Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

Offline Signature Verification and Forgery Detection using Critical Region Matching Method

Shraddha Rao
MS in Information Management
Syracuse University
Syracuse, USA.

Shraddha Inamdar
Assistant System Engineer
Tata Consultancy Services
Mumbai, India.

***Abstract* – In this era of growing technology, biometric systems are being used for personal identification, amongst those one system is Signature Verification System. The objective of this system is to identify the original and forged signature. In this project, we have implemented an Offline Signature Verification and Forgery Detection using Critical Region Matching Method. In this method, the handwritten signatures are scanned and they undergoes a series of image processing techniques and the critical region of the signatures are obtained. The extracted regions are then compared using the Critical Region Matching Method to detect forgery if any.**

Keywords – Offline Verification of Signature, Critical Point, Contour Image Extraction, Critical Region Matching, Critical Point Extraction, Signature Verification.

I. INTRODUCTION

A signature is a personal name written in one's hand on a document, to signify acceptance or approval. Many documents necessitate a handwritten signature. This being a very important area the problem of signature verification and forgery detection becomes crucial [1].

Handwritten signatures vary in their shapes and sizes due to many factors like physical and psychological state of mind, body position, environment conditions, the writing surface, the material of the paper [2]. Due to such variations, sometimes it becomes difficult to differentiate between an original and forged signature just by having a glance at the signature. Hence, we need a Signature Verification System that will help us to authenticate the signatures.

The two most widely used approaches for signature verifications are:

a. Offline Signature Verification
b. Online Signature Verification

In Online Signature Verification System, the user has to do his signature with a pen-based tablet that records the signature in the system. The signature trajectory, pen pressure, pen downs and pen ups are captured by the tablet and then sent to the system. The system then verifies the signature against the database to check whether it's original or forged [3].

In Offline Signature Verification System, the user has to sign on the paper. The signature is scanned and undergoes series of image processing techniques and then is stored in the system. The system applies the algorithm to the signature images and checks whether it's genuine or forged.

In this project, we have implemented an Offline Signature Verification System that detects forged signatures using Critical Region Matching Method [4].

II. SYSTEM OVERVIEW

Offline Signature Verification System is mainly divided into the following steps:

A. Data Acquisition

The first step in the design of an Offline Signature Verification system is data acquisition [5]. Handwritten signatures are collected on a paper. No restrictions are placed on the colour of the ink used or the type of the paper. The signatures are then captured using 8 MP camera. Five samples of each signature are collected. Out of these five samples, four signatures are genuine and one is forged. Three samples of genuine signatures are stored in the database as reference signatures.

B. Pre-Processing

The signature images go through following image processing techniques:

1. Binarization
2. Noise Removal
3. Rotation of Signature
4. Cropping of Signature

C. Contour Image Extraction

The contour of the image helps to identify the critical points of the signature [4].

D. Critical Point Extraction

The critical points of the signature are obtained using the Critical Point Extraction Algorithm [4]. This algorithm uses the contour image as input to extract the critical points.

E. Critical Point Match

After the critical points are obtained, the signatures are processed to find out the number of matching critical points. This can be obtained using the Critical Point Matching algorithm [4].

F. Verification

The critical points of the referenced and the questioned signatures are compared to judge the overall similarity between the input and sample signatures using Critical Region Extract and Match Algorithm.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

III. PRE-PROCESSING

Pre-Processing is a set of operations applied to the images in order to improve the quality of the image. Improved quality of the image helps us to improve the accuracy of the steps that help us to distinguish between the genuine and forged signature.

The various sub-processes of Pre-Processing are as follows:

A. Binarization

In this process, the image is first converted to a grey scale image. The grey scale image is then converted to a binary image. In this process, each pixel of a grey scale image is converted into one bit. It is assigned a value '1' or '0'depending upon the mean value of all the neighbouring pixel. If the value of the pixel is greater than mean value then it is assigned '1' else it is assigned '0'.

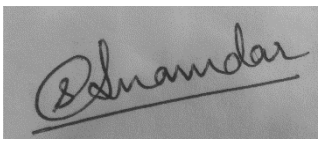

Fig 1. Original Image

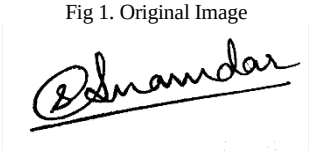

Fig 2. Binary Image

B. Noise Removal

Once a binary image is obtained, the noise components, if present, are removed. The noise components are removed by using Morphological Filters.

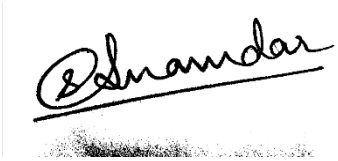

Fig 3. Image with Noise

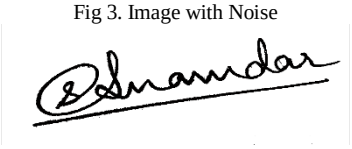

Fig 4. Image after removing noise

C. Rotation of Signature

The image is rotated w.r.t. the most bottom pixels of the signature image and the orientation line through them is fitted.

Fig 5. Image after rotation

D. Cropping of Signature

The image which is obtained after rotation is cropped. Cropping of Signature removes the unwanted background and thus allows the algorithm to concentrate only on the signature. It also helps to reduce the processing time.

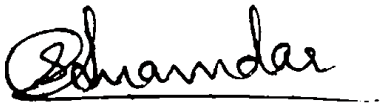

Fig 6. Image after cropping

IV. CONTOUR IMAGE EXTRACTION

A contour-based approach is used to obtain the critical points. A contour can be defined as the outer boundary of a signature. To obtain the contour of an image, the disconnected components of the signature and joined. All those pixels which have a four connected boundary pixel define a contour of the signature [4].

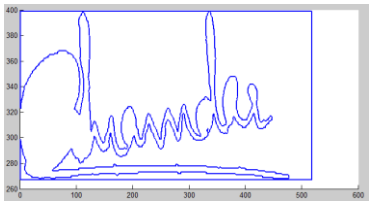

Fig 7. Contour Image

V. CRITICAL POINT EXTRACTION

Critical points can be defined as the set of those points which when modelled accurately, gives us the basic structure of the signature. Critical points are the minimum set of points required to define the shape of the signature. In order to obtain the critical points, the contour of the signature is traversed and any sharp change in the curve is marked as the critical point of the signature.

The critical points are marked as per the following algorithm [4]:

Algorithm: Critical Point Extraction
Input: Contour Image.
Output: Vector x and y containing a sequence of critical points.

for an in 1 to the size of the contour
// (x (a), y (a)) being the coordinates of contour increment count;
//x: vector x containing sequence of points x (a- count: count)
//y: vector y containing sequence of points y (a- count: count)

if count is greater than 5
[p s mu] <-- polyfit(x, y, 2);
//[p,s,mu] = Polyfit(x,y,n) finds the coefficients of a polynomial in x = (x-mu(1))/mu(2) where mu(1) = mean(X) and mu(2)=std(X). This centering and scaling transformation improves the numerical properties of both the polynomial and the fitting algorithm.
else
s.normr<- 0;

if abs (s.normr) is less than 10
check <- track_error_peak (s.normr (vector));

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

if check is equal to 1 then peak is encountered
x_crit (crit_pt) <- x (a); // x coordinate: critical point
y_crit (crit_pt) <- y (a); //y coordinate: critical point
increment crit_pt;
clear x y;
count ps;
continue;
else
x_crit(crit_pt) <- a1; // x coordinate :critical point
y_crit(crit_pt) <- a2; //y coordinate :critical point
increment crit_pt;
clear x y;
count ps;

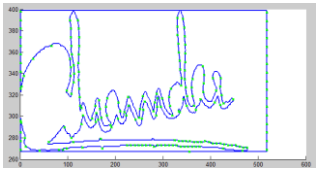

Fig 8. Critical Point Extraction

VI. CRITICAL POINT MAPPING

After Critical Point Extraction of the signatures, the next step is to find which critical point of Signature A (Questioned Signature) corresponds to which critical point of Signature B (Reference Signature).

The critical points are matched using the following algorithm [4]:

Algorithm: Critical Point Match
Input: cp_sigA[ ], cp_sigB[ ]
Output: M[ ]-a set of matched critical point coordinates

Begin
Initialize sigA_block[800][800] to 0
Initialize sigB_block[800][800] to 0
Initialize overlap[sA][sB] to 0
Initialize max[sA] to 0
Initialize stucture M(x_s1,y_s1,x_s2,y_s2)

for i = 0 to size of cp_sigA[], loop
All points in sigA_block[x coordinate of cp_sigA[i] to x coordinate of cp_sigA[i + 20]][y coordinate of cp_sigA[i] to y coordinate of cp_sigA[i + 20]] are made 1

for j= 0 to size of cp_sigB[], loop
All points in sigB_block[x coordinate of cp_sigB[j] to x coordinate of cp_sigB[j + 20]][y coordinate of cp_sigB[j] to y coordinate of cp_sigB[j + 20]] are made 1

overlap_value=sigA_block[ ][ ] AND sigB_block[ ][ ]
overlap[i][j]=overlap_value

Reinitialize sigB_block[ ][ ]
Next
Reinitialize sigA_block[ ][ ]
Next
Initialize Count to 0
for i = 0 to size of cp_sigA[ ],loop
max_val = maximum value in overlap[i][0 to j]
if max_val exceeds a predetermined threshold value for matching, then
j1 = index, where max_val was found
M(count) = (x of cp_sigA[i],y of cp_sigA[i],x of cp_sigB[j1],y of cp_sigB[j1])
Increment count by 1
End If
Next
Return structure M
End

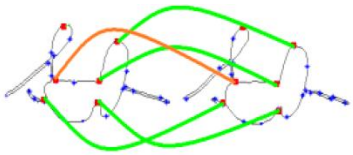

Fig 9. Critical Point Mapping [4]

VII. VERIFICATION

After determining the one-to-one corresponding matched critical points in the sample signatures, their respective critical regions are extracted. Instead of using the entire signature image, its critical portions are extracted and corresponding regions are compared to judge the overall similarity between the input and sample signatures.

Consider the Questioned Signature as Signature A and the Sample Signature as Signature B.

The algorithm is as follows [4]:

Algorithm: Critical Region Extract and Match
Input :- Contour Images sigA[ ][ ],sigB[ ][ ],
structure Match(x1,y1,x2,y2)
Output: - Optimal Distance Matrix[|M|]

Begin
Initialize cr1[31][31],cr2[31][31] to 0
Initialize set1[ ],se2[ ] to 0
Initialize dist_mat[ ][ ] to 0
Initialize optimal_dist[|M|] to 0

for i=0 to size of structure
Match(x1,y1,x2,y2),loop
X1,Y1 and X2,Y2 = ith entry of Match()
cr1[ ][ ]=values in sig1[X1 - 15,X1,X1+15][Y1-15,Y1,Y1+15]
set1[ ]=coordinates in cr1[ ][ ] which are black
cr2[ ][ ]=values in sig2[X2 - 15,X2,X2+15][Y2-15,Y2,Y2+15]
set2[ ]=coordinates in cr2[ ][ ] which are black
Reinitialize dist_mat[|set1|] [|set2|] = 0

for d = 0 to |set1| , loop
for g = 0 to |set2| , loop
dist = Euclidean distance between set1[d] and set2[g]
dist_mat[d][g] = dist
Next
Next min_dist = Hungarian_algorithm(dist_mat[ ][ ])
optimal_dist[i] = min_dist
Next
Return optimal_dist [ ]
End

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

VIII. RESULTS

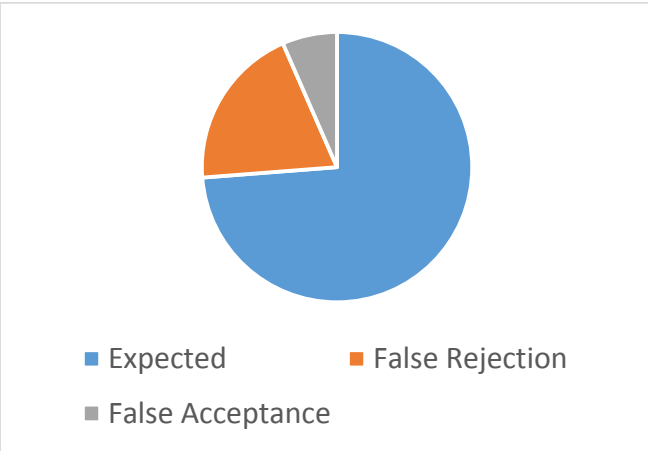

Fig 10. Results

Result	Percentage
Expected	75
False Rejection	20
False Acceptance	6.67
Total:	100

IX. GRAPHICAL USER INTERFACE

1. Start the application
2. Upload the Reference Signature by using the 'Browse Ref Sign' option.

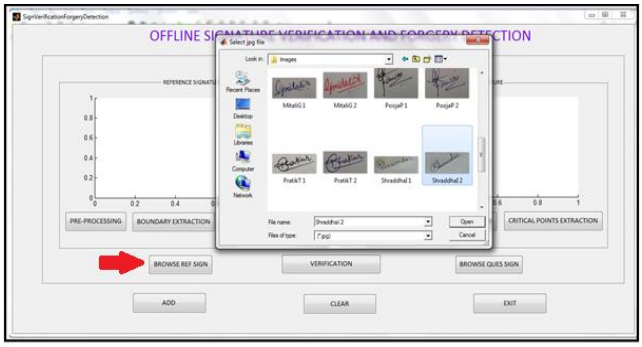

Fig 11: Browse for Reference Signature

3. Obtain the Questioned Signature image and add it to the database using the 'Add' option.

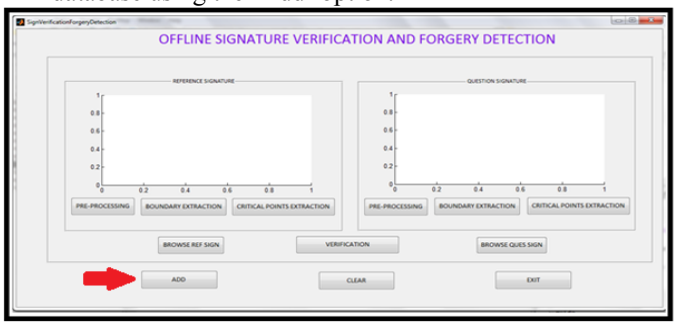

Fig 12. Adding Questioned Signature

4. Upload the Questioned Signature by using the 'Browse Ques Sign' option

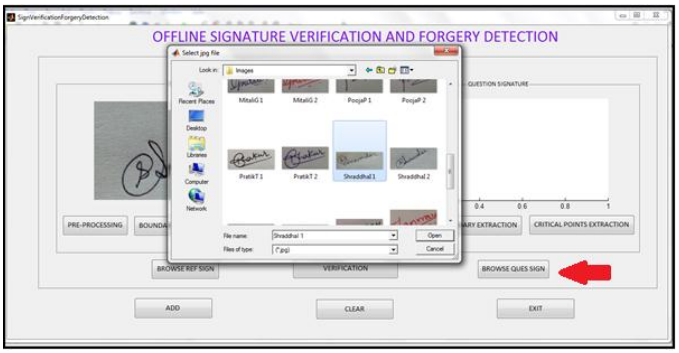

Fig 13. Browse for Questioned Signature

5. Click on Pre-Processing followed by Boundary Extraction and Critical Points Extraction for both the images.

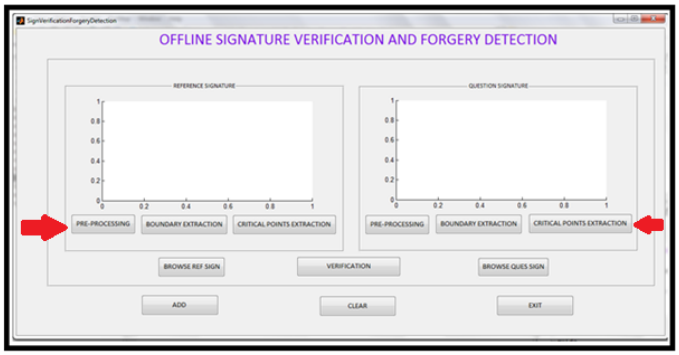

Fig 14. Pre-processing, Boundary Extraction and Critical Points Extraction

6. Click on Verification button for the Verification Process.

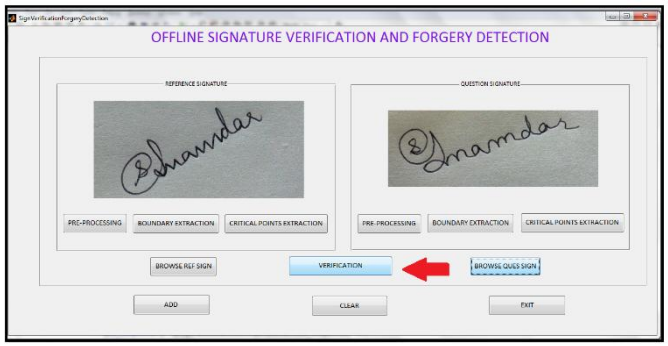

Fig 15. Verification of Signature

7. After the results are obtained, click on 'Clear' option to compare other signatures.

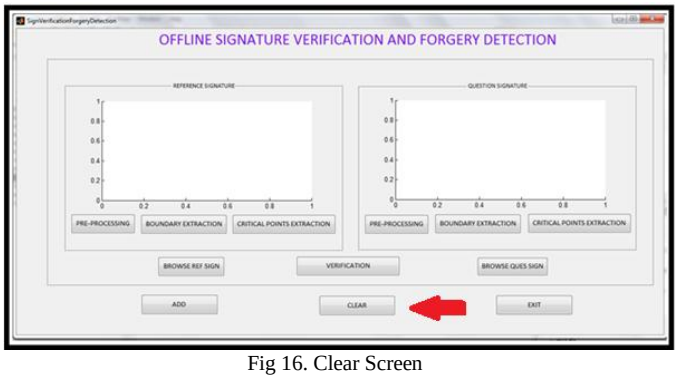

Fig 16. Clear Screen

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

8. After comparing the required signatures, click on 'Exit' option to exit from the system.

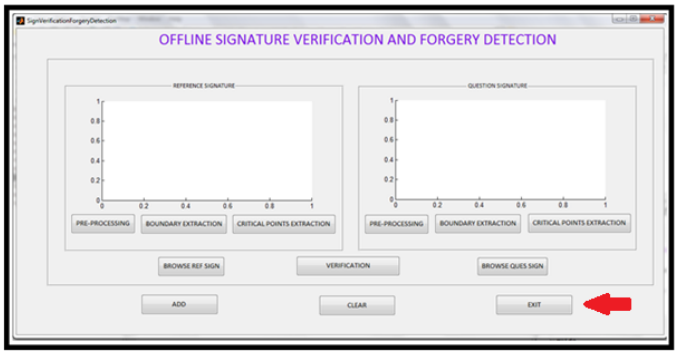

Fig 17. Exit from the System

X. CONCLUSION

In Critical Region Matching, we follow a graph matching approach, which compares these critical regions, taking care of the inevitable intra- personal variations. The results show significant improvement over other approaches for detecting skilled forgery.

The proposed system gives the 75% success rate by recognizing the all signature pattern correctly for all that signature which is used in training. Generally the failure to recognize/verify a signature was due to poor image quality and high similarity between 2 signatures.

The main disadvantage of this system is the non-repetitive personality of variation of the signatures, because of age, sickness, geographic location and some extent the emotional state of the person, accentuates the problem. So, the signature may vary but to resolve this problem threshold value is set and thus, the signature threshold below the threshold will be simply not accepted.

In future, this system can also be used for e-wallets and e-cheques. Also for capturing the signature image, instead of using a mobile camera or a digital camera, one can use image scanners, webcam or a portable scanner.

REFERENCES

[1] S.Maheswaran, "Fuzzy Logic Based Off-line Signature Verification and Forgery Detection System", IIT Kharagpur.

[2] Tomislav Fotak, Petra Koruga, Mirsolav Baca, "On-line Handwritten Signature Identification: The Basics", Faculty of Organization and Informatics, Centre for Biometrics, University of Zagreb.

[3] Yogesh V.G, Abhijit Patil, "Offline and Online Signature Verification Systems: A Survey", International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Special Issue 3, May 2014.

[4] Abhay Bansal,Prasad Nemmikanti, Pramod Kumar, "Offline Signature Verification Using Critical Region Matching" , 2008 Second International Conference on Future Generation Communication and Networking Symposia.

[5] Vamsi Krishna Madasu, Brian C. Lovell, "An Automatic Offline Signature Verification and Forgery Detection System", University of Queensland

[6] Madhuri Yadav, Alok Kumar, Tushar Patnaik, Bhupendra Kumar, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013