Offline Handwritten Signature Verification using CNN inspired by Inception V1 Architecture

Ramesh Kumar Mohapatra
Department of CSE
NIT Rourkela
Rourkela, India
mohapatrark@nitrkl.ac.in

Kumar Shaswat
Department of CSE
NIT Rourkela
Rourkela, India
ku.shaswat1996@gmail.com

Subham Kedia
Department of CSE
NIT Rourkela
Rourkela, India
subhamkedia799@gmail.com

***Abstract*—In the field of behavioral biometric, signature verification is most referenced procedure for authentication of a person. A signature is considered to be the "seal of approval" for verifying the approval of a user and remains the most preferred means of authentication. This verification system mainly aims at verifying the discriminating the forged signature (forged by an imposter) from the genuine signatures. In this paper, Convolutional Neural Networks (CNN) have been utilized to learn features from the pre-processed genuine signatures and forged signatures. The CNN used is inspired by Inception V1 architecture(GoogleNet). The architecture uses the concept of having different filters on same level so that the network would be wider instead of deeper. In this paper, the proposed model is tested on few publicly available datasets such as CEDAR, BH-Sig260 signature corpus, and UTSig.**

Keywords*—*Handwritten Signature, Biometrics, Convolutional Neural Network, Inception V1

I. Introduction

A signature is a handwritten (and often stylized) depiction of someone's name, nickname, or even a simple "X" or other mark that a person writes on documents as a proof of identity and intent. It acts as a proof of an individual. The writer of a signature is a signatory or signer. Signatures have been a significant piece of human innovation and character for truly thousands of years, and in advanced we use them to do everything from sign receipts to authenticate documents, sign autographs and write birthday cards.

In this modern era, biometrics is applied on every domain for security purpose. The point of such frameworks is to perceive an individual dependent on physiological or behavioral traits. The proof is finished by estimations of natural traits, for example, the unique finger impression, face, iris, and so forth. The later case is concerned with behavioral traits such as voice and the handwritten signature. Biometric systems are mainly employed in two scenarios:

1) Verification, and
2) Identification.

In the first case, the user provides a biometric sample and claims his/her identity. The verification system checks for the authenticity of the user. The situation is different in the case of identification. The main objective in case of identification is to recognize an individual from the available batch.

Signature verification systems automatically distinguish if the biometric sample is really of a claimed individual. They are used to categorize query signatures as genuine or forgeries. Forgeries are commonly categorized into three types:

1) Random forgeries
2) Simple forgeries
3) Skilled forgeries

The random forgeries cases are the ones in which the imitator is devoid of any information about the user or the signature. The imitator uses his/her own signature to commit forgery. The signature produced is different in the semantics and overall has a different style to it. The simple forgeries are the ones in which the imitator has a vague idea about the name but is devoid of the signature format. This case may resemble a bit with the real signature, mostly those cases in which the users use their full signatures. The skilled forgeries are the ones in which the imitator has an idea of the users name as well as the users signature. They observe and practice reproducing the same signature time and again. These forgeries have lot many similarities to the genuine signatures and are really tough to distinguish.

Signature verification systems are sub categorized into two groups: Online (dynamic) and Offline (static). When the signature is being produced, the motion of the stylus is taken into account in the Online method. It also uses the location, pressure exerted by the pen, speed at which the signature is done and the acceleration produced as functions of time. The dynamic features are unique to each individual and also repeat themselves over the course of the duration. The algorithms analyse the shape created, the stroke order, the speed achieved, the pen pressure and the timing information captured while signature is being done. The method used in the offline verification process is different. Here, the signature acquisition is done only after the writing process is over. Digital image is used for representation of the signature.

Signature Verification System (SVS) helps in recognizing a customer by offering the organization the signature or customer image to an operator. The easiest and most secure identification method is hand written signature. Signatures have always played a significant role in many sectors which are as follows:

1) Financial,
2) Commercial and
3) Legal transactions.

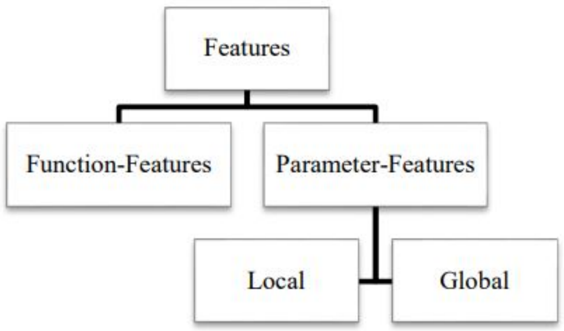

Fig. 1. Feature categories

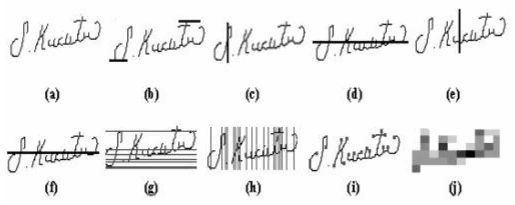

Fig. 2. Feature extraction steps: (a) preprocessed signature and (b) height, (c) maximum vertical histogram, (d) maximum horizontal histogram, (e) horizontal center, (f) vertical center, (g) horizontal local maxima numbers, (h) vertical local maxima numbers, (i) edge points, (h) grid features of the signature.

In this paper, we focus on the geometric features based on gray scale information from image containing handwritten signatures.

The remainder of this paper is organized as follows: Section II reviews the related work on signature verification. Section III provides the proposed algorithm. Section IV presents and discuss the results of our experiment. Section V concludes the paper and describes the future scope.

II. RELATED WORKS

A. Pre-Processing

Likewise with most pattern recognition issues, pre-processing assumes a significant job in signature check. Signature samples may exhibit varieties as far as pen thickness, scale, pivot, and so forth., even among genuine signatures of an individual. This step removes noise and converts the image to the desired format. According to Han [7], this step removes background, performs thresholding, noise cleaning, etc.

B. Feature Extraction

Offline signature verification has been examined from many perspectives, yielding numerous choices for feature extraction [14], [6], [8], [13], [21]. As shown in Fig. 1, the parameter feature extraction are of two types: Global and Local features. Global features considers the signature as whole and it includes features such as height, width, skew angle, and many more. Where as local features are so-called because of their relation to each point of the signature and it describes a part of image. The local features are obtained by segmenting the image or using grids over the image to get features from each grids.

The overall shape of the signature can be obtained from the geometric properties which incorporates essential descriptors, for example, the basic geometric property of the signature like width, height or properties related to the overall signatures. Increasingly unpredictable descriptors incorporate the tally of endpoints and closed loops [2]. As shown in Figure 2, Signature height, Width, Image area, Maximum horizontal projection, Maximum vertical projection, Number of edge points, Number of closed loops, Vertical center of the signature, Global slant angle, Local slant angle are common features used. The grip representation of the signature to extract few features from the local cells in the grid is also one of the used methodologies used by researchers for the verification. For instance, utilizing the pixel thickness inside cell [2]. Graphology or graphometry as defined as analysis of physical characteristics and patterns of handwriting are commonly used by forensic document examiners for detecting the authenticity of the signer. Graphology and graphometry features were investigated by Oliveira et al [9]. They selected few features in respect to differentiate the signature and proposed a feature vector for the following static features:

1) Calibre - Ratio of height of the signature to width of the signature.
2) Spacing - describing empty spaces between strokes.
3) Alignment to baseline - describing the angular displacement to an horizontal baseline.

The geometric feature extractor or hand-engineered feature extractor have been used less in the recent years because of the rise of the Deep Learning models for classification. Rantzsch et al. [15] used metric learning by calculating the distance between the signatures in a writer independent approach.

Amir Soleimani et al. [18] proposed Deep Multitask Metric Learning (DMML) which uses the knowledge from the similarities and dissimilarities between the genuine and forged samples of other classes too. They have mixed the concept of Writer-Independent and Writer-dependent approaches multitask and transfer Learning with Discriminative Deep Metric learning (DDML) method.

III. PROPOSED MODEL

In this paper, features are learned from the signatures and make feature set, train the model for a writer independent classifier and then evaluate the model from the test set. In this work, writer independent features for each user have been trained using the Convolutional Neural Network. Convolutional neural networks have been used to exploit the fact the forged signature and genuine differ from some key points.

Here, the genuine and forged signature are considered as two classes. The inherent properties of a handwritten signature are the main features that has to be extracted for feature space. Accordingly, the train classifiers for both the classes with the aim of finding the unique differences, using this feature space, that model the characteristics of each user. We consider two preprocessed images of the signature and then use an architecture which is inspired by GoogLeNet(Inception V1) [19] architecture as shown in Fig. 3.

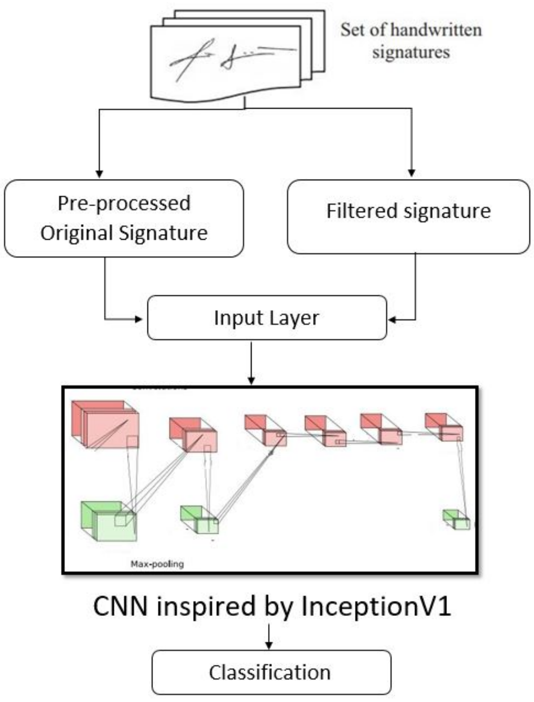
Set of handwritten signatures
Pre-processed Original Signature
Filtered signature
Input Layer
Max-pooling
CNN inspired by InceptionV1
Classification

Fig. 3. The Proposed algorithm

A. *Pre-Processing*

The sample signature images are pre-processed with different filters, so that CNN can learn the image properly. In the processing section, two images from the original signature are being generated which is fed to the network.

1) Filtered Signature: The image is transformed into grayscale image for better understanding since color doesn't matter in signature. The image is passed through a dilate filter using a kernel size of 3×3 and then through a Gaussian blur to reduce the noise. The image has been binarized using the Otsu methodology [10]. After applying the above filters, the edges get prominent, so the signature area is extracted using drawing a rectangle along the signature. Further, the image is resize to 256×128. The contour is found out from the signature to highlight the edges of the signature. As shown in Fig. 4, the normalized image is fed to the CNN.

2) Grayscale Signature: Similar to the filtered signature, the image is transformed to grayscale image for better understanding. From the grayscale image, the signature is extracted along the same dimension of the rectangle found in the first image. Then the image is resize to 256×128. The extracted image is normalized as shown in Fig. 5 and is fed to the CNN .

B. *Writer Independent CNN Training*

Convolutional Neural Networks are commonly used for the problems like signature which needs classification. The CNN architecture have smaller number of trainable parameter which makes then better to use for this case. The limitation that we can reduce the signatures image above some threshold of losing the particulars of the signatures makes CNN architecture favourable for this problem statement. We also note that this type of architecture shares some properties with handcrafted feature extractors used in the literature, as features are extracted locally (in an overlapping grid of patches) and combined in non-linear ways (in subsequent layers).

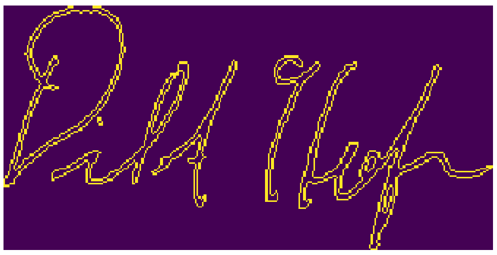

Fig. 4. The filtered signature

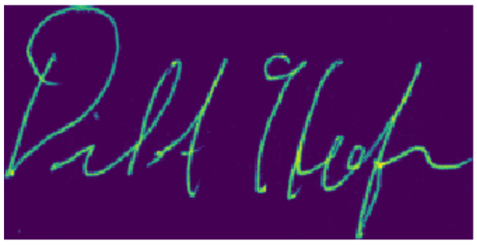

Fig. 5. The grayscale signature

Inception V1 was a significant achievement in the improvement of CNN classifiers. Before its commencement, most prominent CNNs simply stacked convolution layers deeper and deeper, wanting to show signs of improvement execution. On the same level, the networks has more than one filters parallel which makes it more "widers" rather going than deeper like previous networks. The authors designed the inception module to reflect the same. The below Figure 6 is the "naive" inception module. The convolution happens with three sizes of filter (1×1, 3×3, 5×5) on same level along with a max pooling. The outputs from this wider layer are concatenated and feed as input to next layer.

In the below architecture, we use the inputs of both the genuine and forged signatures for the feature learning process. The approach considers the forgeries as one class and genuine signature as another class.

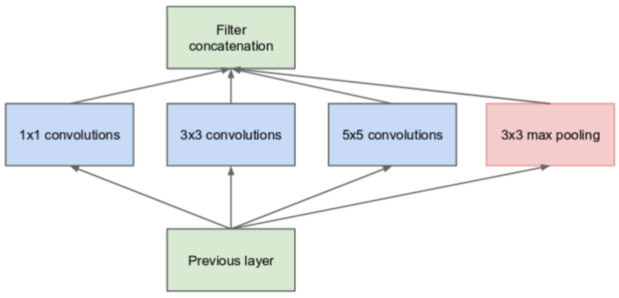

Fig. 6. The Inception V1 Ideology [20]

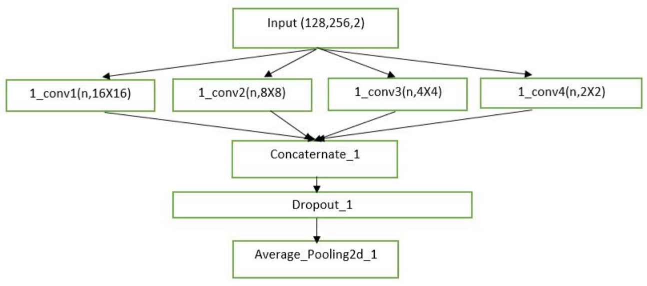

Fig. 7. The proposed CNN Inception Network

TABLE I. INCEPTIONSVGNET MODEL SUMMARY

Layer (type)	Output Shape	Number of Parameters
Input	(None, 128, 256, 2)	0
1_conv1,1_conv2, 1_conv3,1_conv4	(None, 128, 256, 16)	8208,2064,528,144
concatenate_1 (Concatenate)	(None, 128, 256, 64)	0
dropout_1 (Dropout)	(None, 128, 256, 64)	0
average_pooling2d_1 (pooling)	(None, 64, 128, 64)	0
2_conv1,2_conv2, 2_conv3,2_conv4	(None, 64, 128, 24)	393240,98328, 24600,6168
concatenate_2 (Concatenate)	(None, 64,128,96)	0
dropout_2 (Dropout)	(None, 64,128,96)	0
average_pooling2d_2 (pooling)	(None, 32,64,96)	0
3_conv1,3_conv2, 3_conv3,3_conv4	(None, 32,64,32)	786464,196640, 49184, 12320
concatenate_1 (Concatenate)	(None, 32,64,32)	0
dropout_3 (Dropout)	(None, 32,64,32)	0
average_pooling2d_3 (pooling)	(None, 16,32,128)	0
f_conv1,dropout_4 (Dropout)	(None, 16,32,64)	131136,0
average_pooling2d_4 (Flatten)	(None, 8,16,64)	0
flatten1 (Flatten)	(None, 8192)	0
f_dense1 (Dense),dropout5 (Dropout)	(None, 256)	2097408,0
f_dense2 (Dense), dropout6 (Dropout)	(None, 256)	65792,0
dense_final (Dense)	(None, 2)	514
activation1 (Activation)	(None, 2)	0
-	**Total Parameters**	**3,878,178**

C. InceptionSVGNet Architecture

We have used an architecture inspired by Inception Model(GoogLeNet). Figure 7 shows the layer which is used for widening the network instead of going deeper. We have used 3 such layers in the network with filter numbers as 16,24 and 32 as shown in the Figure 8(n denotes filter number). There are 4 convolutional layer with kernel size as 16×16, 8×8, 4×4 and 2×2 respectively. After the three wide layers, we have one more convolutional layer with 16 filters along with 2 flatten layers and 3 dense layer. At the end, by using the softmax we have done the classification. The Table I shows the detailed CNN Model summary along with the parameters.

IV. EXPERIMENTS

InceptionSVGNet has been evaluated on four most used datasets in this field of research, viz., (1) CEDAR, (2) BHSig260 signature corpus (3) UTSig [17].

A. Dataset

1) CEDAR: It contains signatures of 55 signers having a place with the different social and expert background. Here for every user, 24 forged signatures and 24 genuine signatures are considered.

2) BHSig260: This dataset contains Hindi and Bengali language signatures. This has 100 signers from Bengali and around 160 from Hindi therefore a total of 260 signers. Each user comprises of 24 genuine and 30 forged signatures.

3) UTSig: UTSig dataset comprises of signature signed in Persian language and contains a total of 8280 signatures. There are 115 individuals whose signatures are recorded. For each individual, there are 27 genuine signatures and 45 forged signatures.

B. Experiment Protocol

The Convolutional Neural Networks are trained on a set $\mathcal{L}_c$ (denoting Learning set for classification) consisting of K users for each dataset. We monitor the progress on a validation set $\mathcal{V}_c$ (Validation set for classification). The dataset has been split into 70% of the signatures for training, and 30% for this validation set and test set. The batch size is 160 and dropout rate is 0.5.

C. Experimental Setup

For the above experiments to be conducted following computational setup were done.

1) *System Configuration* : The CPU platform is Intel Skylake which has 8 vCPUs, 32 GB memory. GPU used is NVIDIA Tesla V100. The above configuration has been set up in Google Cloud Platform(GCP).
2) *Operating System*: Debian GNU/Linux 9.7
3) *Software Package*: Jupyter Notebook for Python Scripting, Numpy, Scikit-learn [12], OpenCV [3], Tensorflow [1] and Keras [4].

D. Results & Discussion

From Table III, it is evident that for most of the dataset our network has outperformed the other methods. In the case of CEDAR dataset, our model has the same accuracy as the other methodology. The major significant improvement has been seen in the Bengali and Hindi dataset in which accuracy has significantly improved. UTSig is a challenging dataset which is at par with the existing network. Overall, the InceptionSVGNet has performed well in the above conditions set for it.

TABLE II. COMPARISON OF THE INCEPTIONSVGNET WITH THE STATE-OF-THE-ART METHODS ON VARIOUS SIGNATURE DATABASES

Database	State-of-the-art Methods	#Signers	Accuracy	FAR	FRR
CEDAR	Graph matching (Chen and Srihari [16])	55	92.10	8.20	7.70
	Dutta *et al.* [5]	55	100.00	0.00	0.00
	InceptionSVGNet	55	**100.00**	**0.00**	**0.00**
Bengali	Pal *et al.* [11]	100	66.18	33.82	33.82
	Dutta *et al.* [5]	100	84.90	15.78	14.43
	InceptionSVGNet	100	**97.77**	**3.88**	**2.22**
Hindi	Pal *et al.* [11]	100	75.53	24.74	24.47
	Dutta *et al.* [5]	100	85.90	13.10	15.09
	InceptionSVGNet	100	**95.40**	**6.38**	**3.33**
UTSig	Amir Soleimani *et al.* [18]	115	84.04	16.15	12.96
	InceptionSVGNet	115	80.44	23.34	14.01

V. CONCLUSION AND FUTURE SCOPE

Throughout the most recent decade, there are many strategies for Offline Signature Verification. While recognizing genuine signatures and forged signatures remains a difficult task, false acceptance rates have dropped fundamentally in the most recent couple of years because of progressions in Deep Learning. We presented different formulations for learning representations for offline signature verification. Analyzing the above results, it shows that Convolutional Neural Networks is better for classification of signature. The intuition to classify between genuine signatures and forgeries (regardless of the user), by learning the visual cues has improved the accuracy. The significant improvement in the accuracy is also due to the new architecture inspired from GoogleNet which worked more widely than going into deeper. Hence from the above experimental results, it is clear that due to the InceptionSVGNet Architecture is more efficient in identifying patterns in image by using the wider networks.

This pattern will proceed for future work, with researchers proceeding to investigate better feature set (using Deep Learning networks), and approaches to improve arrangement with limited number of tests. Techniques based on ensembles of classifiers, specifically methods for dynamic choice are likewise encouraging procedure in this field.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.

[2] H. Baltzakis and N. Papamarkos, "A new signature verification technique based on a two-stage neural network classifier," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 1, pp. 95 – 103, 2001.

[3] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*, 2nd ed. O'Reilly Media, Inc., 2013.

[4] F. Chollet *et al.*, "Keras (2015)," 2017.

[5] A. Dutta, U. Pal, and J. Llads, "Compact correlated features for writer independent signature verification," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 3422–3427.

[6] S. Ghandali and M. E. Moghaddam, "A method for off-line persian signature identification and verification using dwt and image fusion," in *2008 IEEE International Symposium on Signal Processing and Information Technology*, Dec 2008, pp. 315–319.

[7] K. Han and I. K. Sethi, "Handwritten signature retrieval and identification," *Pattern Recognition Letters*, vol. 17, no. 1, pp. 83 – 90, 1996.

[8] K. Huang and H. Yan, "Off-line signature verification based on geometric feature extraction and neural network classification," *Pattern Recognition*, vol. 30, no. 1, pp. 9 – 17, 1997.

[9] L. S. Oliveira, E. Justino, C. Freitas, and R. Sabourin, "The graphology applied to signature verification," in *12th Conference of the International Graphonomics Society*, 2005, pp. 286–290.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.

[11] S. Pal, A. Alaei, U. Pal, and M. Blumenstein, "Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, April 2016, pp. 72–77.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[13] M. Piekarczyk, "Hierarchical random graph model for off-line handwritten signatures recognition," in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, Feb 2010, pp. 860–865.

[14] M. R. Pourshahabi, M. H. Sigari, and H. R. Pourreza, "Offline handwritten signature identification and verification using contourlet transform," in *2009 International Conference of Soft Computing and Pattern Recognition*, Dec 2009, pp. 670–673.

[15] H. Rantzsch, H. Yang, and C. Meinel, "Signature embedding: Writer independent offline signature verification with deep metric learning," in *Advances in Visual Computing*. Cham: Springer International Publishing, 2016, pp. 616–625.

[16] Siyuan Chen and S. Srihari, "A new off-line signature verification method based on graph," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, Aug 2006, pp. 869–872.

[17] A. Soleimani, K. Fouladi, and B. N. Araabi, "Utsig: A persian offline signature dataset," *IET Biometrics*, vol. 6, no. 1, pp. 1–8, 2017.

[18] A. Soleimani, B. N. Araabi, and K. Fouladi, "Deep multitask metric learning for offline signature verification," *Pattern Recognition Letters*, vol. 80, pp. 84 – 90, 2016.

[19] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[20] S.-H. Tsang. (2018) Review: Googlenet (inception v1) winner of ilsvrc 2014 (image classification).

[21] E. zgndz, T. entrk, and M. Elif Karslgil, "Off-line signature verification and recognition by support vector machine," in *2005 13th European Signal Processing Conference*, Sep. 2005, pp. 1–4.