

Signature Verification Using Support Vector Machine and Convolution Neural Network

Kritika Vohra¹, S. V. Kedar²

¹JSPM's Rajarshi Shahu College of Engineering, PG Student,

²JSPM's Rajarshi Shahu College of Engineering, Professor

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021;
Published online: 05 April 2021

Abstract: Signature is used for recognition of an individual. Signature is considered as a mark that an individual write on a paper for his/her identity or proof. It is used as a unique feature for identifying an individual. It is highly used in social and business functions which gives rise to verification of signature. There are chances of signature getting forged. Hence, the need to identify signature as genuine or forged is utmost important. In this paper, identification of signature as genuine or forged is done using two approaches. First approach is using SVM and second is using CNN. For SVM, pre-processing of signature image is done and feature extraction is performed. Features extracted are histogram of gradient, shape, aspect ratio, bounding area, contour area and convex hull area. Further, SVM is applied to classify signature as genuine or forged and accuracy is determined. In the second approach, signature image is pre-processed, CNN is used to classify signature as genuine or forged and accuracy is determined. Dataset used here is ICDAR Dutch dataset along with 80 signatures taken from 4 people. Dutch dataset consists of 362 signature images and signature images taken from 4 people consists 10 genuine and 10 forged signatures which sums to 442 signature images. The proposed system provides accuracy of 86.39% using SVM and around 83.78% using CNN.

Keywords: signature, genuine, forged, pre-processing, accuracy, verification.

1. Introduction

Signature is used for identity verification of an individual. As there is a great use of signature, there are chances of signature getting forged. Forgery of signature in banks can lead to huge monetary loss. Forgery of signature can lead to huge losses for an individual too. Thus, there is a great need for recognition and verification of signature. Signature recognition and verification is treated as challenging work in biometrics. Hence, the problem can be defined as to find a solution that can differentiate between real signature along with forged signature from a given set of signatures. The important applications of signature include banking, finance, security, examination institutions, etc. where the chances for forgery is at its peak. Disguised signatures are considered difficult to identify where signature is done by genuine authors, but they are of intention to deny the signature. This is mostly done for fraudulent purposes. Types of forgeries are random forgery, skilled forgery, and simple forgery. In random forgeries, person uses its own signature to copy another user. Whereas skilled forgery is one where forger imitates another user. Simple forgery is one where person is familiar with only the shape of the signature, but forger does not have much practice about the shape. Distinguishing random forgeries are considered as an easy task than distinguishing skilled forgeries[1]-[2].

For classification of signature, shape of signature is considered which shows the vertical and horizontal trajectories formed due to the author's hand mobility. For online signature verification and recognition, static and dynamic features were identified from pen-based tablets. Some of the dynamic features are velocity, strokes order, acceleration, angle, pressure, etc. These features are exclusive and challenging to forge[3]. Handwritten signatures are captured by scanning images using scanner or camera. Offline signature is considered complex as there is no dynamic characteristics[4]. Scanned signature images contain noise. To remove this noise, spatial and frequency domain techniques are used. Muhammad Imran Malik et al.[2] presents signature analysis along with its verification by applying Speed Up Local Features. Based on the results signature verification is done. This system achieved EER of 15%. Saeede Anbaee Farimani et al.[3] presents online signature verification and recognition by applying Hidden Markov Model (HMM) technique. It segments signature curve which depends on pen's velocity value. FAR achieved is 4.8% and FRR is 5%. Clustering is another approach for verification of signature. Elements in a cluster are most identical than the elements in another cluster. This technique is also useful in other fields like face recognition and recognition of thumb impression[5].

Mohitkumar A. Joshi et al.[6] presents use of low-level key stroke features and recognition is performed using Support Vector Machine which has 3-fold cross validation. ERR achieved in this research is 15.59%. Derlin Morocho et al.[7] calculates the performance of correlative attributes for verification of signature. EER achieved ranges 5.5% to 21.2%. Features which affect the accuracy are local binary pattern, histogram of gradient, gray level co-occurrence level matrix, SURF, etc. These features help in verification of signature image[8]. Research in paper [9]-[10] refers to signature recognition using SVM. This process includes preprocessing, feature extraction and further SVM is applied. Features such as centroid, centre of gravity,

calculation of number of loops, normalized area are extracted[10]. In paper [11]-[12], signature verification is done using Convolution Neural Network model. Results obtained by extracting features from Deep Convolution Neural Network and SVM as classifier are better than decision tree[11].

Anjali.Ret al.[13] worked on combination of SVM and Neural Network . They calculated gray level features from signature images. Neural network is used for training of imagesby using feed forward back propagation algorithm along with Support Vector Machine. For cursive nature of signature, curve based as well as gradient based feature extraction methods are used such as histogram of curvature along with histogram of gradient[14]. Features such as underscore beneath signature, presence of dot on signature, ending strokes, etc. can be used for recognition. Next step includes splitting of signature to calculate the performance.

Splitting is performed in five categories such as left, right, bottom, up and middle. ANN including structural identification algorithm are used for prediction[15]. Nan Li et al.[16] worked to authenticate user with electronic signatures on mobiles. Coordinates, contact area, pressure and other biometric data was collected. Algorithms used for classification were SVM, Logistic Regression, Random Forest and AdaBoost. Then signature verification is done. Adaboost performance was best among other three algorithms with error rate of 2.375%. For multiscript signature verification generalized combined segmentation verification technique is used using multi-script signature dataset and SVM is used[17].

Moises Diaz et al.[18] presents algorithm for generation and detection of duplicated offline signature images.It is based onlinear as well as nonlinear transformations. It simulates human spatial cognitive map. Duplicator is classified by increasing a training sequence artificiallywhich further leads to verifying performance of signature. Muhammad Imran Malik et al.[19] used Fast Retina Key points (FREAK), which represents local features and uses human visual system such as retina. For performance comparison, Features from Accelerated Segment Test (FAST) and Speed Up Local Features (SURF) were used. This system achieved error rate of 30%.

The remainder of the paper is as follows: Section II briefs about the proposed system, Section III explains the proposed algorithm, Section IV discusses the experimental results, and Section V concludes the paper.

2. Proposed System

In this system, verification of signature is done using Support Vector Machine and Convolution Neural Network.This research involves identifying the genuineness of signature, so it requires a dataset of genuine and forged signature.Dataset used here is ICDAR Dutch dataset along with 80 signatures taken from 4 people. Dutch dataset consists of 362 signature images and signature images taken from 4 people consists 10 genuine and 10 forged signature of each person which sums to 442 signature images.Dutch dataset includes signature of 10 reference writers and skilled forgeries of these signatures.Figure. 1 demonstrates the working of the proposed system using SVM.

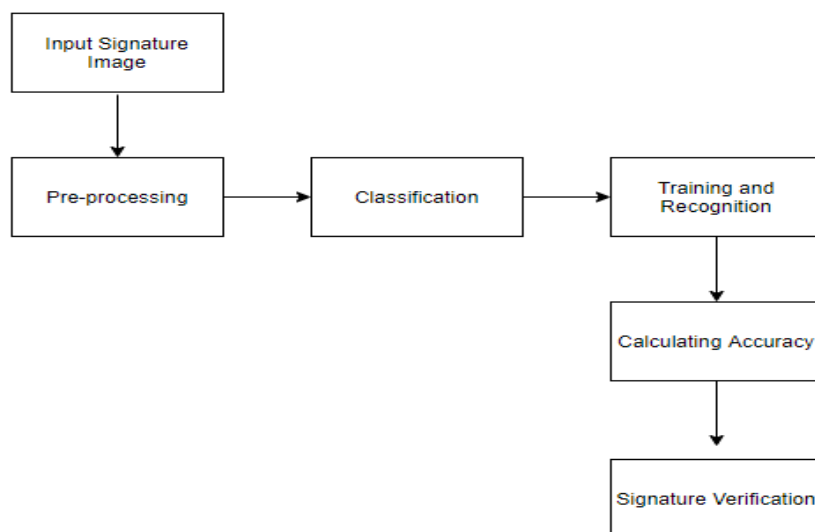


Figure. 1. Working of the proposed system using SVM

This system works in three phases namely training, testing and classification. Training phase acquires PNG

images of known genuine and forged signatures. Testing phase acquires unlabelled signature images. Initially the system operates in training phase. Training phase passes through pre-processing and feature extraction steps respectively. In pre-processing, grayscale conversion occurs. Grayscale image contains only gray shade with no colour. Reason for distinguishing these images from other colour image is less information is given to each pixel. Then, it resizes image to [200,200] size. Next step includes thresholding. Thresholding is considered as easiest method for segmenting. Using this gray scale image, thresholding is done to form binary image. It replaces every pixel of an image by black pixel when image intensity is found to be less than a specific fixed constant. Feature extraction step uses pre-processed signature image as input and extracts features such as shape, histogram of gradient, aspect ratio, bounding area, contour area and convex hull area [22-24]. The description of each feature is mentioned below.

- **Hu Moments:** It is calculated for describing the shape of signature image. They are constant to scale, rotation, and translation. Hu moments includes set of seven numbers which are computed by applying central moments. They are constant to transformation of images. Initial six moments are proved constant to translation, scale, reflection, and rotation whereas the seventh moment's sign differs for image reflection. The seven moments are calculated using the following formula:

$$\phi_1 = \eta_{20} + \eta_{02} \quad \dots(1)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad \dots(2)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad \dots(3)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad \dots(4)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} - \eta_{12})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

...(5)

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03}) \dots(6)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

...(7)

02201

η

η

φ

02201

η

η

φ

02201

η

η

φ

φ

+=

2

11

2

02202

4)(

ηηηφ

+-=

2

0321

2

12303

)3(3(

μηηηφ

+-=

2

0321

2

12304

)(
μηηηφ
+++=
1)0(31)(3(
1)(3))1(0(3(
2
0321
2
1230)03210321
2
0321
2
1230123012305
ηηηηηηηη
ηηηηηηηηφ
+---++
+---+=
) (4
1)(-))((
0321123011
2
0321
2
123002206
ηηηηη
ηηηηηηφ
+++
++-=
1) (-))1(3(0(-3(-
1)3(-))1(0(3(
2
0
φ
+=
2
11
2
02202
4)(
ηηηφ
+-=
2
0321
2
12303
)3(03(
μηηηφ
-+-=
2
0321
2
12304
)(
μηηηφ
+++=
1)0(31)(3(
1)(3))1(0(3(
2
0321
2

1230)03210321
 2
 0321
 2
 1230123012305
 ηηηηηηηη
 ηηηηηηηηφ
 +++++
 ++++-=
))(4
])(-)[((
 0321123011
 2
 0321
 2
 123002206
 ηηηηη
 ηηηηηηηφ
 +++
 +++-=
])(-)[(3)(-3(-
]3(-)[()3(
 2
 0

• Histogram of gradient: Histogram of gradient is used for describing structure and appearance of an image. It is mostly used for object classification. HOG captures local intensity gradients and edge directions. HOG returns a real valued feature vector. Here, appearance of an object is modelled by distribution of intensity gradients in the rectangular regions of the image. Then image gradient is calculated in both x and y directions using formula:

$$G_x = I * D_x \quad \dots(8)$$

$$G_y = I * D_y \quad \dots(9)$$

where I is the input image, D_x is our filter in the x -direction, and D_y is our filter in the y -direction. Final gradient module is calculated using the formula:

$$G = \sqrt{G_x^2 + G_y^2} \quad \dots(10)$$

Finally, the orientation of the gradient for each pixel in the input image can then be computed by:

$$\theta = \arctan2(G_y, G_x) \quad \dots(11)$$

Given both $|G|$ and θ , we can now compute histogram of gradients, where the bin of the histogram is based on θ and the contribution or weight added to a given bin of the histogram is based on $|G|$.

• Aspect ratio of bounding rectangle: It is the width to height ratio of a bounding rectangle. Aspect ratio is used to ensure that the images are displayed in the correct ratio irrespective of its size. So if the object is scaled, the signature image inside the object's bounding box will still be displayed with the correct aspect ratio. Formula for calculating aspect ratio is as follows:

$$\text{Aspect Ratio} = \text{Image Width} / \text{Image height} \quad \dots(12)$$

• Area of bounding rectangle: It calculates the area of bounding rectangle by the formula:
 $\text{Area} = \text{Width} * \text{Height} \dots(13)$

• Contour Area: Contour area is the area inside the contour. Contour should be closed otherwise it will consider it as closed. If it is a line, it will consider the area inside the contour defined by the line. It will consider the area between first point and last point.

• Convex Hull Area: Consider a set of points S convex hull is the intersection of all half spaces that contain S . Convex hull of a set is a closed solid region which includes all the points in its interior. Convex hull C for N points P_1, \dots, P_N is given by the expression:

$$C \equiv \{ \sum_{j=1}^N \lambda_j \geq 0 \text{ for } j \text{ and } \sum_{j=1}^N \lambda_j = 1 \} \dots(14)$$

• HaralickFeatures: They are considered as texture features which depends on adjacency matrix. Adjacency matrix are positioned in co-ordinates (i, j) , i.e. the number of times a pixel takes the value i next to a pixel with value j . Gray LevelCo-occurrence Matrix(GLCM) are used by haralick features for adjacency. Four

GLCM matrixes are constructed for a single image. From this, 14 textural features were computed which are as follows: Angular Second Moment, Contrast, Correlation, Sum of Squares:Variance, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Entropy, Difference Variance, Difference Entropy, Information Measures of Correlation and Maximal Correlation Coefficient[25].

After completion of training phase, system enters next phase i.e. testing phase. Dataset is split into training and testing. Training dataset contains 80% of signature images. Testing dataset contains 20% of the images. This phase acquires PNG images of unknown signature. These signature images are then passed to pre-process and feature extraction is done. Classifier predicts the label for testing images and compares with actual labels and determine accuracy. Pass a random image to the saved model and model predicts the genuineness of signature. Figure. 2 demonstrates the working of the proposed system using CNN.

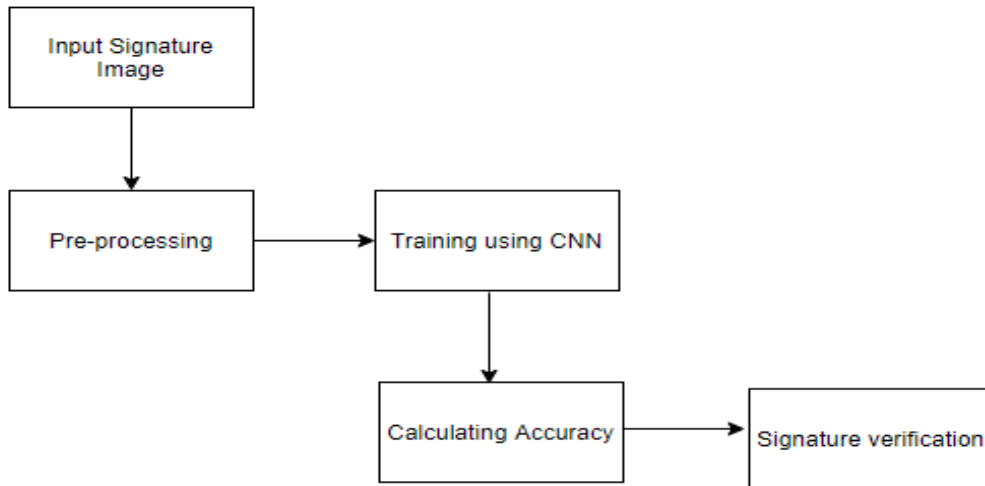


Figure. 2. Working of the proposed system using CNN

Initially, pre-processing of the input signature image is done. Pre-processing includes loading the image, resizing it to 32*32 pixels, converting to array, load the pixel intensities in the range [0, 1] and updating the image list. Dataset is partitioned into training and test dataset. Performed training and testing split of dataset by using 80% of dataset for training purpose and 20% of dataset for testing purpose. Further, 2D Convolution Neural Network architecture is defined and the model is trained using Adam optimizer. Model is trained with 400 epochs. Model is evaluated and its accuracy is computed. Now pass an unknown image to the model and the model predicts the genuineness of signature.

3. PROPOSED ALGORITHM

Signature recognition is done using two algorithms i.e. SVM and CNN. Proposed algorithm for SVM is mentioned below.

A. SVM works on a dataset which has been partitioned into training and testing dataset. Training dataset contains known genuine and forged PNG signature images and testing dataset contains unknown signature images. The steps for proposed algorithm using SVM are as follows:

- Repeat the steps A1 to A3 for all images in training dataset to prepare reference feature vector.
 1. Select a PNG image from training dataset.
 2. Pre-process the selected image using following steps:
 - a. Convert the image to grayscale image.
 - b. Resize the image to [200,200] size.
 - c. Convert the image to binary form.
 3. Perform feature extraction of pre-processed image using following steps:
 - a. Compute shape, histogram of gradient, aspect ratio, bounding area, contour area convex hull area and haralick features.
 4. Now pass the image to the SVM Classifier.
- Select a PNG image from testing dataset.
- Prepare testing feature vector by applying steps A2 and step A3 on testing image.

- Pass the testing dataset to the running model to evaluate the accuracy.
- Using testing dataset make predictions and initialize a dictionary to accumulate computed metrics.
- Load the saved model which will find label of signature images based on its features.
- Now pass a random image to the saved model and the model predicts the genuineness of signature.

Proposed algorithm for CNN is mentioned below.

B. CNN works on a dataset which has been partitioned into training and test dataset. The steps for proposed algorithm using CNN are as follows:

- Repeat the steps A5 and A6 for all images in training dataset to prepare reference feature vector.
- 5. Select a PNG image from training dataset.
- 6. Pre-process the selected image using following steps:
 - a. Load the input image from disk and resize it to 32*32 pixels.
 - b. Scale the pixel intensities to the range [0, 1] and update the image list.
 - c. Define the 2D Convolution Neural Network architecture and train the model using Adam optimizer.
 - d. Train the model with 400 epochs
- 7. Evaluate the model and compute its accuracy of training set.
- Prepare testing feature vector by applying steps A5 and A6 on testing dataset.
- Pass the testing dataset to the running model to evaluate the accuracy.
- Calculate confusion matrix and find accuracy, specificity, and sensitivity.
- Now pass a random image to the saved CNN model and the model predicts the genuineness of signature.
- Using testing dataset make predictions and initialize a dictionary to accumulate computed metrics.

4. Experimental results

Performance comparison of signature verification using SVM and CNN is discussed below:

- A. Results obtained using SVM:
Overall accuracy of 86.39% is obtained using SVM. Confusion matrix for SVM is given in Table I.
- B. Results obtained using CNN:
Overall accuracy of 83.76% is obtained using CNN. Confusion matrix for CNN is given in Table II.

Figure. 3 demonstrates CNN model accuracy graph and Figure. 4 demonstrates CNN Epoch Model Loss graph.

- C. TABLE III. represents EER obtained by using different features in SVM

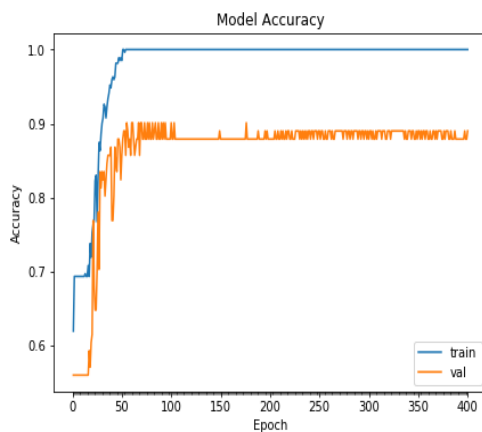


Figure. 3. CNN model accuracy graph

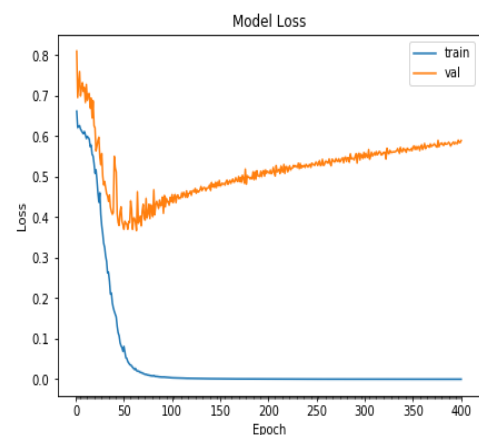


Figure. 4. CNN model loss

Table 1. Confusion matrix for svm

	Precision	Recall	F1-score	Support
Forged	0.78	0.93	0.85	60
Genuine	0.95	0.82	0.88	87
Accuracy			0.86	147
Macro Average	0.86	0.87	0.86	147
Weighted Average	0.88	0.86	0.87	147

Table 2. Confusion matrix for cnn

	Precision	Recall	F1-score	Support
Forged	0.87	0.87	0.87	23
Genuine	0.79	0.79	0.79	14
Accuracy			0.84	37
Macro Average	0.83	0.83	0.83	37
Weighted Average	0.84	0.84	0.84	37

Table 3. Results obtained using different features in svm

System	EER[%]
Signature Recognition based on Comparative Attributes[7]	21.20
Offline Signature Verification based on low level key strokes[6]	15.59
Offline Signature Recognition using Support Vector Machine[10]	7.16

Table III. shows results obtained using different features in SVM. Signature Recognition based on Comparative Attributes[7] showed EER of 21.20%. Offline Signature Verification based on low level key strokes[6] showed EER of 15.59% whereas Offline Signature Recognition using Support Vector Machine[10] showed EER of 7.16%. In this research, accuracy obtained with SVM as a classifier is 86.39% and accuracy obtained using CNN is 83.78%.

5. Conclusion

To avoid forgery of signature in any of the public, private or other sectors, signature is recognized as genuine or forged based on two different approaches. An approach to identify the genuineness of signature using Support Vector Machine and Convolution Neural Network is discussed here. The proposed system provides overall

accuracy of 84.80% using SVM and 87.00% using CNN. Performance comparison of both the approaches is discussed. Our next objective is to improve accuracy by adding more features.

Acknowledgment

Authors would like to thank researchers, publishers for their resources along with teachers for their support. We are also thankful to reviewers for their valuable suggestions.

References

1. Avani Rateria, and Suneeta Agarwal. "Off-line Signature Verification through Machine Learning." UPCON 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics.
2. Muhammad Imran Malik, Marcus Liwicki, Andreas Dengel, Seiichi Uchida, and Volkmar Frinken. "Automatic Signature Stability Analysis And Verification Using Local Features." 14th International Conference on Frontiers in Handwriting Recognition, 2014.
3. Saeede Anbaee Farimani, and Majid Vafaei Jahan. "An HMM for Online Signature Verification Based on Velocity and Hand Movement Directions." 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), 2018.
4. Htight Wai, and Soe Lin Aung. "Feature Extraction for Offline Signature Verification System." International Journal of Computer & Communication Engineering Research (IJCCER), Volume 1 - Issue 3 September 2013.
5. Samit Biswas, Debnath Bhattacharyya, Tai-hoon Kim, and Samir Kumar Bandyopadhyay. "Extraction of Features from Signature Image and Signature Verification Using Clustering Techniques." T.-h. Kim, A. Stoica, and R.-S. Chang (Eds.): SUCOMS 2010, CCIS 78, pp. 493–503, 2010. © Springer-Verlag Berlin Heidelberg 2010.
6. Mohitkumar A. Joshi, Mukesh M. Goswami, and Hardik H. Adesara. "Offline Handwritten Signature Verification Using Low Level Stroke Features." IEEE, 978-1-4799-8792-4/15/\$31.00_c 2015.
7. Derlin Morocho, Aythami Morales, Julian Fierrez, and Ruben Vera-Rodriguez. "Human-Assisted Signature Recognition based on Comparative Attributes." 14th IAPR International Conference on Document Analysis and Recognition, 2017.
8. Kamlesh Kumari, and V.K. Shrivastava. "Factors Affecting The Accuracy of Automatic Signature Verification", India, IEEE, 978-3805-4421-2/16, 2016.
9. Rashika Shrivastava and Brajesh Kumar Shrivash. "Offline Signature Verification Using SVM Method and DWT-Gabor Filter Feature Extraction." IJSTE - International Journal of Science Technology & Engineering, (IJSTE/ Volume 2 / Issue 07 / 051).
10. Kruthi.C, and Deepika.C.Shet, "Offline Signature Verification Using Support Vector Machine." Fifth International Conference on Signal and Image Processing, 2014.
11. Kamlesh Kumari, and Sanjeev Rana. "Offline Signature Recognition using Pretrained Convolution Neural Network Model." International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019.
12. A.Bhanu Sronothara, and M. Hanmandlu. "Offline Signature Verification using CNN.", IJFRCSCE, September 2018.
13. Anjali.R, and Manju Rani Mathew. "Offline Signature Verification based on SVM and Neural Network." International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Special Issue 1, December 2013.
14. Amir Soleimani, Kazim Fouladi, and Babak N Araabi. "Persian Offline Signature Verification Based on Curvature and Gradient Histograms." 6th International Conference on Computer and Knowledge Engineering (ICCKE 2016), October 20-21, 2016.
15. Vaishali R. Lokhande, and Bharti W. Gawali. "Analysis of Signature for the Prediction of Personality Traits." India, IEEE, 978-1-5090-4264-7/17/\$31.00 ©2017.
16. Nan Li, Jiafen Liu, Qing Li, Xubin Luo, and Jiang Duan. "Online Signature Verification Based on Biometric Features.", IEEE, 1530-1605/16 \$31.00, 2016.
17. Keigo Matsuda, Wataru Ohyama, Tetsushi Wakabayashi, and Fumitaka Kimura. "Effective Random-impostor Training for Combined Segmentation Signature Verification.", IEEE, 2167-6445/16 \$31.00, 2016.
18. Moises Diaz, Miguel A. Ferrer, George S. Eskander, and Robert Sabourin Member. "Generation of Duplicated Off-line Signature Images for Verification Systems.", IEEE, 0162-8828, 2016.

19. Muhammad Imran Malik, Sheraz Ahmed, Marcus Liwicki, and Andreas Dengel. "FREAK for Real Time Forensic Signature Verification.", IEEE, 1520-5363/13 \$26.00, 2013.
20. Feriel Boudamous, Hassiba Nemmour, and Yasmine Serdouk, Youcef Chibani. "An-Open System for Off-line Handwritten Signature Identification and Verification using Histogram of Templates and SVM.", IEEE, 978-1-5386-0551-6/17/\$31.00 ©2017.
21. Sergei G. Chernyi, Vladimir E. Marley, and Aleksandr S. Bordug. "Systems of Identification Authentication and Encoding in Maritime Industry.", IEEE, 978-1-5386-4340-2/18/\$31.00 ©2018.
22. M. A. Jayaram, and Hasan Fleyeh. "Convex Hulls in Image Processing: A Scoping Review.", American Journal of Intelligent Systems, 2016.
23. V A Bharadi, and H B Kekre. "Off-Line Signature Recognition Systems.", International Journal of Computer Applications, Volume 1 – No. 27, 2010.
24. Aravinda C.V, Lin Meng, and Uday Kumar Reddy K.R. "An approach for signature recognition using contours based technique.", IEEE, 978-1-7281-3480-2/19/\$31.00/ ©2019.
25. Robert M. Haralick, K. Shanmugam, And Its'hak Dinstein. "Textural Features For Image Classification.", Ieee Transactions On Systems, Man, And Cybernetics, \O Smc(-3, No. 6, November1973.