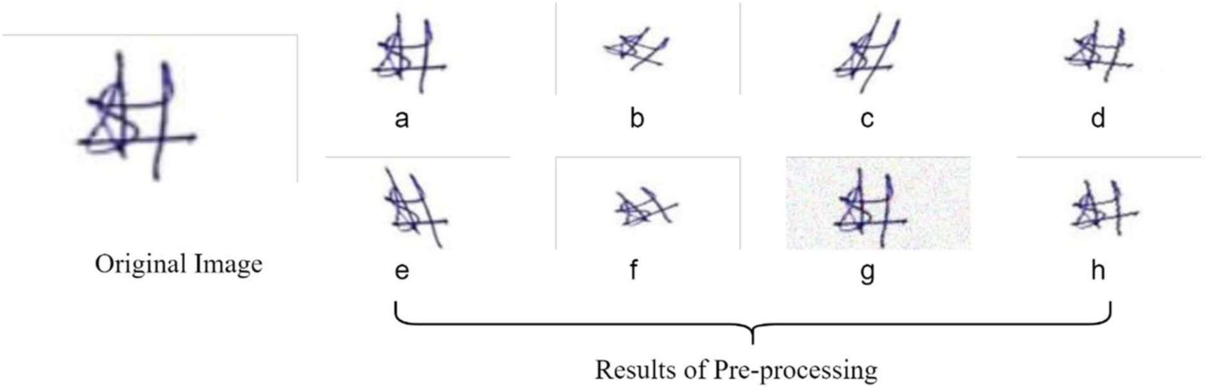

Fig. 13 Results of Pre-processing on CVBL-V2

a. Change in Aspect ratio **b.** Rotation by 30 degrees in clockwise direction **c.** Right Shear **d.** Rotation by 10 degrees clockwise **e.** Left Shear **f.** Rotation by 30 degrees in anticlockwise direction **g.** Gaussian Noise **h.** Rotation by 10 degrees anticlockwise

Table 2. Training parameters

Parameter	**Value**
Learning rate	10^{-4}
Momentum	0.9
Batch Size	32
Gradient Threshold	7

4.4 Experimental setup

Experiments were conducted on a workstation that had the following configuration. Intel(R) Core (TM) i7-7820 x CPU @3.60 GHz, 16 GB RAM, and each have three 11 GB NVIDIA 1080 Ti. We used MATLAB 2018a for the experiment.

Section 5

Results and Discussion

This research focuses on developing a verification scheme of signatures using the convolutionary neural network. We have broken down all datasets into different ratios for this experiment. These data distributions can be seen as the test images: train images [2:8, 3:7, 4:6, 5:5]. Such dataset distributions are trained and evaluated on the proposed model. Figure 14 shows the accuracy of the relation on our model between those distributions. It is evident from Fig. 14 That we have been given greater accuracy across all distributions when we use the distribution dataset [2:8]. When we divided the test and train data set into 2:8 ratios, MCYT-100 gave the highest precision of 98.40 percent among all distributions. By Fig. 14 It is clear that a large number of training samples are required to improve accuracy and generalization compared to the test samples.

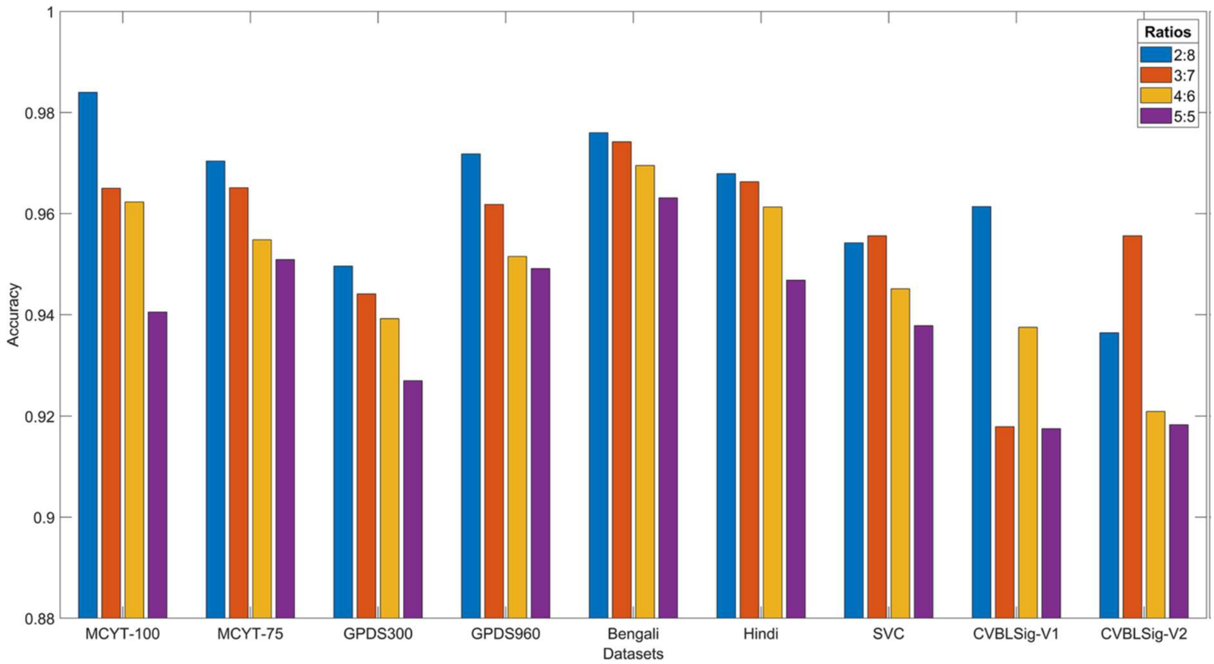

Fig. 14 Accuracy vs Data Distribution for sCNN

Table 3. Accuracy of sCNN on different dataset using SGDM optimizer

Dataset	Accuracy (%)
MCYT - 100	99.40
MCYT - 75	98.93
GPDS300	96.87
GPDS960	97.19
BHSig Bengali	98.40
BHSig Hindi	97.12
SVC - 2004	97.00

Table 3 summarizes the accuracy of the proposed methods by taking into account a number of data sets which are publicly accessible and self-created. The proposed network is trained from scratch in all the experiments, thus randomly choosing five test images and remaining images as the training images. It is evident from Table 3 that the accuracy of identification obtained is 99.40 percent, 98.93 percent, 96.87 percent, 97.19 percent, 97.00 percent, 96.78 percent, and 93.15 percent, respectively, for MCYT-100, MCYT-75, GPDS-300, GPDS-960, SVC-2004, CVBLSig-V1, and CVBLSig-V2 datasets. We also did this experiment on the BHSig260 national dataset, which includes two Hindi and Bengali languages. On BHSig Bengali and BHSig Hindi dataset, respectively, we have achieved accuracies of 98.40 percent, and 97.12 percent. Our model provides the highest accuracy on the MCYT-100 dataset, since after augmentation the sample per class is higher in MCYT-100 data set. It can be noticed that the model generalizes well, and performed better, when we have a large number of training samples. The convergence of the loss and accuracy with respect to the iterations has been shown in Fig. 15 for all the datasets.

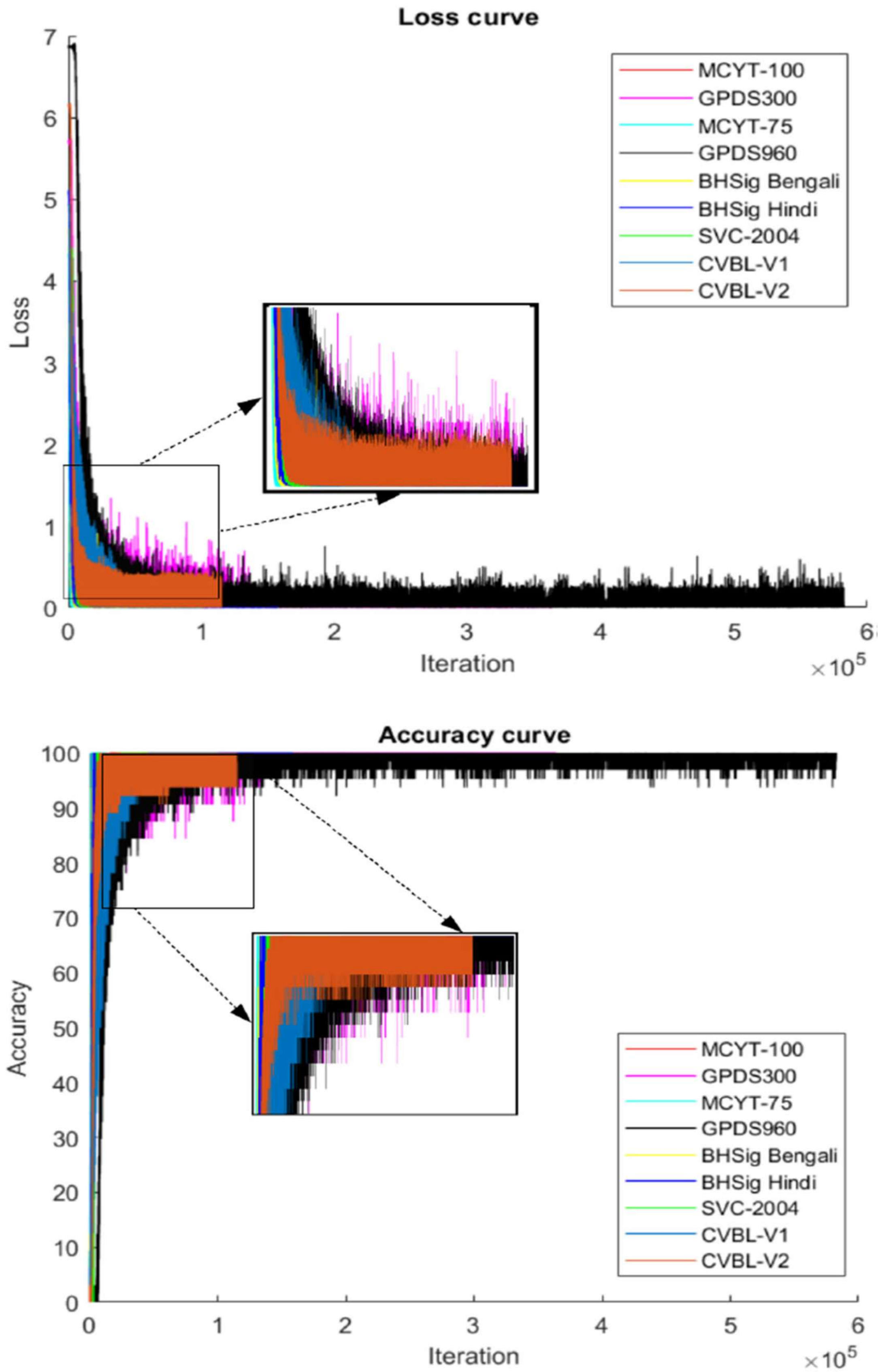

Fig. 15 Convergence of Loss and Accuracy Curve

The proposed design has also been tested with the two separate optimizers, i.e. rmsprop and adam. Table 4 indicates results for the same. It can be seen from Tables 4 and 3, in comparison with the adam and rmsprop, SGDM works better in the signature domain.

In literature, most authors find EER, respectively, as a performance parameter for MCYT, SVC data collection, and accuracy for GPDS, BHSig260 datasets. So, we considered EER as a performance parameter for MCYT and SVC datasets and accuracy for GPDS and BHSig260 data sets for a fair comparison. Table 5 displays the EER about the proposed MCYT architecture. EER can be defined where both the false-negative (FNR) and the false-positive (FPR) rate are equal. It is not always possible to get the same value of the FNR and FPR when we have a large number of classes; hence, we have taken a difference of 0.01 between FPR and FNR to find the correlation between FPR and FNR. The average will be based on occurrence of multiple EER points.

Table 4. Accuracy of sCNN using adam and rmsprop optimizer

Dataset	Optimizer	Accuracy (%)
MCYT-100	adam	96.00
MCYT-75	adam	94.40
GPDS300	adam	96.27
GPDS960	adam	94.17
Bengali	adam	97.60
Hindi	adam	97.13
CVBL-V1	adam	95.47
CVBL-V2	adam	92.91
MCYT-100	rmsprop	93.60
MCYT-75	rmsprop	95.73
GPDS300	rmsprop	89.33
GPDS960	rmsprop	78.60
Bengali	rmsprop	94.80
Hindi	rmsprop	93.50
CVBL-V1	rmsprop	93.71
CVBL-V2	rmsprop	87.26

From Table 5 it is clear that our model yields better results than [8, 12, 13, 16, 17, 19, 21]. By using the combination of different classifiers (Neural Network and PCA), Manjunatha et al. reported EER of 0.80 percent [19]. Authors in [19] have also increased the difficulty of using specific classifier forms for each speaker. In [12], authors used five convolution layers (AlexNet architecture) and two fully linked layers, reducing EER on the MCYT-100 dataset to 2.87 per cent. At the other hand, we used just three convolution layers and one completely connected layer, and on the MCYT-100 dataset we succeeded in reducing EER to 0.2 per cent. The number of parameters in the proposed system is less as compared to [12]. The method proposed has 41,370 before fully connected layer and [12] has 3,747,200 parameters. This work can be used in real-time applications and mobile devices, due to the reduced number of layers and parameters.

Table 5. EER Comparison on MCYT Dataset

Dataset	Method	EER
MCYT-100	W-Dependent feature [19]	7.75
	W-Dependent feature (NN+PCA) [19]	0.80
	Signet (SVM) [21]	2.87
	DTW and GMM [28]	2.12
	Curvature feature + Torsion features [16]	1.20
	sCNN (Proposed)	**0.2**
MCYT-75	GLCM [8]	2.30
	GLCM+WT [8]	2.44
	FV with fused KAZE features [21]	5.47
	Fixed size representation [13]	3.64
	BoVW with KAZE features [20]	6.4
	WI using asymmetric pixel relation [32]	3.5
	sCNN (Proposed)	**0.4**

There is also a comparison of our approach on the MCYT-75 dataset with state-of-the-art methods in Table 5. Compared to the most state-of-the-art system [8, 13, 14, 20, 21, 32], sCNN did better. Authors in [8] used GLCM and wavelet transform to train and evaluate their system along with SVM. Compared to the [8] proposed model, there is better performance than GLCM and Wavelet features because of the consistency of CNN's extracted features. In [14], authors used CNN along with the SPP to increase the EER by 3.4 per cent. SPP provides the vector with a fixed size before fully attached layer. We have managed to reduce the EER to 0.4 per cent on the MCYT-75 dataset. One explanation behind reduced EER is the higher accuracy obtained for the proposed model, and the lowest FNR and FPR.

For the current architecture, GPDS dataset accuracy is high compared to state-of-the-art methods [4, 6, 9, 14] [see Table 6].

Table 6. Recognition accuracy comparison on GPDS dataset

Dataset	Method	Accuracy (%)
GPDS-300	Compact corelated feature [7]	88.79
	Curvlet Transform [9]	94.96
	Signet (Siamese) [6]	76.83
	sCNN (Proposed)	**96.87**
GPDS-960	Compact corelated feature [7]	73.67
	Signet (Siamese) [6]	77.76
	One-class SVM [4]	95.68
	sCNN (Proposed)	**97.19**

Authors in [6] used the Siamese network to differentiate between genuine and fake signatures and achieved accuracy of recognition of 76.83 percent, and 77.76 percent, respectively, of GPDS300 and GPDS960. In [7], authors used correlated features to differentiate the inter-class samples and achieved 88.79 percent accuracy, and 73.67 percent accuracy on GPDS300, respectively, and GPDS960 data set. Table 7 shows how the BHSig260 dataset compares with the state-of-the-art process. It is evident from Table 7 that our model performed much better on BHSig260 dataset than the state-of-the-art methods [4, 14].

Table 7. Recognition accuracy comparison on BHSig260 dataset

Dataset	Method	Accuracy (%)
BHSig Bengali	LBP and ULBP [23]	66.18
	Compact Corelated Features [7]	84.90
	Signet (Siamese) [6]	86.11
	sCNN (Proposed)	**98.40**
BHSig Hindi	LBP and ULBP [23]	75.53
	Compact Corelated Features [7]	85.90
	Signet (Siamese) [6]	84.64
	sCNN (Proposed)	**97.12**

Table 8 displays sCNN findings on dataset SVC-2004. We achieved the lowest EER amongst state-of-the-art methods. ROC defines the discriminative power of the classifier, and that's not contingent on the class distribution. Figure 16 displays the Dataset Receiver Operator Characteristics (ROC) curve considered by the model being proposed. Since we calculated True Positive Rate (TPR) and False Positive Rate (FPR) for one class, we calculated ROC for each class and took an average of both. By Fig. 16, It can be found that our model has outperformed the GPDS960 dataset, because the region under the curve is the highest for GPDS960. By Fig. 16, The AUC is maximum for GPDS960, which has 960 items, that is to say a large number of people. The total number of samples after augmentation is very high as compared to other datasets. And for various datasets total samples vary, though test samples are the same. So, it can be concluded that the proposed model exhibits good output when a large number of samples are present in the dataset, and the ratio of train and test set samples is very small.

Table 8. EER Comparison on SVC-2004 Dataset

Method	EER
DTW [30]	5.50
DTW [24]	3.38
HMM [28]	4.83
RNN+LNPS [17]	2.37
sCNN (Proposed)	**1.01**

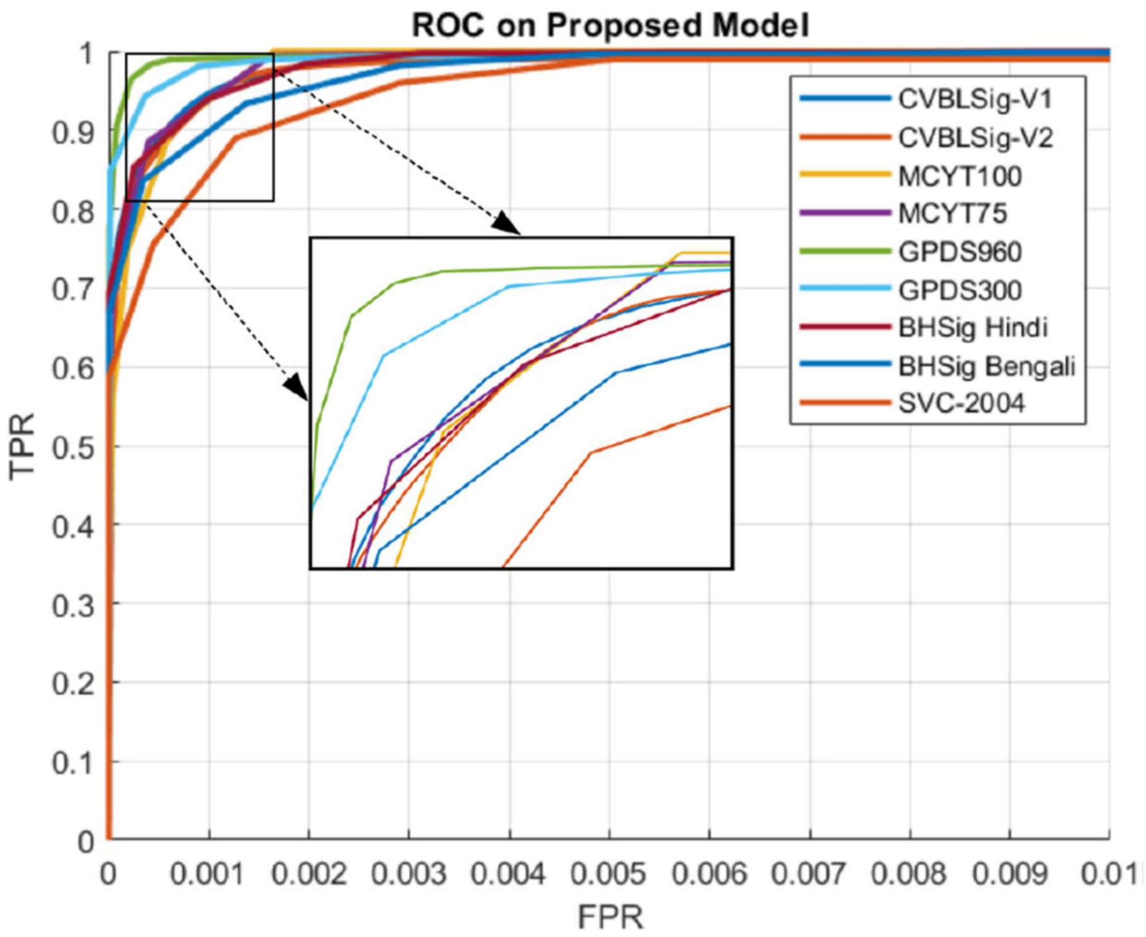

Fig. 16 ROC on sCNN for Different Dataset

Section 6

Conclusion

CNNs' recent success in classifying and identifying images in almost every area of computer vision research has motivated the creation of the proposed model. Easy and efficient convolutionary, language-independent, neural network-based signature verification architecture has been proposed in this paper. In comparison to the other state of the art models, the proposed model is very simple in terms of the number of basic layers (Conv and Pool); therefore, the weight parameters to be optimized are smaller in size. The proposed sCNN has less layers and parameters; therefore, the time taken by proposed system in training and testing would be less. Compared with other state of art signature biometric approaches, the sCNN obtained better efficiency in terms of accuracy and EER. We investigated the random forgery in this paper and achieved a marked accuracy of 98.93 per cent and 96.87 per cent respectively on MCYT-75 and GPDS- 300 datasets. EER is stated to be 0.2 percent using sCNN on the MCYT-100 dataset which is the lowest of all published literature, as shown in Table 5. On CVBLSig-V1 and CVBLSig-V2, respectively, we earned 96.78 per cent and 93.15 per cent accuracy. It can be deduced from the accuracy that the collected CVBLSig-V1 and CVBLSig-V2 datasets are quite challenging compared to the other publicly available datasets. Various researchers will also use the dataset, as stated in this paper, to benchmark the architectures and algorithms used for biometric signature recognition. Compared to other state of the art approaches and architectures, the accuracy obtained by the proposed architecture is 98.40 percent and 97.12 percent for Bengali and Hindi signature datasets (BHSig260); hence the proposed model is language independent and can be used for various regional languages too. It is also concluded that very deep network verification might not be suitable for the task of offline signature verification, and shallower networks can deliver better or comparable levels of accuracy with the reward of fewer parameters and reduced complexity in space-time. The signature dataset doesn't have a huge number of samples, so the model will over-fit when using the deep learning methods. We have tried to reduce the overfitting through data increase in this job. We would be reducing the probability of over-fitting in the near future by using data increase. This work also covered the random forgery of the signatures, and we'll focus on the proposed network's professional forged signature photos.

References

1. Alcantarilla PF, Bartoli A, Davison AJ (2012) Kaze features. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C (eds) Computer vision – ECCV 2012. Springer, Berlin, pp 214–227
2. Azmi AN, Nasien D, Omar FS (2017) Biometric signature verification system based on freeman chain code and k-nearest neighbor. Multimed Tools Appl 76(14):15341–15355
3. Berkay Yilmaz M, Ozturk K (2018) Hybrid user-independent and user-dependent offline signature verification with a two-channel cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 526–534
4. Bouamra W, Djeddi C, Nini B, Diaz M, Siddiqi I (2018) Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Syst Appl 107:182–195
5. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/convolutionalnetworks/
6. S, Dutta A, Toledo JI, Ghosh SK, Llad´os J, Pal U (2017) Signet: Convolutional siamese network for writer independent offline signature verification. arXiv:1707.02131
7. A, Pal U, Llad´os J (2016) Compact correlated features for writer independent signature verification. In: 2016 23Rd international conference on pattern recognition (ICPR), pp 3422–3427. https://doi.org/10.1109/ICPR.2016.7900163
8. Ferrer MA, Vargas JF, Morales A, Ordonez A (2012) Robustness of offline signature verification based on gray level features. IEEE Trans Inf Forensic Secur 7(3):966–977. https://doi.org/10.1109/TIFS.2012.2190281
9. Hadjadji B, Chibani Y, Nemmour H (2017) An efficient open system for offline handwritten signature identification based on curvelet transform and one-class principal component analysis. Neurocomputing 265:66–77. https://doi.org/10.1016/j.neucom.2017.01.108. http://www.sciencedirect.com/science/article/pii/S0925231217310159. New Trends for Pattern Recognition: Theory and Applications
10. Hafemann LG, Sabourin R, Oliveira LS (2016) Analyzing features learned for offline signature verification using deep cnns. In: 2016 23rd international conference on Pattern recognition (ICPR). IEEE, pp 2989–2994

11. Hafemann LG, Sabourin R, Oliveira LS (2016) Writer-independent feature learning for offline signature verification using deep convolutional neural networks. In: 2016 international joint conference on Neural networks (IJCNN). IEEE, pp 2576–2583

12. Hafemann LG, Sabourin R, Oliveira LS (2017) Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recogn 70:163–176

13. Hafemann LG, Oliveira LS, Sabourin R (2018) Fixed-sized representation learning from offline handwritten signatures of different sizes. Int J Doc Anal Recogn (IJDAR) 21(3):219-232. https://doi.org/10.1007/s10032-018-0301-6

14. Hafemann LG, Oliveira LS, Sabourin R (2018) Fixed-sized representation learning from offline handwritten signatures of different sizes. Int J Doc Anal Recogn (IJDAR) 21(3):219–232

15. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

16. He L, Tan H, Huang ZC (2019) Online handwritten signature verification based on association of curvature and torsion feature with hausdorff distance. Multimedia Tools and Applications 78(14):19253–19278

17. Lai S, Jin L, Yang W (2017) Online signature verification using recurrent neural network and length normalized path signature descriptor. In: 2017 14Th IAPR international conference on document analysis and recognition (ICDAR). IEEE, vol 1, pp 400–405

18. Liu W, Wen Y, Yu Z, Yang M (2016) Large-margin softmax loss for convolutional neural networks. In: ICML, pp 507–516

19. Manjunatha K, Manjunath S, Guru D, Somashekara M (2016) Online signature verification based on writer dependent features and classifiers. Pattern Recogn Lett 80:129–136

20. Okawa M (2018) From bovw to vlad with kaze features: Offline signature verification considering cognitive processes of forensic experts. Pattern Recogn Lett 113:75–82

21. Okawa M (2018) Synergy of foreground–background images for feature extraction: Offline signature verification using fisher vector with fused kaze features. Pattern Recogn 79:480–489. https://doi.org/10.1016/j.patcog.2018.02.027. http://www.sciencedirect.com/science/article/pii/S0031320318300803

22. Ortega-Garcia J, Fierrez-Aguilar J, Simon D, Gonzalez J, Faundez-Zanuy M, Espinosa V, Satue A, Hernaez I, Igarza JJ, Vivaracho C et al (2003) Mcyt baseline corpus: a bimodal biometric database. IEEProc-Vis Image Signal Process 150(6):395–401
23. Pal S, Alaei A, Pal U, Blumenstein M (2016) Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. In: 2016 12Th IAPR workshop on document analysis systems (DAS). IEEE, pp 72–77
24. Pascual-Gaspar JM, Carde˙noso-Payo V, Vivaracho-Pascual CE (2009) Practical on-line signature verification. In: Tistarelli M, Nixon MS (eds). Springer, Berlin, pp 1180–1189
25. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747
26. Sae-Bae N, Memon N (2014) Online signature verification on mobile devices. IEEE Trans Inf Forensic Secur 9(6):933–947. https://doi.org/10.1109/TIFS.2014.2316472
27. Sharma A, Sundaram S (2017) A novel online signature verification system based on gmm features in a dtw framework. IEEE Trans Inf Forensic Secur 12(3):705–718. https://doi.org/10.1109/TIFS.2016.2632063
28. Van BL, Garcia-Salicetti S, Dorizzi B (2007) On using the viterbi path along with hmm likelihood information for online signature verification. IEEE Trans SystMan Cybern Part B (Cybern) 37(5):1237–1247. https://doi.org/10.1109/TSMCB.2007.895323
29. Vargas JF, Travieso CM, Alonso JB, Ferrer MA (2010) Off-line signature verification based on gray level information using wavelet transform and texture features. In: 2010 12Th international conference on frontiers in handwriting recognition, pp 587–592. https://doi.org/10.1109/ICFHR.2010.96
30. Yeung DY, Chang H, Xiong Y, George S, Kashi R, Matsumoto T, Rigoll G (2004) Svc2004: First international signature verification competition. In: Zhang D, Jain AK (eds) Biometric authentication. Springer, Berlin, pp 16–22
31. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2016) Understanding deep learning requires rethinking generalization. arXiv:1611.03530
32. Zois EN, Alexandridis A, Economou G (2019) Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets. Expert Syst Appl 125:14–32
33. Ferrer MA, Diaz-Cabrera M, Morales A (2013) Synthetic off-line signature image generation. In: 2013 International conference on biometrics (ICB), pp 1–7. https://doi.org/10.1109/ICB.2013.6612969