SIGNATURE RECOGNITION AND FORGERY DETECTION

SUBMISSION DATE / POSTED DATE

07-10-2022 / 13-10-2022

CITATION

R, Anagha; Kumar G, Chandan (2022): SIGNATURE RECOGNITION AND FORGERY DETECTION. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.21293121.v1

DOI

10.36227/techrxiv.21293121.v1

SIGNATURE RECONITION AND FORGERY DETECTION

Dr. Chayadevi M L
Computer Science and Engineering
BNM Institute of Technology
Bangalore, India
chayadevi1999@gmail.com

Anagha R
Computer Science and Engineering
BNM Institute of Technology
Bangalore, India
Anagharavindra@gmail.com

Chandan Kumar G
Computer Science and Engineering
BNM Institute of Technology
Bangalore, India
chandankumarg35@gmail.com

***Abstract*— Every person has a distinctive signature primarily used for personal identification and to confirm the authenticity of important papers or legal transactions. Static and dynamic signature verification are the two types available. While dynamic (online) verification occurs as a person makes their signature on a digital tablet or a similar device, static (offline) verification occurs after an electronic or document signature has been completed. Therefore, using the Support Vector Machine (SVM) algorithm and the K-Means algorithm, an intelligent model is built that gets trained on the real signature data sets and can identify forged signatures based on the available forged signature data sets. Using this model, we can attain maximum accuracy in detecting forgery in signatures. The proposed system achieves an accuracy of 95.83% for forgery detection.**

***Keywords*—** Signature Recognition, Forgery Detection, Support Vector Machines, K – Means, Feature Extraction.

I. Introduction

Offline signatures are significant today. Offline signature verification and forgery detection are complex and fraught with serious problems. The forging of signatures causes cooperating and commercial organizations to suffer substantial financial losses and damages their security reputation. Forgery is frequently observed in the banking industry since it involves sensitive information, official paperwork, and government regulations (LIC) that could be vulnerable to fraud and its effects. Therefore, a system that can tell the difference between a real signature and a forgery is needed to reduce the likelihood of theft or fraud.

The two broad categories of biometrics are physiological and behavioral. A behavioral biometric uses a handwritten signature. It was the first biometric technology to be utilized before PCs and laptops were introduced. The use of handwritten signatures for identity verification in the banking and financial industries dates back many years. The verification procedure is typically carried out manually, either by someone knowledgeable about the signature database or through comparison against a few signature templates. The identification of the signature owners and the determination of whether the signature is authentic, or fake are two different problems that the signature verification system can vigorously address. Not only is the study of signature verification necessary in the field of image processing and pattern recognition. Besides being extensively utilized in money, access control, legal issues, and security, signature verification systems and other signature verification procedures are classified using two unique classes: online and offline. A pointer and an electronic tablet attached to a PC that gathers dynamic signature data are needed for online verification. These emotional traits are individual to each person, sufficiently stable, and repeating. An online signature system recognizes the action of the pen while signing, and these signatures can be validated depending on several factors, including pen pressure and writing speed. These characteristics are unique and difficult to forge.

While in offline systems, signatures are nothing more than 2D images created through scanning or camera capture. Because signatures lack dynamic geometry, the offline signature procedure is a complex operation. Additionally, it is more challenging to separate signature strokes because of various modern and atypical writing styles. The type of signature obtained may also depend on the nature and varied pen patterns. Sometimes genuine people's signatures cannot be adequately completed because of illness, mood swings, aging, or emotional behavior. Significant intra-personal and interpersonal variances are a result of developing.

It is necessary to create an intelligent system that can take these elements into account and quickly identify different kinds of forgeries. Neither the system's sensitivity nor its coarseness should be excessive. Low false acceptance and low false rejection ratios should be acceptable trade-offs. The system's design should seek features that minimize the amount of storage required and computation time needed.

Our suggested method trains the model using the Support Vector Machine (SVM) technique. SVMs can use kernels to do both linear and non-linear classifications. For this classification, a gaussian kernel will be used. SVMs are far superior to humans at extracting significant/relevant data for type, making them exceptionally successful systems for recognition tasks. Additionally, the K-means method has assisted us in obtaining the most similar signature image.

This paper's contribution is as follows:

- Pre-processing technique proposed to simplify signature verification.
- K-means algorithm-based model for the proposed SVM-based Signature verification system.

- A Support Vector Classifier (SVC) for signature forgery detection has been proposed.

II. LITERATURE SURVEY

Data collection, picture processing, normalization, clustering, and evaluation are carried out in the paper "Offline Signature Recognition and Verification System utilizing Efficient Fuzzy Kohonen Clustering Network (EFKCN) Algorithm" by Dewi Suryani, Edy Irwansyah, and Ricki Chindra. RGB to Grayscale Format conversion, binary image conversion, binary image inversion, border removal, and bounding box extraction were the pre-processing techniques used. An accuracy of roughly 70% was attained utilizing this strategy.

In Tejas Jadhav's paper titled "Handwritten Signature Verification using Local Binary Pattern Features and KNN," the pre-processing methods used: RGB to Gray Scale conversion, Otsu Thresholding, and Boundary box cropping, and feature extraction methods used: LBP image generation, texture features, and name features are used as the feature extraction methods. KNN is used in this methodology together with Euclidian distance. The accuracy of this strategy is 73.34 %.

Thresholding, Edge Thinning, Noise Removal, and Noise Removal with Adaptive Filtering are the pre-processing techniques used in the Tansin Jahan, Md. Shahriar Anwar, and S. M. Abdullah Al-Mamun paper titled "A Study on Preprocessing and Feature Extraction in offline Handwritten Signatures." The methods for extracting features from a signature include finding loops and converting the signature's pixels into binary numbers. MATLAB is used in this paper to run the model.

The Deep Convolutional Neural Networks (DCNN) and Explainable Deep Learning are used to implement the model in the paper titled "An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach" by Hsin-Hsiung Kao and Che-Yen Wen. The pre-processing techniques include rotation, block-based data augmentation, and RGB to grayscale conversion. The accuracy of this model is 94.37%.

The pre-processing techniques suggested in the paper "OFFLINE SIGNATURE VERIFICATION" by Dr. M. Narayana, L. Bhavani Annapurn, and K. Mounika included binarization, noise removal, thinning, locating the picture boundary box, solving angular problems, and size normalization. Height/width ratio, OTSU's method, linked component, and threshold are employed as feature extraction techniques. The primary way advocated for detecting signature forgery is Euclidian Distance. The efficiency of this approach was 85.42%.

Jivesh Poddar, Vinanti Parikh, and Santosh Kumar Bharti's paper, "Offline Signature Recognition and Forgery Detection using Deep Learning," suggests using Convolutional Neural Networks (CNN), the Crest-Trough method, the SURF algorithm, and the Harris corner detection algorithm for the model training. The accuracy of the suggested system is between 85 and 89 percent.

III. METHODOLOGY

The four-stage technique used by the verification system is Data Pre-processing, Feature Extraction, Model Training, and Model Testing. The system architecture below shows the data flow and whole verification mechanism.

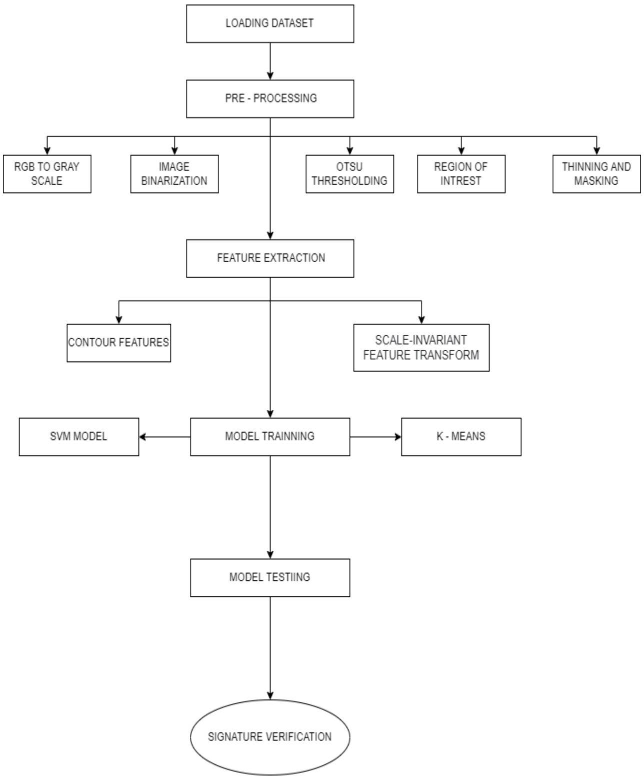


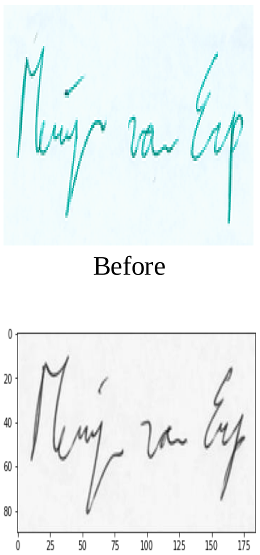
Figure 1: Proposed Model

Figure 1 proposes the model used for our verification system. The pre-processing method involves RGB to Gray Scale Conversion, Image Binarization using OTSU Threshold and Gaussian Blurring, finding the Region of Interest, and Thinning and Masking. The Feature extraction methods used are contour features and the scale-invariant feature transform (SIFT) features. Support Vector Machine (SVM) and K-Means algorithm are implemented for model training.

IV. IMPLEMENTATION

The first stage of methodology is Data Pre-Processing, which includes RGB to Grayscale conversion, Image Binarization using Gaussian Blurring and OTSU Thresholding, Finding the Region of Interest, and then finally applying Thinning to the image.

1. *RGB to Grayscale Conversion*: This step will convert the signature image in color to a grey scale to occupy less space and help faster processing in the later stages. The result obtained after this stage:

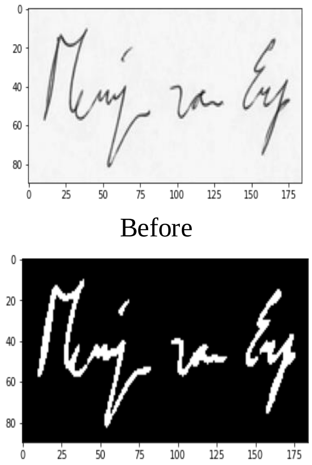


Before

After

Figure 2: RGB to Grayscale conversion results

2. *Image Binarization using Gaussian Blurring and OTSU Thresholding*: This method will blur the image and convert it into a binary image using the Grayscale image by applying the OTSU thresholding to find the threshold point and display the picture automatically.

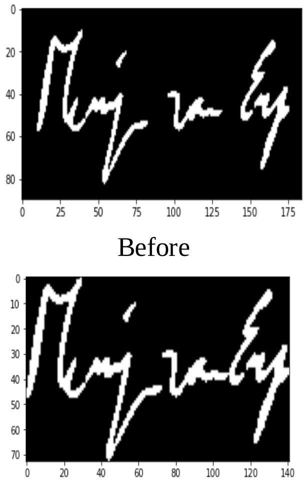


Before

After

Figure 3: Image Binarization results

3. *Region of Interest (ROI):* This phase involves getting the area of interest.

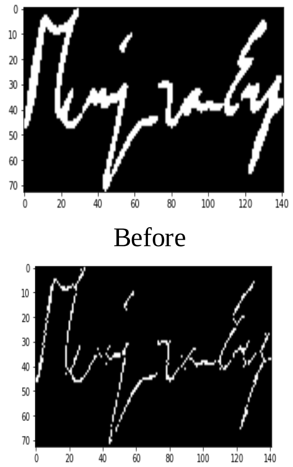


Before

After

Figure 4: Extraction of Region of Interest results

4. *Thinning*: Thinning Technique is applied to the image obtained from the 3rd step.



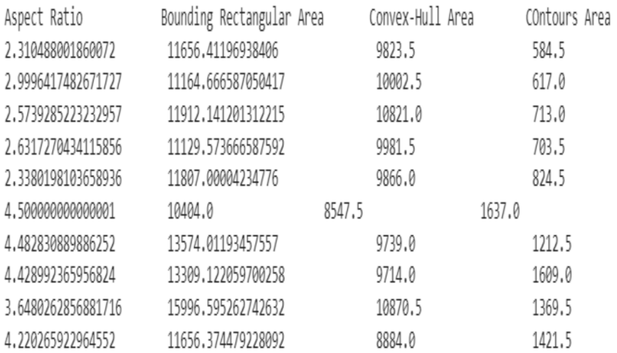
Before

After

Figure 5: Thinning and masking results

The next step is feature extraction. The methods applied are obtaining the contour features of the image and using Scale Invariant Feature Transform (SIFT) techniques.

1. *Obtaining Contour Features*: Contouring is a feature extraction step where contours onto the signatures are drawn as we cannot get a geometrically perfect signature to train the model. Hence this step is necessary.
2. *Scale Invariant Feature Extraction (SIFT) techniques*: This is one of the techniques for extracting features from an image regardless of its scale and rotation angle.



Aspect Ratio	Bounding Rectangular Area	Convex-Hull Area	COntours Area
2.310488001860072	11656.41196938406	9823.5	584.5
2.9996417482671727	11164.666587050417	10002.5	617.0
2.5739285223232957	11912.141201312215	10821.0	713.0
2.6317270434115856	11129.573666587592	9981.5	703.5
2.3380198103658936	11807.00004234776	9866.0	824.5
4.500000000000001	10404.0	8547.5	1637.0
4.482830889886252	13574.01193457557	9739.0	1212.5
4.428992365956824	13309.122059700258	9714.0	1609.0
3.6480262856881716	15996.595262742632	10870.5	1369.5
4.220265922964552	11656.374479228092	8884.0	1421.5

Figure 6: Contour Features for the pre-processed sample image

Figure 6 gives the contour features of the signature images, such as the aspect ratio, bounding rectangular area, convex-hull area, and contour area.

- A pre-processed image's width to height ratio is known as the aspect ratio.
- For object detection, a bounding box is a hypothetical rectangle that establishes a collision box for the target item.
- The convex hull of a Euclidean space is often defined as the set of all convex combinations in the subset of the intersection of all convex sets encompassing the subset.
- The contour area is defined as a curve connecting all straight points and borders with the same color and intensity.

The features obtained from the Scale Invariant Feature Transform (SIFT) is given below:

```
[('data/valid/001001_001.png', array(
[[ 26.,  0.,   0., ...,   0.,   0.,   0.],
[ 37.,  0.,0., ...,1.,0.,0.],
[ 21.,   0.,   0., ...,   0.,   0.,   0.], ...,
[ 47.,   0.,   0., ...,   1.,   0.,   0.],
[ 54.,  96.,   0., ...,  28.,   0.,   0.],
[0.,   0.,   0., ..., 126.,   0.,   0.]], , dtype=float32))
```

Signature datasets can be divided into 70:30, where 70% of the signatures are trained to the model, rest, 30%, is used to test the model.

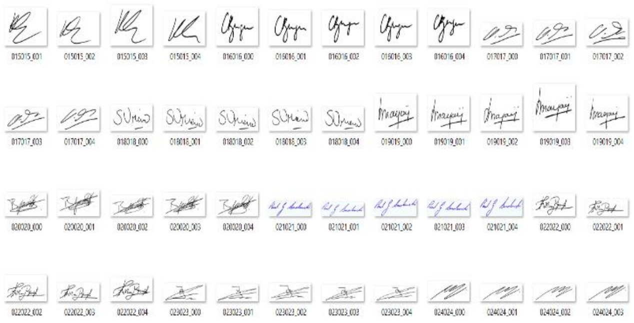


Figure 7: Datasets loaded to train the model

Figure 7 shows the datasets taken for model training.

The second phase is training our model, which employs the Support Vector Machine (SVM) classification algorithm and the K-Means technique for clustering.

Support Vector Machine is a classification method used in Supervised Machine Learning that creates decision boundaries to divide n-dimensional space into several categories that may be utilized in the future to classify a new test image into the appropriate class.

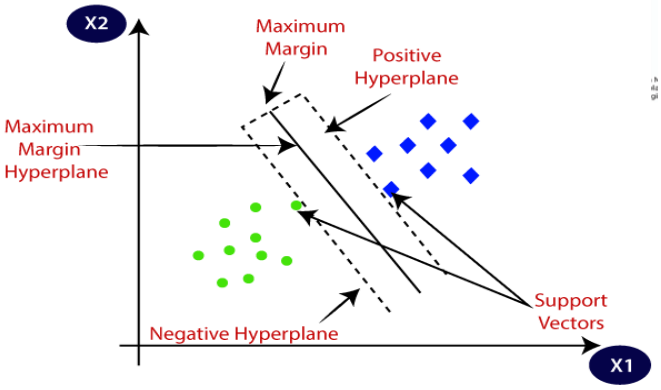


Figure 8: Support Vector Machine
Image Courtesy: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

Linear Support Vector Classifier is used as it provides a linear kernel, and we can create a custom loss function for the model to give it better accuracy.

The following algorithm is K-Means clustering which is used to cluster data based on the value of k provided. Here, for our model, we need to press the training images into real and forged values.

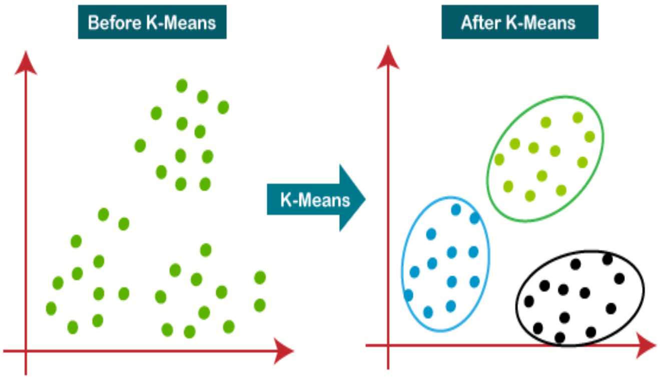


Figure 9: K – Means Clustering
Image Courtesy: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

For Testing the model, 30% of datasets that include both genuine and forged signature images are used to check whether the model predicts the result accurately.

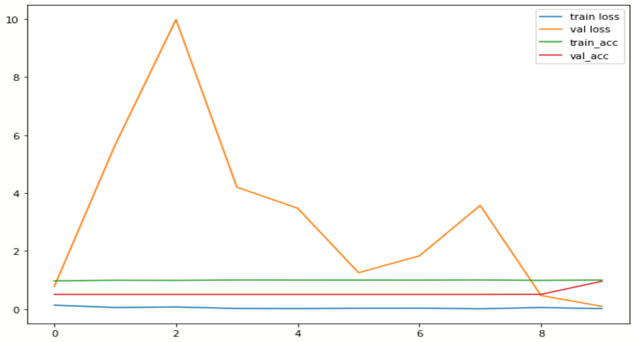
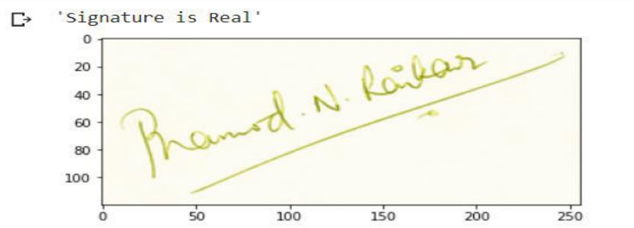


Figure 10: Test accuracy chart

Figure 10 gives an accuracy chart considering the training accuracy, train loss, value accuracy, and value loss.

V. RESULTS

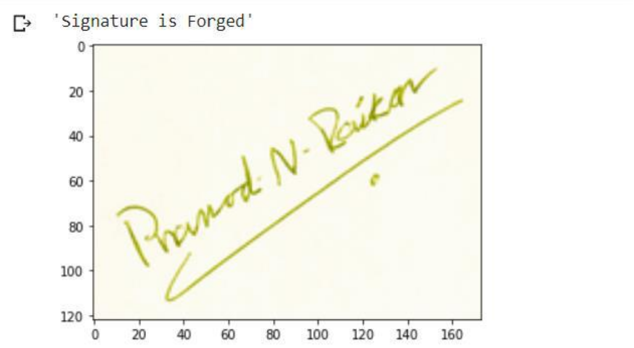
The testing model will determine whether a signature image is authentic or fake when it is given as input. The following are the various findings from several test instances:



'Signature is Real'

Figure 11: Test Case 1

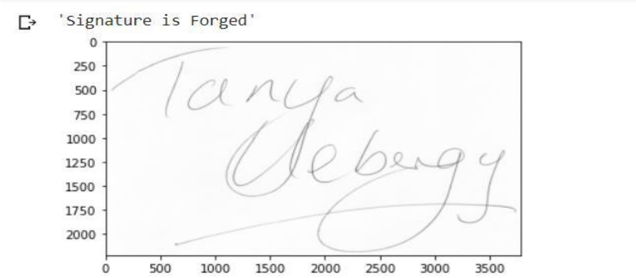
In the above output result, this is one of the valid test signatures passed to the model testing giving an output saying the signature is genuine.



'Signature is Forged'

Figure 12: Test Case 2

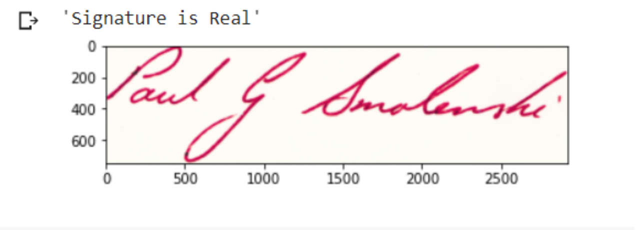
The above result is for an invalid image present in the test datasets giving an output saying that the signature is forged.



'Signature is Forged'

Figure 13: Test Case 3

The above output is outside, not a part of the testing database. This image was classified as forged.



'Signature is Real'

Figure 14: Test Case 4

The above result image in the test datasets can classify this image as accurate.

VI. CONCLUSION

In today's world, authentication and verification are critical as they can help prevent fraudulent activities, especially signature forgery. Hence it is necessary to build an efficient verification system to classify a signature as real or forged efficiently.

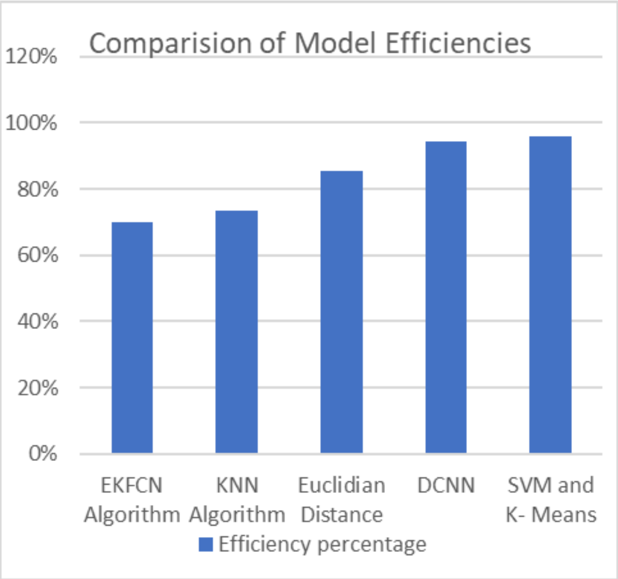


Figure 15: Comparison Results

Figure 15 shows the comparison results for the other implemented methods and the development led by SVM and K-Means. The model implemented above helps us to efficiently recognize fraudulent activities in the signature of a person using the Support Vector Machine (SVM), which will help us to find complex relationships between data without having to perform any complex transformations on our own. Here, the model is executed in google colabs mainly to see the results required.

The efficiency obtained here is 95.83% compared to all the other proposed systems in the chart.

VII. FUTURE ENHANCEMENTS

This model can be further enhanced by adding more feature extraction methods to accurately get the features and compare them to give optimal results.

This model can be implemented in banks to verify signatures on cheques, loan documents, etc.

REFERENCES

[1] Offline Signature Recognition and Verification System using Efficient Fuzzy Kohonen Clustering Network (EFKCN) Algorithm 2017: Dewi Suryani, Edy Irwansyah∗, Ricki Chindra.

[2] Handwritten Signature Verification using Local Binary Pattern Features and KNN 2019: Tejas Jadhav.

[3] A Study on Preprocessing and Feature Extraction in offline Handwritten Signatures 2015: Sm Abdullah Al-Mamun and Tansin Jahan Daffodil.

[4] An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach 2020: Hsin-Hsiung Kao * and Che-Yen Wen.

[5] Offline signature Verification 2017: Dr. M. Narayana and L. Bhavani Annapurna, K. Mounika.

[6] Offline Signature Recognition and Forgery Detection using Deep Learning: Jivesh Poddara, Vinanti Parikha, Santosh Kumar Bhartia,∗ .

[7] Shahane P.R., Choukade A.S., & Diyewar A.N. (2015) "Online biometric authentication mistreatment Matlab." International Journal Of Innovative analysis in Electrical, Physics, Instrumentation, and management Engineering

[8] Zagoruyko, S., & Komodakis, N. (2015). "Learning to compare image patches via convolutional neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4353-4361).

[9] Fahmy, M. M. (2010). "Online handwritten signature verification system based on DWT features extraction and neural network classification." Ain Shams Engineering Journal, 1(1), 59–70.

[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems (pp. 1097-1105).

[11] Khalajzadeh, H., Mansouri, M., & Teshnehlab, M. (2012). "Persian signature verification using convolutional neural networks." International Journal of Engineering Research and Technology, 1(2), 7-12.

[12] Batista, L., Granger, E., & Sabourin, R. (2012). "Dynamic selection of generative discriminative ensembles for offline signature verification." Pattern Recognition, 45(4), 1326-1340.

[13] Liwicki, M., Malik, M. I., Van Den Heuvel, C. E., Chen, X., Berger, C., Stoel, R., ... & Found, B. (2011). "Signature verification competition for online and offline skilled forgeries (sigcomp2011)". In 2011 International Conference on Document Analysis and Recognition (pp. 1480-1484). IEEE.

[14] Arena, F., & Soares, C. G. (2009). "Non-linear crest, trough, and wave height distributions in sea states with double-peaked spectra." Journal of Offshore Mechanics and Arctic Engineering, 131(4), 041105.

[15] Zhu, G., Zheng, Y., Doermann, D., & Jaeger, S. (2008). "Signature detection and matching for document image retrieval." IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(11), 2015-2031.

[16] "Offline signature verification and recognition by support vector machine." In 2005 13th European Signal Processing Conference (pp. 1-4). IEEE.

[17] Baltzakis, H., & Papamarkos, N. (2001). "A new signature verification technique based on a two-stage neural network classifier." Engineering applications of Artificial intelligence, 14(1), 95-103.

[18] Justino, E. J., El Yacoubi, A., Bortolozzi, F., & Sabourin, R. (2000). "An offline signature verification system using HMM and graphometric features." In Proc. of the 4th international workshop on document analysis systems

[19] Hsiao, P. Y., Lu, C. L., & Fu, L. C. (2010). "Multilayered image processing for multiscale Harris corner detection in digital realization." IEEE Transactions on Industrial Electronics, 57(5), 1799-1805.

[20] Pang, Y., Li, W., Yuan, Y., & Pan, J. (2012). "Fully affine invariant SURF for image matching." Neurocomputing, 85, 6-10.