INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 11.10.2021

Accepted: 20.01.2022

Published: 16.02.2022

Citation: Raj AD, Venugopal P, Padmapriya N (2022) Identification of Proper and Improper Signatures Using Graph Theory Techniques. Indian Journal of Science and Technology 15(6): 227-236. https://doi.org/10.17485/IJST/v15i6.1913

* **Corresponding author**.

danielraja@ssn.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Identification of Proper and Improper Signatures Using Graph Theory Techniques

A Daniel Raj[1]*, P Venugopal[2], N Padmapriya[1]

1 Department of Mathematics, Sri Sivasubramaniya Nadar College of Engineering (Autonomous), 603110, Kalavakkam, Chennai, India
2 Mathematics, School of Science and Humanities, Shiv Nadar University Chennai, Chennai, India

Abstract

Objectives: To propose an automatic signature identification for off-line signature utilising graph theory approaches. **Methods:** Scanned signatures (Kaggle, https://www.kaggle.com/divyanshrai/handwritten-signatures/data) are collected for off-line signature data. The method follows pre-processing, vertex point extraction by midpoint traverse method, features extraction using edge, average edge and average edge D-distance and Support Vector Machine (SVM) to classify and predict the true label for the genuine and forged signatures. False Acceptance Ratio (FAR) and False Rejection Ratio (FRR) give the accuracy of the proposed methods. This off-line signature verification method is compared with the deep learning techniques existing in the literature. **Findings:** Support Vector Machine (SVM) used for classification and results on standard signature datasets like ICDAR (International Conference on Document Analysis and Recognition). The results demonstrate how the proposed strategy outperforms the state-of-the-art already available. **Novelty:** The proposed approach use the edge distance, average edge distance, and average edge D-distance inbuilt graph structures to extract the feature points.

Keywords: Signature images; grid approach; bipartite graph; complete bipartite graph; mid point traverse method

1 Introduction

In the digital world, the signatures have to be uploaded with most of the online documents. But these signatures can be easily scanned or impersonated to create counterfeit or improper documents. The proper signatures are forged based on the details that are handy or obtainable to the forger. Using the available data, a forger can imitate a signature. Such imitated signatures are said to be improper signatures that can be differentiated only by the signer. But every time, a signer cannot be called to identify the originality of the signature. Hence, identifying the valid or genuine signature from an invalid signature has become an important area of study. Signature identification of scanned or photographed images is still an important unsolved problem in pattern recognition[1].

This article proposes a novel algorithm for signature identification. This algorithm constructed a structural graph using mid-point traverse method (MPTM). From the structural graphs, the features of the signatures are computed and classified using SVM. False Acceptance Ratio (FAR), False Rejection Ratio (FRR) of genuine and forged signatures results in the accuracy of the identification[(2)].

In general, there are two major methods of signature identification. The first one is an online method and the second one is an off-line method. The online method identifies and measures sequential data such as handwriting and pen pressure with a special device. In contrast, the off-line method uses an optical scanner (scanner, mobile camera, etc.,) to obtain handwritten data on paper[(3)]. In general, there are two significant ways to tackle this pivotal step, viz., the statistical and the structural approach. The graph-based pattern representation is commonly used in structural pattern recognition[(4)]. Nowadays, off-line signatures are used mostly than online signatures. These off-line signatures are more vulnerable to duplication and are used to create forged documents. Manual identification of signatures is time-consuming; hence the automation of identification of signatures is essential. The task of an automated handwritten signature identification system is to verify a person's identity based on his signature[(5)].

There are many identification methods to verify whether a signature in a document is genuine or not. Vohra K et, al[(6)] proposed a novel approach by classifying signatures and accuracy determined using SVM and CNN methods. Features extracted are histogram of gradient, shape, aspect ratio, bounding area, contour area and convex hull area in data set ICDAR. Xamxidin N et, al[(7)] proposed fusion of TAS (Threshold adjacency statistics) and HOG (Histogram of oriented gradients) feature for off-line handwritten signature verification. It's can effectively describe the threshold feature and gradient feature signature image. Zhou Y et al[(8)] proposed a score fusion method based on accuracy (SF-A), which combines off-line and online features through fusion and effectively utilises the complementarity among classifiers. Sharif et al.[(9)] proposed a genetic algorithm to select the appropriate features for signature verification. Then, they used the SVM classification by using the selected features. Maergner et al.[(10,11)] described the graph-based signature verification system. This system was a combined model consisting of key points graphs with approximate graph edit distance and inkball models. Bhunia AK et al.[(12)] presented the writer dependent (WD) method for signature verification by combining the hybrid texture features based on one SVM.

For structural analysis of signature, graph theory techniques can be employed as it analyses the signature more accurately as each point in the signature is considered for verification; this helps to identify the genuineness of a signature more exactly. A summary of the existing methods and their classification techniques is given in Table 1.

Table 1. Summary of the existing methods and their classification techniques

Methods	Features	Classifier
Guerbai et al. 2015[(13)]	Curvelet transform	RBF-SVM, MLP
Pham et al. 2015[(14)]	Geometry-based features.	Likelihood ratio
Serdouk et al. 2016[(15)]	Gradient Local Binary Patterns (GLBP) and LRF	k-NN
Pal et al. 2016[(16)]	Uniform Local Binary Patterns (ULBP)	Nearest Neighbor
Loka et al. 2017[(17)]	Long range correlation (LRC)	SVM
Zois et al. 2019[(18)]	Lattice arrangements and Pixel distribution	Decision tree
Sharif et al. 2020[(19)]	Local pixel distribution	GA, SVM
Batool et al. 2020[(20)]	GLCM, geometric features	SVM
Ajij M et al. 2021[(21)]	Quasi-straight-line segments	SVM
Proposed Method	Average edge distance	SVM

This paper is organised as follows. Section 2 talks about the pre-processing of the image of the input signature and then extraction of the points of the image using grid merging algorithm and Mid-Point Traverse Method (MPTM). Section 3 discusses constructing the bipartite graph and complete bipartite graph based on the extracted points followed by classification by SVM. Section 4 validates the proposed method.

2 Methodology

2.1 Proposed method

This section proposes an off-line signature automatic identification method that can be used to identify the signature's authenticity

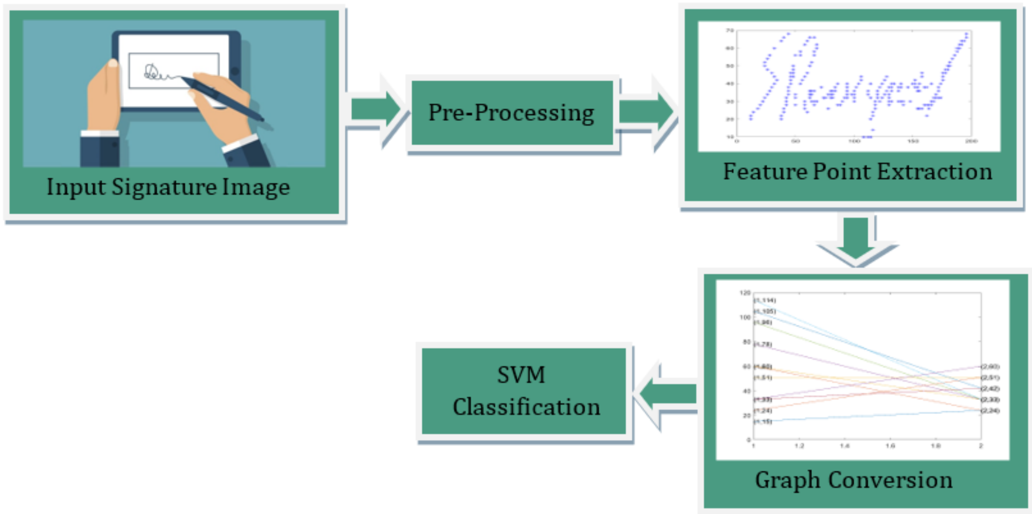


Fig 1. Block diagram of the off-line automatic identification method

[Figure 1] describes the work flow of the off-line automatic identification method. The input image is the signature image. Pre-processing has been applied to this image to extract the points. The set of points is extracted using MPTM. In the XY plane, the points are denoted by (x,y). Construct the bipartite graph and complete the bipartite graph between the two sets X and Y. These feature points are fed into the SVM classifier to classify the genuine and the fake signatures. FAR and FRR are calculated for each signature.

2.2 Point Extraction of Signature

The extraction of points and the construction of the graphs are illustrated in the following steps:

2.2.1 Pre-processing

Fig 2. Input signature image

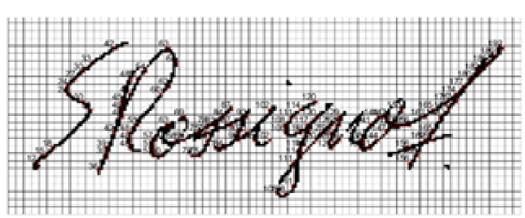


Fig 3. Mid-Point Traverse Method (MPTM) of the signature in Fig 2

The pre-processing involves the following steps:

- **Input** - Input a scanned copy of the signature images.
- **Image Resize** - Resize the image, identity scale factor and use the default interpolation method.
- **Binarisation** - The grey-scale image of the signature is binarised using intensity-based thresholding.
- **Grid Merging** - Binarised image merged with a grid by choosing an appropriate scale for the coordinates can be done by a grid merging algorithm.

Grid Merging Algorithm

Step 1: Find the (X, Y) coordinates bound based on the size of the binarised images.
Step 2: Draw the horizontal lines for the values of X.
Step 3: Draw the vertical lines for the values of Y.
Step 4: Set some constant value to the grid sizes so that the feature points coincide with the grid points.

2.2.2 Mid-Point Traverse Method (MPTM)

Extract the vertex points from the grid merged images. These points are used to draw a bipartite graph and complete a bipartite graph of the input image. These points are extracted using the MPTM.

MPTM Algorithm

Step 1: (row, column) = size (image)
Step 2: Z=size (grid)
Step 3: For i = 1: Z: (row − Z)
Step 4: For j = 1: Z: (column − Z)
Step 5: Grid = image (i: i + Z, j: j + Z)
Step 6: If d_i + Z/2e, d_j + Z/2e == 0
Signature traversed to the Mid-Point else
Signature not traversed to the Mid-Point
Step 7: Choose a Mid-Point
Step 8: Repeat step 1 to step 7.

Illustration

The size of the grid depends on the size of the input image. The width of each row and column is constant and chosen according to system requirements. The points of the signature images that coincide with the grids' centre point are extracted as points. The points are extracted by traversing the signature starting from left to right. [Figure 3] shows the extraction of the vertex points using the Mid-Point Traverse Method of the image given in [Figure 2]. Grid size can be adjusted and any number of vertex points can be extracted. [Figure 4] shows the vertex points of the input image.

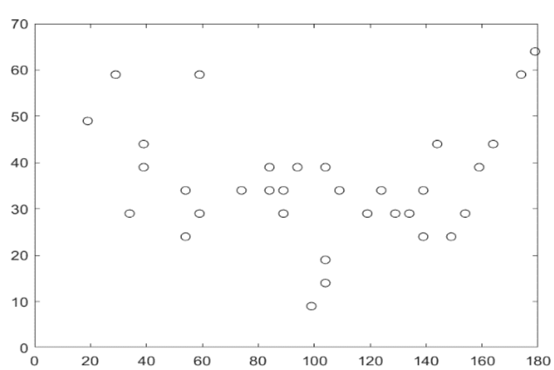


Fig 4. Vertex points of the signature

3 Construction of the Graphs

A bipartite graph is constructed based on the extracted vertex points and their interactions among them. Each point is (x, y) in XY plane. The horizontal line represents the y-axis, and the vertical line represents the x-axis. From these points, form two sets, X and Y, such that $x \in X$ and $y \in Y$. If (x, y) is a point in a plane, then we draw an edge between x and y. For convenience, represent the x-values as vertex set 1 and y-values as vertex set 2. A bipartite graph is constructed for the signature image in [Figure using the adjacency between the two sets X and Y. 2]. The bipartite graph of the given signature is given in [Figure 5].

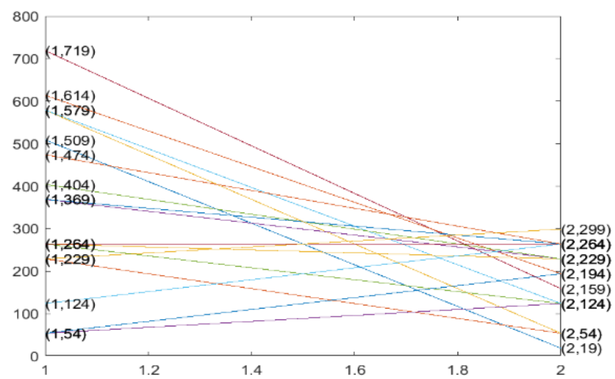


Fig 5. Bipartite graphs for image in Figure 2

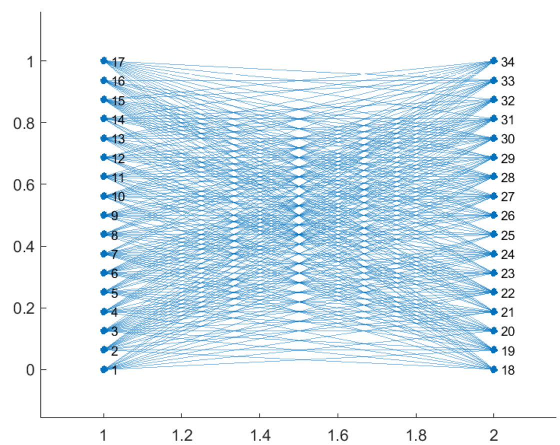


Fig 6. Complete bipartite graph for image in Figure 2

The complete bipartite graph of the input signature is given in [Figure 6] Two vertices x and y are said to be adjacent if there is an edge between them. If every vertex $x \in X$ is adjacent to at least one vertex $y \in Y$, then the graph obtained is said to be a bipartite graph. This adjacency list of vertices of each signature can be saved as data. In this way, a bipartite and complete bipartite graph is constructed from points of an input image. Every signature can be uniquely represented by a bipartite graph and a complete bipartite graph.

4 Experimental Results and Discussion

The dataset has 21 different sets of 60 signatures each [22], which sums to 1260 different signatures. Each set of 60 signatures has 30 genuine and 30 forged signatures. Skilled forgeries are considered in this data. In our proposed approach, 26 genuine and

25 forged signatures are kept as the training set, and the remaining 4 genuine and 5 forged are considered for testing. [Table 2] shows the dataset partitioning for training and testing.

Table 2. Data set partitioning

Signatures	Training Set	Testing Set	Total
Genuine Signatures	26	4	30
Forged Signatures	25	5	30
Total	51	9	60

All image data selected for testing were tested in MATLAB R2018b. The program was run on WINDOWS-TVFERPR, 2.30GHz with 8.00 GB Ram.

4.1 Testing Process and Feature Extraction

For 26 authentic and 25 forged training signatures, a threshold T was calculated. A threshold high (TH) is the highest value of the feature point in the collection of signatures, and a threshold low (TL) is the lowest value of the feature point in the set of signatures (TL). For each signature, the midpoint is computed, which is the threshold (T) value for detecting false acceptance and rejection.

$$T = \frac{TH + TL}{2}$$

For the bipartite graph and complete bipartite graph constructed from the extracted vertex points of the input signature, the pairwise distance $(P\_distance\,([X,\,Y]))$ is calculated for all the (x,y) pairs. Using this $P\_distance$, we have calculated the feature points of each signature by the following three methods and the accuracy, FAR, and FRR were compared among these methods. For each method, 60 feature values are computed.

4.1.1 Edges in Bipartite graph method (EBGM)

This method is based on the number of edges, $E_{x,y}$ - of the given bipartite graph. The score value W_1 of each set is computed as follows.

$$E_{x,y} = \sum_{x\in X} deg(x) = \sum_{y\in Y} deg(y)$$

Let $D = \; P\_distance((X,Y])$
Then $W_1 = D + E_{x,y}$

4.1.2 Average edge D-distance in Complete bipartite graph method (AEDDCBGM

Theorem 1 [23]: The average edge D-distance of a complete bipartite graph is
$\mu_3^D\left(K_{m,n}\right) = \frac{mn(n-1)(m-1)(m+n+1)}{(m+n)(m+n-1)}$
The score value W_2 of each signature based on the average edge, D-distance is computed as follows.
Let $D_1 = \; P\_distance([X,Y])$
Then $W_2 = D_1 + \mu_3^D\left(K_{m,n}\right)$.

4.1.3 Average edge distance in Complete bipartite graph method (AEDCBGM)

Theorem 2. [24]**:** $\mu'\left(K_{m,n}\right) = \frac{mn(m-1)(n-1)}{(m+n)(m+n-1)}$
The score value W_3 of each signature based on the average edge, distance is computed as follows.
Let $D_2 = \; P\_distance([X,Y])$
Then $W_3 = D_2 + \mu'\left(K_{m,n}\right)$

4.2 SVM classifier

Vapnik et al. [25,26] introduced SVM to classify signatures. SVM draws a hyperplane between two classes. The vectors adjacent to the hyperplane are called support vectors. The signature features obtained by different distance methods are used to train the SVM model. Since the feature set has large features, linear kernel offers the best hyperplane compared to polynomial, radial basis kernel, and sigmoid on our feature set. The linear kernel function is given by,

$$G(X_j, X_k) = X'_j, X_k$$

The proposed model uses 10-fold cross-validation to set the parameter cost. This cross-validation reduces the misclassification error of training and testing sets. The model is trained with the ICDAR dataset[27]. Each set contains 26 signatures from the genuine class and 25 signatures from the forged class. For each person, a set of 60 random signatures was taken into consideration and divided into 51 signature images used for training, and 9 were tested. The training and testing classes have been made with all possible mixtures, which gives a complete set of 60 results (considering all different training samples). The cost values had been updated in a loop with a small variation (0.0001), starting at 0.0001, and checked up to at least one. The corresponding cost has been selected for the genuine or forged signature to obtain maximum accuracy.

4.3 Performance Evaluation

As an initial setup, each individual had a training size of <26+25>, i.e., 26 genuine signatures and 25 forged signatures were used for training, as mentioned in 4.2. In addition, to record the error rates (FAR, FRR), its finished the experiments for 21 distinctive sets of signature images with the training size <26+25>. The results obtained for six sets of signatures verified using the features such as edges in the bipartite graph method, Average edge D-distance in the complete bipartite graph method and Average edge distance in the complete bipartite graph method from the ICDAR dataset are shown in Table[3–5].

Table 3. Edges in Bipartite graph method

Training Size 26+25 (Edges in Bipartite graph method)			
Set	**FAR (%)**	**FRR (%)**	**Accuracy (%)**
1	7.69	10	91.15
2	7.69	3.33	94.48
3	7.69	0	96.15
4	19.23	0	90.38
5	3.84	0	98.07
6	3.84	3.33	96.41
Mean	8.33	2.11	94.44

Table 4. Average edge D-distance in complete bipartite graph method

Training Size 26+25 (Average edge D-distance in Complete bipartite graph method)			
Set	**FAR (%)**	**FRR (%)**	**Accuracy (%)**
1	3.84	0	98.07
2	0	3.33	98.33
3	0	3.33	98.33
4	7.69	0	96.15
5	7.69	3.33	94.48
6	0	13.33	93.33
Mean	3.20	3.88	96.44

For each set, error rates are recorded, and the resultant FAR, FRR are described. Further, the training size <26+25>, as revealed in Sect. 4.2, we have tested the performance of our algorithm on different distance methods also. The performance of the proposed method is compared with the existing methods are shown in [Table 3], [Table 4] and [Table 5]. FAR and FRR are important two evaluation parameters for any signature identification system.

[Table 6] summarises the average performance of existing methods and proposed methods based on FAR and FRR. It is evident that from this Table that the average performance of our proposed method is better than all the existing methods. The average performance based on accuracy, FAR and FRR are discussed in [Table 7].

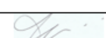
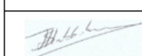
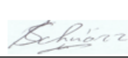
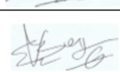
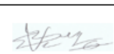
Table 5. Average edge distance in complete bipartite graph method

Training Size 26+25 (Average edge distance in Complete bipartite graph method)			
Set	**FAR (%)**	**FRR (%)**	**Accuracy (%)**
1	3.84	0	98.07
2	7.69	0	96.15
3	3.84	0	93.07
4	0	3.33	98.33
5	3.84	0	98.07
6	0	3.33	98.33
Mean	3.20	1.11	97.00

Table 6. Evaluation of Performances of Different Methods

Dataset		**Technique & Features**	**FAR (%)**	**FRR (%)**	**Performance**
ICDAR	Sharvari, K.S et al., [28] [2021]	VGG16 ResNet50 MobileNetV2 DenseNet121 Xception	-	-	67.9 62.32 59.52 60.73 60.29
	KAO HH et al., [29] [2020]	Deep learning	-	-	94.37
	Navid SM et al., [30] [2019]	VGG19	-	-	94
	Proposed Methods	SVM (Mean) & Edges in Bipartite graph method	8.33	2.11	94.44
		SVM (Mean) & Average edge D-distance in complete bipartite graph method	3.20	3.88	96.44
		SVM (Mean) & Average edge distance in Complete bipartite graph method	3.20	1.11	97.00

Table 7. Predicted accuracy for the proposed method using SVM

Signatures	**AEDDCBGM and AEDCBGM**		
	True Label	**Predicted Label**	**Accuracy**
	0	0	99.46
	1	1	99.60
	1	1	99.47
	1	1	98.99
	0	0	99.78
	0	0	98.14
	1	1	99.81
	0	0	95.75
	0	0	98.37
	1	1	99.90

[Table 7] summarises the true label of genuine and forged signatures. This proposed method extracts the features using edge distance methods for bipartite and a complete bipartite graph in all signatures. The set of all signature parameters divides the genuine and forged sets labelled as 0 and 1. The SVM model helps predict genuine and forged signatures with an accuracy level of more than 95%.

5 Conclusion

The graph-based method is developed for the identification of the genuineness of the off-line signature. In this paper, a set of proper and improper signatures were collected from a signatory. The signatures' points were extracted and the bipartite graph and complete bipartite graphs were constructed. The features were calculated from these graphs using edges, average edge distance and average edge D-distance methods. From these features of the proper and improper training signatures, a threshold value T is computed. Using this T value with the features of the testing signature, accuracy, FAR and FRR of the given method is computed. This paper found that the average edge distance gives better accuracy than the other two methods and accurately predicted all labelled features for signatures using the SVM classification method. The predicting level is more than 95%. These methods are compared with the different techniques and the results are satisfactory. This work can be extended to other graph structures using some new techniques to improve the accuracy of identification of genuineness of the signature.

6 Acknowledgement

The authors wish to thank the management of Sri Sivasubramaniya Nadar College of Engineering for their continuous support and encouragement.

References

1) Diaz M, Ferrer MA, Impedovo D, Malik MI, Pirlo G, Plamondon R. A Perspective Analysis of Handwritten Signature Technology. *ACM Computing Surveys*. 2019;51(6):1–39. Available from: https://dx.doi.org/10.1145/3274658.
2) Jagtap AB, Sawat DD, Hegadi RS, Hegadi RS. Verification of genuine and forged offline signatures using Siamese Neural Network (SNN). *Multimedia Tools and Applications*. 2020;79(47-48):35109–35123. Available from: https://dx.doi.org/10.1007/s11042-020-08857-y.
3) Lai S, Jin L, Zhu Y, Li Z, Lin L. SynSig2Vec: Forgery-free Learning of Dynamic Signature Representations by Sigma Lognormal-based Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021;p. 1–1. Available from: https://dx.doi.org/10.1109/tpami.2021.3087619.
4) Riesen K. Structural pattern recognition with graph edits distance. In: and others, editor. Advances in computer vision and pattern recognition. Springer International Publishing. 2015. Available from: https://link.springer.com/book/10.1007/978-3-319-27252-8.
5) Radhika KS, Gopika S. Online and Offline Signature Verification: A Combined Approach. *Procedia Computer Science*. 2015;46(46):1593–1600. Available from: https://dx.doi.org/10.1016/j.procs.2015.02.089.
6) Vohra K. Signature Verification Using Support Vector Machine and Convolution Neural Network". *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*. 2021;12(1):80–89. Available from: file:///C:/Users/User/Downloads/1564-Article%20Text-2912-1-10-20210407%20(1).pdf.
7) Xamxidin N, Mamat M, Kang W, Aysa A, Ubul K. Off Line Handwritten Signature Verification Based on Feature Fusion. *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*. 2021;p. 260–265.
8) Zhou Y, Zheng J, Hu H, Wang Y. Handwritten Signature Verification Method Based on Improved Combined Features. *Applied Sciences*. 2021;11(13):5867–5867. Available from: https://dx.doi.org/10.3390/app11135867. doi:10.3390/app11135867.
9) Sharif M, Khan MA, Faisal M, Fernandes YM, L S. A framework for off-line signature verification system: best features selection approach. *Pattern recognition letter*. 2018. Available from: https://doi.org/10.1016/j.patrec.2018.01.021.
10) Maergner P, Howe N, Riesen K, Ingold R, Fischer A. Offline Signature Verification Via Structural Methods: Graph Edit Distance and Inkball Models. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2018;p. 163–168. doi:10.1109/ICFHR-2018.2018.00037.
11) Maergner P, Howe NR, Riesen K, Ingold R, Fischer A. Graph-based off-line signature verification. 2019. Available from: https://arxiv.org/pdf/1906.10401.pdf.
12) Bhunia AK, Alaei A, Roy PP. Signature verification approach using fusion of hybrid texture features. *Neural Computing and Applications*. 2019;31(12):8737–8748. Available from: https://dx.doi.org/10.1007/s00521-019-04220-x.
13) Guerbai Y, Chibani Y, Hadjadji B. The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*. 2015;48(1):103–113. Available from: https://dx.doi.org/10.1016/j.patcog.2014.07.016.
14) Pham TA, Le HH, Do NT. Offline handwritten signature verification using local and global features. *Annals of Mathematics and Artificial Intelligence*. 2015;75(1-2):231–247. Available from: https://dx.doi.org/10.1007/s10472-014-9427-5.
15) Serdouk Y, Nemmour H, Chibani Y. New off-line Handwritten Signature Verification method based on Artificial Immune Recognition System. *Expert Systems with Applications*. 2016;51:186–194. Available from: https://dx.doi.org/10.1016/j.eswa.2016.01.001.
16) Pal S, Alaei A, Pal U, Blumenstein M. Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. *In2016 12th IAPR workshop on document analysis systems (DAS)*. 2016;p. 72–77. Available from: https://doi.org/10.1016/j.eswa.2016.01.001.
17) Loka H, Zois E, Economou G. Long range correlation of preceded pixels relations and application to off-line signature verification. *IET Biometrics*. 2017;6(2):70–78. Available from: https://dx.doi.org/10.1049/iet-bmt.2016.0046.
18) Zois EN, Alexandridis A, Economou G. Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets. *Expert Systems with Applications*. 2019;125:14–32. Available from: https://dx.doi.org/10.1016/j.eswa.2019.01.058.
19) Sharif M, Khan MA, Faisal M, Yasmin M, Fernandes SL. A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*. 2020;139(39):50–59. Available from: https://dx.doi.org/10.1016/j.patrec.2018.01.021.

20) Batool FE, Attique M, Sharif M, Javed K, Nazir M, Abbasi AA, et al. Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM. *Multimedia Tools and Applications*. 2020. Available from: https://dx.doi.org/10.1007/s11042-020-08851-4.
21) Ajij M, Pratihar S, Nayak SR, Hanne T, Roy DS. Off-line signature verification using elementary combinations of directional codes from boundary pixels. *Neural Computing and Applications*. 2021;(1-8). Available from: https://dx.doi.org/10.1007/s00521-021-05854-6.
22) Kaggle. . Available from: https://www.kaggle.com/divyanshrai/handwritten-signatures/data.
23) Babu DR, Varma PLN. Average D-Distance Between Edges of a Graph. *Indian Journal of Science and Technology*. 2015;8(2):152–152. Available from: https://dx.doi.org/10.17485/ijst/2015/v8i2/58066.
24) Balcı M, Ali P, Dündar. Average edge-distance in graphs. 2010. Available from: https://www.academia.edu/528133/Average_Edge-Distance_in_Graphs.
25) Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995;20(3):273–297. Available from: https://dx.doi.org/10.1007/bf00994018.
26) Vapnik VN. The Nature of Statistical Learning Theory. New-York. Springer New York. 2000. Available from: https://link.springer.com/book/10.1007/978-1-4757-3264-1.
27) Liwicki M, Malik MI, Van Den Heuvel CE, Chen XI, Berger CD, Stoel R, et al. Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). *2011 International Conference on Document Analysis and Recognition*. 2011. Available from: http://www.dfki.de/~liwicki/SigComp2011/trainingSet.zip.
28) Sharvari KS. A Comparative Study of Transfer Learning Models for Offline Signature Verification and Forgery Detection. *Journal of University of Shanghai for Science and Technology* . 2021;23:1129–1139. Available from: https://jusst.org/a-comparative-study-of-transfer-learning-models-for-offline-signature-verification-and-forgery-detection/.
29) Kao HH, Wen CY. An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach. *Applied Sciences*. 2020;10(11):3716–3716. Available from: https://dx.doi.org/10.3390/app10113716.
30) Navid SMA, Priya SH, Khandakar NH, Ferdous Z, Haque AB. Signature Verification Using Convolutional Neural Network. *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*. 2019;p. 35–39.