Signature Verification

CS-1390: Introduction to Machine Learning

Payal Basu, Raunak Basu, Asheesh Kumar Singh, Aryan Sharma

Abstract

Signature verification is an important biometric authentication technique that involves comparing the signature of a person with their signature stored in the database. It is an essential part of verification of a person as authorization based on a forgery can have serious consequences. Our paper attempts to efficiently carry out offline signature verification using a Siamese Neural Network, a deep learning model that involves a twin network to compare sample input pairs and classify them as similar or dissimilar . Using the ICDAR SigComp 2011 Dataset to train and test our model, we are able to achieve an accuracy of 67%.

1. Introduction and Motivation

The world of biometric technology has continued to grow over the last decade. Mainly used for identification and verification, there have been great strides taken in terms of research, development as well as implementation.

Verification technology has been used widely in order to prevent fraudulent behaviour and ensure that an individual's identity is in fact who they claim to be. Signature verification is one such prominent subsection within the realm of biometrics. It has been extensively researched and proven to be effective, seeing use in important bank and governmental document authentication as well as being easy enough to use for other documents and fields as well.

Signature verification systems aim to distinguish genuine signatures from their forged counterparts. In the process, generally, a photo of the signature in question is put through a signature verification system that compares it to the original signature on record. Various factors may be considered when comparing the two, from checking how closely it matches in terms of Euclidean distance to characteristics like pressure, pen lifts, speed and direction of pen strokes. The systems used can usually be divided into two categories, online signature verification and offline signature verification systems. The focus of this paper shall be on the latter: Offline

Signature Verification is static by nature wherein digital images or scans of written signatures are used and analysed. Offline signature verification systems are imperative for further research and evolution due to its growing relevance in fraud detection in various fields.

In order to construct accurate signature verification systems, deep learning algorithms such as Convolutional Neural Networks(CNN) and Siamese Neural Networks have often been used, especially for offline signature verification. With the accuracy afforded by Siamese Networks in comparing objects in machine learning and visual imagery today, while utilising a robust GPU-based architecture, this paper will attempt to employ this technique appropriately.

2. Background and Previous Work

There have been huge advancements in the field of offline signature recognition in the past few years. Deep learning methods have especially shown promising results with data models with high accuracy. Neural Networks being used as models for signature verification systems have been widely researched for both writer-independent and writer-dependent systems. A model that can classify inputs from any user, it would be referred to as writer-dependent, whereas a model specifically trained for each and every user would be called a writer-dependent model. Writer independent models ensure efficiency and consistency across various inputs and datasets, and replicating such a model would be the main focus of the paper.[1]

In 1997, Huang and Yan trained multiple neural networks based on geometric feature extraction. The combined outputs at each scale were then used to generate a match rating and make a decision through a second neural network. Results showed 90% accuracy by using such an approach. [2]

More recently, in 2019, Hadjadj et al used an offline signature verification technique that exploits textural information of images in order to distinguish the real signatures from forgeries. In their system, signature images are characterized using two different textural descriptors, and

[1] Hafemann, Sabourin, Luiz S. Oliveira, “Offline Handwritten Signature Verification - Literature Review” *arXiv preprint arXiv:1507.07909* (2017).
[2] Huang, Kai, and Hong Yan. "Off-line signature verification based on geometric feature extraction and neural network classification." *Pattern Recognition* 30.1 (1997): 9-17.

the distance between genuine signature images and their fake counterpart pairs is used to train SVM classifiers. Decisions from both such classifiers are then used to arrive at a conclusion on the validity of a signature brought before the system. The results showed high accuracy on the ICDAR 2011 dataset which is a common benchmark in terms of signature verification models.[3]

In 2016, Alvarez et al worked on Offline Signature Verification utilising Convolution Neural Networks using VGG16 architecture[4], which was also proven to be effective on the ICDAR dataset. Their method reached an accuracy of 97% for Dutch Signatures on the dataset and 95% for Chinese signatures. Furthermore, they experimented with several features like unseen identities, and altering the training data as well as the prediction methods to mirror practical applications to improve their accuracy. The final accuracy of their model was near comparable to the other state of the art models present at that time, although the results were not much higher than the baseline reached from a naive method.

In 2017, Dey et al modelled an offline writer-independent signature verification system using a convolutional Siamese network[5]. The network was introduced to similar and dissimilar (real and forged signatures) observation pairs. The machine had two identical sub-networks that processed both the images simultaneously. Each sub-network had a similar layered architecture of 4 Convolutional layers, pooling functions, local response normalization layers and two fully connected layers. The Euclidean distance between the similar pairs were minimized and that between the dissimilar pairs were maximized. The constructed twin networks, called SigNet, were exposed to signatures in several languages and different handwritings across four data sets (CEDAR, GPDS300, GDPS Synthetic, BHSig260) and displayed excellent results. The SigNet method provided better results than most other offline signature verification networks that preceded it.

Due to its remarkable results, the SigNet network has been taken as the main source of inspiration for the architecture of the network presented in this paper.

---

[3] Hadjadj, Ismail, et al. "Offline signature verification using textural descriptors." *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, Cham, 2019.

[4] Alvarez, Gabe, Blue Sheffer, and Morgan Bryant. "Offline signature verification with convolutional neural networks." *Technical report, Stanford University* (2016).

[5] Dey, Sounak, et al. "Signet: Convolutional siamese network for writer independent offline signature verification." *arXiv preprint arXiv:1707.02131* (2017).

3. CNN and Siamese Networks

3.1 Convolutional Neural Network

Convolutional Neural Networks are multilayered Deep-Learning Neural Networks that are mostly used in image recognition and classification. These are very effective in differentiating an image from another. They consist of convolution layers and pooling layers that detect features and reduce or downsample the output of the convolutions. The output from the third layer is then normalized in turn and sent to a softmax activation function which generates the final output prediction.

3.2 Siamese Neural Network

The Siamese Neural Network (SN) is capable of learning with a relatively small dataset. This means that with even bigger datasets it should be able to give even better results while maintaining a reasonably good amount of computational efficiency. This network architecture consists of two identical sub-networks. These networks are often convolutional neural networks with the exact same parameters and layer structure and are thus used to find how similar or dissimilar the inputs are as it is perfectly set up for comparison between two items.

Figure 1 below is the state of the art SigNet network implemented by Dey et al (2017) Network structured specifically for the purpose of Offline verification.

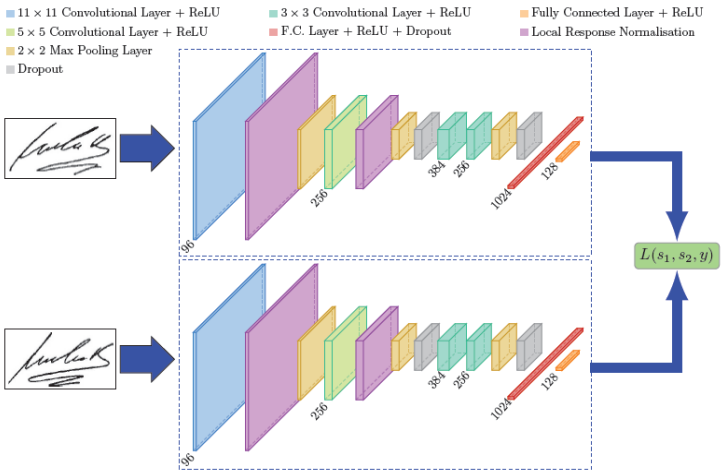


Figure 1 : Model from the SigNet reference paper

When images channels are input, both the CNNs run simultaneously and both the networks resemble the same parameter updates. At the end of the networks, they are then introduced to a loss function. Commonly the choice of loss function for a Siamese Network is the Contrastive loss Function. The Contrastive loss function uses the Euclidean distance between two similar points of the feature representation of both the incoming networks.

Represented mathematically:

Contrastive loss: $L(x1, x2, F) = \alpha(1 - y)Dw\,2 + \beta ymax(0, n - Dw)2$

where x1 and x2 are two samples (here signature images), F is a binary indicator function indicating if both the sample images belong to the same class or different classes, α and β are constants and n here represents the margin (1). Dw is the measure for the Euclidean distance between the feature space of both the sides of the Siamese network.

Because of the way that the contrastive loss function is structured, if two images are the same as each other (original signatures in the context of offline signatures), the loss is lower. If the two images are different to each other, the loss is higher

.

4. Our Approach

4.1 Datasets

The ICDAR 2011 SigComp dataset is used in this paper to test our model's accuracy. The Dataset has two subsets of Chinese and Dutch signature images. Each of these language-specific signature sets have a training set as well as a test set. While both Offline and Online signatures are available, for the purpose of this paper, just the Offline signature dataset in Dutch will be considered.

The ICDAR 2011 SigComp dataset has been widely used for its versatility and has a large number of observations. The dataset is divided into train and testing folders. In each folder, there are two folders, xxx and xxx_forg for each person where xxx represents the person number. The folder xxx contains the genuine signature and xxx_forg contains images of forced signature. The dataset we used also provided us with training and testing CSV files. In each file, there are three columns: two for image paths and one that contains the label when the two signatures are compared.

4.2 Preprocessing

In order to feed in the training data to our model, the data set is preprocessed to maintain uniformity and so as to eliminate edge and corner cases that might throw up anomalies and inaccuracies in the data. First, the images in the dataset taken are all resized to (100 x 100) pixels to capture the subject of the image (the signature) better and to maintain the same size and format for all the images. This eliminates issues with processing and computing. The images are also converted to grayscale to improve on processing time, while increasing accuracy and efficiency. Thanks to these tasks, the images can now be vectorised easily and accurately which in turn leads to more accurate results.

4.3 Architecture of chosen model

The model architecture and layer structure chosen in this paper have specifically been inspired by the SigNet model (Dey, 2017), which is a state of the art convolutional Siamese Network implemented specifically for offline signature verification.

Table 1 below shows the parameters that our network uses for both the individual CNNs.

Layer	Size	Parameters
Convolution layer 1	50 x 5 x 5	Stride = 1
Local Response Norm	-	n = 50
Max Pooling Layer	50 × 2 × 2	Stride = 2
Convolution layer 2	60 × 5 × 5	Stride = 1
Local Response Norm	-	n = 60
Max Pooling Layer	60 × 2 × 2	Stride = 2
Convolution layer 3	80 × 5 × 5	Stride = 1
Fully Connected Layer	32000	

Fully Connected Layer	128	

Table 1

In the convolution layers, the size can be seen as k x d1 x d2 where k is the number of filters in the respective layer and d1,d2 are the dimensions of each filter. Our first convolutional layer helps filter the 100 x 100 images with 50 kernels of size 5 by 5. The second convolutional layer takes the max-polled and normalized output from the first convolutional layer. It filters the output with 60 filters of size 5 by 5. Our third and final convolutional layer takes in the output from the second layer and has 80 filters of dimensions 5 by 5. As the layers progress, the number of filters in each convoluted layer consecutively increases.

The output from the last convolutional layer is then introduced to two fully connected layers of 32000 and 128 respectively. These are the final layers determining the classification before the outputs are introduced to the Contrastive loss function.

4.4 Training the Network

Using the robust deep learning framework afforded by the PyTorch library, a neural network was set up and our model was trained for 10 epochs. The CUDA-enabled GPU for training that was integrated with PyTorch was used, which was vital for improving accuracy and effectively reading and translating the data present in the images into tensors which were more effective in training the model.

Scanned images of written signatures are considered, and with the graphical computational ability of the PyTorch library, they were appropriately preprocessed, vectorised and helped train the neural network that was implemented. The RMSProp optimizer was also used in addition for increased efficiency and smoothness in running the program. The program took 45 minutes in total to run 10 epochs and train the model based on the ICDAR 2011 dataset.

4.5 Testing the Network

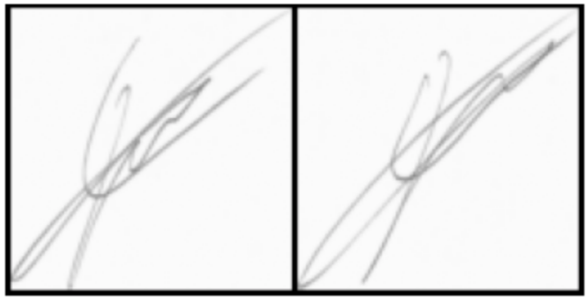
In the testing phase, the model takes a pair of images from the test data and compares them on the basis of the euclidean distance between them. In order to differentiate between the forged and real signatures a threshold value of 0.5 is adhered to. We used the pairwise_distance function to calculate the same and a value below or equal to 0.5 labelled the images as real. On the other hand, a value greater than 0.5 predicted the images as forged.

For the accuracy calculation on test data, our program takes a pair of images, where both the images belong to the same id. The machine is then tested with 500 different pairs of images within the test data provided by the ICDAR 2011 dataset. We increment a count variable that tracks if the predicted result (decided on the basis of the Euclidean distance between them) and the actual result are the same.

4.6 Results

Examples of output:

```
Predicted Eucledian Distance:- 0.8270631432533264
Actual Label:- Forged
Predicted Label:- Forged
```



```
Predicted Eucledian Distance:- 0.2795881927013397
Actual Label:- Both are original
Predicted Label:- Both are Original
```

```
Accuracy:  67 %
```

The model used two images: a reference signature image and an image in question. We used euclidean distance between the images as a performance metric. The threshold was 0.5. In the above images, The image on the left is a genuine/ reference image and the second image is the image in question. Based on the Euclidean distance from *"Predicted Euclidean distance", "Predicted label"* shows what the model predicts about whether the signatures are similar or not (hence forged). A distance of less than 0.5 implies that the image in question is a real image, or else it predicts to be fake. Although in the both above images, the predicted class matches the actual class, this was often not the case. Hence this paper has accounted for the accuracy of the chosen model due to the above mentioned factors.

To find the accuracy, the following formula was used:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ test\ image\ pairs}$$

The final accuracy of the model ranged between 63- 69 %

5. Other Attempted models

Before working on the Siamese network, a CNN model for signature verification was considered. Three convolution layers with kernel size of 5 × 5 in the first and second layers and 3 × 3 in the third layer were used to achieve this. However, the results fluctuated and the accuracy was generally around 40%. Since Siamese models are essentially two CNNs running together and hence a lot more useful when comparing images, the loss of accuracy in a simple CNN model is accounted for. While CNN models have been used commonly in Signature verification, Siamese neural networks have been regarded as the more accurate models in Offline Signature verification literature.

6. Conclusion and Discussion

The model worked upon in this paper is an attempt to find various ways of improving the methods being used currently. However, certain techniques adapted in the model limited the accuracy of the model. Although the choice of loss function in this paper was the Contrastive loss function, using a *binary cross entropy* function could have potentially helped to improve our results. It is important to note that the training data was limited to samples of Dutch signature pairs. In order to implement a successful network, most Siamese networks test on multiple datasets across several languages and scripts. Since training data was restricted to the Dutch Language in the interest of efficient processing speeds, it is likely that the model will not be able to be tested on Signatures in other languages or scripts. Consequently, the model would have performed better if the input dataset was larger. To prevent overfitting, different combinations of images in our dataset were attempted which required significantly more time to train the model. Another limitation on the methodology were resources and the time constraint. The accuracy would have likely been significantly higher if a greater number of epochs had been used (more than 10 in this case) and if state-of-the-art computational resources were available. This model uses GPU for computation, but the systems it was run on did not have enough computational capacity to effectively do so . Even though the GPU available on Google Colaboratory's programming interface was also utilised, restricted access to the GPU for limited periods of time meant that the computational barriers were less but still remained. This also meant that the time spent in training and processing increased because of which the earlier mitigating factors such as smaller dataset inputs and lower number of epochs were necessitated.

While following the SigNet paper's network architecture, the Siamese network model has been adapted and utilised for the ICDAR 2011 dataset. The resultant accuracy of an average 67% in a range of 63-69% is promising but could potentially be increased with different resources being utilised.

7. References

Dey, Sounak, et al. "Signet: Convolutional siamese network for writer independent offline signature verification." *arXiv preprint arXiv:1707.02131* (2017).

Alvarez, Gabe, Blue Sheffer, and Morgan Bryant. "Offline signature verification with convolutional neural networks." *Technical report, Stanford University* (2016).

Huang, Kai, and Hong Yan. "Off-line signature verification based on geometric feature extraction and neural network classification." *Pattern Recognition* 30.1 (1997): 9-17.

Hadjadj, Ismail, et al. "Offline signature verification using textural descriptors." *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, Cham, 2019.

Hafemann, Sabourin, Luiz S. Oliveira, “Offline Handwritten Signature Verification - Literature Review” *arXiv preprint arXiv:1507.07909* (2017).

“What Is Signature Verification? - Definition from Techopedia.” *Techopedia.com*, www.techopedia.com/definition/30499/signature-verification.

“ICDAR 2011 Signature Verification Competition (SigComp2011).” *TC11*, www.iapr-tc11.org/mediawiki/index.php/ICDAR_2011_Signature_Verification_Competition_(SigComp2011).