International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

Signature Verification using Edge Detection and Oc-Svm

Mokshith Reddy A.V, Harsha Nimmagadda, Venkatesh Pallapothu, Jasmine Pemeena Priyadarsini.M

Abstract— ***There are many researches going on in the field of Image processing to find an accurate model for detection of forged signatures. It is compulsory to select appropriate algorithm for achieving best results. Many have implemented the available algorithms like LBP, HOG, geometric features and some have designed their own extraction techniques like Histogram of template and mixture of many complex algorithms to detect the forgeries. In our model we extracted edge features of the image using canny edge detection and then extracted features using HOG and then we calculated area, standard deviation, centroid, kurtosis etc. for the edge image which give a feature vector of length 262. There is little research towards one class SVM or outlier detection. We trained our model with different kernels of SVM to find which kernel gives best result. This OC-SVM will be very helpful than a multi class SVM as we will be having only original signatures of the users and not forged so it will be best to use though it is very sensitive it can be used even in the real world.***

Keywords— ***Average Error Rate (AER), False Acceptance Rate (FAR), Feature extraction, False Rejection Rate (FRR), Histogram of Oriented Gradients (HOG), Preprocessing, Support Vector Machine (SVM), Radial Basis Function (RBF)***

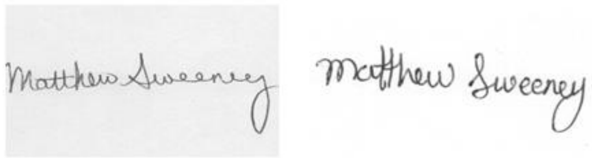
I.INTRODUCTION

Even in today's world we use paper for signing signature for many things as we did not get digitalized completely like for bank checks, contracts, agreements and many crucial things. We have many other developed biometrics like finger print, eye scanner etc. but we still use signature as biometric because everyone has their own hand writing and if we use an efficient algorithm, we can differentiate one user's signature with other. Also cost of production for this kind of biometric is less compared to others. There are two kinds of verification system one is offline and other is online verification. We have taken offline verification as many important transactions happen offline mode. In online verification we have measurements that are taken while keeping the signature like pressure, number of lifts, Smoothness, acceleration pattern etc.

Manuscript received on April 02, 2020.
Revised Manuscript received on May 21, 2020.
Manuscript published on May 30, 2020.

* Correspondence Author
Mokshith Reddy A.V*, Department of Electronics and Communication Engineering, Vellore Institute of Technology, Vellore, India
Harsha Nimmagadda, Department of Electronics and Communication Engineering, Vellore Institute of Technology, Vellore, India
Venkatesh Pallapothu, Department of Electronics and Communication Engineering, Vellore Institute of Technology, Vellore, India
Jasmine Pemeena Priyadarsini.M, Department of Electronics and Communication Engineering, Vellore Institute of Technology, Vellore, India

But we don't have these information as for many transactions are done offline so we use only static features of signature rather than dynamic which are used in online verification. Some of the static features are image texture, geometry, no of corners, topology of signature. Offline verification of signature is difficult when compared to online because there will be less repetitiveness of comparable features due to the area available for signing, emotional status at that situation, age etc. There are two main kinds of features for image one is global features and other is local features. The global features represent the whole image where local features represent the image based on its neighbor.



a) original signature **b) forged signature**
Fig.1 Sample signatures from database

Then for verification there are two types of approaches when it comes to biometric. They are as follows:

1) *Writer Independent:* It is a one to many Comparison

approach. The signature that needs to be tested we will be compare with every other signature present in database. We need to train the model only once. The problem with this is that if a forged signature features match with others then also it will produce output as original.

2) *Writer Dependent*: This is like one to one approach.

The verifying signature needs to be compared with only the specific user's database. We need to train model for every user and store them separately. In our paper we will be comparing different kernels of SVM like gaussian, linear and polynomial to find the best model with high accuracy and low FAR, FRR and AER. FAR is no of false signatures that got accepted and FRR is no of genuine signature that got rejected and AER is the average error rate of both FAR and FRR. If a mode gives low FAR then we can reduce the frauds and if even if it has high FRR as we will be asking the user for another verification but to it is necessary to have both FAR and FRR to be low to have the faith of the user.

II.PREVIOUS WORK

There were many researchers that contributed a lot towards this field of biometric. Few of the verification systems are taken into account and will be briefly discussed here. Feriel Boudamous *et al.* [1] introduced a verification system based on Histogram of Templates (HOT) which consists of 20 templates that describe all possible orientations that can take a segment centered on a pixel P.

Then we will be sliding through the signature with 3x3 window and find no of pixels that fit with corresponding template.
Templates will be bins and
No of pixels for that correspond to that template will be values of bins and later used SVM for verification. He got an accuracy of 99.29%. But this method fails by influence of disturbance such as illumination changes and rotation.
Snehal K. Jadhav, M.K. Chavan [2] implemented Local Binary Pattern as feature vector. It is a local feature extraction method. In his we will be considering a 3x3 window and will be comparing the intensity values of center with neighboring values if it is greater than center then it will be updated with 1 else 0 then a binary code is generated based on this. Then the center pixel will be updated with the new value generated from the binary code. The histogram of the new image will be taken as feature vector. Then he did a comparison between SVM and KNN and got SVM as better classifier with an accuracy of 67% for Writer dependent approach and 90.36% for Writer Dependent approach.

Srikanth Pal *et al.* [3] Used Zernike moments which represent image when combined using Zernike polynomial. These polynomials are orthogonal to unit circle which are complex valued polynomial, so this can represent even complex valued data. Then he also used Gradient features as another descriptor which uses Robert filter for extraction feature vector which is 576 long. It contains histogram of the edge image which are divided into 9x4 blocks each containing 16 directions. Then verified both using linear SVM different values of gamma and found the optimum values for both out of which gradient features performed better with accuracy of 90.69%. They did an effective contribution to the field of non-western signature verification.

Ashok Kumar, Karamjit Bhatia [4] introduced a combination method which used both local and Global features as feature vector. First, we will be generating global features namely area, mean, standard deviation etc. First, we will be extracting 10 features as whole then divide the image into 4 parts and again do the extraction we will be getting 40 local and 10 global features. This will be used for training the neural network and then he also used LBP for the image and computed histogram by dividing the image into blocks. He then generated a feature vector of length 256. Then he used only genuine and random forged signatures for training as the organization may not have skilled forged signatures using neural network. He then trained using both neural network and SVM-poly and calculated the time taken by both approaches and also their accuracy. It is found that SVM-poly took less time in both training and testing time. It is seen that Local Binary Pattern (LBP) performs better than geometric features with SVM-poly as classifier.

Miguel A. Ferrer *et al.* [5] used a bit complex algorithm for feature extraction. They first dilated the original signature and then filled the total signature and took the outline of the signature. They took 3 features for as feature vector for 64 points which they found to be a good trade-off between recognition ratio and computation. First feature will be radius to the point from geometric center. Then the next feature for a point is angle of point, which is calculated by arctan. The last feature will be no of black pixels that the radius crosses when sweeping from this point to next. The next feature vector is also based on envelope signature stroke density, but here will be using cartesian coordinates. He then used Hidden Markov Model (HMM), SVM and Euclidean Distance based signature models for verification. It is found that HMM performed better than SVM and Euclidean distance performed worst.

Weiwei Pan, Guolong Chen [6] have implemented a simple yet effective algorithm keeping both time and accuracy into mind. First, they partitioned the image into equal region grids and then found the sum of black pixels in that partition. This will be the feature vector which will be normalized in order to avoid large numeric ranges dominating smaller. Then he trained the data using two kinds of SVM one is linear kernel
And other is radial kernel. Linear kernel is less sensitive to parameter selection whereas radial kernel is performing best when accurate parameters are used which may require many experiments. A.B.M. Ashikur Rahman *et al.* [7] proposed a model of applying Harris corner detection which detects corners or isolated maximum and minimum of curve and then orientation of these points is found using 16x16 neighbors and then SIFT is used to find the descriptor. Then the key points for matching using distance ratio selection to find the thresholding distance also no of matches between two signatures variance must be less than thresholding variance. Madhuri R. Deore *et al.* [8] used parameter features which he extracted by applying DWT (Discrete Wavelet Transform) to the image which produces wavelet coefficients and then verified the model using different verification systems like HMM, SVM, neural network etc.

III.METHODOLOGY

We have followed writer dependent approach in our model. We will be comparing the input signature with the specific user rather than with all as mentioned in introduction.
The block diagram of our model is shown in Fig.3 and the below are each step explained detailly.

A. Preprocessing:

We will be preprocessing the signature to remove un-wanted information from the image like noise and we will also remove color as it does not contain any information. Here we will be converting the colored image into gray scale image as we don't have any information present in color, as one person can put signature with different ink. Then we will be resizing the image to 256x256 to trade-off between computation and accuracy. Then the image is applied with gaussian filter to remove any noise present in the image. We have used 5x5 gaussian filter coefficients to smooth the image.

B. Local feature extraction:

For local feature extraction we will be using edge detection using sobel operator. For edge detection there are three main steps first is calculation of gradient magnitude.
We will be using horizontal and vertical direction filters to get the boundary of the image using the below by sliding the window of the image of size 3x3 and find the resulting horizontal gradient and vertical gradient.

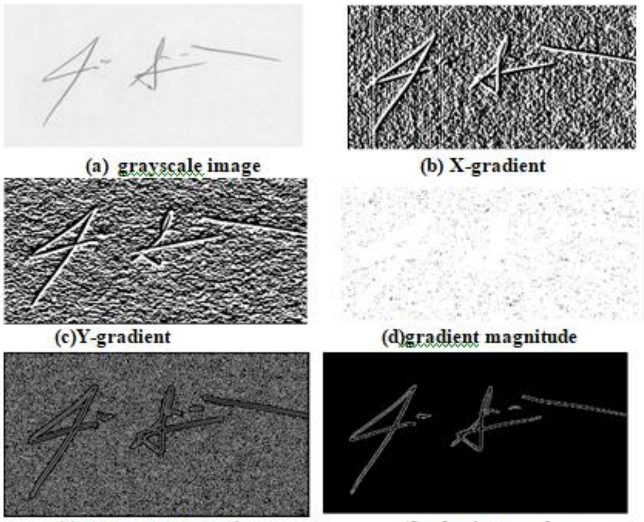
$$\mathrm{Gx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Gy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using (1) and We can also find the direction at each point as we know the x and y direction magnitude.
We can use slope to find the direction in which the edge is going with the formula (2)

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\Theta = \tan^{-1}(Gy/Gx) \quad (2)$$

As we will be getting the gradient direction in negative, we will be converting them into positive by adding 360 to the negative ones. Now we want to remove the duplicate edges identified by Sobel Edge Detection. We want just one line to show the edge rather than having multiple lines for the same edge. This can be done by the Non-Max Suppression Algorithm.



(a) grayscale image (b) X-gradient (c)Y-gradient (d)gradient magnitude (e)non-max suppression (f) edge image after thresholding

Fig.2 Process of converting original image to edge image

A pixel can have total 4 directions for the gradient since there are total 8 neighboring pixels. We want to make sure no adjacent pixel is referring the same edge. We will be comparing the gradient magnitude in the direction of the referenced pixel and check if the before and after pixel are less than or equal to the referenced pixel if it is more then we will be updating the value else leave as it will be representing the same edge of some pixel.

We will be taking 8 direction vectors for a pixel and then find its direction value. Suppose say the gradient direction of a given pixel is 0 degree. So, we will compare the magnitude of gradient of the right and left pixel with it. In this example, clearly the magnitude of gradient of the selected pixel is higher than the other two, hence we update our output pixel value by the magnitude of gradient of the selected pixel. Now as we don't get every direction of pixel as exactly 0,45,90,135,180,225,270,315, we will be dividing the interval and if they are between them then they will be updated with new direction. As 0 and 180,45 and 225,90 and 270,135 and 315 for the sake of computation. (e) in Fig.2 shows the image after applying the above steps. This process of removing the pixels that represent same edge is called non-max suppression.

For separating foreground from background, we use hysteresis thresholding. In this we will be using two thresholding values one if pixel value is above high threshold value then it is given 1 and if it is between low and high threshold and if it is connected to any edge pixel then it will be given 1 else 0. (f) in Fig.2 is the final image after applying thresholding and this image will be used for extracting local features and global features. We will be dividing the image into 8x8 blocks with each block containing 32 pixels and then we will be finding the histogram of each block and add them together to get the feature vector. The feature vector is of length 8*8*4 i.e. 256. We will now extract global features.

C. Global feature extraction:

The edge produced in the last step will be used for extracting the global features. we will be extracting area, centroid, standard deviation, skewness and kurtosis for the edge image that will give area of edge center of the signature and other statistical parameters are discussed in [7][8]. These features will be combined with the local features so in total we will be having a 262-length feature vector we will be using this to train Machine learning model.

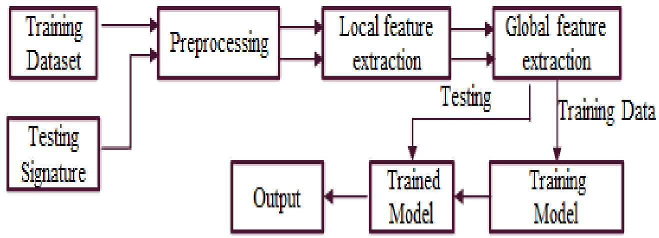


Fig.3 Signature verification system

D. Training and Testing:

For training we will be using different kinds of kernels with SVM. They are linear, rbf, polynomial [11]. As we don't have forged signatures data with the verification department we need to rely on original database. There are three parameters that varies the accuracy of the model. They are kernel function,
Outlier fraction and Nu. Kernel function is used to solve non-linear problems using linear classifiers. Outlier fraction is the percentage of data that has to removed. The removed data corresponds to large gradients.
Nu is the value that gives tradeoff between ensuring that most training examples are in the positive class and minimizing the weights in the score function. A value closer to 0 will give a fewer support vectors and crude decision boundary and high value gives more support vectors and a curvy, flexible boundary. We will be training using different functions, outlier fractions and Nu and find the optimum values for high accuracy. We will be generating the scores of each testing signature and if the score is greater than 0 then it belongs to original class else forged.

IV.EXPERIMENTAL RESULTS

For training we will be using 15 original signatures and for testing we will be using 150 random forgeries, 24 skilled forgeries and 9 original signatures. We will be setting outlier fraction close to zero as we need to include almost all signature's for training. Now for finding the optimum value of Nu we will iterate through each value from 0 to 1 and find for which value we get high accuracy.

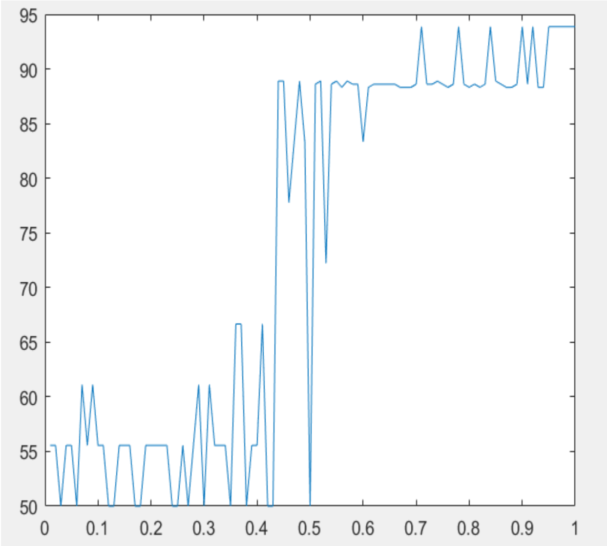
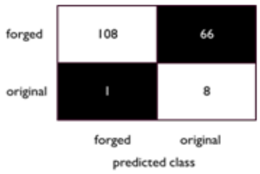


Fig.4 Nu vs ACCURACY with outlier fraction=0.01

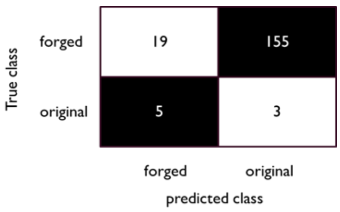
It is found to choose Nu closer to 1 as we get more flexible boundary as shown in Fig.4. The table.1 is comparison between different kernels with optimum Nu, outliner fraction.

It can be seen that gaussian or RBF kernel outperformed the other two. Linear and polynomial kernels work very good with two class classification problems but are very poor when it comes to outliner detection. No of signatures available for training also plays as important role . The below figures are the confusion matrix which gives all the information about False Rejection Ratio, False Acceptance Ratio and accuracy. In



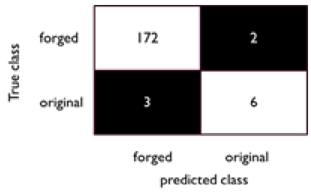
(a) Gaussian

(b) linear

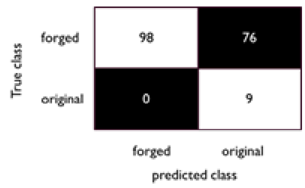


(c) Polynomial

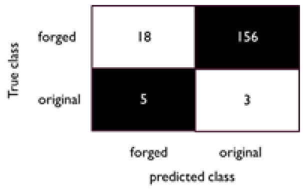
Fig.5 Confusion matrix for different kernels with training signatures=15



(a) Gaussian



(b) linear



(c) Polynomial

Fig.6 Confusion matrix for different kernels with training signatures=13

True class represent the known label and predicted class represents the output label after giving them to trained model. White region indicates the no of signatures that are correctly predicted and black region represent the no of signatures that are incorrectly predicted. FAR will be ratio of no of forged accepted to be original the total forged signatures used. FRR will be ratio of no of original signatures predicted to be forged to the total original signatures used for testing. AER is the total error of the system that is it is the ratio of total incorrect predictions to total signatures for testing

$$Accuracy = \frac{(100 - FAR) + (100 - FRR)}{2}$$

Here in Table.1 we used confusion matrices from Fig.5 to calculate FAR, FRR, AER, Accuracy and Table.2 we used confusion matrices in Fig.6. which will help in better understanding of the results.

kernel	FAR	FRR	AER	ACCURACY
gaussian	1.14%	11.11%	6.125%	93.86%
linear	25.29%	0%	24.04%	87.36%
polynomial	99.43%	11.11%	95.08%	44.73%

Table.1 Accuracy of different kernels with Nu=0.9, outliner fraction=0.01

kernel	FAR	FRR	AER	ACCURACY
gaussian	1.15%	33.33%	2.73%	82.76%
linear	41.95%	0%	39.89%	79.02%
polynomial	98.85%	11.11%	94.54%	45.02%

Table.2 Accuracy of different kernels with Nu=0.9, outliner fraction=0.01

It can be clearly seen that no of signatures used for training will have a huge impact on the model. But in both Tabel.1 and Table.2 it can be seen gaussian or RBF performs best with One class classification.

Retrieval Number: A3121059120/2020©BEIESP
DOI:10.35940/ijrte.A3121.059120
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V.CONCLUSION

So, we have implemented edge detection along with global edge features for feature extraction and used different kernels for training the model and found that gaussian kernel outperformed others as it is one-class classification. Confusion matrix gave a better understanding of FAR, FRR and also in calculating accuracy. Gaussian kernel gave an accuracy of 93.86% when we use 15 training signature and an accuracy of 82.76% when 13 signatures are used for training. So, we can say that no of signatures available at training will also impact the model prediction. we don't have forged signature data so we need to solely depend on original so it is best to use gaussian kernel. This is an outlier detection problem. In our future work we will try to implement an efficient feature extraction technique that will work even with small training dataset.

REFERENCES

1. F. Boudamous, H. Nemmour, Y. Serdouk and Y. Chibani, "An-open system for off-line handwritten signature identification and verification using histogram of templates and SVM," *2017 International Conference on Advanced Technologies for Signal and Image Processing* (ATSIP), Fez, 2017, pp. 1-4
2. S. K. Jadhav and M. K. Chavan, "Symbolic Representation Model for Off-Line Signature Verification," *2018 9th International Conference on Computing, Communication and Networking Technologies* (ICCCNT), Bangalore, 2018, pp. 1-5.
3. S. Pal, U. Pal and M. Blumenstein, "Off-line verification technique for Hindi signatures," in *IET Biometrics*, vol. 2, no. 4, pp. 182-190, December 2013.
4. A. Kumar and K. Bhatia, "A robust offline handwritten signature verification system using writer independent approach," *2017 3rd International Conference on Advances in Computing, Communication & Automation* (ICACCA) (Fall), Dehradun, 2017, pp. 1-6.
5. M. A. Ferrer, J. B. Alonso and C. M. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 993-997, June 2005.
6. B. Fang, C. H. Leung, Y. Y. Tang, P. C. K. Kwok, K. W. Tse and Y. K. Wong, "Offline signature verification with generated training samples," *in IEE Proceedings - Vision, Image and Signal Processing*, vol. 149, no. 2, pp. 85-90, April 2002.
7. A. B. M. A. Rahman, G. Mostaeen and M. H. Kabir, "A statistical approach for offline signature verification using local gradient features," 2016 2nd *International Conference on Electrical, Computer & Telecommunication Engineering* (ICECTE), Rajshahi, 2016, pp. 1-4.
8. R. Wajid and A. B. Mansoor, "Classifier performance evaluation for offline signature verification using local binary patterns," *European Workshop on Visual Information Processing* (EUVIP), Paris, 2013, pp. 250-254.
9. S. Chandra and S. Maheskar, "Offline signature verification based on geometric feature extraction using artificial neural network," 2016 3rd *International Conference on Recent Advances in Information Technology* (RAIT), Dhanbad, 2016, pp. 410-414.
10. Htight Htight Wai, Soe Lin Aung, "Feature extraction for offline signature verification system", *IJCCER*, vol. 1, no. 3, pp. 84-87, 2013.
11. K. -. Muller, S. Mika, G. Ratsch, K. Tsuda and B. Scholkopf, "An introduction to kernel-based learning algorithms," in *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181-201, March 2001.

AUTHORS PROFILE

Mokshith Reddy A. V pursuing his B. Tech in Electronics and Communication Engineering from Vellore Institute of Technology, Vellore. His areas of interest are Machine Learning and Image Processing. He has done many projects based on Machine Learning

Harsha Nimmagadda pursuing his B. Tech in Electronics and Communication Engineering from Vellore Institute of Technology, Vellore. His areas of interest are Image Processing and Neural Networks. He has done many projects based on Image Processing

Venkatesh Pallapothu. V pursuing his B. Tech in Electronics and Communication Engineering from Vellore Institute of Technology, Vellore. His areas of interest are Digital Logic Design and Cryptography

Jasmine Pemeena Priyadarsini. M completed her M. Tech in Communication Engineering. Her research interest includes Digital Image Processing. She is currently working as Associate Professor Sr. in Vellore Institute of Technology, Vellore. She assisted as a guide for so many students to publish their research papers.