

Offline handwritten signature verification using various Machine Learning Algorithms

Chinmay Lokare^{1,*}, Rachana Patil^{1,**}, Saloni Rane^{1,***}, Deepakkumar Kathiraseen^{1,****}, and Dr.Yogita Mistry^{1,†}

¹Ramrao Adik Institute of Technology, Nerul, India

Abstract. In today's world it is necessary to protect one's authenticity in order to ensure the protection of personal information that only the authenticate credentials of a person can have access to. Nowadays there is an increase in number of malpractices like signature forgery to access the important information of a person. To encounter signature verification problem, there have been a number of advances in verifying the authenticity of signature using various techniques including Machine Learning and Deep Learning. This paper introduces a novel approach to verify the signatures using difference of gaussian filtering technique, gray level co-occurrence matrix feature extraction technique, principle component analysis and kernel principal component analysis associated with various machine learning algorithms. The publicly available Kaggle offline handwritten signature dataset is used for training. This article compares the accuracy of the dataset on various machine learning algorithms. After training datasets the lowest accuracy achieved is 56.66% for Naive Bayes algorithm. The highest accuracy achieved is 82% for K-Nearest Neighbour (KNN) and 81.66% for Random Forest using principle components and kernel principle components of the dataset.

Keywords: Difference of Gaussians(DoG), GLCM, Principle Component Analysis (PCA), KPCA, K- Nearest Neighbour(KNN)

1 Introduction

The process of biometric verification is utilized for verifying the identity of people dependent on their special characteristics. It has gotten a famous norm for admittance to high security systems. One of the execution of biometric approval is Signature verification, which expects to perceive whether or not a given signature is certifiable(genuine) or fashioned(forged).

Authentication should be possible in two distinct approaches: Online and Offline based on how the signature is obtained. In the online strategy, the signature is captured while writing in this manner providing the dynamic data including area, speed, acceleration, pen pressure, pen lift, pen down, edge and time. Within the offline methodology, the signature is scanned once the signature is written so resulting in the static image of the signature known as the scanned signature. It's tough to differentiate a signature within the offline mode than in online mode that provides a lot of estimations.

Handwritten signatures have diverse dimensions and shapes and the varieties in them are so large at a time that it's miles difficult to test the authentic people. Also, the signature of an individual differs from time to time. Small varieties are intrinsic and these can be endured by the verification system. Here we have examined the effective us-

age of Offline signature verification using various filtering techniques.

2 Literature Survey

The field of handwritten signature verification has gotten a lot of attention in recent decades, but it's still a work in progress. The goal of offline signature verification, which is characterised by using images of signatures that are static in nature, is to determine whether a signature was created by the claimed person or by an impostor. This section gives an overview of how the issue has been approached by various researchers over the years, as well as recent advances in the field. Batista et al. [1] proposed a hybrid generative–discriminative ensemble of classifiers for building a writer-dependent HSV framework that dynamically selects the classifiers. During the generating stage, the signatures are separated into a grid structure and numerous independent left-to-right directions. To function at various levels of perception, HMMs(Hidden Markov Models) are trained with varying numbers of states and codebook sizes. Each enrolled signature's HMM likelihoods are then computed and aggregated into a feature vector, which is then used to build a pool of two class classifiers using a specialised Random Subspace Method (discriminative stage).

One-Class SVM (OC-SVM)[2], which aims to alleviate the difficulties associated with large user populations. The proposed approach models only one class as a one-class classification problem (genuine signatures). This is a use-

* e-mail: chinmaylokare9@gmail.com

** e-mail: rachanakp22@gmail.com

*** e-mail: salonirane29@gmail.com

**** e-mail: deepakkathir15@gmail.com

† e-mail: yogita.mistry@rait.ac.in

ful feature because, in most cases, the device only has legitimate signatures for each writer to train the classifier with. Nonetheless, a significant obstacle remains the low number of genuine signatures.

Offline Handwritten Signature Recognition using Biometrics, which refers to identifying an entity based on physiological or behavioural characteristics, has the potential to reliably differentiate between an approved person and an imposter, according to Gulzar A. Khuwaja and Mohammad S. Laghari [3]. The device is trained with low-resolution scanned signature images and uses a neural network to recognise offline handwritten signatures.

Off-Line Signature Recognition Systems were suggested by V Bharadi [4]. Handwritten signatures are one of the most commonly used biometric traits for document and individual authentication. The performance metrics of typical systems, as well as their feature extraction mechanisms, are contrasted.

Bertolini et al. [5] suggested a writer-independent method for handwritten signature verification in the WI scenario. Dissimilarity representation and SVMs as classifiers are used in this method [5]. The writers' two major contributions are as follows: I develop a new collection of graphometric features centred on the curvature of the most important segments, which is simulated with Bezier curves. (ii) Using an ensemble of classifiers structure to boost anti-forgery resistance.

3 Proposed Methodology

The authors have imported the handwritten signature dataset from Kaggle which consisted of 700 images for the training purpose and 300 images for the testing purpose which are further divided into genuine and forged signature images in both the training dataset and the testing dataset. Further the authors have pre-processed the images where they have reduced the noise from the gaussian denoising filter and further took the difference of gaussian. Now extracted the features from these noise reduced signature images using Gray-Level Co-occurrence Matrix or GLCM and then reduce the dimensions of these extracted features using the Principle Component analysis (PCA) and Kernel Principle Component analysis (KPCA). Further the accuracies are predicted on the test dataset on the basis of training of the train dataset on 10 different machine learning algorithms.

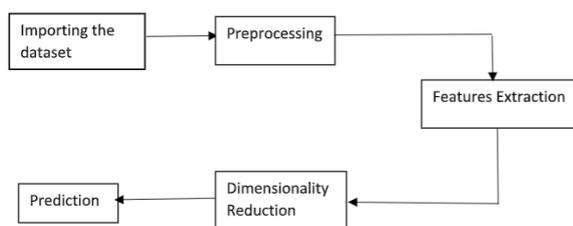


Figure 1. Proposed Methodology

3.1 Gaussian Denoising Filter

Gaussian blurring is a smoothing technique which makes use of a low pass filter whose weights are derived from a Gaussian function. It basically blurs the images and removes noise. The kernel represents a Gaussian hump (bell-shaped). The core concept is to use this 2D distribution as a "point-spread" function, which is accomplished using convolution. Because the image is stored as a collection of discrete pixels, first discrete approximation to the Gaussian function must be created before proceeding with the convolution. Gaussian smoothing is achieved using normal convolution methods after a sufficient kernel has been computed. A denoised image can be produced by performing Convolution of a smoothing kernel with required noisy images. Different denoising results can be obtained on the basis of the property of these kernels. Gaussian smoothing is achieved using normal convolution methods after a sufficient kernel has been computed. The Gaussian Distribution in 1D has formula:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where σ is the distribution's standard deviation. The authors have also concluded that the distribution has a mean of zero (that is, it's based on $x=0$). Figure below illustrates the distribution.

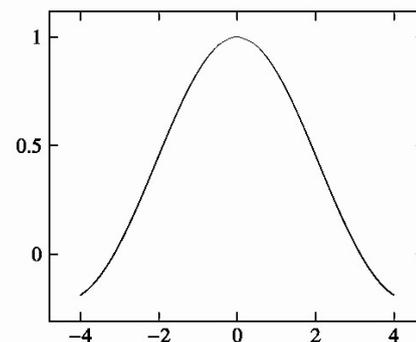


Figure 2. Gaussian distribution with 0 mean

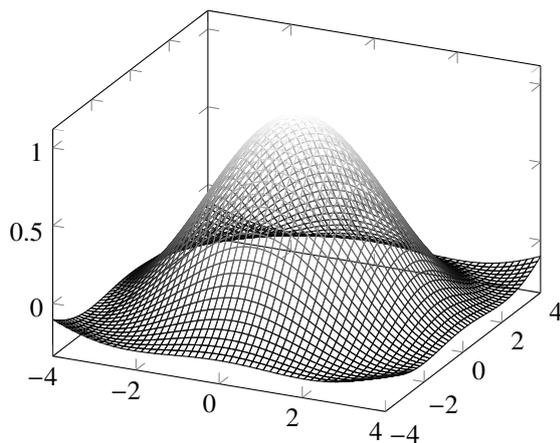
The Gaussian Distribution in 2D has formula:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where,

$$\sigma = 1 \text{ and mean}=(0,0)$$

Gaussian distribution in 3D



Gaussian smoothing is achieved using convolution, which involves employing this 2-D distribution as a point-spread function. Because the image is stored as a series of discrete pixels, first a discrete approximation to the Gaussian function must be built before proceeding with the convolution. The Gaussian distribution is non-zero everywhere in principle, requiring an infinitely large convolution kernel, but it is essentially zero beyond about three standard deviations from the mean in reality, allowing us to truncate the kernel at this stage.

3.2 Difference of Gaussians(DoG)

As stated earlier the Gaussian Blurring uses Gaussian kernels for image smoothing, further the authors have highlight certain high-frequency parts in an image. In Gaussian Blurring the standard deviation of the Gaussian affects the degree of smoothing. More high frequency components will be suppressed with larger standard deviation that is there will be more blurring. Thus, if two Gaussian Kernels are considered having different standard deviation of the same image and when the difference of them is taken, an output is received with a particular high-frequency components based on the standard deviations used. The basic logic is to remove high-frequency components by blurring that represent noise, and by taking the difference some low-frequency components of the homogeneous areas of the image are removed. Here, the difference of Gaussian acts like a band-pass filter. Here the authors have taken two Gaussian kernels with a Standard Deviation. $\sigma_1 \geq \sigma_2$ Here the kernel with signal1 when is convoluted with the signature image, the high frequency components will be blurred more as compared to the other kernel. When the subtraction of these is done the information that lies between the frequency range which is not blurred or suppressed is gained.



Figure 3. DoG Output

3.3 GLCM

Also referred to as co-occurrence distribution, a co-occurrence matrix is the distribution of co-occurring values at a given offset which is defined over an image or we can say it represents the distance and angular spatial relationship of specific size over an image. A gray-scale image is used to build the GLCM. It determines how frequently a pixel with the gray-level value x appears horizontally, vertically, or diagonally to adjacent pixels with the value y . Horizontal (0 degree), vertical (90 degree), diagonal with bottom left to top right (-45 degree), and top left to bottom right (-135) are all possible directions in GLCM (P0, P45, P90, and P135, respectively, are the directions. For example:

Lets take a point $P(x,y)$ Where x and y are the gray level values (tone) in the image

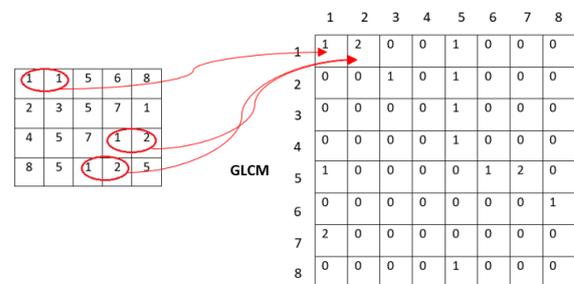


Figure 4. process used to create the GLCM

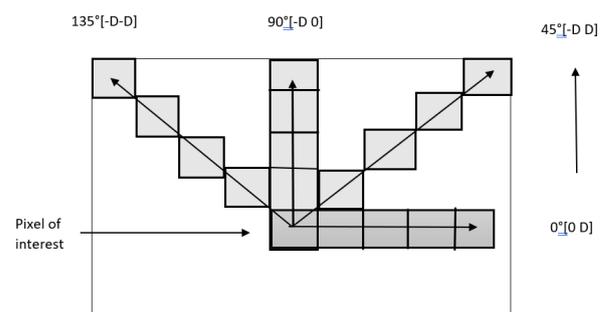


Figure 5. GLCM angles

Here you can see the element (1,1) in GLCM matrix has a value 1 as in the original matrix there is only one because there is only one scenario in the input image when two horizontally adjacent pixels have the values 1 and 1, the GLCM matrix element (1,1) has the value 1 because the original matrix only had one. Element (1,2) in the GLCM has the value 2 because there are two, the value is two occurrences of two pixels that are horizontally adjacent with the values 1 and 2, whereas in the GLCM, Element

(1,3) has the value 0 since there are no horizontally contiguous pixels with the values 1 and 3.

GLCM features:

The authors have used 5 GLCM features namely Energy, Correlation, Homogeneity, Contrast and Dissimilarity. And are used on 4 different offset distances and 3 angles combination. Finally total of 25 different features with these combinations were gained. Let's have a look at these features one by one.

• **Energy :-**

Energy(ene) =

$$\sum_x \sum_y g_{xy}^2$$

where,

$$\sum_x = \sum_{i=1}^{N_g}$$

$$\sum_y = \sum_{j=1}^{N_g}$$

Ng = Number of distinct gray levels in the image

g_{xy} = (x, y)th entry in GLCM

$$g_{xy} = \sum_{y=1}^{N_g} g(x, y)$$

It is also known as Uniformity or Angular second momentum which measures the textural uniformity, that is pixel pair repetition. It is responsible for the detection of disorders in textures. The maximum value Energy reaches to is 1.

• **Correlation –**

It is the measure of gray tone linear dependencies present in the image.

$$\text{Correlation (cor)} = \frac{\sum_x \sum_y (xy)g_{xy} - \mu_i \mu_j}{\sigma_i \sigma_j}$$

where,

$$\sum_x = \sum_{i=1}^{N_g}$$

$$\sum_y = \sum_{j=1}^{N_g}$$

Ng = Number of distinct gray levels in the image

g_{xy} = (x, y)th entry in GLCM

$$g_{xy} = \sum_{y=1}^{N_g} g(x, y)$$

μ_i, μ_j, σ_i and σ_j are the means and standard deviations of g_i and g_j

• **Homogeneity –**

$$\text{Homogeneity (hom)} = \sum_x \sum_y \frac{1}{1+(x-y)^2} g_{xy}$$

It's also known as Inverse Difference Moment, and it's a measurement of image homogeneity that assumes greater values for smaller grey tone variations between pair elements. In the presence of near diagonal elements in GLCM its sensitivity is high. When all the elements in the image are same it has a maximum value. GLCM Homogeneity and GLCM contrast are substantially but inversely associated in terms of equivalent distribution in the pixel pairs population. This means that homogeneity decreases if contrast increases while the energy is constant. where,

$$\sum_x = \sum_{i=1}^{N_g}$$

$$\sum_y = \sum_{j=1}^{N_g}$$

Ng = Number of distinct gray levels in the image

• **Contrast:**

Contrast measures the spatial frequency of an image

$$\text{Contrast (con)} = \sum_x \sum_y (x - y)^2 g_{xy} \text{ where,}$$

$$\sum_x = \sum_{i=1}^{N_g}$$

$$\sum_y = \sum_{j=1}^{N_g}$$

and is difference momentum in GLCM. It is a difference between the highest and the lowest values of a contiguous pixel sets. The amount of local variations in and image is measured by Contrast. Low spatial frequency and a low contrast image exhibit the GLCM concentration term around the principle diagonal.

• **Dissimilarity-**

It is basically a measure of distance between the pairs of pixels. $\text{dissimilarity} = \sum_x \sum_y |(x - y)|p(x, y)$

Below are 5 features of two signature images with distance 3 and angle 0.

	Energy	Corr	Diss_sim	Homogen	Contrast
0	0.5706	0.3507	34.0768	0.6689	8328.8668
1	0.5874	0.4373	28.0754	0.7051	6905.2941

3.4 PRINCIPLE COMPONENT ANALYSIS (PCA)

PCA is a dimensionality reduction technique which is used to reduce the dimensionality of larger dataset. It does so by transforming a large dataset of variables into smaller one that usually retains most of the information. The

purpose of reducing the dimensions is to make a dataset more simple because it is usually much easier to visualize smaller datasets and explore them that the smaller ones which eventually helps for faster learning for the Machine Learning algorithms. The Principle Components are the new variables that are created by combining the basic variables in a linear fashion. The calculation of these combinations is done in such a manner that the newly generated variables, or Principle Components, are not correlated, and the majority of the information contained in the starting variables is squeezed or compressed into the first components. Steps to perform PCA over a dataset.

1. Standardization – Standardization of the range of the original continuous variables from 0 to 1 so that they all contribute equally to the analysis is the primary purpose of this step.
2. Covariance Matrix Computation – The purpose of this step is to be aware of how the input variables are varying from mean with respect to each other. This is done because in some cases the variables might contain redundant information as they are correlated on a higher degree.
3. Computation of Eigen Vectors and Eigen values – The authors have computed the Eigen Vectors and Eigen Values from the covariance matrix to determine the Principle components.
4. Feature Vector – A feature vector is a matrix with the eigenvectors of the components as columns.
5. Recast the Data – The data is reoriented to the ones represented by the Principle Components using the Feature Vector from the original axes. This is accomplished by multiplying the original dataset's transpose by the feature vector's transpose. We have got 25 components in the GLCM. We further reduce these 25 into 5 components.

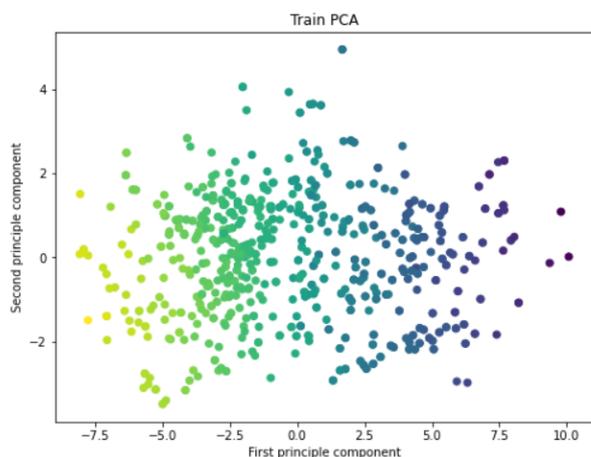


Figure 6. Train PCA

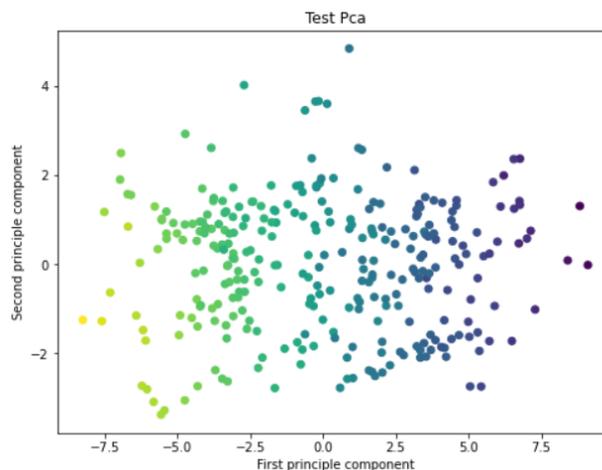


Figure 7. Test PCA

3.5 KERNEL PRINCIPLE COMPONENT ANALYSIS (KPCA)

Since the data has been retrieved from images there is also certain non-linearity in the structure of the extracted features. The PCA is used to reduce the dimensions of the linearity in the structure of the images. Further KPCA is used to reduce the dimensions of the non-linear structure of the data. The theory behind the KPCA is that by projecting data into a higher dimension space, many data that are not linearly separable in space can be made linearly separable. Simple arithmetic operations on the original data dimensions are used to create the extra dimensions. Here also the authors have reduced the 25 components into 5 components retrieved from the GLCM.

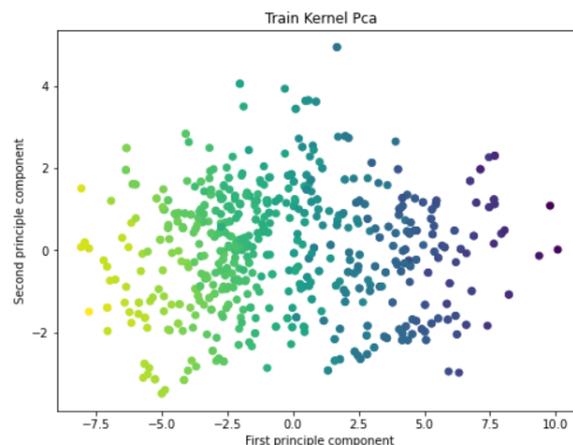


Figure 8. Train Kernel PCA

4 Result

Standardizing the GLCM data results are received in a scaled dataset. Further this scaled dataset is applied to the PCA, KPCA and received reduced dataset. Later these

Table 1. Result

	Scaled dataset	PCA Dataset	Kernel PCA Dataset
ANN	76.00	74.67	74.00
SVM	70.00	70.33	70.33
Logistic Regression	63.00	62.00	62.00
KNN	82.00	75.66	75.66
Decision Tree	62.66	74.33	75.33
Bagging Decision Tree (Ensemble Learning I)	75.33	75.33	62.66
Boosting Decision Tree	62.66	75.33	75.33
Naive Bayes	56.66	62.33	62.33
Random Forest	67.66	81.66	81.66
Voting classification	69.66	71.66	72.00

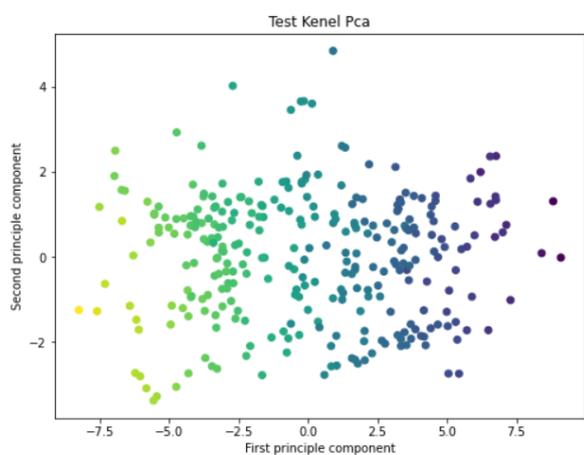


Figure 9. Test kernel PCA

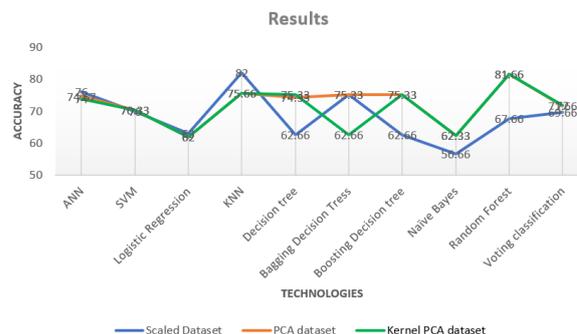


Figure 11. Technologies vs Accuracy %

three datasets are applied to 10 different Machine Learning algorithms and got the results which are displayed in Table 1. Amongst these, the scaled dataset performed well with the accuracy of 82% on the KNN and the PCA, KPCA datasets performed almost the same with the Random Forest.

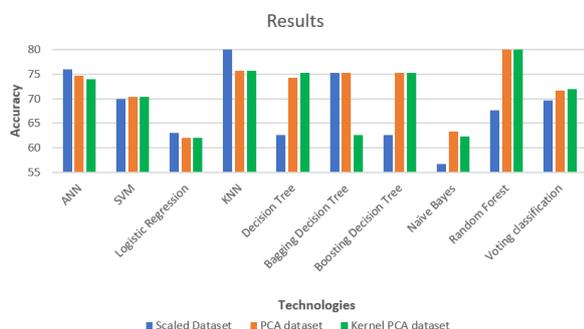


Figure 10. Technologies vs Accuracy

5 Conclusion

This paper proposed a novel approach to verify the signatures using difference of gaussian filtering technique, gray

level co-occurrence matrix feature extraction technique, principle component analysis and kernel principal component analysis associated with various machine learning algorithms. Signature images are trained with image processing algorithms, machine learning and deep learning algorithms. The K-Nearest Neighbour algorithm is assessed under the following conditions: varying circumstances, increasing or decreasing the quantity of multiple signing sessions, signature training database with a variety of inks and expanding the number of possible combinations signatories. By comparing with the existing techniques, KNN gives better result with 82% accuracy. In this field, numerous methodologies are used, but accuracy has to be improved. The accuracy achieved thus far by existing systems is not particularly high, and much more study into offline signature verification is required. Future work will be focused on improving the accuracy and determining the ideal approach.

References

- [1] L. Batista, E. Granger, and R. Sabourin, "Dynamic selection of generative–discriminative ensembles for offline signature verification," *Pattern Recognition*, vol. 45, no. 4, pp. 1326–1340, 2012.
- [2] Y. Guerbai, Y. Chibani, and B. Hadjadj, "The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognition*, vol. 48, no. 1, pp. 103–113, 2015.

- [3] Gulzar A. Khuwaja and Mohammad S. Laghari Off-line Handwritten Signature Recognition using Biometrics April 2011 at International Scientific Academy of Technology and Engineering
- [4] V A Bharadi Off-Line Signature Recognition Systems Proceedings of the International Conference and Workshop on Emerging Trends in Technology, ICWET 2010
- [5] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, 2010.
- [6] A. Bhanu Srothara, M. Hanmandlu, Offline Signature Verification using CNN, *International Journal on Future Revolution in Computer Science & Communication Engineering* Volume: 4 Issue: 9, September 2018.
- [7] Sameera Khan, Avinash Dhole, A Review on Offline Signature Recognition and Verification Techniques Vol. 3, Issue 6, June 2014