Siamese Triple Ranking Convolution Network in Signature Forgery Detection

Ojaswini Chhabra [*], Souradip Chakraborty [*]

[*] Data & Analytics, Walmart Labs, Bangalore

Abstract- Identifying a credible signature match based on a *base* signature of a person is an age-old problem. Despite recent automation and advances in this field using image recognition, a lot remains to be explored. In this paper, we develop an intelligent framework which can automatically detect a forged signature even if it is highly skilled, based on the developed feature embeddings and the corresponding algorithm. Siamese Triplet Convolution Neural Network is used to generate the feature embeddings for the signature images followed by a generalized Logistic Regression model to detect forgery. On the widely used SigComp dataset, our system achieves an accuracy of 96% in detecting forged signatures. Once the model is trained, it requires just one *base* image to determine whether another signature image is genuine or fraudulent with one shot learning. This algorithmic framework can be used in multiple commercial settings. One such example is testing customer or employee signatures on documents against a corresponding base signature saved beforehand.

Index Terms- Active and real-time vision, Fraud detection, Off-line signature recognition, Triplet loss.

I. INTRODUCTION

Signatures are widely relied upon for identity verification by business, financial organizations and governments to authorize transactions and documents. Accurate signature verification is imperative since forgery and fraud can cost organizations money, time, and their reputation. In the last few years, a lot of progress has been made in the field of automating signature forgery detection using machine learning and image recognition-based concepts.

Signature forgery can be broadly of two types:

- Blind Forgery: Where the forger has no idea what the signature to be forged looks like. This is easy to detect by machine because it is usually not very close to the appearance of a genuine signature.
- Skilled Forgery: Either simulation or tracing, in which the forger has a sample of the signature to be forged. In this case, detecting fraud requires more sophisticated tools to differentiate minute but critical details between genuine and forged signatures.

In this paper, an automatic off-line signature verification and forgery detection system using image processing and Deep Convolutional Siamese networks is proposed wherein a deep triplet ranking network is used to calculate the image embeddings. This is coupled with generalized linear model architecture with logistic loss functions and cross validation to arrive at the final model to label images as authentic or forged.

Training the model requires significant computation resources, but once the model is trained, it requires only one base image to determine whether another signature image is genuine or not with one shot learning. This process is instantaneous and can be carried out in real time. The main contribution of this paper is to enhance the robustness of the signature image embeddings using a FaceNet [1] based triplet network architecture with transfer learning using MobileNet CNN architecture [2]. The triplet loss

architecture with fine- tuned MobileNet CNN embeddings not only takes care of the separation between the genuine and forged signatures but also ensures the relative positioning of the genuine (positive) signatures in the embedding dimension.

An overview of the rest of the paper is as follows: in section II we review the literature in this area; section III delves into the method and framework developed in selecting triplets, learning the embeddings and the logistic loss parameters. Finally, in section IV we present some quantitative results of our embeddings.

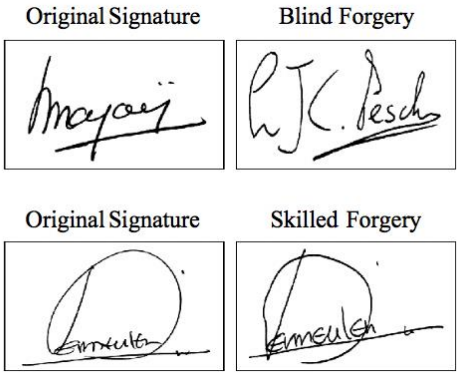

Fig. 1. Examples of blind and skilled forgeries from SigComp dataset

II. RELATED WORK

Offline signature verification is one of the most challenging tasks in the field of forgery detection and lot of work has been done in this field. Earlier methods to handle the problem include creating hand crafted features like block codes, wavelet, Fourier transformation etc. [3]. There is also a literature of work considering the geometrical and topological characteristics of local attributes such as position, tangent direction, blob structure, curvature as in Munich et al. [4].

Projection and contour based methods as shown in Dimauro *et al.* [5] gained a lot of popularity in this field. Other interesting approaches on direction profile [6], surroundedness features [7], grid-based features [8] etc. have also gained a lot of momentum in the past. There is also some literature related to structural methods where relationship between the local features is explored using graph-based matching [9].

Srinivasan *et al.* [10] explores the person-dependent and person-independent learning tasks which tend to capture both type of variances and thereby giving greater accuracy, but the method is highly sensitive to the number of samples per individual for consistency in estimation of the distribution.

Justino *et al*. [11] presents a very interesting learning process based on HMM where the objective is to get the best model that is able to represent each writer's signature while differentiating the intra-personal variation and interpersonal variation.

Drouhard *et al.* [12] proposes a neural network-based approach which uses a directional probability density function as a global shape factor wherein its discriminating power is enhanced by reducing its cardinality via filtering.

Alvarez *et al.* [13] focuses on building systems trained using VGG Convolutional neural network architecture with varying degrees of information, as well as experimenting with different objective functions to obtain optimal error rates.

Dey *et al*. [14] emphasizes on an offline writer independent signature verification task using a Siamese twin architecture which has the ability to learn complicated features to detect forgery which can't be achieved through hand-crafted features.

The architecture proposed in this paper uses depth-wise separable convolutions which significantly reduces the number of parameters when compared to normal convolutions with the same depth in the networks resulting in light weight deep neural networks. The triplet loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity as explained in Schroff *et al.* [1], thereby giving state of the art results in face recognition problems.

We use this triplet loss function in conjunction with logistic regression model to efficiently identify the complex patterns in the signatures to detect blind as well as skilled forgery.

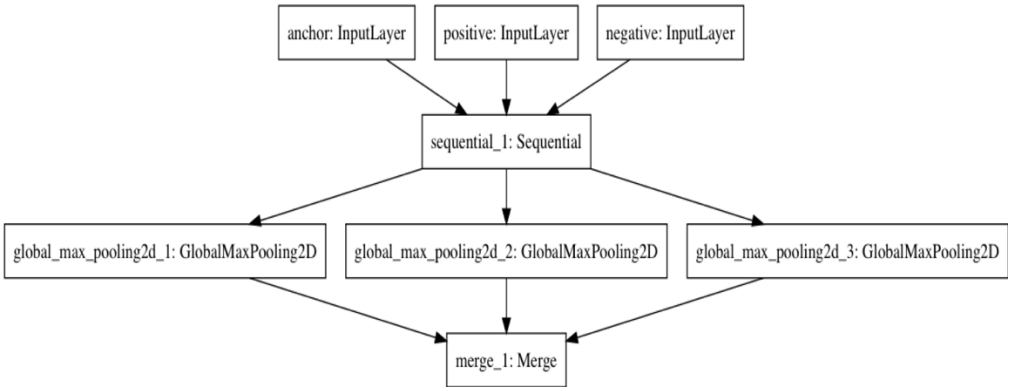

Fig. 2. Triplet loss architecture based on FaceNet [1]

III. METHOD

A. CNN Network Architecture

We use a deep convolutional Siamese network. Siamese convolution networks are twin networks with shared weights, which can be trained to learn the feature embed- dings where similar observations are placed in proximity and dissimilar are placed apart.

Triplet Loss function is used, wherein we contrive the data set in a triplet formation. This is done by taking an *anchor* image (genuine signature of a person) and placing it in conjunction with both a *positive* sample (another genuine signature of the same person) and a *negative* sample (a forged signature by someone else of the same person). This kind of framework ensures that the squared distance between two genuine signatures of the same individual is small, whereas the squared distance between a genuine and forged signature of an individual is large.

In implementing the CNN architecture, we have modified the pre-trained *MobileNet* CNN model [2] with additional layers and have used *Transfer Learning* in building the same. We have trained the last few layers and built dense layers on top of it to extract the embeddings with triplet loss function. Figure 3 shows the structure of convolution neural network based on [2] and the corresponding *trainable* and *non- trainable* parameters.

Layer (type)	Output Shape	Param #	Connected to
anchor (InputLayer)	(None, 128, 128, 3)	0	
positive (InputLayer)	(None, 128, 128, 3)	0	
negative (InputLayer)	(None, 128, 128, 3)	0	
sequential_1 (Sequential)	multiple	3228864	anchor[0][0] positive[0][0] negative[0][0]
global_max_pooling2d_1 (GlobalM	(None, 1024)	0	sequential_1[1][0]
global_max_pooling2d_2 (GlobalM	(None, 1024)	0	sequential_1[2][0]
global_max_pooling2d_3 (GlobalM	(None, 1024)	0	sequential_1[3][0]
merge_1 (Merge)	(None, 1)	0	global_max_pooling2d_1[0][0] global_max_pooling2d_2[0][0] global_max_pooling2d_3[0][0]

Total params: 3,228,864
Trainable params: 1,052,672
Non-trainable params: 2,176,192

Fig. 3. Structure and layer configuration of triplet CNN architecture based on MobileNet [2]

We use the Adam Optimizer with mean absolute error for back propagation to get to the final encodings. The image embeddings so obtained are such that the dissimilarity between the anchor image and positive image must be low and the dissimilarity between the anchor image and the negative image must be high for every triplet.

This kind of architecture ensures that even small differences in signatures can be captured in order to flag a skilled forgery.

The loss that is being minimized is then

$$Loss = \sum_{i=1}^{N} [\|f(x_{i=1}^{a}) - f(x_{i=1}^{p})\|_2^2 - \|f(x_{i=1}^{a}) - f(x_{i=1}^{n})\|_2^2 + \alpha] \qquad (1)$$

where,

f(a) refers to the image encoding of the anchor a

f(p) refers to the image encoding of the positive p

f(n) refers to the image encoding of the negative n

α is a constant used to make sure that the network does not try to optimize towards

f(a)- f(p) = f(a)- f(n) = 0

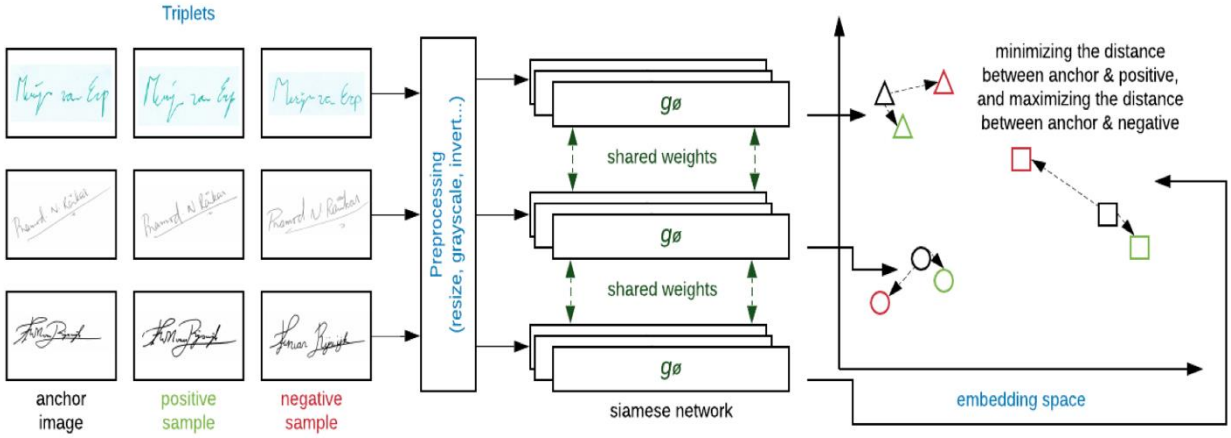

Fig. 4. Training the model using Deep Triplet Ranking CNN Network

B. Logistic Regression Model

Once the final image embeddings for all the training images is attained through the triplet loss architecture, we train a generalized linear model with logistic loss function to get the final model that declares any signature as genuine or forged against a base image signature.
For training this logistic function, we arrange the images in a pairwise manner where each observation is pair of images, either both of a person's genuine signature, or one of person's genuine signature, and the other as person's forged signature.
These will have labels (class) genuine or fraud assigned to them respectively. We use cross validation to get to the final logistic model taking the corresponding differences between the embeddings of each of the pairs (1024 length difference vector of embeddings) as the feature set and the class labels (genuine/ fraud) as the dependent variable y.

The loss function pertaining to this is

$$Loss = g(\sum_{i=1}^{N}(\sigma(W^T X_i(x_i^a, x_i^b) + b)), y_i) \quad (2)$$

where,
$g()$ is the Logistic Loss Function
y_i is the response class (0/1) for image pair i
x_i^a, x_i^b are the image embeddings from the trained CNN network of images in the image pair i, where x_i^a is a person's genuine signature, and x_i^b is either genuine or forged
$X_i(x_i^a, x_i^b)$ represents the entire set of difference based features computed from the corresponding embeddings of image pair i in space IR^m.
W and b represent the final weights obtained from generalized linear logistic loss function.

C. Final Framework Architecture

Once the training process is completed, all we need is the trained model outputs (encodings and logistic model weights). Next, we create a database framework to save the original signatures of every new individual against a unique ID. We pass these base images through the encodings we obtained in the training process and get the corresponding image embeddings. This is in the format of a vector of length 1024. We precompute these embeddings and save them against the individuals' unique ID in the database. Now, whenever we acquire a new signature image against one of the unique ID's that needs to be accessed to determine whether it is genuine or fraud, the framework passes that image through the encoding once again to get its image embeddings.

This new image embedding is then compared to its corresponding embedding of the base image of that individual to determine whether it is genuine or forged. This is done by taking the difference vector of the two embeddings and passing it through the logistic model to get the final prediction. If the resultant probability from the logistic model is low, then the framework declares the new image as a genuine, otherwise it is considered a forgery.
The test workflow is depicted below in Figure 5 and the framework flowchart is depicted in Figure 6.

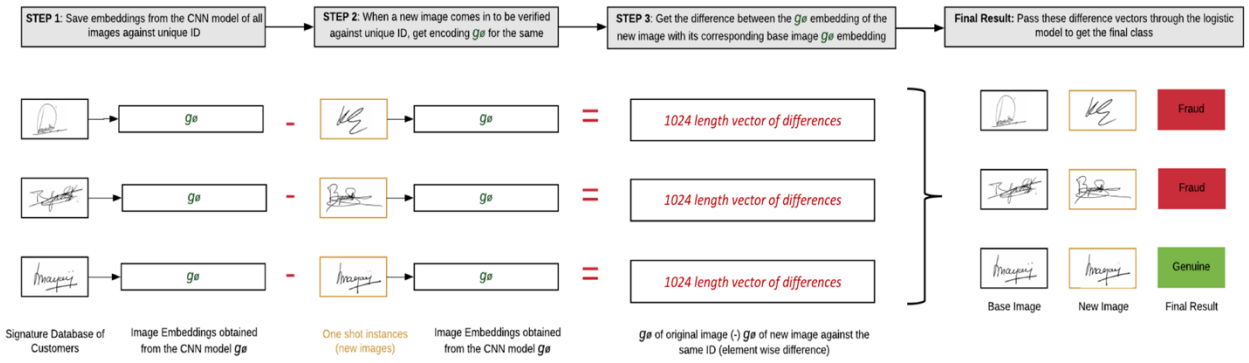

Fig. 5. Testing process of determining whether signature is genuine or fraud

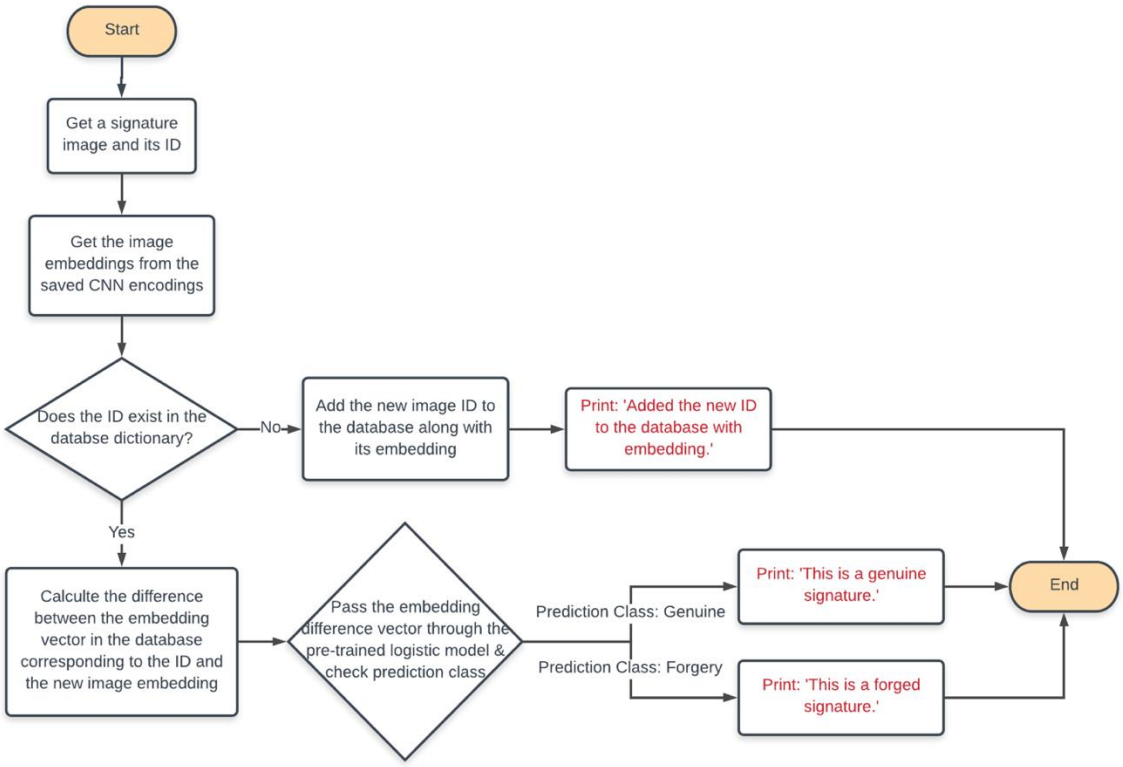

Fig. 6. Execution Framework Flowchart

IV. DATASETS AND EVALUATION

We evaluate our method on the SigComp datasets which are widely used in this field. The entirety of SigComp 2009, SigComp 2010, and SigComp 2012 was used. For SigComp 2011, we only used the Dutch offline dataset.

We create all combinations of triplets from this data: this is done by taking an anchor image (genuine signature of a person) and placing it in conjunction with both a positive sample (another genuine signature of the same person) and a negative sample (a forged signature by someone else of the same person). A total of 120k such triplet combinations was obtained. We resize all the images and convert them to arrays to be passed in the CNN triplet model.

A. Logistic Regression Model

We keep a holdout set of around 20%, that has the same distribution as our training set, but disjoint identities. For training the logistic model, we use 10-fold cross validation on the image pairs obtained.

[1] These datasets are from ICDAR Signature Verification Competition from years 2009-2012

TABLE I
RESULTS OF ACCURACY, PRECISION AND RECALL

Data	Accuracy	Precision	Recall
Training	98.49%		
20% Hold Out Test Set	96.54%	95.28%	96.75%
Average 10-fold CV	95.93%	94.74%	96.01%
Cross Validation fold 1	96.68%	95.83%	96.60%
Cross Validation Fold 2	95.37%	94.35%	95.10%
Cross Validation Fold 3	95.55%	94.37%	95.50%
Cross Validation Fold 4	95.76%	94.75%	95.60%
Cross Validation Fold 5	96.24%	95.07%	96.40%
Cross Validation Fold 6	96.24%	95.70%	95.70%
Cross Validation Fold 7	95.85%	94.66%	95.90%
Cross Validation Fold 8	96.37%	95.08%	96.70%
Cross Validation Fold 9	95.63%	94.20%	95.90%
Cross Validation Fold 10	95.59%	93.42%	96.70%

B. Personal Signature Images

We also curated a database of our personal signatures (both genuine pairs and forged) with clean labels to test the model output. This consisted of more than 50 signature pairs.

C. Results

Our model achieved a training accuracy of 98.5%. On the 20% hold-out set, it attained a test accuracy of 96.5%. The precision and recall for the same was 95.2% and 96.7% respectively.

The results from the 10-fold cross validation are presented in Table I. This gave an overall accuracy of above 96% which denotes that the model has high accuracy and generalizability, and is significant as well.

V. CONCLUSION

The differences between the images of a genuine signature and its skilled forgery is at times very minute and is challenging to detect by even a trained eye. Deep Triplet loss function is a very powerful loss function used in the industry for face recognition [1]. We have created our custom triplet model architecture with modified MobileNet CNN and dense layers with triplet loss function. Based on this loss, the image embeddings are created in such a way that the dissimilarity between the anchor image and positive image must be low and the dissimilarity between the anchor image and the negative image must be high for every triplet. This kind of architecture ensures that even small differences in signatures can be captured in order to flag a skilled forgery effectively.

REFERENCES

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 **IEEE** Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815-823, 2015.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mo- bilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR,* vol. abs/1704.04861, 2017.

[3] M. K. Kalera, S. Srihari, and A. Xu, "Off-line signature verification and identification using distance statistics," *International Journal of Pattern Recognition and Artificial Intelligence,* pp. 228-232, 2003.

[4] M. Munich and P. Perona, "Visual identification by signature tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vo1. 25, pp. 200— 21 7, 03 2003.

[5] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, "A multi-expert signature verification system for bankcheck processing," *International Journal of Pattern Recognition and Artificial Intelligence,* vo1. 11, pp. 827-843, 08 1997.

[6] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 993-997, June 2005.

[7] R. Kumar, J. D. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," *Pattern Recogn. Lett.,* vo1. 33, no. 3, pp. 301-308, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2011. 10.009 .

[8] R. Chandra and S. Maheskar, "Offline signature verification based on geometric feature extraction using artificial neural network," *2016 3rd International Conference on Recent Advances in Information Technology (RAIT1,* pp. 410—414, March 2016).

[9] S. Chen and S. Srihari, "A new off-line signature verification method based on graph" vo1. 2, 01 2006, pp. 869–872.

[10] H. Srinivasan, S. Srihari, and M. J. Beal, "Machine learning for signature verification," vol. 4338, 01 2006, pp. 761–775.

[11] E. Justino, M. El Yacoubi, F. Bortolozzi, P. Pontifcia, and R. Imaculada Conceio, "An off-line signature verification system using hmm and graphometric features," 08 2002.

[12] J.-P. Drouhard, R. Sabourin, and M. Godbout, "A neural network approach to off-line signature verification using directional pdf," *Pattern Recognition,* vol. 29, pp. 415–424, 03 1996.

[13] G. Alvarez, B. Sheffer, and M. Bryant, "Offline signature verification with convolutional neural networks," Tech. Rep.

[14] S. Dey, A. Dutta,J. IgnacioToledo, S. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional Siamese network for writer independent offline signature verification," 07 2017.