Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-08728-6

Handwritten signature verification using shallow convolutional neural network

Anamika Jain[1] · Satish Kumar Singh[1] · Krishna Pratap Singh[1]

Received: 2 April 2019 / Revised: 21 December 2019 / Accepted: 1 February 2020 /
Published online: 07 April 2020

Abstract
Handwritten signatures are an undeniable and unique way to prove the identity of persons. Owing to the simplicity and uniqueness, it finds an essential place in the area of behavioral biometric. Signatures are the most widely accepted biometric trait by law enforcement agencies/personnel for verification purposes, especially in financial institutions, legal transactions, etc. and hence secured authentication becomes imperative. In the era of a digital age, numerous transactions are taking place, where handwritten signature verification is required by the agencies, e.g., banks, etc. In such scenarios, the process of signature verification, besides being accurate and secure, should be very fast, i.e., the real-time verification can be done. In this paper, we have proposed a convolutional neural network-based language-independent shallow architecture (sCNN(Shallow Convolutional Neural Network)) for signature verification. The proposed architecture is very simple but extremely efficient in terms of accuracy. A custom shallow convolution neural network is used to automatically learn the features of signature from the provided training data. Another contribution of the research work, which is the handwritten signature data collection for 137 subjects and 467 subjects, which are named as CVBLSig-V1 and CVBLSig-V2 respectively, has been reported in this paper. The performance of the proposed architecture has been evaluated on publicly available datasets, i.e., MCYT-75, MCYT-100, and GPDS, as well as CVBLSig-V1 and CVBLSig-V2. The performance was also compared with state of the art reported methods and shown improved, while considering the accuracy and equal error rate (EER) as performance metrics.

Keywords Behavioral biometrics · Offline signature Verification · CNN

✉ Anamika Jain
anamika06jain@gmail.com

Satish Kumar Singh
sk.singh@iiita.ac.in

Krishna Pratap Singh
kpsingh@iiita.ac.in

[1] Indian Institute of Information Technology Allahabad, Uttar Pradesh, India

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 [27]. Biometrics is categorized into two types, namely physiological and behavioral, where the physiological traits include the face, fingerprint, iris, palm-print, etc. [8, 9, 11, 39, 42, 43], whereas the signature, gait, speech, etc. [6, 44] comes under the behavioral biometrics [26]. Signatures have been used for ages for this purpose and are the preferred biometric modality. With the advancements of the technology, today, a large number of financial transactions are taking place, which needs to be verified against the genuinity of such transactions. Even today, most of the institutions are using conventional methods to verify the signatures of the persons. Mostly the traditional techniques are manual, and experience persons are required for the said purposes. The manual verification takes significant time, and it is an entirely subjective process that is highly dependent upon the expertise of the human verifier. Signatures are highly influenced by the mental state of the subject and hence become even more difficult for the verifier to verify. While designing the automatic state of art computer-based method for identification and verification, biometric plays a vital role [14].

A signature verification system verifies a person based on his/her handwritten content and is being used in government and private financial organizations to test the genuinity of the person as an owner of some document, record, or account [14]. Signatures, as biometric is popular because of their simple acquisition process, requiring fewer efforts. Moreover, people are familiar with the process [14]. Automatic verification of the signature can be done in online or offline modes. In an online process, the signatures are collected electronically, i.e., on the signature pad, and also contains some other auxiliary information, e.g., pen-up, pen-down, pressure, angle, etc. In offline mode, signatures are collected with the help of pen and paper only, and the auxiliary information is not present, hence fewer features and flexibility with the offline signature verification, making the process more challenging [19].

Offline Signature verification aims to verify an individual automatically based on one's signature and the outcome of a positive or negative decision [19]. Befooling the signature verification system by forgery is possible. Mainly two types of forgery have been reported in the literature, namely the random and skilled forgery [20]. In random forgery, all genuine signatures are considered as the forged signatures for all other users in the dataset while in skilled forgery, a skilled person practices for the signature of a particular person's signature.

The rest of the paper has been divided into Six sections. In Section 2, there is a discussion about the related state of the art methods. 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.

2 Related work

Lots of research work has been done in recent past and reported in literature [2–5, 12, 13, 18–20, 22, 30, 32, 34, 37, 38, 41]

Napa et al. [37] proposed a histogram-based method to verify a signature by using mobile devices. Signatures that were collected through a mobile device can be represented as a set of histograms. Before histogram extraction, signatures are normalized in the time domain, and strokes of the signature are concatenated. Histogram features can capture the required attributes and can also preserve the attribute relationship. Histograms can be used as features for recognition in many fields, e.g., object recognition, signature verification in offline mode. Depending on the length of the signature, a feature vector is calculated and quantized in O(n) time, where n is the length of the signature. The length of the feature vector is not uniform for all the signatures, posing the problem while matching. Manjunatha et al. proposed a writer-specific online signature verification algorithm [30]. For each writer, dedicated features and a classifier has been learned. In the later stage of the process, the EER of all classifiers specific to the user is calculated, and the classifier for each trial is selected based on minimum EER. The classifier has the highest frequency among the trials considered as the best classifier. Publicly available dataset MCYT-100 has been used in their experiment. Time complexity is higher in their approach because they have used different classifiers for each writer.

Abhishek et al. proposed an online method for signature verification that uses features of the Gaussian Mixture Model (GMM) [38]. These features are used for alignment of signature that was derived from dynamic time warping (DTW). DTW is a measure that shows the similarity between reference signature and test signature. Warping path score can be obtained by taking the similarity between histograms that are generated from the cost matrix. Warping path score and score of DTW are fused using the sum rule for validating the authenticity of signatures. The objective of the fusion of DTW and warping path score is to increase the discrimination between genuine and forged signature. MCYT-100 dataset has been used in their experiment. Signatures suffer from a major problem called intra-class variation, so there is a need to improve the signature verification system. Songxuan et.al. [28] came up with an idea to use a recurrent neural network(RNN) to verify a signature. Authors aim to minimize intra-class variation between genuine and forged samples. The authors have proposed a descriptor named Length Normalized Path Signature (LNPS) to preprocess the signatures. With this descriptor signature path is normalized. LNPS descriptor is scale and rotation invariant. The authors evaluated their method on publicly available dataset SVC-2004. RNN needs a large dataset to train and also requires high computational power. Negative examples are also necessary to train RNN, but it cannot be possible for each dataset because some dataset does not have forged sample.

In [25], authors have presented an online verification method that uses curvature and torsion features. These features are described in [25]. Signature has been considered as the spatial curve. On the basis of peaks and valley points in the signature curve, the curvature and torsion features have been calculated, and an eight-dimensional feature set has been formed. Based on the Hausdorff distance, verification of the test sample with respect to the reference sample has been performed.

In offline mode, there is the absence of dynamic information like pressure, angle, pen-up pen-down, etc. The lack of information makes the offline verification process a challenging task as compared to the online methods. The online mode of the signature has other auxiliary information to support the feature vector of signatures. There is the need to work on an offline mode of the signature because many organizations like banks support the offline mode of signature. So researchers have to bend the literature towards offline signature verification.

Hafemann et al. has proposed a Convolutional Neural Network-based offline signature verification method [20]. They investigated that sometimes handcrafted features have no or very less resemblance with a signature in the verification process. The authors reported that CNN gives relevant features than handcrafted features. Publicly available datasets GPDS, PUC-PR are used to evaluate the performance of the method. They have stated that their approach achieved the lowest EER, but there is an imbalance between False Positive Rate (FPR) and False Negative Rate (FNR). Later, the authors extended their work [19] and analyzed the deep learned features that were extracted in [20]. They have explored different architecture and reported the lowest EER in the literature on the GPDS dataset. In 2017 Sounak et al. reported a Siamese convolutional network for signature verification [12]. Siamese network has two identical networks with shared weights, same parameters, and configuration that take different image pair as an input. These two networks are connected using contrastive loss function. According to the loss function similarity score between two images was calculated by using Euclidean distance, at the time of back-propagation, parameters updated in a similar manner in both the networks. The network has been trained with the objective to reduce the distance between genuine to genuine pair and increase the distance between genuine to imposter pair. Authors have evaluated their method on profoundly different datasets, e.g., BHSig260, GPDS, CEDAR. This method requires an ample amount of time and high computational power because two networks trained simultaneously.

Vargas et al. reported an offline mode based signature verification method [41]. This method works on wavelet and textural statistical features on local and global image levels. Some of the features extracted in this method are based on the ink type used while capturing a signature. The gray-level co-occurrence matrix (GLCM) was created based on local features. Support Vector Machine (SVM) has been used for training. They have performed training and testing ten times to achieve optimal results. The false-positive rate (FPR) reported was quite high, 28% for the MCYT dataset.

YILMAZ et al. [5] proposed a method that uses two-channel CNN for signature verification. Authors have used CNN for feature extraction and verification. The Network accepts signatures as a pair, the first channel takes a reference signature, and the second channel takes the query signature. They have maintained the performance while decreasing the dimensionality up to 200 at the last locally connected layer. They successfully reduced the EER to 4.13% on the GPDS dataset. Later they have used score level fusion, and EER is decreased further to 1.73%. HADJADJI in [18] has proposed a writer dependent method that used curvelet transform (CT) and principal component analysis (PCA). They have used curvelet transform for generating of the features, and PCA is for reducing the feature size produced by CT. To combine the results of multiple scores generated by the system, they have used Choquet fuzzy integral. They have conducted their experiments on the GPDS dataset and reported the accuracy of 94.96%. Manabu et al. [32] proposed a method that combined featured from fisher vector and fused KAZE [1] features. KAZE features have extracted from the foreground and background signature images. They have shown that KAZE features have good results in signature verification. They have also discussed that the fisher vector gives a more accurate distribution of the characteristics per writer in the spatial domain. In their experiment, they have used the MCYT-75 dataset. Dutta et al. proposed a method that uses the local feature and global statistics of the local features [13]. In their approach, Histogram of Oriented Gradients (HOG) based features have been used, and weight has been applied to the local features based on the height of the signature. They used the concept that local features play a significant role in the geometry of signature, and by using these features, one can distinguish between genuine and imposter signatures. To advert, the access size of the signature, authors created a local pairwise feature, and this feature represented as a collective descriptor. They have tested their method on GPDS and CEDAR dataset.

Authors in [34] used the most robust texture extraction algorithm, namely local binary pattern (LBP) and uniform local binary pattern (ULBP), to extract the features of signature images. For similarity measure, authors have used the nearest neighbor algorithm. They found that there was no exceptional difference in the results obtained by LBP and ULBP on the publicly available dataset BHSig260, GPDS100. Neural Network has one constraint the all the inputs are given to the network must be of the same size. Authors in [22] have used the concept of spatial pyramid pooling (SPP) [24] to deal with the variation in the signature size. SPP is used to get the fixed vector size from the different sizes of the images. They have also investigated that by using fine-tuning on SigNet [21] performance of the system increased.

Authors in [2] have presented a method that uses freeman chain code(FCC) for the representation of the data. From the longest connected path, the FCC was extracted. A detailed description of the FCC is presented in [2]. Later the FCC was divided into various(four, eight, sixteen) number of parts. Authors have also calculated six global feature that has been described in [2]. Verification of the signature images has been done by using the K-nearest neighbor. They have performed their experiment on the MCYT dataset. Signature images have significant intra-class variation. To deal with the problem, the authors in [7] proposed a method based on run-length features extraction. The signature image was converted into a binary image to calculate the run-length features. In the binary image, the black pixels were considered. Run can be defined as the chain of connected points in a particular direction; all the pixels are having the same level of intensity. Authors have used one-class SVM for classification purpose. Authors in [31] tried to improve the performance of the automated signature verification. In this method, the authors used the bag of visual words(BoVW) and vector of locally aggregated descriptors(VLAD). A detailed description of BoVW and VLAD has been presented in [31]. Along with the BoVW and VLAD, KAZE features have been extracted from the strokes of the signature image. For the classification, SVM was used.

Despite the advancements in the field of signature verification, the system fails to verify the signature images when the inter-class similarity is higher.

Offline signature verification is widely used by many organizations and is a challenging problem because of the non-presence of much auxiliary information, like pressure, pen up, pen down angle, etc. Moreover, accurate signature verification architecture development is the need of the hour, and this motivated us to work on an offline signature biometric verification. The objective of this manuscript is to design a very simple model that outperforms the state of the art methods in terms of accuracy. The convolutional neural network has proved its superior feature extraction process over most of the traditional methods. This inspired us to use CNN in our proposed research work reported in this paper. We planned to design a simple yet effective model that can fulfill our requirements. Since the signature images are not the texture-rich images and only lines, curves, and edges are present; hence, very deep features are not present. In the proposed work, we have used shallow neural network architecture for signature verification, because on increasing the layers, accuracy saturates for some dataset and for some dataset performance degrades. This work contributed in two ways. Firstly we have created an offline signature dataset (CVBLSig) in two sessions. Sessions one and two are having 137 and 467 individuals, respectively. Detailed information about this dataset is present in Section 5-A. Second, we have developed sCNN (shallow Convolutional Network) that contains less number of layers yet performed well.

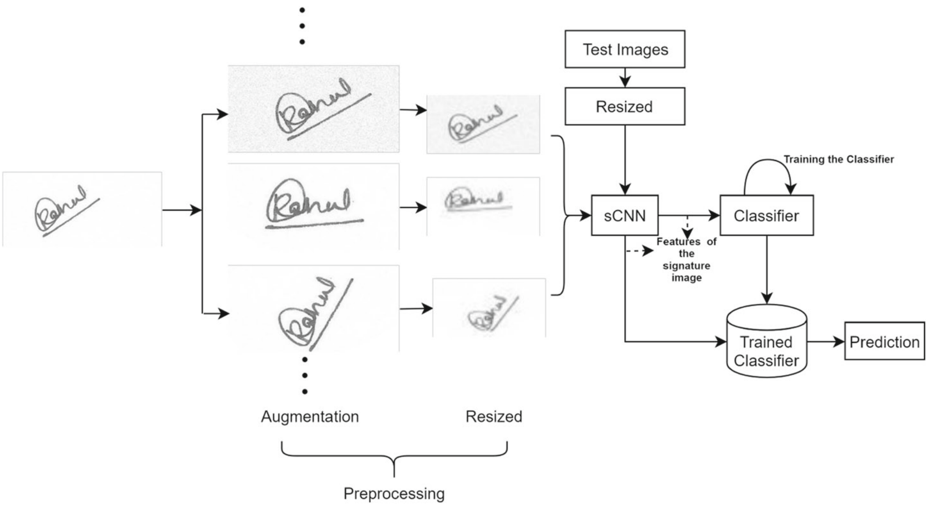

Fig. 1 Methodology of Proposed work

3 Proposed method

3.1 Methodology

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 Extracted features from sCNN are fed to a softmax classifier. At the end of the training and classification phase, we have a trained model, and with the help of that trained model, the test signature image is verified based on the threshold. 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, sometimes handcrafted features are not sufficient to recognize an individual, gives us the motivation to use such a model that will find the relevant features from the signature image. These features will help us to identify a person correctly.

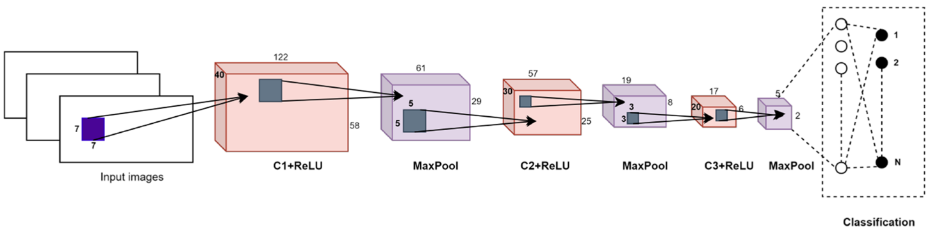

Fig. 2 Network Architecture of Shallow Convolutional Neural Network

To extract the relevant features of the signature, we have used CNN as a base architecture in this proposition. CNN is a layered architecture, which has several layers like convolutional, pooling, and fully connected layers. The detailed description of the CNNs has been presented in the [10].

The proposed model has three convolutional layers and one fully-connected layer. The network architecture of the proposed network is shown in Fig. 2. Raw signature images are given to the input layer of the network, which takes input of size 64x128. The input layer is connected to the first convolutional layer (C_1). The input image will be filtered by forty random filter kernels of size 7x7 while using the stride one. According to the number of filters and channels, the convolutional layer will generate the respective feature maps. After the first convolutional layer, we have forty feature maps, each of size 58x122. The feature maps generated by C_1 are being fed to the activation function Rectified Linear Unit (ReLU). ReLU introduces some non-linearity to the features to overcome the problem of vanishing gradients. Activation function of ReLU is: $R(x) = max(0, x)$. Where R is a ReLU activation function, and x is feature values from the feature maps. This function will check the feature space and remove the negative feature value if any. After the non-linearity layer, there is the pooling layer ($Maxpool_1$). The purpose of the pooling layer is to reduce the size of the output from the ReLU layer and to introduce the non-linearity in feature space. We have used max-pooling in our proposed architecture. In this layer, the window size of 2x2 is used with stride two over the image and takes maximum value from the window. After the maxpool layer size of each feature map became 29x61. Feature maps generated by the $Maxpool_1$ layer will be filtered by thirty filters with kernel size 5x5 using stride 1 using the second convolutional layer (C_2). We have maintained symmetry in the proposed architecture, i.e., convolution layer followed by non-linearity followed by the maxpool layer. Feature maps generated by C_2 are further passed to the ReLU, and then $Maxpool_2$ layer with window size 3x3 and stride 3. $Maxpool_2$ is connected to the third convolutional layer C_3, which has twenty kernels of size 3x3 with stride 1. Feature maps of C_3 will be fed to the ReLU, and $Maxpool_3$ will be applied with window size 3x3 with stride 3. After $Maxpool_3$, we will have a fully connected layer that layer will contain some specific neurons according to the number of classes in the database. Up to a fully-connected layer, there are 41370 parameters in the network. According to [46], for a two-layer neural network, $2n + d$ parameters are sufficient to express the dataset with n samples having d dimension. Parameters in the proposed network are enough to label the dataset used in the proposed work as the required number of parameters are within the range of the available parameters. For classification, we have used a softmax classifier, that has a cross-entropy loss function [29]. That loss function is interpreted by (1).

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

In k_j, k signifies class score, and j signifies the mean of the vector k. The softmax classifier gives the probability of each test sample belonging to all classes. Initially, values of filters are assigned and generated randomly by a Gaussian distribution, and at the time of back-propagation, these weights get adjusted. We have used a stochastic gradient descent optimizer with momentum (SGDM) 0.9 [36].

Table 1 summarizes the parameters, output shape of the activations, and the number of parameters. In Table 1, F represents the filter size, and S represents the stride. The second column in the table represents the shape of a convolutional and pooling layer activation map as M x W x H, where M is the number of filters and W, H is the height and width of the

Table 1 Network architecture of sCNN

Layer	Activation shape	Number of learnable parameters	Total count
Input Layer	$64 \times 128 \times 3$	0	0
Convolutional 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) ReLU	$20 \times 6 \times 17$	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			

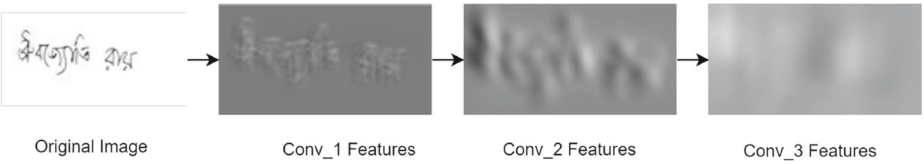

Fig. 3 Visualization of the features at different layers on Bengali dataset

activation map, respectively. The third column in the table contains the number of weights and bias parameters.

4 Experiment

This experiment aims to design a model that can extract the relevant features from the signature image. The extracted feature set should be robust enough to discriminate between the two different samples. The proposed architecture extracts the feature set that can distinguish between the different samples. The one feature map and one filter after the different convolutional layers have been shown in Figs. 3 and 4, respectively. All the feature maps and filters have been shown in Appendix A Figs. 17, 18, 19, 20, 21 and 22. It can be noticed from Fig. 3, that through the starting convolutional layers, the edges of the signature image have been extracted. After two convolutional operations, there are no such details present in the feature map. The Fig. 3 justifies the uses of the shallow network in case of the signature images.

With extensive experiments, we have fixed that three convolutional layers are sufficient to works with signature images. We have tried to vary the number of convolutional layers and found that on increasing the number of convolutional layers, there is no effect on the performance of the system.

4.1 Dataset

The proposed architecture has been trained on various publicly available datasets, namely MCYT-100 [33], MCYT-75 [17, 33], and GPDS dataset [15, 16], SVC-2004 [45], and CVBLSig-V1, CVBLSig-V2 which have handwritten signatures images where English is

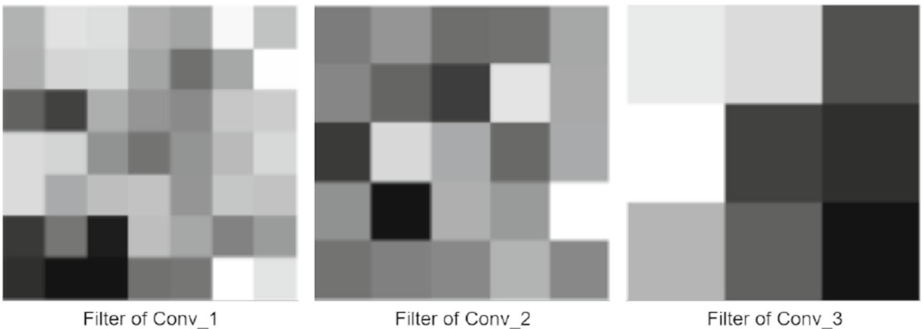

Fig. 4 Visualization of the filters at different Conv layers on Bengali dataset

the basic language used. To show the wide applicability of the proposed method, two other datasets, namely BHSig260 Hindi [34], and BHSig260 Bengali [34] has also been considered where the basic language used is Hindi and Bengali respectively. The details about the various signature data sets considered by us have been elaborated in the following subsections.

4.1.1 MCYT-100

The MCYT-100[1] was collected on WACOM ink tablet and consists of 25 genuine and 25 forged signatures for 100 individuals, so total 5000 signature images only [17, 33]. This dataset contains other auxiliary information like an azimuthal angle, pressure, etc. But we have not considered these pieces of information in our experiments because; these are not required for the offline signature verification process.

4.1.2 MCYT-75

Dataset MCYT-75[2] is a variation of MCYT-100 but contain offline mode of signature of some individuals of MCYT-100. This dataset has 75 individuals, and each has 15 genuine signature and 15 forged signature [17, 33].

4.1.3 BHSig260

This dataset[3] has a signature in two regional languages, Bengali and Hindi. In the Bengali dataset, there were 100 signers, and each signer has 24 genuine signature and 30 forged signature [34]. There were total 24 x 100=2400 Genuine signature and 30*100=3000 forged signature. In the Hindi dataset, signatures of 160 individuals were collected, and each person has 24 genuine and 30 forged signatures.

4.1.4 GPDS

This dataset has an offline signature of the individuals. GPDS[4] dataset has 4000 individuals. GPDS is the largest signature dataset present to date. GPDS Dataset has versions like GPDS-160, GPDS-300, GPDS-960. When we use the GPDS-300 dataset, it signifies that out of 4000 individuals, we are choosing the first 300 for the experiment. Dataset also contains forged signatures that are collected by skilled persons. Each has 24 genuine images and 30 forged images. Since the dataset is collected in offline mode, there is no information about the angle, pressure, etc. [15, 16].

4.1.5 SVC-2004

SVC-2004[5] [45] was the first international signature verification competition to benefit to the field of the signature verification process. This dataset collected in two tasks. Each task contains 100 individual, but authors have published 40 individual's signature because the

[1] http://atvs.ii.uam.es/atvs/mcyt100s.html
[2] http://atvs.ii.uam.es/atvs/mcyt75so.html
[3] https://drive.google.com/file/d/0B29vNACcjvzVc1RfVkg5dUh2b1E/view
[4] http://www.gpds.ulpgc.es/downloadnew/download.htm
[5] https://www.cse.ust.hk/svc2004/download.html

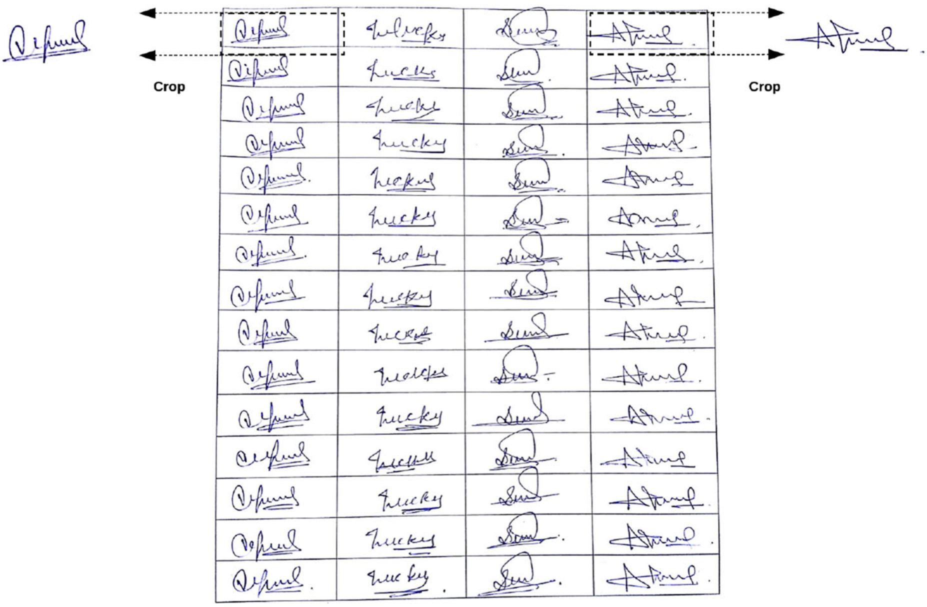

Fig. 5 Setup of Signatures form CVBLSig Dataset

remaining 60 includes the same person but with different ink and strokes. WACOM Intuos tablet was used for signature acquisition and stored in a separate text file. Each person has 20 genuine and 20 forged signatures. These test files contain the sequence of points in the signature, and the first line of each text file will have the number of sequence points of signature. Task 1 and Task 2 signature files have X-coordinate, Y-coordinate, time-stamp, button status. Additionally, task 2 signature file will have azimuth, altitude, and pressure.

4.1.6 CVBLSig

As a contribution to the research community, we have created an offline signature dataset CVBLSig.[6] This dataset has been collected on a sheet of paper. Figure 5 shows the setup that was followed to collect the signature of the CVBLSig dataset. Each individual is required to sign in one column of the sheet. Maximum four people can sign one sheet of the paper. These sheets are scanned using a digital scanner, and signatures are being cropped. CVBLSig is collected in two sessions; session one contains 137 (CVBLSig-V1) individual, and session two has 467 (CVBLSig-V2) signers. Each signer in CVBLSig-V1 has a minimum of 20 images, and in CVBLSig-V2, there will be 15 images from each signer. There are no forged signatures present in this dataset, but we are also planning to include the forged signature of each individual. Since this dataset is collected in offline mode so, no supportive information like pressure, angle, pen-up, pen-down is present.

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.

[6] https://cvbl.iiita.ac.in/dataset.php

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

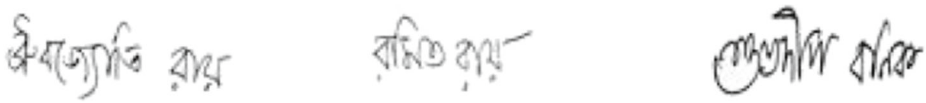

Fig. 10 Sample Signature of BHSig260 Bengali Dataset

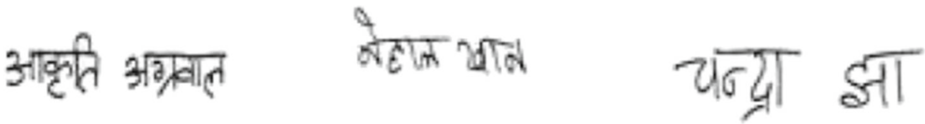

Fig. 11 Sample Signature of BHSig260 Hindi Dataset

Fig. 12 Sample Signature of SVC-2004 Dataset

4.2 Pre-processing

We know very well that being the behavioral trait, the signature biometric poses substantial intra-class variations. The appearance of the handwritten signatures depends upon many intrinsic (mood, mental state and cognitive behavior of the person), and extrinsic factors (Background, Pen, Ink Color, physiological style, etc.). Due to the intrinsic and extrinsic effects, the signature images captured, which are mostly aligned horizontally, there can be variation in size, angular diversion, orientation, the thickness of lines, background paper color, etc. So a lot of possibilities for the captured signatures, and hence it is required to have the signatures with all possible variations in the training dataset for the better generalization and accuracy. Since most of the publicly available datasets, i.e., MCYT-75, MCYT-100, and GPDS, BHSig260, and SVC-2004 are not having a large number of samples per object, so we have performed several data augmentation, i.e., clockwise and anticlockwise rotation by 10 and 30 degrees, shear by the factor 0.6, added Gaussian noise (zero mean and variance 0.01) and changed the aspect ratio by 3/5. The detailed description of the datasets has been given in Section 4.1. The signatures images in the different datasets are of different sizes, so the resizing to standard 64x128 pixels have been done. The result of preprocessing has been shown in Fig. 13 on CVBLSig-V2. These preprocessed images are fed to the proposed sCNN for further processing.

4.3 Parameters

Table 2 shows the parameters that are used to train the network shown in Fig. 2. To regenerate the results of this experiment, we have provided all the parameters used in the experiment. We have used a stochastic gradient descent optimizer with momentum [36]. SGDM has momentum, learning rate, gradient threshold as a parameter. We have used a non-adaptive learning rate equal to 10^{-4}. Momentum is 0.9 with gradient threshold 7. If

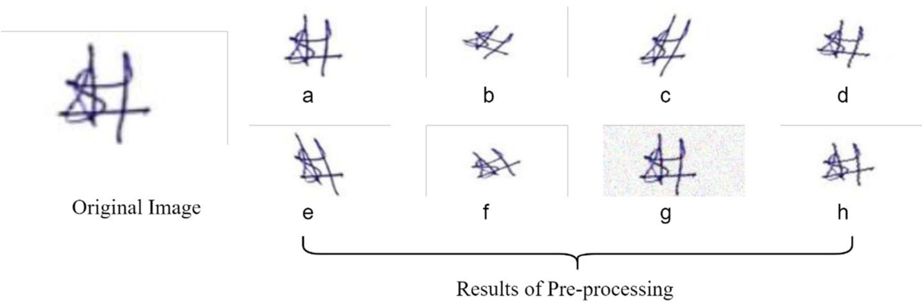

Fig. 13 Results of Pre-processing on CVBL-V2 **a.** Change in Aspect ratio **b.** Rotation by 30 degrees in clockwise direction **c.** Right Shear **d.** Rotation by 10 degrees clockwise **e.** Left Shear **f.** Rotation by 30 degrees in anticlockwise direction **g.** Gaussian Noise **h.** Rotation by 10 degrees anticlockwise

Table 2 Traning parameters

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

chosen the small value of momentum, for a new sample, the optimizer will fluctuate a lot between local minima and maxima. Other parameters affect the performance of the network, i.e., mini-batch size, number of epochs. We have considered batch size to 32, and the experiment has been completed after 150 epochs.

4.4 Experimental setup

Experiments have been performed on a computer workstation having the following configuration. Intel(R) Core (TM) i7-7820 x CPU @3.60 GHz, 16 GB RAM and have three NVIDIA 1080 Ti of 11 GB each. For this experiment, we have used MATLAB 2018a.

5 Results and discussion

This work focuses on designing a signature verification system using the convolutional neural network. For this experiment, we have divided all datasets in different ratios. These distributions of data can be represented as the test images: train images [2:8, 3:7, 4:6, 5:5]. On the proposed model, these distributions of the dataset are trained and tested.

Figure 14 shows the comparison accuracy between these distributions on our model. It is clear from Fig. 14 that among all distribution, we have received higher accuracy when we use the dataset in the distribution [2:8]. MCYT-100 gave the highest accuracy of 98.40%

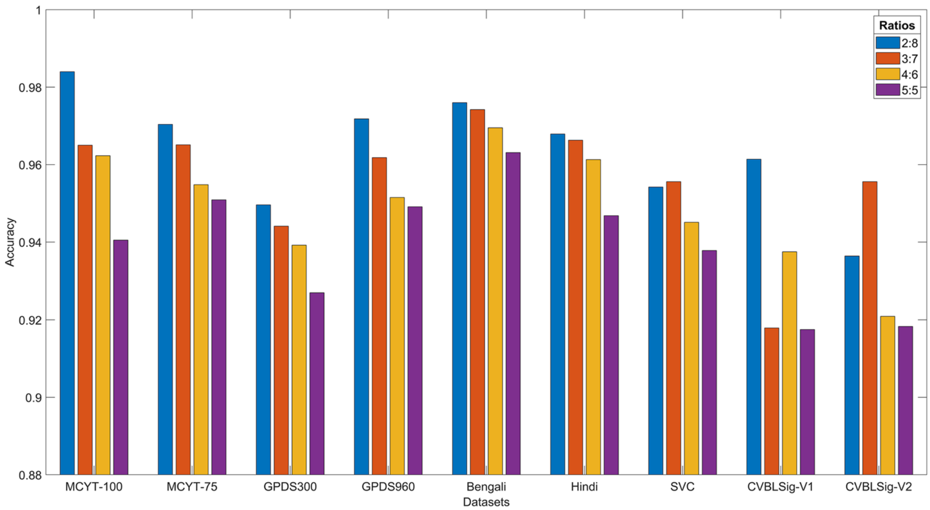

Fig. 14 Accuracy vs Data Distribution for sCNN

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
CVBLSig-V1	96.78
CVBLSig-V2	93.15

among all distributions when we divided the test and train dataset into 2:8 ratios. From Fig. 14 it is evident that large numbers of training samples are required for getting better accuracy and generalization compared to the test samples.

Table 3 summarizes the accuracy of the proposed methods while considering a variety of publicly available and self-created data sets. In all the experiments, the proposed network is trained from scratch while randomly selecting five test images and remaining images as the training images. From the Table 3, it is clearly visible that achieved recognition accuracies are 99.40%, 98.93%, 96.87%, 97.19%, 97.00%, 96.78%, and 93.15% on MCYT-100, MCYT-75, GPDS-300, GPDS-960, SVC-2004, CVBLSig-V1, and CVBLSig-V2 datasets respectively. We also performed this experiment on the regional dataset BHSig260, which contains two languages Hindi and Bengali. We have achieved accuracies of 98.40%, and 97.12% on BHSig Bengali and BHSig Hindi dataset, respectively. On the MCYT-100 dataset, our model gives the highest accuracy, because the sample per class is higher in MCYT-100 dataset after augmentation. It can be noticed that when we have a large number of training samples, the model generalizes well, and performed better. The convergence of the loss and accuracy with respect to the iterations has been shown in Fig. 15 on all the datasets.

We have also tested the proposed architecture with the two different optimizers, i.e., rmsprop and adam. Performance for the same is shown in Table 4. It can be noticed from the Tables 4 and 3, SGDM works better on the signature domain as compared to the adam and rmsprop.

In literature, most of the authors consider EER as a performance parameter for MCYT, SVC dataset, and accuracy for GPDS, BHSig260 datasets, respectively. So for a fair comparison, we have considered EER as a performance parameter for MCYT and SVC datasets and accuracy for GPDS and BHSig260 datasets. Table 5 shows the EER on the proposed architecture on MCYT. EER can be defined where the false-negative rate (FNR) and the false positive rate (FPR) are equal. When we have a large number of classes, it is not always possible to get the same value of the FNR and FPR; therefore, we have taken a difference of 0.01 between FPR and FNR to find the similarity between FPR and FNR. On the occurrence of multiple EER points, the average will be calculated.

It is clear from the Table 5 that our model gives better performance than [14, 21, 22, 25, 28, 30, 32]. Manjunatha et al. has reported EER of 0.80% [30] by using the combination of different classifiers (Neural Network and PCA). Authors in [30] have also increased the complexity by using different types of classifier for each writer. In [21], authors have used five convolutional layers (AlexNet architecture) and two fully connected layers and

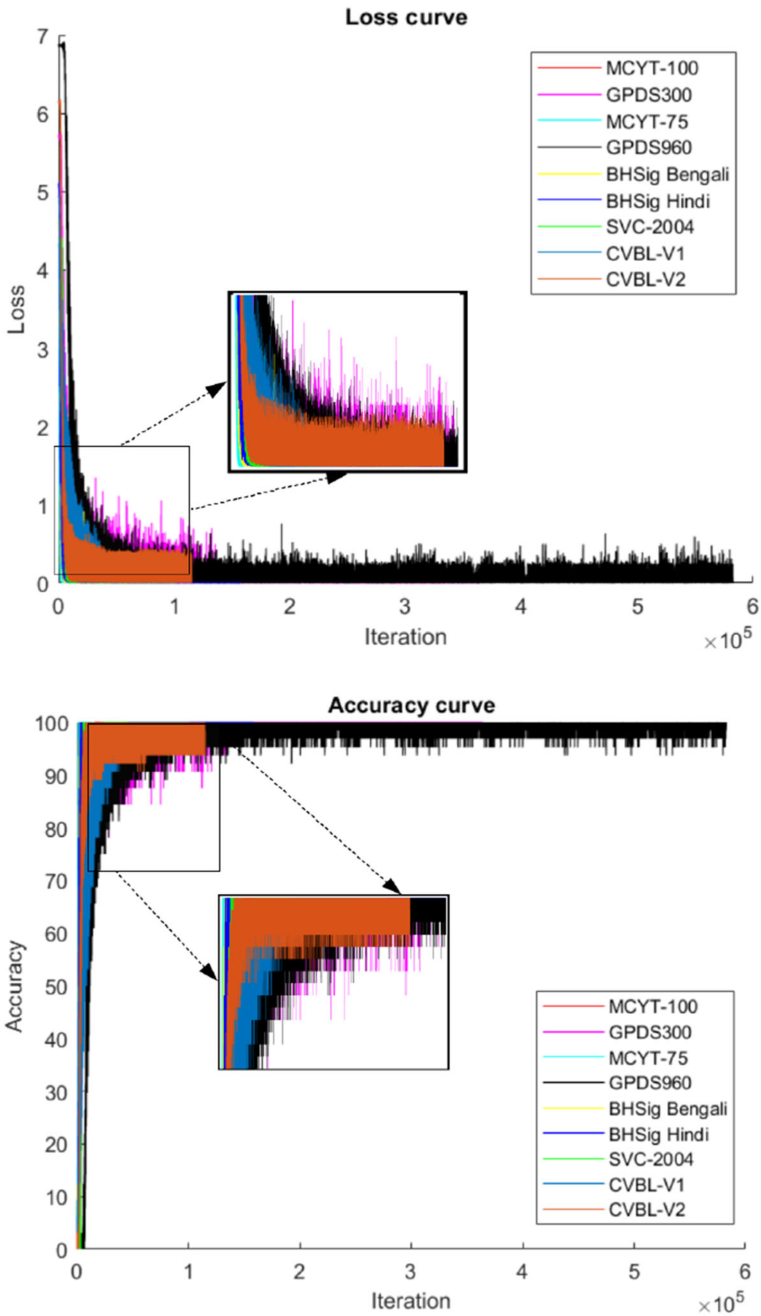

Fig. 15 Convergence of Loss and Accuracy Curve

reduced EER to 2.87% on the MCYT-100 dataset. On the other hand, we have used only three convolutional layers and one fully-connected layer, and we have managed to reduce EER to 0.2% on the MCYT-100 dataset. As compared to [21], the number of parameters in

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

the proposed method is less. Before fully connected layer proposed method has 41,370 and [21] has 3,747,200 parameters. Due to the less number of layers and parameters, this work can be used in real-time applications and hand-held devices.

In Table 5, there is also a comparison of our method with the state of art methods on the MCYT-75 dataset. sCNN performed better in comparison to the most state of the art method [14, 22, 23, 31, 32, 47]. Authors in [14] used GLCM and wavelet transform along with SVM to train and test their method. In comparison with the [14] proposed model exhibits better performance, because of the quality of extracted features by CNN are superior over GLCM

Table 5 EER Comparison on MCYT dataset

Database	Method	EER
MCYT-100	W-Dependent feature [30]	7.75
	W-Dependent feature (NN+PCA) [30]	0.80
	Signet(SVM) [21]	2.87
	DTW and GMM [28]	2.12
	Curvature feature+Torsion features [25]	1.20
	sCNN(Proposed)	**0.2**
MCYT-75	GLCM[14]	2.30
	GLCM+WT[14]	2.44
	FV with fused KAZE features[32]	5.47
	Fixed size representation[22]	3.64
	BoVW with KAZE features[31]	6.4
	WI using asymmetric pixel relation[47]	3.5
	sCNN(Proposed)	**0.4**

Table 6 Recognition accuracy comparison on GPDS dataset

Database	Method	Accuracy%
GPDS-300	Compact Corelated Features [13]	88.79
	Curvlet Transform [18]	94.96
	SigNet(Siamese) [12]	76.83
	sCNN(Proposed)	**96.87**
GPDS-960	Compact Corelated Features [13]	73.67
	SigNet(Siamese) [12]	77.76
	One-class SVM [7]	95.68
	sCNN(Proposed)	**97.19**

and Wavelet features. In [23], authors have reduced the EER 3.4% by using CNN along with the SPP. SPP provides the fixed-size vector before fully connected layer. On the MCYT-75 dataset, we have managed to reduce the EER to 0.4%. One reason behind reduced EER is the higher achieved accuracy and the lowest FNR and FPR for the proposed model.

For the proposed architecture the accuracy is high for GPDS dataset with compare to state of the art methods [7, 12, 18, 34][refer Table 6].

Authors in [12] have used the Siamese network to distinguish the genuine and forged signature and achieved the recognition accuracy of 76.83%, and 77.76% on GPDS300 and GPDS960, respectively. In [13], authors have used correlated features to distinguish the inter-class samples and achieved an accuracy of 88.79%, and 73.67% on GPDS300, and GPDS960 dataset respectively. Table 7 shows the comparison with the state of art method on the BHSig260 dataset. It is visible from the Table 7 that on BHSig260 dataset, our model performed much better than the state of art methods [12, 13, 34].

Table 8 shows the results of sCNN on SVC-2004 dataset. We have achieved the lowest EER among the state of the art methods.

ROC describes the classifier's discriminative power, and that is not dependent on the distribution within the classes. Figure 16 shows the Receiver Operator Characteristics (ROC) curve of the dataset considered by the proposed model. Since True Positive Rate (TPR) and False Positive Rate (FPR) calculated for one class, we have calculated ROC for each class and take an average of all. From Fig. 16, it can be noticed that our model outperformed on dataset GPDS960 because the area under the curve is highest for GPDS960. From Fig. 16, the AUC is maximum for GPDS960, which is having 960 objects, i.e., a large number of persons. After augmentation, the total number of samples are quite high compared to other datasets. So total samples are varying for different datasets, whereas test samples are the same. So it can be inferred that when a large number of samples are present in the dataset, and the ratio of train and test set samples quite high, the proposed model exhibits good performance.

Table 7 Recognition accuracy comparison on the BHSig260 dataset

Database	Method	Accuracy%
BHSig Bengali	LBP and ULBP [34]	66.18
	Compact Corelated Features [13]	84.90
	SigNet(Siamese) [12]	86.11
	sCNN(Proposed)	**98.40**
BHSig Hindi	LBP and ULBP [34]	75.53
	Compact Corelated Features [13]	85.90
	SigNet(Siamese) [12]	84.64
	sCNN(Proposed)	**97.12**

Table 8 EER Comparison on SVC-2004 dataset

Method	EER%
DTW [45]	5.50
DTW [35]	3.38
HMM [40]	4.83
RNN+LNPS [28]	2.37
sCNN(Proposed)	**1.01**

6 Conclusion

The recent success of CNNs in classification and recognition of images in nearly all the domains of computer vision research inspired the development of the proposed model. In this paper, simple and effective convolutional neural network-based language-independent signature verification architecture has been proposed. The proposed model is quite simple in terms of the number of basic layers (Conv and Pool) in contrary to the other state of the art methods; hence the weight parameters to be optimized are lesser in number. The proposed sCNN has less number of layers and parameters; hence, the time taken in training and testing by proposed method will be less. The sCNN achieved better performance in terms of accuracy and EER compared to the other State of Art signature biometric methods. In this paper, we have investigated the random forgery and achieved a marked accuracy of 98.93% and 96.87% on MCYT-75 and GPDS- 300 datasets, respectively. EER is reported 0.2% on the MCYT-100 dataset using sCNN, which is the lowest among all the results as reported in various literature, as shown in Table 5. We have received an accuracy of 96.78% and 93.15% on CVBLSig-V1 and CVBLSig-V2, respectively. On the basis of the accuracy, it can be deduced that the collected CVBLSig-V1 and CVBLSig-V2 datasets are quite challenging compared to the other publicly available datasets. The dataset, as reported through this paper, may also be used by various researchers for bench-marking the architectures and algorithms used for signature biometric recognition. The accuracy achieved by the proposed architecture is 98.40% and 97.12% for Bengali and Hindi signature datasets

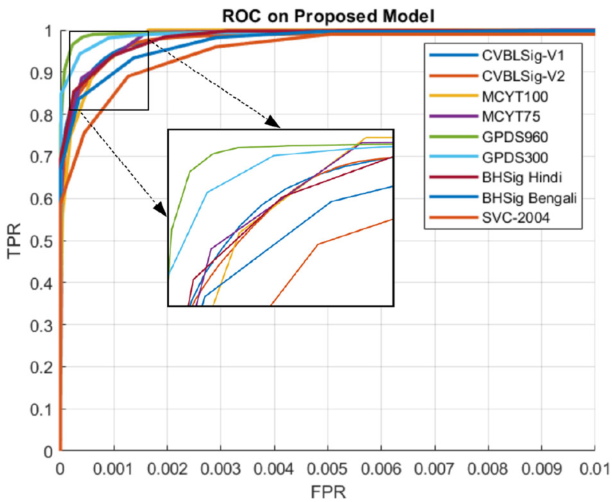

Fig. 16 ROC on sCNN for Different Dataset

(BHSig260) compared to the other state of the art methods and architectures; hence the proposed model is language independent and can be used for various regional languages too. It is also concluded that for the task of offline signature verification very deep network may not be suitable, and shallower networks may produce better or similar accuracy levels with the reward of fewer parameters and reduced space-time complexity.

The signature dataset does not have a large number of samples, so when using the deep learning methods, the model can over-fit. In this work, we have tried to minimize the over-fitting by data augmentation. In the near future, we will reduce the chances of over-fitting without using data augmentation. Also, this work addressed the random forgery of the signatures, and we will work on the skilled forged signature images on the proposed network. The program of the proposed work will be available on request.[7]

Acknowledgments This research is conducted in Indian Institute of Information Technology Allahabad and supported by the Ministry of Human Resource and Development, Government of India. We are grateful to the support of NVIDIA Corporation. NVIDIA Corporation has donated TITANX(PASCAL) GPU with 3584 CUDA cores.

Appendix

A Feature Maps and Filters after each Convolutional Layer

In Figs. 3 and 4 we have shown only one filter and one activation map of each convolutional layer. The detailed features and filter from each convolutional layers has been shown in Figs. 17, 18, 19, 20, 21 and 22.

Fig. 17 Visualization of all the feature maps after first convolution layer

[7] anamika06jain@gmail.com, rsi2016005@iiita.ac.in

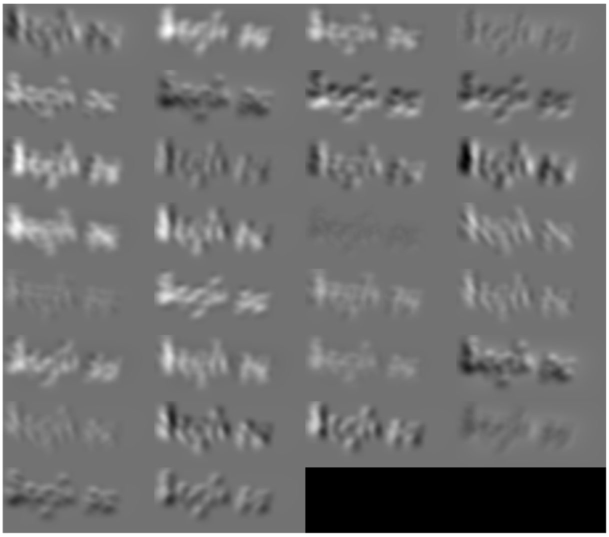

Fig. 18 Visualization of all the feature maps after second the convolution layer

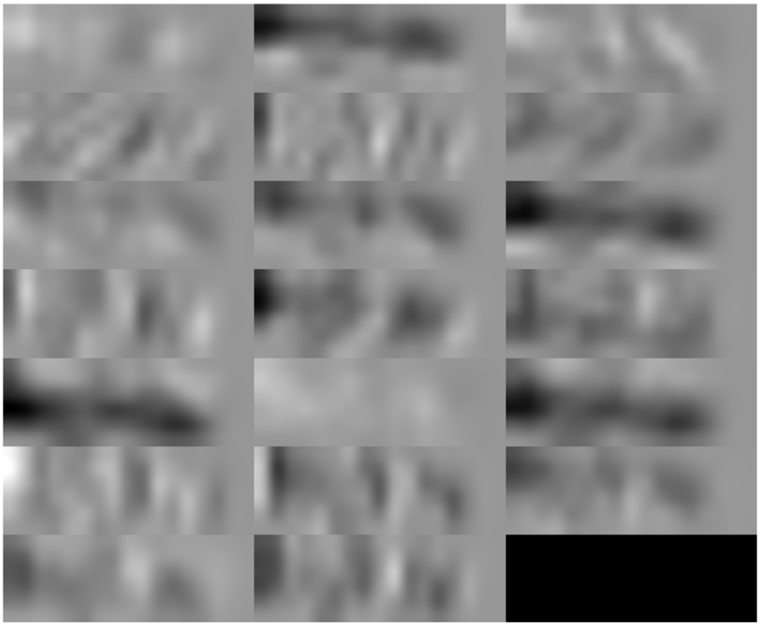

Fig. 19 Visualization of all the feature maps after third convolution layer

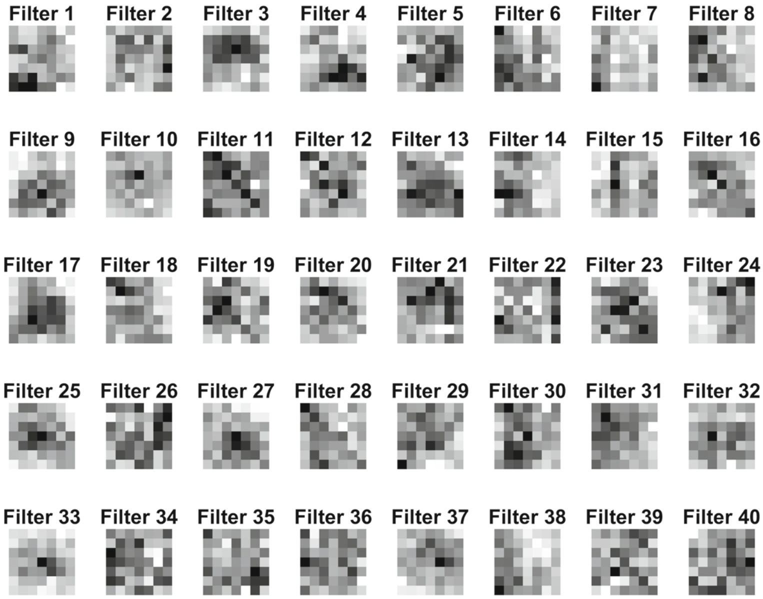

Fig. 20 Visualization of all the Filters of first convolution layer

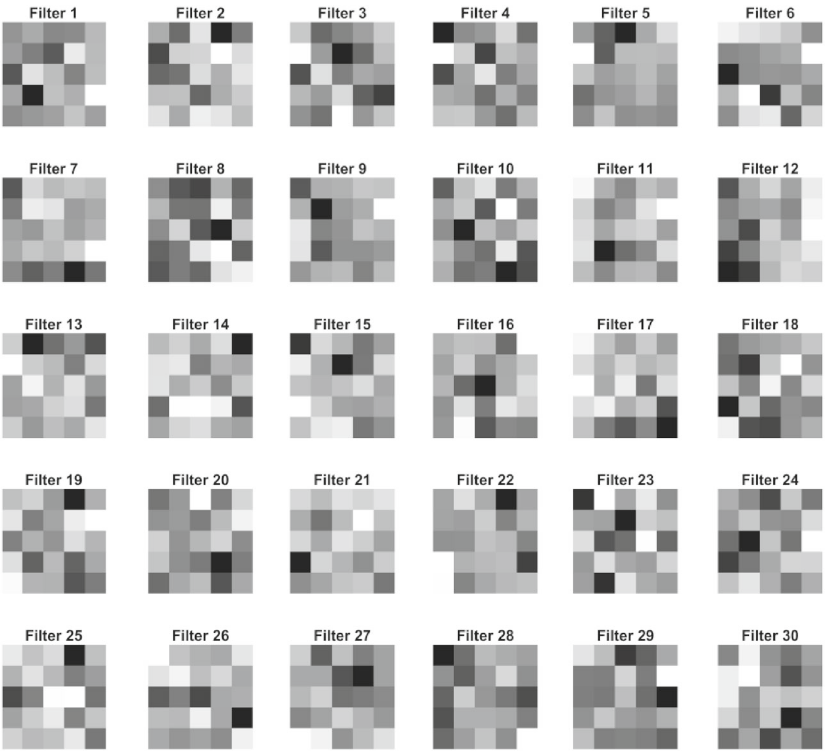

Fig. 21 Visualization of all the Filters of second convolution layer

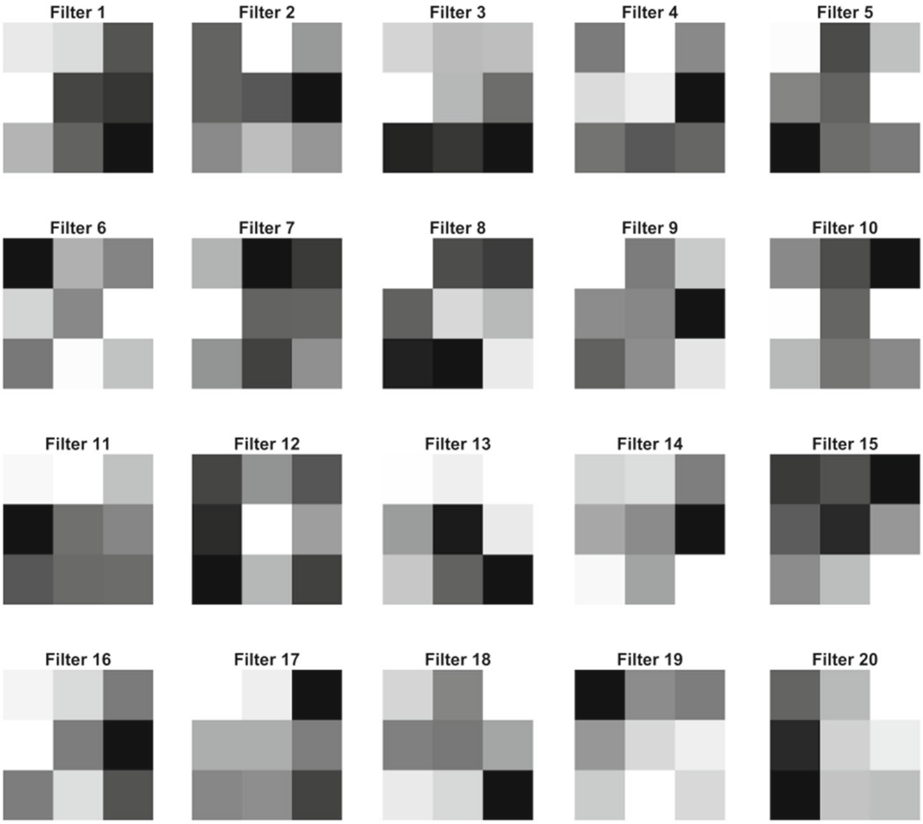

Fig. 22 Visualization of all the Filters of third convolution layer

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. Bajaj R, Chaudhury S (1997) Signature verification using multiple neural classifiers. Pattern Recogn 30(1):1–7
4. Baltzakis H, Papamarkos N (2001) A new signature verification technique based on a two-stage neural network classifier. Eng Appl Artif Intell 14(1):95–103. https://doi.org/10.1016/S0952-1976(00)00064-6. http://www.sciencedirect.com/science/article/pii/S0952197600000646
5. 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
6. Bhukya RK, Sarma BD, Prasanna SM (2018) End point detection using speech-specific knowledge for text-dependent speaker verification. Circ Syst Signal Process 37(12):5507–5539
7. 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
8. Charfi N, Trichili H, Alimi AM, Solaiman B (2017) Bimodal biometric system for hand shape and palmprint recognition based on sift sparse representation. Multimed Tools Appl 76(20):20457–20482
9. Chaudhry SA, Naqvi H, Khan MK (2018) An enhanced lightweight anonymous biometric based authentication scheme for tmis. Multimed Tools Appl 77(5):5503–5524

10. Cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/
11. De Marsico M, Petrosino A, Ricciardi S (2016) Iris recognition through machine learning techniques: a survey. Pattern Recogn Lett 82:106–115
12. Dey S, Dutta A, Toledo JI, Ghosh SK, Lladós J, Pal U (2017) Signet: Convolutional siamese network for writer independent offline signature verification. arXiv:1707.02131
13. Dutta A, Pal U, Lladós 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
14. 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
15. 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
16. Ferrer MA, Diaz-Cabrera M, Morales A (2015) Static signature synthesis: a neuromotor inspired approach for biometrics. IEEE Trans Pattern Anal Mach Intell 37(3):667–680
17. Fierrez-Aguilar J, Nanni L, Lopez-Peṅalba J, Ortega-Garcia J, Maltoni D (2005) An on-line signature verification system based on fusion of local and global information. In: Kanade T, Jain A, Ratha NK (eds) Audio- and video-based biometric person authentication. Springer, Berlin, pp 523–532
18. 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
19. 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
20. 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
21. Hafemann LG, Sabourin R, Oliveira LS (2017) Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recogn 70:163–176
22. 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
23. 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
24. 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
25. 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
26. Jain AK, Ross A, Prabhakar S (2004) An introduction to biometric recognition. IEEE Trans Circ Syst Video Technol 14(1):4–20
27. Jain AK, Nandakumar K, Ross A (2016) 50 years of biometric research: Accomplishments, challenges, and opportunities. Pattern Recogn Lett 79:80–105. https://doi.org/10.1016/j.patrec.2015.12.013. http://www.sciencedirect.com/science/article/pii/S0167865515004365
28. 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
29. Liu W, Wen Y, Yu Z, Yang M (2016) Large-margin softmax loss for convolutional neural networks. In: ICML, pp 507–516
30. 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
31. 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
32. 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

33. 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. IEE Proc-Vis Image Signal Process 150(6):395–401
34. 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
35. Pascual-Gaspar JM, Cardeṅoso-Payo V, Vivaracho-Pascual CE (2009) Practical on-line signature verification. In: Tistarelli M, Nixon MS (eds). Springer, Berlin, pp 1180–1189
36. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747
37. 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
38. 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
39. Sharma S, Dubey SR, Singh SK, Saxena R, Singh RK (2015) Identity verification using shape and geometry of human hands. Expert Syst Appl 42(2):821–832
40. Van BL, Garcia-Salicetti S, Dorizzi B (2007) On using the viterbi path along with hmm likelihood information for online signature verification. IEEE Trans Syst Man Cybern Part B (Cybern) 37(5):1237–1247. https://doi.org/10.1109/TSMCB.2007.895323
41. 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
42. Vezzetti E, Calignano F, Moos S (2010) Computer-aided morphological analysis for maxillo-facial diagnostic: a preliminary study. Journal of Plastic. Reconstruct Aesthet Surgery 63(2):218–226
43. Vezzetti E, Moos S, Marcolin F, Stola V (2012) A pose-independent method for 3d face landmark formalization. Comput Methods Programs Biomed 108(3):1078–1096
44. Wan C, Wang L, Phoha VV (2019) A survey on gait recognition. ACM Comput Surv (CSUR) 51(5):89
45. 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
46. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2016) Understanding deep learning requires rethinking generalization. arXiv:1611.03530
47. 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

Anamika Jain is perusing Ph.D. from Indian Institute of Information Technology Allahabad India. She received her M.Tech in Advanced Computing from MANIT Bhopal in 2015. Her research interest includes Image processing, Biometric, pattern recognition.

Satish Kumar Singh is with the Indian Institute of Information Technology Allahabad India as Associate Professor at Department of Information Technology from 2013. Before joining the IIIT Allahabad, he served the Department of Electronics and Communication Engineering, Jaypee University of Engineering and Technology Guna, India from 2005 to 2012. He is having about 15 years of academic and research experience in various capacities at JUET Guna and IIIT Allahabad. Presently he is heading the Computer Vision and Biometrics group and in-charge of the Computer Vision and Biometrics Lab (CVBL) at IIIT Allahabad from 2015 onwards. His is involved in the research and development of the Signal & Image Processing, Vision, and Biometrics algorithms and system. His areas of interest include Image Processing, Computer Vision, Biometrics , Deep Learning, and Pattern Recognition. Dr. Singh and his group published about 60 research articles in various IEEE/ACM/Elsevier/Springer Transactions and Journals and International conferences of repute.

Krishna Pratap Singh is currently an Associate Professor in Indian Institute of Information Technology Allahabad India. He has received his Ph.D. and M.Sc from IIT Roorkee in 2009 and 2004 respectively. His research areas are research interests are Evolutionary Algorithms, Machine Learning, Transfer Learning and Differential Privacy.