Proxy Signature Detection

Prof N.R. Talhar[1], Kamal[2], Shivam Kumar Singh[3], Uday Deshmukh[4], Faiz Ahmad[5]
Department of Computer Science Engineering
[1,2,3,4,5] AISSMS College of Engineering ,Pune

Abstract- *We are all aware of a number of biometric methods available for verification of user identification. Examples can be fingerprints, iris or signatures. Among them signatures is the least costly and most popular whereas fingerprints and iris verification require costly instruments. This makes signatures preferred choice for banks cheque processing and marking attendance in educational institutions and other coaching classes. But this less cost comes along with risk factors. It is because its very difficult for humans to distinguish between genuine and forged signatures. Hence a proxy signature detection system is needed which could distinguish between genuine and forged signatures with high accuracy. For achieving this, we have used convolutional neural network on GPDS Synthetic signature dataset for the purpose of feature learning. Then, we have applied transfer learning and have tried to learn the patterns in student signature database collected by us.*

Keywords- Deep Learning, Convolutional Neural Network, Signature Verification, Handwritten Signature

I. INTRODUCTION

There are various biometric methods for verification of identity. Vision based methods of identification include face recognition, fingerprint and retina scanning. Non vision methods include signature verification and voice recognition.Signatures are especially useful in financial, commercial and legal matters. For any legal transaction authorization is done by signatures.

So, the need for signature verification increases. The primary advantage of signature verification over other methods is that it is already an accepted method of identification. Signature is a special case of handwriting where there is abnormality between the ratio of heights of letters. In some signatures, the letters are not even readable by humans. Hence, it is very important for us to deal with signatures not thinking of them as a collection of words but checking each signature at pixel level and try to find pattern in signatures which belong to genuine class and those belonging to fake class. We know that the patterns of signature in both genuine class and fake class are going to vary probably not by much but there still has to have some difference. In this project, we are looking to exploit that difference. It has been observed that people have two mindsets before signing at a particular document. There are some documents which are not important for people so they tend to take it lightly and hence do signatures casually. Also there are some fake signature makers which try to copy the exact signature by slowly copying the same.

Hence, there are two parameters on which the authenticity of signatures can be ascertained. One is how fluent signature is and the other is how much similar the signature is to the actual genuine signature. We are using the similarity criteria in our project. Since we are not looking signatures as a collection of letters, it is difficult to manually look for patterns. Hence, we have used Convolutional neural network to find patterns and bring loss assmall as possible.

We have used transfer learning approach in our project.We used GPDS Synthetic Signature dataset to train our model and learn generalised features. We also created our own dataset for training and testing of real world data. We took 12 signatures each from 100 individuals and also tried to fake those signatures and hence got 12 fake signatures as well. First, we split our dataset into training and testing dataset. Then we have tried to learn features from GPDS dataset using Convolutional Neural Network. Finally, we use this feature extractor to train on our own dataset for each user again using neural network. We have tested our data on testing part of the same dataset which we collected. We have used GPU's in google colab for training and testing of data.

II. DATA COLLECTION AND PRE-PROCESSING

We asked students in our college to give signatures as it was required for research purpose. Other than students from our college, we also asked other people to give signatures. They readily agreed to it. Then, we scanned our data using a photocopy machine so that we would get all signatures to their actual size.

Figure 1: Sample Signatures

In the GPDS dataset, we have images of different sizes varying from around 300 x 600 pixels to 1200 x 900 pixels.For training a convolutional neural network we need all input images to be of the same size.Hence, we have resized the signatures to a value of 200 x 400 pixels. It is because too big images take too much memory and time to train and it is not feasible for GPU used in google colab. Also resizing images to very small pixels may lead to loss in important information.

Since, all the signatures were taken on a white sheet of paper.We kept the background as it is. We performed normalization on our input pixels by subtracting from mean and dividing by standard deviation.

From the GPDS Dataset which consists of about 2 lakh signature images (both genuine and fake), we have tried to learn as many features as possible. In GPDS Synthetic Dataset, we have 24 genuine and 30 fake signatures of 4000 individuals.

We considered all the images as a whole and divided them into folders each of size 704 which consisted of both fake and genuine signatures. By doing this, we were able to train all the images in batches. We did this because it was not feasible for us to train all the images at once due to memory constraints of GPU in google colab. We had also done the same for creating testing set of the individuals by keeping 704 signatures in each folder.We had to train all the images in batches in multiple epochs to overcome memory constraints.

III. PROPOSED METHOD

Various methods have been proposed for the verification of signature. Machine learning methods have been proposed after some data preprocessing using HOG and SIFT methods. People have proposed various graphical models for verification of signatures. Various neural network models have also been proposed whether be it using GANs or using Convolutional Neural Network. Transfer Learning methods have also been proposed using Conovlutional Neural Network and SVM.

Our method uses Convolutional Neural Network in both phases of transfer learning. First, we have used CNN to learn features from the publicly available GPDS Synthetic signature dataset. Then, we have used the learnt features in our own dataset to get better accuracy. We have used the same CNN for this purpose.

IV. EXPERIMENTATION

The objective of using Convolutional neural network used on GPDS Synthetic Dataset was to minimize the loss function and learn as many featuers as possible. We have used cross entropy as our loss function since it is better suited for classification tasks. We used leaky Relu as our activation function since it is more robust to exploding and vanishing gradients. Leaky Relu or simple Relu has been the preferred choice for most of the computer vision problems and problems solved using convolutional Neural Network.The accuracy that we got using these parameters were the best.

We have used Adam as our optimization or learning algorithm. The of 0 to get 32 feature maps at layer 1. Our motive behind using 32 filters was to make our model learn as many minute features (curves) present in our signature. We kept a max pooling layer of size 3 x 3 and a stride of 3 to reduce the number of dimensions. At the second layer again we used 16 filters of size 3 x 3 and a max pooling layer of size 3 x 3 with the same stride and padding. At the third layer, we used 8 filters of size 3 x 3 with same stride and padding. For max pooling, we picked 2 x 2 filter with a stride of 2. Then we used two fully connected layers with 10 neurons. reason behind this choice is again being well suited to coming out plateaus in the error function . Adam also has this ability to not depend heavily on a single and very influential parameter. We have tried various learning rates, we picked 0.001. It gave us the best loss and was giving a smooth loss curve. When going for the structure of convolution neural network, we had to make sure that we dont build a very heavy and deep model due to our memory and processor constraints. Hence, we tried different combinations.

We decided to put 32 filters of size 3 x 3 with a stride of 1 and a padding of 0 to get 32 feature maps at layer 1. Our motive behind using 32 filters was to make our model learn as many minute features (curves) present in our signature. We kept a max pooling layer of size 3 x 3 and a stride of 3 to reduce the number of dimensions. At the second layer again we used 16 filters of size 3 x 3 and a max pooling layer of size 3 x 3 with the same stride and padding. At the third layer, we used 8 filters of size 3 x 3 with same stride and padding. For max pooling, we picked 2 x 2 filter with a stride

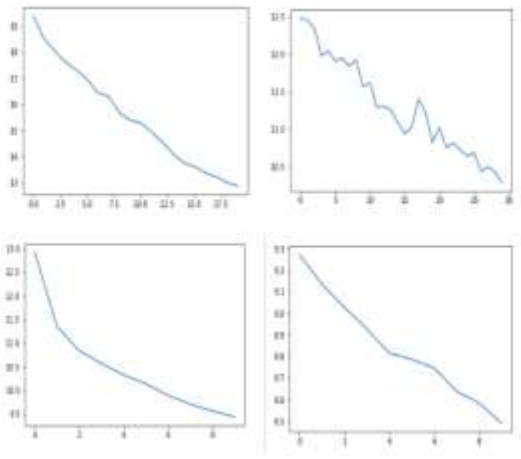
of 2. Then we used two fully connected layers with 10 neurons.

The reason behind keeping filter of size 3 x 3 is that it reduces the number of learnable parameters and hence reduces the amount of memory used to store them. A max pooling filter of size 2 x 2 has been taken because we dont want to degrade our image too much. We have used small filters thorughout to catch very small variations in our signatures. 32 filters have been used because we could not have gone beyond that as it was crossing our memory constraints. We have used dropout to make sure our CNN model doesnt overfit the data. To initialize the model while training on GPDS Synthetic Dataset, we have used He initialization. This convolution neural network was trained using pytorch library on GPU provided by Google Colab.

After training on GPDS Dataset, we needed to train our learnt model on signatures of each user. The Dataset collected did not contain very skilled forgeries of genuine signatures. It was more practical dataset where fake signatures were created by us only. Again we split our dataset into training and testing with each part containing genuine and fake signatures of each individual. We use the same Convolutional Neural Network and try to learn even more features for each individual. During the course of experiment, we chose the hyperparameters that were giving us the best results.

V. RESULTS AND DISCUSSION

There were two sets of results we will discuss, first is the loss we got on training our CNN model on GPDS Dataset. Following are our results i.e. graph of loss with iteration on GPDS dataset.



Loss on GPDS signature dataset reduced form 19.37 to 7.22. Following figures show reduction in error rate when input was fed in batches. Second is the results after training our data on our own dataset. Following are our results:
Cross validation mean Accuracy per user = 0.80779

VI. CONCLUSION

With our processor and memory constraints, our CNN model works very well for practical and real world datasets. Such a model can be used in educational institutions for verifying attendances and in some offices where signatures are still prominent. But due to memory constraints, it is still not able to distinguish between very skilled fake signatures and very casually done genuine signagtures. Still, there is a lot can be done to improve such a model to get better accuracy.As we mentioned above in the paper, we can build a dataset which will check even the fluency parameter of each signature along with similarity with genuine signature to get even better results.

REFERENCES

[1] Hemanta Sakia, Kanak Chandra Sharma, “Approaches and issues in offline signature Verification ”, Published in International Journal of Computer Applications (0975 - 8887) Volume 42- No. 16, March 2012

[2] Ali Karouni, Bassam Daya, Samia Bahlak, “Offline signature recognition using neural networks approach”, 1877-0509 2010 published by Elsevier Ltd.

[3] Reena Bajaj, Shantanu Chaudhury,“Signature verification using multiple neural classifiers”, received 14 octoer 1993; in revised form 19 march 1995;received for publication 15 april 1996

Table 1: Structure of CNN

Layer	Size	Other Parameters
Convolution	32x198x398	n=32,kernel=3x3,stride=1,padding=0
Pooling + Relu	32x66x132	kernel=3x3,stride=3
Convolution	16x64x130	n=16,kernel=3x3,stride=1,padding=0
Pooling + Relu	16x21x43	25 kernel=3x3,stride=3
Convolution	8x19x41	n=8,kernel=3x3,stride=1,padding=0
Pooling + Relu	8x9x20	kernel=2x2,stride=2
Fully Connected + Dropout	1440	
Fully Connected	10	
Fully Connected + softmax	2	

[4] Luiz g. Hafemann, Robert Sabourin, Luiz S. Oliveira,“Writer-independent Feature Learning for Offline Signature Verification using Deep Convolutional Neural Networks ”, accepted as a conference paper for IJCNN 2016 [cs:CV] 4 Apr 2016

[5] Pallavi V. Hatkar, Prof. B.T. Salokhe, Ashish A.Malgave,“Offline Hand-written Signature Verification using neural network ”, published in International Journal

of Innovations in Engineering Research and Technology [IJERT], volume 2 Issue 1 Jan 2015