

International Journal of Control Theory and Applications

ISSN: 0974-5572

© International Science Press

Volume 10 • Number 11 • 2017

VLSI Implementation of SVM Based Offline Signature Verification Algorithm

Sangeeta¹ and Manpreet Kaur²

¹ M.Tech Student, ECE Department, CEC, Landran, Mohali ² Assistant Professor, ECE Department, CEC, Landran, Mohali

Abstract: This paper presents the VLSI architecture for the signature recognition in biometrics. In the field of Bankcheque processing, document authentication, ATM access etc. automated signature verification has many applications. Who is signing the document is a matter of concern today. In this paper a SVM based offline signature verification algorithm has been proposed. The local and structural parameters are utilized for the purpose. Every signature has a unique variation in the features. These form the basis for training the SVM classifier. The proposed method is extracting many such features which make the algorithm robust. The hardware results were simulated using ModelSim and realized on Spartan 6 FPGA.

Keywords: VLSI, FPGA Implementation, Signature Verification, Image Security, SVM.

INTRODUCTION

Biometrics refers to the automatic identification of a person based on his/her physiological or behavioral characteristics. Thus, biometrics can be defined as the science and technology of measuring and statistically analyzing biological data. Physiological characteristics are based on measurements of data derived from direct measurement of a part of the human body. Fingerprints, hand geometry, and retina, iris, and facial images are leading physiological biometrics. Behavioral characteristics are based on an action taken by a person. Behavioral biometrics, in turn, are based on measurements of data derived from an action, and indirectly measure characteristics of the human body. Signatures, voice recordings (which also has a physiological component), and keystroke rhythms are leading behavioral biometric technologies, the terms "Biometrics" and "Biometry" have been used since early in the 20th century to refer to the field of development of statistical and mathematical methods applicable to data analysis problems in the biological sciences. Recently, these terms have also been used to refer to the emerging field of information technology devoted to automated identification of individuals using biological traits especially for authentication purposes.

In general, the verification system falls into two broad categories according to the biometric technology which is applied. One is the online signature verification system and the other is the finger vein system. Even for the signature verification system, the system splits into separate systems depending on the classification method.

Figure 1 shows the three separated signature verification systems each one of them using different classification methods while Figure 1.3 shows the two biometric systems signature verification and finger vein.

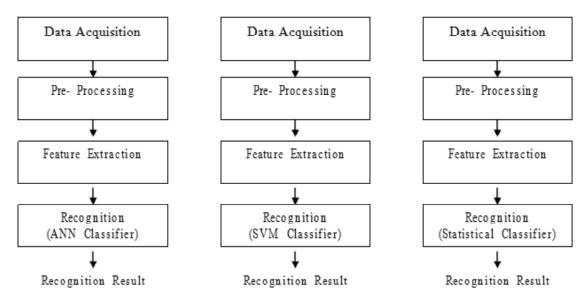


Figure 1: Signature verification systems using different classification methods

PROPOSED WORK

The proposed approach aims at developing automatic offline signature verification and forgery detection system. Fig. 1 shows the algorithm that is used in order to build the automated signature verification and forgery detection system. The proposed system has been divided into two parts namely:

- Training
- Testing

(A) Training Phase

In the training part of the system, the following steps are performed:

- 1) Image Database: The images are collected for training and are stored in a database. The images are collected by scanning them from a physical paper source. The database used is a self-created database which contains signatures of three different people. The database consists of fifteen signatures belonging to each person, and summing up to be forty-five signatures in total. More signatures can be added to the database easily and also the number of signatures per person can also be increased or decreased.
- 2) Pre-Processing: In this step, each of the scanned signature goes through a series of pre-processing steps which include the following[15]:
 - a) Image Resizing: The image is resized to a predefined size of 128 x 128 pixels.
 - b) Binarization: After resizing the image, the image is binarized, i.e. it is converted to black and white [14].
 - c) Thinning: After the process of binarization, the image goes through the process of thinning, i.e. the thickness of the strokes of the signature is thinned down to a single pixel. It is done is order to exclude the variations in thickness of signature which may occur due to the use of different types of pens.

International Journal of Control Theory and Applications

- d) Rotation: The image is then rotated on the basis of the lower most pixels. When a person signs a document, depending on the writing style of the person, there is a certain angle to the signature in which it is done. This process straightens out the signature.
- e) Cropping of the image: After the image is rotated, the excess area around the signature is removed and the image is cropped to the outer most pixels in four directions, i.e. top, bottom, left and right.
- 3) Feature Extraction: After the image has gone through the pre-processing, various features are extracted from the image. The extracted features out of each image are then stored in a MATLAB file. Following unique features are extracted from each the images:
 - a) Height-Width Ratio: After the image is cropped, height-width ratio of the signature is calculated.
 - b) Centroid of Signature: The centroid or the barycentre of the image is calculated. The centroid gives the central point of the signature which is a unique signature characteristic. The signature is broken down vertically into two halves, and the centroid of the each half is calculated.
 - c) HOG: We first divide the image into 9 blocks. For each block we find the 1st derivative of the image is found in the x and y direction. We find the orientation of the image. Further we find the histogram of the orientation in 9 bins. These histograms are stacked for all the 9 blocks.
 - d) Quadrant Areas: The image is broken down into four quadrants, and then the area of the signature pixels in each quadrant is calculated. This area is the area of strokes of the signature in that particular quadrant and does not include the area of the background.

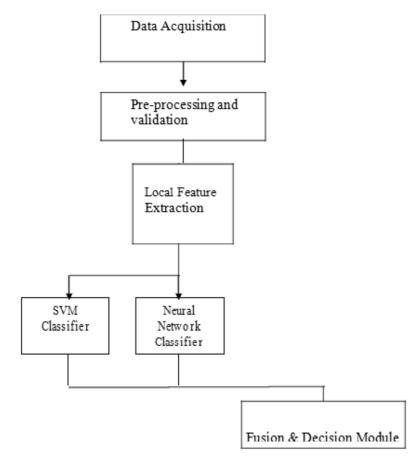


Figure 2: Block Diagram for Proposed Method

Sangeeta and Manpreet Kaur

- e) COM Matrix: COM Matrix or Co-Occurrence Matrix refers to the distribution of the co-occurring values at a given offset. It is used to measure the texture on the image. What is does is, as our image is in black and white after the pre-process, that means the image matrix has values either 0 or 1. It looks for pattern distribution of these values and looks where the patterns 00, 01, 11 and 10 occur. The cooccurrence matrix is also calculated for the signature.
- Edge Point Calculation: The number of edge points in the signature are calculated which gives a f) distinct characteristic about the signature.
- Horizontal and Vertical Histogram: Each row and each column of the signature is gone through and **g**) the number of black pixels is calculated. The row and the column with the maximum number of black pixels is recorded and used as a feature. All these features give out unique characteristics about the signature and are used for classification of the signatures.
- 4) Generate Training Feature Set and training svm: All the features calculated are stacked together to form the feature set. This is calculated for all the training images. All the feature sets are then used to train the SVM classifier. This gives us a hyperplane which can distinguish between the signatures.

RESULTS

The algorithm has been developed in Matlab on intel i3 with 4GB RAM. The algorithm has been tested for the training set as well as testing set. In the training set we get 100% accuracy and in the testing set we get some statistics which is shown in table 1.

	Table 1 Comparison Table for Neural Networks and SVM						
	Neural Network	SVM with modified features					
tp	50	50					
fn	7	2					
fp	7	2					
recall	87.7	96.15					
precision	87.7	96.15					

Table 1

Some results are shown in the figures below. The signatures have lot of variance in the images. Figure 3 and 4 can be compared and it can be seen clearly that the signatures have a lot of variance. Similar analysis can be done in figure 5 and 6.



Figure 3: Signature 1

VLSI Implementation of SVM Based Offline Signature Verification Algorithm

2	3 -	for	i=1:	2								
Z	4 -		p_no	=p_nol(1)	;							
2	5 —	¢.		ii=6:6								_
2	6 —			📣 Figure 4	ļ					-	- 🗆	×
2	7	ql		51 F.D.	16 1 1	. .		147 1				~
2	в —				View Insert							
2	9		_	1 C 🖬	🗟 🗛 🔍	Q 🕐	9 🖓	1 - 2				
<												
Co		nd Windo										
		n main	_	7								
		ning:				1						
		n feat	-									
		n main	_	7	1	/	~	<i>,</i> ,				
	War	ning:	Integ		1 ~	j Cm	ć i		80		1	
	> I	n <u>feat</u>	ure e	4	1 million		~ (-n	00	へしへ	~~~	
	I	n <u>main</u>	at 3		}							
					1	~						
	res	ult =										
		1										

It is no. 1s sign ft >>

Figure 4: Signature 2

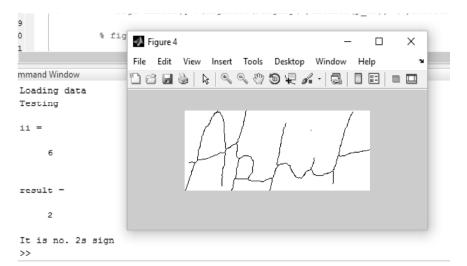


Figure 5: Signature 1

28 - sign=imread (I' \signdata\Eorgerv\'int2str(n no) '\'int2str(n no), 'i 29 Figure 4 - 30 % fi File File Edit View Insert Tools Desktop Window Help -
30 % fil File Edit View Insert Tools Desktop Window Help ~ 31 □
31 File Edit View Insert Tools Desktop Window Help ~ C C C C C C C C
≤ □러┠७ ҟ ९९७७ ₽ ⊀ ⋅ ₿ □ ⊟ = □
Command Window
A AN ACCOUNT OF A A A A A A A A A A A A A A A A A A
In main at 38
Warning: Integer o
> In feature_extra
In main at 38
Warning: Integer o
> In feature_extra
In main at 38
result =
2
2
It is no. 2s sign
ft 15 h5. 25 51gh

Figure 6: Signature 2

Sangeeta and Manpreet Kaur

Hardware Realizations

For the hardware realizations the code was developed in Verilog HDL and realized over Spartan 6 FPGA. The HDL code was simulated using ModelSim. The results are demonstrated below.

Preprocessing and Feature Extraction

	В	С	D	E	F	G	н	1	J	К	L	M	N
evice			On-Chip	Power (W)	Used	Available	Utilization (%)		Supply	Summary	Total	Dynamic	Quiescent
mily	Spartan6		1Os	0.000	9	296	3		Source	Voltage	Current (A)	Current (A)	Current (A)
rt	xc6sbx45t		Leakage	0.036					Vccint	1.200	0.015	0.000	0.015
ckage	fgg484		Total	0.036				- [Vecaux	2.500	0.005	0.000	0.005
mp Grade	C-Grade	\sim							Vcco25	2.500	0.002	0.000	0.002
ocess	Typical	~				Max Ambient	Junction Temp						
eed Grade	-3		Thermal	Properties	(C/W)	(C)	(C)				Total	Dynamic	Quiescent
					19.1	84.3	25.7		Supply	Power (W)	0.036	0.000	0.036
vironment	-												
bient Temp (C		_											
custom TJA		\sim											
stom TJA (C/V		_											
luw (LFM)	0	~											
at Sink	None	~											
tom TSA (C/V	() INA												

Figure 7: Power Report for Preprocessing and Feature Extraction

Device Utilization

Device Utilization Summary (estimated values)							
Logic Utilization	Used	Available	Utilization				
Number of Slice Registers	75	54576	0%				
Number of Slice LUTs	133	27288	0%				
Number of fully used LUT-FF pairs	67	141	47%				
Number of bonded IOBs	10	296	3%				
Number of Block RAM/FIFO	36	115	31%				
Number of BUFG/BUFGCTRLs	1	16	6%				

Figure 8: Device Utilization Results for PreProcessing and Feature Extraction

Testing by SVM classifier

Device Utilization

Device Utilization Summary (estimated values)							
Logic Utilization	Used	Available	Utilization				
Number of Slice Registers	133	54576	0%				
Number of Slice LUTs	10664	27288	39%				
Number of fully used LUT-FF pairs	71	10726	0%				
Number of bonded IOBs	3	296	1%				
Number of Block RAM/FIFO	4	116	3%				
Number of BUFG/BUFGCTRL/BUFHCEs	1	16	6%				

Figure 9: Device Utilization Results for Testing by SVM

International Jo	ournal of Contro	Theory and	Applications

VLSI Implementation of SVM Based Offline Signature Verification Algorithm

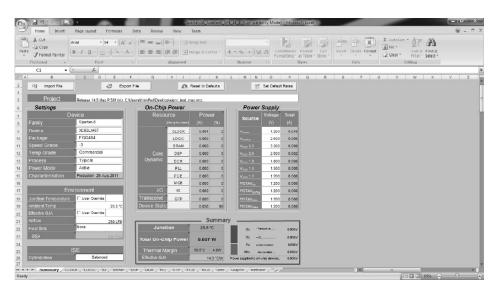


Figure 10: Power Report for Testing by SVM

SIMULATION RESULTS

Simulations were carried out for 4 different feature sets, two for each signature.

166 features for one signature. SVM classify is used for signature recognition.

Na	me	Value	990 us 995 us	1,000
	l <mark>e</mark> dk	1		
	orm_feature[0:165,51:0]	[00000000000000000000000000000000000000	[00000000000000000000000000000000000000	00011000110111100
	supportvector[0:165,0:6,31:0]	[[00000000000000000]]	[[0000000000000000000000000000000000000	000000000000000000000000000000000000000
V.	svm_dot[0:6,31:0]	[0100000111000111010	[0 100000 1 1 1000 1 1 10 1000 100 10000 100,0 100000 1 100 1 10 100 1 1 1 1	00010110011001110
	[0,31:0]	01000001110001110100	0 100000 11 1000 11 10 1000 100 1000 100	
	[1,31:0]	01000001100110100111	010000011001001101001101101	
	[2,31.0]	01000001001001000101	0 100000 100 100 100 100 100 100 1100 1110 1	
	[3,31:0]	01000001001110110000	0 100000 100 1 1 10 1 1000000 1 1 10 1 0 1 10 1	
	[4,31:0]	1100000011011111100	11000000011011111100110100011010	
	[5,31:0]	11000000010110001001	1 10000000 10 1 1000 100 100 11 1 100 110	
	[6,31:0]	11000010111000010001	1 10000 10 1 1 10000 1000 1 10 10 10 10000 1	
V.	alpha[0:6,31:0]	[1011110010111010100	[10111100101110101000000110101111,10111100001001	11101101100011100
	[0,31:0]	10111100101110101000	10 1 1 1 100 10 1 1 10 10 1000000 1 10 10	
	[1,31:0]	10111100001001001101	10 1 1 1 10000 100 100 1 10 1 10 1000 100 10 1	
	[2,31:0]	10111100100011111011	1011110010001111101101001111000	
	[3,31:0]	10111011010011110010	10 1 1 10 1 10 100 1 1 1 100 10 10 10 10	
	[4,31:0]	00111101000000101011	00111101000000101011100001111000	
	[5,31:0]	00111100101001000011	001011000101000001110000001000000000000	
	[6,31:0]	00111010110011100000	001110101100000001001001111	
	svm_sum[31:0]	101111111010101111110	1011111110101011111100010111111	
	bias[31:0]	10111100100001011101	10 1 1 1 100 10000 10 1 1 10 1 1 100 10 1	
	svm_final[31:0]	10111111101011011111	101111111010110111111010p1100010	
	index[1:0]	01	01	

Figure 11: Simulation Resuts for Signaure 1

Name	Value	0 us	200 us	400 us	600 us	800 us
l <mark>en</mark> cik	0					
Inorm_feature[0:165,31:0]	[00000000000000000000000000000000000000	[00000000000000000000000000000000000000	00000000000000,00111	1010110101001111100	0000100,101111110101	1 100 10 100 1 100 10 10
supportvector[0:165,0:6,31:0]	000000000000000000000000000000000000000	000000000000000000000000000000000000000	0000000000000000,0000	000000000000000000000000000000000000000	00000000,0000000000	
🔻 📷 svm_dot[0:6,31:0]	[0100000100000111110:	[0 100000 100000 11	1101001001011100,110	0000110000111010111	001010111,1100000110	0100111010101010011
▶ ■ [0,31:0]	01000001000001111101	8<	0 1000	0100000111110100100	1011100	
[1,31:0]	11000001100001110101:	8(11000	0110000111010111000	1010111	
[2,31:0]	11000001100100111010:	0X	11000	0110010011101010100	1110000	
[3,31:0]	01000001001010111100	¢X.	0 1000	00100101011110011100	0111000	
[4,31:0]	010000100000000000000	ÓX.	0 1000	0 1000000000000000 10 10	0011100	
[5,31:0]	01000001100001001111	OK	0 1000	0011000010011110100	10011110	
[6,31:0]	11000010000100111001	0X	11000	0100001001110011001	11100111	
🔻 🚮 alpha[0:6,31:0]	[1011110010111010100	[101111001011101010	00000110101111,10111	0000 100 100 1 10 1 10 100	0100101,101111001000	11110110110001110.
[0,31:0]	10111100101110101000	(101111	0 10 1 1 10 10 10 00000 1 10	0101111	
[1,31:0]	10111100001001001101:	(101111	0000 100 100 1 10 1 10 100	100101	
[2,31.0]	10111100100011111011	(101111	0010001111101101100	111000	
[3,31:0]	10111011010011110010:		101110	1 10 100 1 1 1 100 10 10 10 10 1	000101	
[4,31:0]	00111101000000101011:	(001111	0 1000000 10 10 1 110000	1111000	
[5,31:0]	00111100101001000011:	(001111	0 10 100 10000 1 1 100 10	110100	
[6,31:0]	00111010110011100000		001110	1011001110000000100	1001111	
svm_sum[31:0]	0011111111001000001	00	00111	1111100100000011110	10001111	
▶ 📑 bias[31:0]	10111100100001011101:	C	101111	0 10000 10 1 1 10 1 1 100 1	110011	
► 📑 3vm_finel[31:0]	0011111110001100000	0C	00111	1111100011000000111	0011100	
index[1:0]	10	CX		10		

Figure 12: Simulation Results for Signature 2

Sangeeta and Manpreet Kaur

Name	Value		200 us	400 us	600 us	800 us
Ling cik	0					
norm_feature[0:165,31:0]	[00000000000000000000000000000000000000	([0000000000000000000000000000000000000	0000000000000,001111	10111010101000110001	11110,0100000001111	011.11100001000010
supportvector[0:165,0:6,31:0]	[[00000000000000000]]	<pre>%[[00000000000000000000000000000000000</pre>	000000000000000,0000	000000000000000000000000000000000000000	0000000,0000000000000000000000000000000	00000000000000000
▼ 🚮 svm_dot[0:6,31:0]	[11000010000100100	[1100001000010010	0000000110101000,1100	00010011100110011101	10111110,010000011000	001000011101001000
[0,31:0]	110000100001001000	*	11000	0100001001000000011	101000	
[1,31:0]	110000010011100110	X	11000	00100111001100111011	0111110	
[2,31:0]	010000011000010000	ХХ_	0 1000	00110000100001110100	1000011	
[3,31:0]	110000011110101000	Х	11000	00111101010000001010	0010010	
[4,31:0]	110000100011010000	_X	11000	0 1000 1 10 1000000 10 1 1 1	0100001	
[5,31:0]	110000011001011011	_X	11000	00 1 100 10 1 10 1 10 10 10 100 1	1111100	
[6,31:0]	010000110111111010	_X	01000	01101111110100111111	0001111	
🛡 🚮 alpha[0:6,31:0]	[10111100101110101	[1011110010111010100	000110101111,1011110	000 100 100 1 10 1 10 1000 1	00101,101111001000111	11011011000111000
[0,31:0]	101111001011101010		1011110	0101110101000001101	01111	
[1,31.0]	101111000010010011		1011110	000 100 100 110 1 10 1000 1	00101	
[2,31:0]	101111001000111110		1011110	01000111110110110001	11000	
[3,31:0]	101110110100111100		101110	1010011110010101010	00101	
[4,31:0]	001111010000001010		0011110	1000000101011100001	11000	
[5,31:0]	001111001010010000		0011110	0 10 100 10000 1 1 100 100 1	10100	
[6,31:0]	001110101100111000		001110	01100111000000010010	01111	
▶ 🚮 svm_sum[31:0]	101111110010100111		10111	111001010011111110111	100 1000	
▶ 🖬 bias[31:0]	101111001000010111		1011110	0100001011101110010	10011	
svm_final[31:0]	101111110010111000		1011:	11100101110001010101	0101110	
▶ 📷 index[1:0]	01			01		

Figure 13: Simulation Result for Signature 1

N	lame		Value	200 us	400 us	600 us	800 us	
	16	clk	0					
	Ő	norm_feature[0:165,31:0]	[00000000000000000000000000000000000000	[00000000000000000000000000000000000000	00000000,001111010	11010100111110000001	00,101111110	.0101)
۲	0	supportvector[0:165,0:6,31:0]	000000000000000000000000000000000000000	[[0000000000000000000000000000000000000	000000000,00000000	000000000000000000000000000000000000000	000,00000000	0000)
Y	6	svm_dot[0:6,31:0]	[0100000100000111110	[0 100000 100000 11 11 10 100 1	0 10 1 1 100, 1 100000 1 10	00011101011100010101	11,110000011	0100)
		[0,31:0]	01000001000001111101		0100000100000111110	1001001011100		
		[1,31:0]	11000001100001110101		1100000110000111010	1110001010111		
		6 [2,31:0]	11000001100100111010		110000011001001110	0101001110000		$ \rightarrow $
		[3,31:0]	01000001001010111100		0100000100101011110	0111000111000		
		[4,31:0]	01000010000000000000		010000100000000000000000000000000000000	0010100011100		
		[5,31:0]	01000001100001001111		010000011000010011	1010010011110		
		[6, 31 :0]	11000010000100111001		1100001000010011100	1100111100111		$ \rightarrow $
Ŧ	6	alpha(0:6,31:0)	[1011110010111010100	[1011110010111010100000	110101111,1011110000	100 100 1 10 1 10 1000 100 1	01,101111001	0011)
	٠	6 [0,31:0]	10111100101110101000		1011110010111010100	0000110101111		
		[1,31:0]	10111100001001001101		1011110000100100110	1 10 1000 100 10 1		
		o [2,31:0]	10111100100011111011		101111001000111110	1011000111000		
		[3,31:0]	10111011010011110010		101110110100111100	0 10 10 10000 10 1		
		[4,31:0]	00111101000000101011		001111010000001010	1100001111000		
		[5,31:0]	00111100101001000011		0011110010100000	1100100110100		
		6,31:0]	00111010110011100000		0011101011001110000	0001001001111		
	0	svm_sum[31:0]	0011111111001000001		001111111100100000	1111010001111		
	ø	bias[31:0]	10111100100001011101		1011110010000101110	1110010110011		
	6	svm_final(31:0)	00111111110001100000		0011111111000110000	0011100011100		$ \rightarrow $
	0	index[1:0]	10		10			

Figure 14: Simulation Results for Signature 2

CONCLUSION

The paper presented a VLSI implementation for verification of offline signature. The method provides robust results. The system having this algorithm can be used in ATM's, document authentications and banks. The emotion and situation of the person doing signature may affect his/her signature. This can be improved in the future.

REFERENCES

- [1] Er. Sangeeta chopra and Er. Manpreet kaur, "Review on Signature recognition using Neural Network, SVM, Classifier combination of HOG and LBP features", Chandigarh engineering College, Landran, Mohali.
- [2] Md. Iqbal Quraishi, Arindam Das and Saikat Roy (2013), "A Novel Signature Verification and Authentication System Using Image Transformation and Artificial Neural Network", Narula Institute of Technology, Kolkata.

International Journal of Control Theory and Applications

- [3] Othman o-khalifa, Md. Khorshed Alam and Aisha Hassan Abdalla (2013), "An Evaluation on Offline Signature Verification using Artificial Neural Network Approach", International Conference on Computing, Electrical and Electronic Engineering (ICCEEE).
- [4] Rameez Wajid and Atif Bin Mansoor, "Classifier Performance Evaluation For Offline Signature Verification Using Local Binary Patterns", Institute of Avionics & Aeronautics, Air University, Islamabad, Pakistan.
- [5] Muhammad Imran Malik, Marcus Liwicki and Andreas Dengel, "Evaluation of Local and Global Features for Offline Signature Verification", German Research Center for AI (DFKI GmbH).
- [6] Juan Hu and Youbin Chen (2013), "Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudodynamic Features", 12th International Conference on Document Analysis & Recognition.
- [7] Vaibhav Shah, Umang Sanghavi, Udit Shah, "Off-line Signature Verification Using Curve Fitting Algorithm with Neural Networks", Dwarkadas J. Sanghvi College of Engineering, Mumbai.
- [8] M.Nasiri, S.Bayati and F.Safi, "A Fuzzy Approach for the Automatic Off-line Signature Verification Problem Base on Geometric Features", Azad University, Iran.
- [9] Surabhi Garhawal and Neeraj Shukla (2013), "A Study on Handwritten Signature Verification Approaches", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, Issue 8, August 2013.
- [10] L B. Mahanta, Alpana Deka (2013), "A Study on Handwritten Signature", International Journal for Computer Applications (0975- 8887), Volume 79 - No. 2, October 2013.
- [11] Pradeep Kumar, Shekhar Singh, Ashwani Garg and Nishant Prabhat (2013), "Hand Written Signature Recognition & Verification using Neural Network", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013
- [12] Ishita Sharma, Sakshi Goyal and Shanu Sharma, "Sign Language Recognition System for Deaf and Dumb People", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol 2, Issue 4, April- 2013, pp. 382-387.
- [13] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey", IEEE Tran. on Pattern Analysis and Machine Intelligence, vol.22 no.1, pp.63-84, Jan.2000.
- [14] M. Blumenstein. S. Armand. and Muthukkumarasamy, "Off-line Signature Verification using the Enhanced Modified Direction Feature and Neural based Classification," International Joint Conference on Neural Networks, 2006.
- [15] Lal Chandra, Puja Lal, Raju Gupta, Arun Tayal, Dinesh Ganotra: Improved adaptive binarization technique for document image analysis. VISAPP (1) 2007: 317-321.
- [15] Ved Prakash Agnihotri, "Offline Handwritten Devanagari Script Recognition", I.J. Information Technology and Computer Science, 2012, 8, 37-42.