NATIONAL INSTITUTE OF TECHNOLOG WARANGAL

Department of Computer Science and Engineering

Report on Project

Offline Handwritten Signature Verification

Guided by: Murukessan A. P.

Submitted to: Dr. M Srinivas

Submitted by

Ankur Tiwari(21CSM2R02)

MTech I year, Computer Science and InformationSecurity (2022)

Abstract

In the real world, banks have always required a handwritten signature while creating a new account, signing cheques or any other official document. But they never really verify if the signature provided is authentic or just a forgery trying to fraud the person and the bank. Not only banks documents but other official documents like those in courtrooms, hospitals, and property documents also need to authenticate the handwritten signatures. This model introduces a novel and efficient offline signature verification method that provides a quick, secure and reliable real-time technique to verify and authenticate the handwritten signatures. Currently, some systems have been introduced to automate the process of signature verification but they face multiple drawbacks that our proposed solution manages to overcome in an efficient manner. The neural network we used is Siamese neural network which is an artificial neural network which takes in similar input image vectors and produces comparable output vectors.

Introduction

Handwritten signature is one of the most important forms of biometric authorizations that are used universally to authorize and verify documents. Thus, a need for identifying the genuineness of a signature is essential.

The primary difficulty faced by signature recognition systems is the intraclass variation, i.e., an individual's original sign may not be the same each time they sign it because several external and internal factors affect it, such as, the pen used, the surface used, etc. Thus, any model that aims at performing signature recognition must account for these variations

There are two ways to identify the signatures, off-line signature recognition and on-line signature recognition. On-line signature assesses the parameters such as speed of writing, the force employed on the signing instrument, the position of the hand, etc., all of which are physical factors that are outside the scope of this project. The offline signature verification can be done in many ways one of those methods is using the concept of artificial neural networks which gives the result in a lesser time.

In a writer dependent model, the characteristics included are writer dependent threshold, features and classifiers. However, this simple image classifying neural network is not sufficient since a new neural network would have to be trained for each and every individual writer, which is highly inefficient and ineffective.

Dataset

The BHSig260 dataset includes signatures in Bengali and Hindi. There were 100 signers in the Bengali dataset, each with 24 genuine signatures and 30 solid signatures. There were 24 x 100 = 2400 legitimate signatures and 30*a hundred=3000 solid signatures in general. Signatures of 160 people were collected in the Hindi dataset, with each person having 24 genuine and 30 fraudulent signatures.

Hindi signatures were utilized. We develop true-real image names and authentic-solid picture names in this section. We have 24 authentic signatures for each person, thus we have 24 select 2 = 276 actual-genuine photograph pairings for each individual. To create actual-forged pairings, we match each person's proper signature with ten randomly picked forged signatures of the same person.

As a result, for one individual, we create 24 * 10 = 240 proper-cast picture pairings. We have information on a total of 120 people in the training records. 120 * 276 = 33120 total number of genuine-true pairings 120 * 240 = 28800 total number of genuine-cast pairs total number of points = 33120 + 28800 = 61920 This dataset is sufficient for model.

Related work

Handwritten Offline Signature reputation based on biometrics, as proposed by Gulzar A. Laghari, can reliably distinguish between an imposter and a legitimate person. Biometrics is the study of an individual's behavioral and physiological characteristics. Shashi Kumar D R and K B Raja The paper proposes an off-line signature verification system that combines grid and global capabilities with a neural network (SVFGNN). Signature verification function units are created by combining global and grid capabilities.
Ankit Arora, Aakanksha S Choosey Awesome characteristics (set of sections, percentage coefficient, projection, and center of gravity) were analyzed independently in the study, and the outcomes of each element were discussed. It shows how to understand offline signatures using DWT and Angular features (DOSVAF). To extract the functions, the signature is scaled and the blocks are subjected to DWT (Discrete Wavelet Rework). Nilesh Y. offered signature reputation using the Propagation Neural community once more.

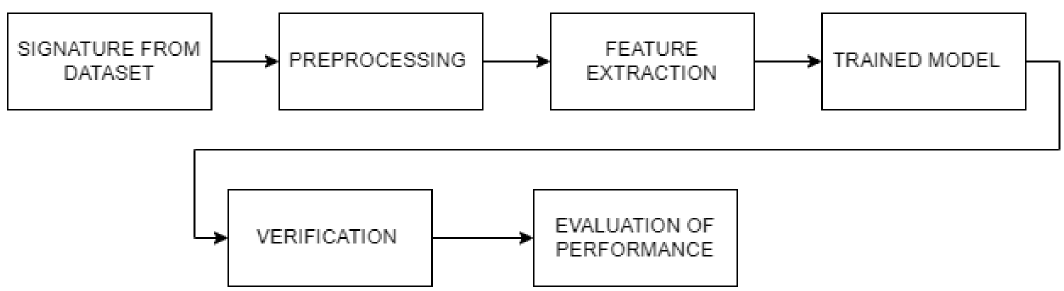
A fully image-based returned propagation neural network with an invariant central second and certain global residencies is suggested.

Neural Networks were proposed by Odeh. It demonstrates how to utilize an MLP neural network to do offline signature verification and authenticity using four picture processing functions: eccentricity, skewness, kurtosis, and orientation. Mujahed Jarad, Nijad Al-Najdawi, and Sara Tedmori provided assistance.

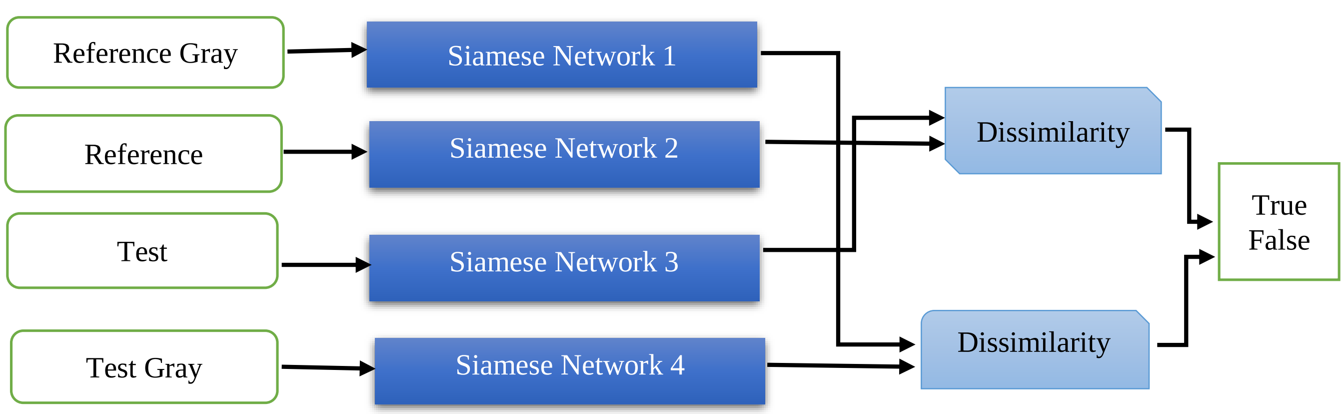
Neural network algorithms have outperformed traditional signature verification systems. However, most current solutions handle the problem of signature verification in the context of a picture class rather than modelling the signature itself, which can lead to inaccurate predictions for complex signature images.

SigNet takes twin Siamese Network and take Reference and Test images. I tried to improve this model to change twin to four Siamese Network. Other 2 Siamese Network take gray image of both Reference and Test images.

Block Diagram



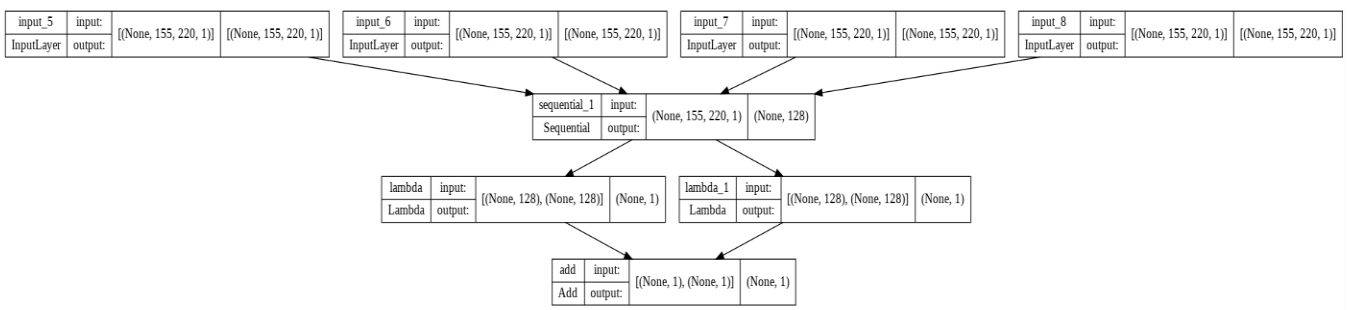
Proposed Model



Siamese Neural Network

A Siamese neural network is a type of community architecture made up of two or more equal subnetworks. here The four CNNs are set up in the same way, with the same parameters and weights. The updating of parameters is mirrored across subnetworks. This framework has been successfully applied to dimensionality discounting and verification in weakly supervised metrics. Those subnetworks are connected at the top by a loss characteristic that computes a similarity metric based on the Euclidean distance between the characteristic representation on each facet of the Siamese community. The contrastive loss is one such loss function that is commonly used in the Siamese community.

Model



Here the model structure takes 4 images, where 2 are RGB reference and test images and the other 2 are gray images of both. After that, image processing in CNN layers for feature vector The vectors in the respective zones show the areas or signature features that are learned by the network for distinguishing between these two signatures. The feature vector is then compared by the Euclidean distance formula. It gives 2 dissimilarity values, one RGB and one Gray image. In the last layer, both dissimilarity values are added for better dissimilarity values.

layers

```
[ ] base_network.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1_1 (Conv2D)            (None, 37, 53, 96)        11712

 batch_normalization (BatchN  (None, 37, 53, 96)       148
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 18, 26, 96)       0
 )

 zero_padding2d (ZeroPadding  (None, 22, 30, 96)       0
 2D)

 conv2_1 (Conv2D)            (None, 18, 26, 256)       614656

 batch_normalization_1 (Batc  (None, 18, 26, 256)      72
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 8, 12, 256)       0
 2D)

 dropout (Dropout)           (None, 8, 12, 256)        0

 zero_padding2d_1 (ZeroPaddi  (None, 10, 14, 256)      0
 ng2D)

 conv3_1 (Conv2D)            (None, 8, 12, 384)        885120

 zero_padding2d_2 (ZeroPaddi  (None, 10, 14, 384)      0
 ng2D)

 conv3_2 (Conv2D)            (None, 8, 12, 256)        884992

 max_pooling2d_2 (MaxPooling  (None, 3, 5, 256)        0
 2D)

 dropout_1 (Dropout)         (None, 3, 5, 256)         0

 flatten (Flatten)           (None, 3840)              0

 dense (Dense)               (None, 1024)              3933184

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 128)               131200

=================================================================
Total params: 6,461,084
Trainable params: 6,460,974
Non-trainable params: 110
_________________________________________________________________
```

The preliminary convolutional layers filter out the 155X220 input signature image with 96 kernels of length 11x11 and a stride of one pixel. The first convolutional layer's (reaction-normalized and pooled) output is fed into the second convolutional layer, which filters it with 256 5x5 kernels. The 1/3 and fourth convolutional layers are linked to each other without any pooling or layer normalization. The output of the second convolutional layer (normalized, pooling, and dropout) is coupled to the 384 3x3 kernels of the third layer. The fourth convolutional layer is made up of 256 3x3 kernels. As a result, the neural community learns fewer lower-level traits for smaller receptive fields while learning more higher-level or abstract functions. The first absolutely linked layer contains 1024 neurons, while the second completely related layer contains 128 neurons. As a result, the highest learned characteristic vector for each facet has a length of 128.

Metrics Used

Adaptive Estimation is a method for optimizing the gradient descent algorithm. When dealing with a large amount of data or parameters, the approach is truly green. It uses less memory and is more efficient. It's essentially a combination of the 'gradient descent with momentum' and the 'RMSP' algorithms.

The Contrastive Loss feature is the version's loss function. Its goal is to reduce dimensionality by learning an invariant mapping that generates high-to-low-dimensional area maps that switch similar input vectors to adjacent locations on the output manifold and different vectors to distant locations.

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)} ,$$

Euclidean The shortest distance between two points is represented by distance. This distance metric is used by most machine learning algorithms, including K-Means, to quantify the similarity of observations. learning rate - 0.001.

$$d(x,\ y)\ =\ \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Results

Base model given 76% accuracy. I am trying to increase this with different methods. I am not showing training accuracy because model needs best threshold for better classification. I got threshold 0.08 gives 66.97% accuracy. I think model is overfitting. I try different ways like change loss function, increase dataset but I didn’t get better accuracy.

Base model accuracy: -

```
[ ]  tr_acc, threshold = compute_accuracy_roc(np.array(pred), np.array(tr_y))
     tr_acc, threshold

     (0.7677416973849314, 0.03073887110920623)
```

Accuracy = 76.77% and Threshold = 0.03

Proposed Approach accuracy: -

```
tr_acc, threshold = compute_accuracy_roc(np.array(pred), np.array(tr_y))
tr_acc, threshold
```

```
(0.6697010869565218, 0.08133484799880535)
```

Accuracy = 66.97% and Threshold = 0.08

Model Tested Sample: -

```
predict_score()
```

Genuine Forged

```
Difference Score =  1.5050793
Its a Forged Signature
```

```
[ ] predict_score()
```

Genuine Forged

```
Difference Score =  0.009561546
Its a Genuine Signature
```

Conclusion

Introduced a Siamese network-based methodology for offline signature verification that uses writer independent feature learning. This technique, unlike its predecessors, does not rely on hand-crafted characteristics; instead, it learns them from data in a writer-independent environment. Experiments on the BHSig260 Synthetic dataset show that this is a first step toward modelling a general prototype for actual forgeries based on synthetically generated data. The writer independent method also used a Siamese Neural Network to create a model that exposed the network to a pair of similar and different observations, reducing the Euclidean distance between similar pairs while increasing it between dissimilar pairs. … Any machine learning classification model can be used in the future to better identify dissimilarity values.

During training and testing, this network takes a long time. This will be another task to improve the model's speed in the future.

References

- Convolutional Siamese Network for Writer Independent Offline Signature Verification (Sounak Deya, Anjan Duttaa, J. Ignacio Toledoa, Suman K.Ghosha, Josep Llad´osa, Umapada Palb)
- Offline Signature Recognition and Forgery Detection using Deep Learning (Jivesh Poddara, Vinanti Parikha, Santosh Kumar Bharti)
- Neural network based offline signature recognition and verification system. Research Journal of Engineering Sciences (Shikha, P., & Shailja S)
- Inverse Discriminative Networks for Handwritten Signature Verification (Ping Wei, Huan Li, Ping Hu) **CVPR 2019**
- SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification (Sounak Dey, Anjan Dutta, J. Ignacio Toledo, Suman K.Ghosh, Josep Llad´os, Umapada Pal) **CVPR 2017**