International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 9
ISSN: 2454-4248
85 – 90

Offline Signature Verification using CNN

A.Bhanu Sronothara,
PG Scholar,
Department of CSE,
MVSR Engineering College.

M. Hanmandlu,
Professor,
Department of CSE,
MVSR Engineering College.

Abstract-This paper presents the convolutional neural network for feature extraction and Support vector machine for theverification of offline signatures. The cropped signatures are used to train CNN forr extracting features. The Extracted features are classified into two classes genuine or forgery using SVM. The the new signature is tested on GPDS signature data base using the trained SVM. The dabase contains signatures of 960 users and for each user there are 24 genuine signatures and 30 forgeries. The CNN network is trained with 300 users and signatures of 400 users are used for feature learning. These 400x20x25 signatures are used 90%to train and 10% to test SVM classifier.

Keywords: *Convolutional Neural Network (CNN), Classification, Forgery, Genuine, Support Vector Machine(SVM), Radial Basis Function Network(RBFN).*

******

1. Introduction

Personal verification and identification are essential for securingpersonal assets. Hand written signatures are important for personal verification and identification. Legality of the most of the documents like bank checks, visa applications can be establishedthrough an authorized signature.

Automatic verification of signatures has been a well-researched problem. In the literature, several approaches such as fuzzy logic based and neural network-based approaches have been suggested. In offline signature verification the dynamic information of the signature writing process is lost, which is difficult in designing good feature extractor.

The signature verification is a pattern recognition problem. It involves finding similarities in patterns. Some typical applications of pattern recognition techniques are classification of text into several categories (e.g., spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes, automatic recognition of human face or handwritten text extraction from medical forms. There are two approaches in medical recognition: one supervised and another, unsupervised classification.

Verification can be done in two modes: On-line and Offline depending on how is thesignature acquired. In the on-line mode, the signature is captured while writing thus providing the dynamic

information comprising location, velocity, acceleration, pen pressure, pen up, pen down, angle and time. In the offline mode, the signature is scanned after the signature is written thus leading to the static image of the signature called the scanned signature. It is more difficult to verify a signature in the offline mode than in the on-line mode that provides more measurements.

Handwritten signatures have different sizes and shapes and the variations in them are so large at times that it is difficult to verify the genuine persons. Moreover, the signature of a person varies from time to time. Small variations are inherent and these can be tolerated by the authentication system. But when there is a significant change in the signature, the verification system should be updated with the new signature database. Signatures are categorized as: simple, cursive and graphical depending on their shapes. Simple signatures are the ones containing the names of persons. Cursive signatures are the ones written in cursive form. Graphical signatures are the ones depicting some geometric patterns. Some ofthe sample signatures are shown in Fig. 1.

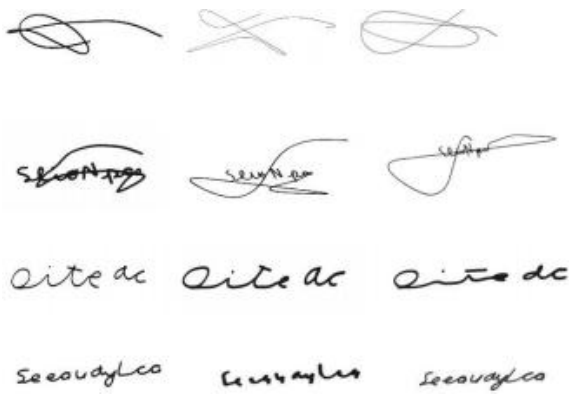

Fig. 1 Signature samples

1.1 Motivation for this paper

The variability in the different signatures of a person in the form of shape, size and orientation is the main motivation. By taking a standard size for all the signatures the variability in size is eliminated. The signatures are written horizontally and very rarely do we have the oriented signatures. But shape variations are common and these are attempted at capturing their effects in form of CNN features which are classified into categories: genuine and forged.

The issue of the categorization of signatures into different kinds like simple, graphical, skilled and unskilled to name a few is not attempted as we are not concerned with their forms rather their classification into genuine or forged.

2. Literature Survey

Offline signature verification is one of most challenging areas of pattern recognition. Being a behavioral biometric trait, which can be imitated, we face a challenge in designing such a system to counter the intrapersonal and interpersonal variations.

The previous works on this type of signature verification are summarized now. Sameera Khan and Avinash Dhole in [1] have reviewed offlinesignature verification and recognition. Different classification approaches such as Template Matching, Statistical, Structural, Spectrum Analysis and Neural Network are discussed.

Tritharaj Dash in [2] has proposed an offline signature verification system based on Associative Memory Net(AMN). The features extracted are trained on the AMN for the detection of forgery with an accuracy of 92.3%.

PallaviPatil and ArchanaPatil in [3] have presented an offlinesignature recognition using global features like area, height, and width. The Euclidean distance is employed while finding a match

between the test signature and the signature stored in the database. The system gives the recognition rate of nearly 89%.

Ranjan Jana [4] has developed an offline signature verification system using the Euclidian Distance.

The topological features such as baseline slant angle, aspect ratio, normalized area, center of gravity of the whole signature image and the slope of the line joining the centers of gravities of two halves of a signature image are used. The Euclidian distance between the claimed signature and the template serves as a measure of similarity between the two. If this distance is less than a predefined threshold, the test signature is said to be the genuine signature otherwise declared as a forgery. The system gives the classification of the genuine and forgery signatures with an accuracy of 100%.

Hafemann et al. [5] have analyzed the features extracted from offline signature using deep Convolutional Neural Networks(CNN) architectures like Alex Net and VGG NET.

Khalajzadeh in [6] has used CNNs for the signature verification on a dataset of Persian signatures by detecting of only the random forgeries.

Several methods have been proposed for feature extraction and classification of offline signatures. One of the most popular features for signature verification is Zonal or Graph based feature which considers the signature as a set of points. It divides the area into grids and estimates different statistics in each zone of asignature.

A variety of feature extractors have been investigated for the problem of signature verification from simple geometric descriptors [7], [8], inspired by graphology and graphometry [9], directional based descriptors such as HOG [10] and D-PDF descriptors based on interest-point, such as SIFT [10], to texture descriptors, such as Local Binary Patterns (LBP) [10] and GrayLevel Co-occurrence Matrix (GLCM) [11]. These features are commonly extracted locally from the signature images, by dividing an image into cells and computing descriptors for each cell(either in Cartesian or in polar coordinates).

3.The Proposed Methodalogy

3.1 Problem statement

The objective is to develop an offline handwritten signature verification system capable of differentiating between the genuine and forged signatures based on features extracted using CNN and classifier SVM.

3.2 Existing system

The existing models use hand-chosen features from an image and these are fed into a classifier. The models are only as strong as the chosen features and they often need large amount ofeffort to construct. The explicit features include geometric, graphometric, directional, wavelet, shadow, and texture features.

3.3 The Proposed system

The proposed signature verification system involves CNN for feature extraction and SVM for classification. In a CNN, the features learned from the dataset are fed into a classifier. The architecture of CNN consists of a number of layers such that each layer performs a simple computation starting at the raw image pixels and feeds the result to the next layer with the final

 ISSN: 2454-4248
85 – 90

result being fed to a classifier. The classifier selected is SVM with the cubic kernel. This is trained using the extracted features and testing is done on the trained SVM.

3.4 Algorithm for Signature Verification

The steps of the algorithm are outlined here:
Step-1: Acquire the signatures.
Step -2: Perform preprocessing on the signature.
Step-3: Extract the features using the trained CNN.
Step-4: Train the SVM using the extracted features.
Step-5: Test the signatures using SVM.

3.4.1 Database

Here we have used the database GPDS960 in [13] consisting of data from 960 individuals: 24 genuine signatures for each individual, plus 30 forgeries of his/her signature. The 24 genuine samples of each signer are collected in a single-day writing session. The forgeries are created from the static images of a genuine signature. Each forger is allowed to practise the signatures long as he/she wishes. Each forger imitates 3 signatures of 5 signers in a writing session. The signatures to be forged are chosen randomly out of 24 genuine signatures. Therefore, for each
genuine signature there are 30 skilled forgeries created by 10 forgers from 10 different genuine samples.

3.4.2 Preprocessing

As we have signature database acquisition of signatures from checks is not needed but for some preprocessing steps. The signatures in GPDS960 vary significantly in shape ranging from small signatures of size 153x258 to large signatures of size 819x1137 pixels.
We first center the signature in a large canvas of size Scanvas = H x W and then remove the background by setting the background pixels to white (intensity 255) and leaving the foreground pixels in grayscale using Otsu's algorithm. The image is then inverted by subtracting each pixel from the maximum brightness I (x; y) = 255 x I (x; y), such that the background pixels have zero gray level. The image is resized into the input size of the network.

3.4.3 Architectures of CNN

Most existing models in the literature use explicit feature extraction that includes extracting geometric, graphometric,bdirectional, wavelet, shadow, and texture features. Only in the recent years the learning of features has been explored. In this paper, CNN is used to learn the relevant features for signature verification by feeding the signature image.

The CNN consists of multiple layers, performing operations such as convolutions, max-pooling and dot products (fully-connectedlayers), where in convolutional layers and fully-connected layers have learnable parameters, that are optimized during training. The kind of architecture of CNN determines how many layers it has, what the function of each layer is and how the layers are connected.
There are four architectures for CNN, viz., LeNet, AlexNet, VGG and GoogleNet. Out of these, we have investigated LeNetand AlexNet. Choosing a good architecture is crucial to successful learning of CNN. The Alex Net of CNN architecture contains a total of 11 layers among which there are convolutional layers, pooling layers and the fully connected layers. The network that
we have used contains a total of 7 layers with learnable parameters as shown in Table 1.

Table 1: Summary of the CNN layers

Layer	Size	Other Parame-ters
Input	1x150x220	
Convolution (c1)	96x11x11	Stride=4, Pad=0
Pooling	96x3x3	Stride=2
Convolution(c2)	256x5x5	Stride=1, Pad=2
Pooling	256x3x3	Stride=2
Convolution(c3)	384x3x3	Stride=1, Pad=1
Convolution(c4)	384x3x3	Stride=1, Pad=1
Convolution(c5)	256x3x3	Stride=1, Pad=1
Pooling	256x3x3	Stride=2
Fully Connected (FC6)	2048	
Fully Connected (FC7)	2048	

Convolutional Layers: They process an input image by sliding a number of small filters across each possible region and output the dot product of the kernel, i.e. the image at each region. In this architecture the image of sizeis fixed at150x220as the input to the network. The first convolutional layer has 96 filters of size 11x11 with the stride of 4 and no padding. The second convolution layer has 256 filters of size 5x5 with stride of 1 and pad of 2.
The third convolution layer has 384 filters of size 5x5 with stride=1 and pad =1. The fourth convolution contains 384 filters of size5x5 with stride of 1 and pad of 1. The fifth convolution layer contains256 filters of size 3x3 with stride of 1 and pad of 1.

Max-Pooling Layer: Spatial Pooling (also called subsampling ordownsampling) reduces the dimensionality of each feature mapbut retains the most important information. Spatial pooling can beof different types: Max, Average, Sum etc. In the case of maxpooling, we define a spatial neighborhood (for example, a 2×2window) and take the largest element from the rectified featuremap within the window. Instead of taking the largest element wecould also take the average or sum of all elements in the window.We have considered maxpooling on the windows of size 3x3 withthe strideof 2.

Fully Connected Layer: This layer employs the traditional Multi-Layer Perceptron that uses a SoftMax activation function in the output layer. Instead of this classifier, we have employed SVM classifier. The term "Fully connected" implies that every neuron in the previous layer is connected to every neuron in the next layer. The outputs from the convolutional and pooling layers represent high-level features of the input image. The purpose of the fully-connected layer is to use these features for classifying the input image into various classes based on the training
dataset. In a fully-connected layer, every output depends on every input according to the weight matrix W, a learnable parameter. Outputs also depend on a bias term b which is learnable but
doesn't depend on the inputs. The Alex Net gives two feature vectors FC6 and FC7 of size 2048.

3.4.4 Training of CNN

We have trained CNN with 500 users, each user having 24 genuine signatures and 30 forgeries. That is, we have 400x54 samples for training the network. With the exception of the last layer in the network, after each learnable layer, we apply batch normalization, followed by the ReLU non-linearity. The last layer is a fully connected layer where feature vector of size 2048 is the
output. The trained CNN is used for feature extraction. Optimization is conducted by minimizing the loss with stochastic gradient descent and Nesterov momentum using mini-batches of size 32,
and momentum factor of 0:9. For the regularization, we have applied L2 penalty with weight decay matrix of size 10x4. The CNN models are trained for 60 epochs, with an initial learning rate of 10x3, which is divided by 10 for every 20 epochs. We have used simple translations for data augmentation using random crops of size 150x220 from 170x242 signature image. Then
the batch normalization terms (mean and variance) are calculated from the mini-batches during training. For the generalization, the mean (E[zi]) and variance (Var[zi]) for each neuron are calculated from the training set. It is worth noting that from our experiments we have found that batch normalization is crucial to train the deeper networks. Without using this technique, we cannot train architectures with more than 4 convolutional layers and 2 fully-connected layers. In these cases, the performance of both thetraining and validation sets remains the same as that of chance thus not indicating overfitting in the optimization process.

3.4.5 Feature extraction

We use 400 users' data to extract features by training CNN. The signatures of size (150x220 pixels) perform feedfor-ward propagation until the last layer before softmax and use the activations at this layer as the feature vector for the image. The feature vector for each image is stored in .mat file which is further used for classification. We have obtained 2048 features for
each image.

3.4.6 Classification

After extracting features, they are stored in .mat file of the 400 users each person having 25 forgeries and 20genuine signatures. We divide it into 90% for training and 25% for testing.
In this work we have used the SVM with the cubic kernel. SVM can be extended to separate a non-linear surface by using a kernel trick. A non-linear function φ can convert the original space into a higher dimensional space. The cubic kernel function used in
SVM given by

$$K(X_i, X_j) = (X_i, X_j + 1)^d$$

 ISSN: 2454-4248
85 – 90

The SVM is trained with genuine, random forgery, skilled forgery and unskilled forgery signatures. Considering the real signatures as positive class and forgeries as negative class during training, the remaining signatures are tested to see which class they belong to. When we give any signature for testing the features of the signature are extracted using the trained CNN and these features are given to the trained SVM for classification. SVM gives theoutput of 1 for the genuine signature and -1 for the forgery signature thus categorizing the signatures into either forgery or genuine.

3.5 Comparision of Results

The results obtained with CNN features discussed here. For the classification we have used SVM with different kernels like linear, Gaussian, quadratic and cubic kernels and RBFN. The best results are obtained with cubic kernel shown in Table 2. The performance of LeNet is found to be inferior to that of Alex Net on the database used. The LeNet has less convolutional layers

than that of AlexNet hence the features obatained from it are less accurate. But LeNet is more favored for digit or character recognition.

Table 2: Results obtained with different kernels

SVM Kernel function	Recognition rate
Linear SVM	87.3%
SVM Gaussian Kernel	63.4%
QuadraticSVM	92.4%
Cubic SVM	96.6%

The classification is also done using Radial Basis Function Network (RBFN) on the same dataset we get the accuracy of 83%. It is lessthan that of SVM. We evaluate the performance on the test

set data using the error rates: False RejectionRate (FRR): the fraction of genuine signatures rejected as forgeries; False Acceptance Rate (FAR): the fraction of forgeries accepted as genuine. We have achieved FRRof3% and FAR of 4% on the dataset as shown in Fig. 2.

Fig. 2 Confusion matrix

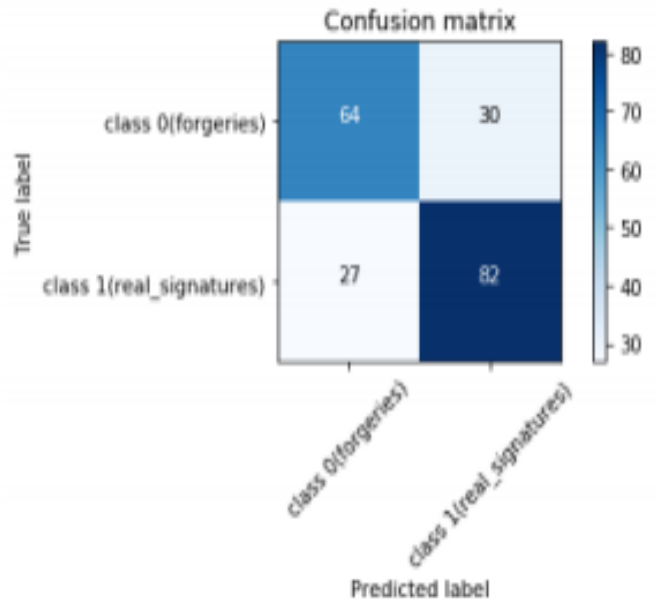

Fig. 3 Confusion matrix using LENET

4. Conclusions

The offline signature verification system is developed to distinguish between genuine or forged signatures. We have used two architectures: LeNet and Alex Net of Convolution Neural Network (CNN) for the extraction of features followed by Support Vector Machine (SVM) for classification. The best

 ISSN: 2454-4248
85 – 90

results are obtained with cubic kernel of SVM and Alex Net of CNN on the database. Though LeNet has proved its mettle on the recognition of handwritten digits, its performance is not impressive in the verification of signatures.

Owing to the lack of skilled signature data, we are unable to verify this class of signatures of a person. Creation of skilled signature data is therefore necessitated for improving the efficiency of the network.

References

[1]. Sameera Khan1, Avinash Dhole, A Review on Offline Signature Recognition andVerification Techniques, International Journal of Advanced Research in Computer and Communi-cation Engineering, Vol. 3, Issue 6, June 2014.

[2]. Tritharaj Dash, TanisthaNayak, SubhagataChattopadhyay, Offline Handwritten Sgnature Verification using Associative Memory Net, International Journal of Advanced Research in Computer Engineering and Technology Vol. 1, Issue 4, June 2012.

[3]. Ms. PallaviPatil, Ms. ArchanaPatil,"Offline Signature Recognition Using Global Features"International Journal of Emerging Technology and Advanced Engineering.Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 1, January 2013)

[4]. Ranjan Jana, RituparnaSaha,DebaleenaDatta, "Offline Signature Verification using Euclidian"Distance,(IJCSIT) International Journal of Computer Science and Information Tech-nologies, Vol. 5 (1) , 2014, 707-710.

[5]. L. G. Hafemann, R. Sabourin, L. S. Oliveira," Analyzing features learned for Offline SignatureVerification using Deep CNNs" arXiv:1607.04573v2 [cs.CV] 26 Aug2016.

[6]. H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Persian Signature Verification using Convolutional Neural Networks," in InternationalJournal of Engineering Research and Technology, vol. 1. ESRSA Publications, 2012.

[7]. R. N. Nagel, A. Rosenfeld, Computer detection of freehand forgeries C-26 (9) 895–905. doi:10.1109/TC.1977.1674937.

[8]. E. J. Justino, A. El Yacoubi, F. Bortolozzi, R. Sabourin, An off-line signature verification system using HMM and graphometric features, in: Fourth IAPR International Work-shop on Document Analysis Systems (DAS), Riode, Citeseer, 2000, pp. 211–222.

[9]. L. S. Oliveira, E. Justino, C. Freitas, R. Sabourin, The graphology applied to signature verification, in: 12thConference of the International Graphonomics Society, 2005, pp. 286–290.

[10]. M. B. Yilmaz, B. Yanikoglu, Score level fusion of classifiers in off-line signature verification, InformationFusion 32, Part B (2016) 109–119. doi:10.1016/j.inffus.2016.02.003.

[11]. J. Hu, Y. Chen, Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudo-dynamicFeatures, in: Document Analysis and Recognition, 12th International Conference on, 2013, pp. 1345–1349.doi:10.1109/ICDAR.2013.272.

[12]. J. Vargas, M. Ferrer, C. Travieso, J. Alonso, Off-line Handwritten Signature GPDS-960 Corpus, in: DocumentAnalysisand Recognition, 9th International Conference on, Vol. 2, 2007, pp. 764–768. doi:10.1109/ICDAR.2007.4377018.

[13]. L. G. Hafemann, R. Sabourin, L. S. Oliveira, Offline Handwritten Signature Verification-Literature Review, arXiv preprint arXiv:1507.07909.

[14]. Luiz G. Hafemanna, Robert Sabourina, Luiz S. OliveirabLearning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks, arXiv:1705.05787v1 [cs.CV] 16 May 2017.