HANDWRITTEN SIGNATURE VERIFICATION

Susheel VK
Department of Electrical Engineering and Computer Science
IISER Bhopal, India
vksusheel20@iiserb.ac.in

ABSTRACT

Despite the advent of sophisticated biometric verification methods such as fingerprint recognition, usage of signatures for verification remains popular since the process to collect signatures is non-invasive, and people are familiar with the use of signatures in their daily life. However signature verification is a difficult task to solve, due to the large high intra-class variability between genuine signatures. Compared to physical biometric traits, handwritten signatures from the same user often show a large variability between samples. This makes detection of authentic signatures, particularly when compared to skilled forgeries, extremely difficult, since skilled forgeries usually show low intra-class variability, meaning that creating a model that is capable of minimizing the differences between genuine signatures while maximizing the difference between genuine signatures and skilled forgeries is not trivial. The project involves two objectives. Primarily, we investigate the effectiveness of two existing models for Offline Handwritten Signature Verification on three publicly available datasets, the CEDAR, BHSig260-Hindi and the BHSig260-Bengali datsets. We also show preliminary results using these models on the 102 genuine signatures collected from IISER students, with 3 signatures per user.

1 MODELS USED

1.1 MODEL 1: DEEP CNN AND SVM

The first model used is the one given in Hafemann et al. (2017), with the implementation given at this Github repository. The approach takes advantage of the inherent feature extraction properties of CNNs, with greater depth allowing for extraction of more specific features. Unlike previous work that used feature extractors like SIFT to extract features, here the pixel intensities of the signature images are fed directly to the CNN. However, the feature extraction by the CNN can be considered similar to that followed by the feature extraction algorithms, since the CNN considers local patches during convolution and combines features in a non-linear fashion.

The approach used in the paper is analogous to transfer learning, where a model that is trained on one set of data can be used to classify another set of data. In this case, one model is formed by training a CNN for Writer-Independent feature learning. This model is then used to train Writer-Dependent classifiers which are SVMs. Since the neural networks expect inputs of a fixed size, while signatures vary significantly in shape, preprocessing is done to standardize the input. First the signatures are centered in a canvas of a fixed size using the signatures' center of mass. To remove the beackground, we set it to white(intensity 255), and leave the foreground pixels in grayscale. The image is then inverted by subtracting each pixel from 255 such that the background is zero-valued. Lastly, the image is resized to 170×242.

To test the model locally, we follow the sequence given above. We modify the backend to include the relevant details of the BHSig260 Hindi and Bengali datasets. We then train the model to learn a Writer-Independent classifier and test the results of the Writer-Dependent classifiers on the same dataset. The results given include the EER using the global and user-dependent thresholds and the mean AUC, given in 1. We observe an improvement on the results given in the original paper for the CEDAR dataset, which is expected since we have trained both the Writer-Independent and Writer-Dependent classifiers on the same dataset, in contrast to the paper where the Writer-Independent model was trained on the GDPS-960 dataset

Table 1: Results of Model 1

Dataset	EER-global threshold	EER-user thresholds	Mean AUC
CEDAR	3.07±0.94	0.39±0.56	1.0
BHSig260-Hindi	4.26±0.41	1.44±0.44	0.992
BHSig260-Bengali	1.54±0.47	0.42±0.22	0.997

1.2 MODEL 2: SIAMESE NETWORK

The second model used is given in this repository. It implements a Siamese Convolutional Network based on Dey et al. (2017). The Siamese network takes two images as input, and learns their representation in the feature space. Although a GitHub repository does exist for Dey et al. (2017), access to the pre-trained model was not provided. For this reason, using the model locally was infeasible. The code from the GitHub repository roughly follows the model from the paper, and uses the contrastive loss with Euclidean distance from training as described in the paper. Note that the code used inverts the definition of the label y, with 1 referring to (genuine, genuine) pairs instead of (genuine, forged) and vice versa for y = 0.
The Siamese network aims to bring the output feature vectors closer for input pairs that are labelled as similar, and push the feature vectors away if the input pairs are dissimilar. Each of the branches of the Siamese network can be seen as a function that embeds the input image into a space. Due to the loss function selected, this space will have the property that images of the same class (genuine signature for a given writer) will be closer to each other than images of different classes (forgeries or signatures of different writers). Both branches are joined together by a layer that computes the Euclidean distance between the two points in the embedded space. Then, in order to decide if two images belong to the similar class (genuine, genuine) or a dissimilar class (genuine, forged) one needs to determine a threshold value on the distance. This value varies for each pair considered.
The pre-trained model used was trained for 20 epochs on the CEDAR dataset. The accuracy of the model is given by the the average of the True Positive and True Negative Rates, considering the Euclidean distance(d) that maximizes these values, with the final accuracy being the average of the maximum accuracy obtained for each batch. The accuracy obtained on the three datasets used is given in 4, along with the optimum d value. The size of the test data set is also given in terms of the number of samples of (genuine, genuine) and (genuine, forged) pairs.

Table 2: Results of Model 2

Dataset	Test data set	Accuracy	d-value
CEDAR	2025	0.79	0.156
BHSig260-Hindi	5891	0.774	0.114
BHSig260-Bengali	3682	0.765	0.103

1.3 APPLICATION ON IISER DATASET

The Writer-Dependent SVM classifiers in the first model aim to distinguish between genuine and skilled forgeries of each user. Since the IISER dataset contains only genuine sample, we cannot follow the same methodology. Instead, we use the pretrained model(on the GDPS 960-dataset) to obtain the feature vectors of each preprocessed signature in the dataset, and train a classifier to perform a 34 class classification, with 3 classes per sample. After performing leave one out cross validation, we obtained an average **accuracy of 0.77**. Note that due to the minimal samples per class, this accuracy is volatile, and not necessarily indicative of good performance. However, it provides a guideline for a similar implementation when more samples are available.
For the second model, instead of considering skilled forgeries as negative samples we consider an arbitrary signature from another user(i.e. a random forgery) to construct the testing dataset. Using the same process as the other datasets, we obtain an **accuracy of 0.721** at **d=0.189**.

REFERENCES

Sounak Dey, Anjan Dutta, J. Ignacio Toledo, Suman K. Ghosh, Josep Llados, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification, 2017. arXiv: 1707.02131.

Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, 2017. ISSN 0031-3203.

A APPENDIX

A.1 DATASETS

Table 3: Datasets

Dataset Name	Users	Genuine Signatures	Forgeries
CEDAR	55	24	24
BHSig260-Hindi	160	24	30
BHSig260-Bengali	100	24	30

A.2 IMPLEMENTATION DETAILS OF MODEL 1

The steps followed for the CEDAR dataset are: First, the dataset is preprocessed following the steps given above. Then, the Writer-Independent classifier is trained using all the 55 users from the dataset. Random crops of size 150×220 are passed to the network to mimic translation as a form of data augmentation. For the Writer-Dependent classifier we consider the first 40 users as forming the exploitation set, with the remaining 15 forming the exploitation set. The RBF kernel is used in the SVM. The batch size is 32, and the training is perfomed 10 times for different splits of the data. We consider 12 genuine signatures per user during training. For testing, we consider 10 genuine signatures per user, and 14 signatures from the development set are considered forgeries. A similar process is followed for the BHSig260-Hindi and Bengali datasets according to the specifications given below in 3:

Table 4: Specifications for Model 1

Dataset	Writer-Independent	Development set	Exploitation set
CEDAR	55	40	15
BHSig260-Hindi	140	100	60
BHSig260-Bengali	100	60	40

A.3 TEST SET FORMULATAION FOR MODEL 2

In the orignal code, for training the CEDAR dataset a large number of samples(13500) are randomly chosen out of all possible (genuine, genuine) and (genuine, forged) pairs. These samples are then split into the train and test datasets in the ratio 85:15. Since we are only forming the test data set, we consider only $0.15 \times 13500 = 2025$ samples to form the test set. A similar ratio is used for all datasets, leading to the values given in 2