International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Performance Comparison of Devanagari and Gurmukhi Handwritten Numerals Recognition

Gita Sinha[1], Rakesh Kumar Roshan[2]

[1]Department of CSE, Asst. Professor, L.N.M.U. women's Institute of technology darbhanga

[2]Department of CSE, Asst. Professor, Government Engineering College, Dumka, India

Abstract: ***In this paper we proposed two different methods for Numeral Recognition are and compared their performance. The objective of this paper is to provide an efficient and reliable method for recognition of handwritten numerals. First method employs Image Centroid Zone feature extraction and recognition algorithm. In this method the features of the image are extracted and these feature set is then compared with the feature set of database image for classification. While second method contains Zone Centroid Zone algorithms for feature extraction and the features are applied to support vector machine [SVM] for recognition of input image. Handwritten Optical Numeral recognition[HONR] is important research area because of its wide applications in many areas like Cheque Reading in Bank, postcode reading ,form processing, Post office, Hospitals , signature verification etc.***

Keywords: Handwritten Numeral Recognition, Grid Technique, ANN, Feature Extraction, Classification.

1. Introduction

OCR - optical character recognition is the branch of computer science that involves reading text from paper and translating the images into a form that the computer can manipulate (for example, into ASCII codes). An OCR system enables you to take a book or a magazine article, feed it directly into an electronic computer file, and then edit the file using a word processor.

All OCR systems include an optical scanner for reading text, and sophisticated software for analyzing images. Most OCR systems use a combination of hardware (specialized circuit boards) and software to recognize characters, although some inexpensive systems do it entirely through software. Advanced OCR systems can read text in large variety of fonts, but they still have difficulty with handwritten text.

2. Litreture Survey

M. Hanmandlu, J. Grover, V. K. Madasu, S. Vasikarla. [1] Proposed a system to recognize Handwritten Hindi Numerals based on the modified exponential membership function. They have used 3500 dataset. The overall recognition rate is found to be 95%.

Omid Rashnodi, Hedieh Sajedi, Mohammad Saniee [2]. They proposed box approach method to achieve higher recognition accuracy and decreasing the recognition time of Persian numerals. In classification phase, support vector machine (SVM) with linear kernel has been employed as the classifier. 98.945% correct recognition rate was obtained.

Kartar Singh, Siddharth, Renu Dhir, Rajneesh Rani [3]. They have used projection histograms, zonal density and Background Directional Distribution (BDD) features. The SVM classifier with RBF (Radial Basis Function) kernel is used for classification. They have obtained 99.2%, 99.13% and 98% accuracy.

S.L.Mhetre, Prof.M.M.Patil [4] they have used Image Centroid Zone and Zone Centroid Zone algorithms for feature extraction and the features are applied to Artificial Neural Network for recognition of input image. And they have applied Grid techniques and ANN techniques on 500 data then obtained 83.6% and 86.4% result accuracy.

Reena bajaj et al. [5] This paper is concerned with recognition of handwritten Devnagari numerals . They have used Density features, Moment features of right, left, upper and lower profile curves, Descriptive component features and he got 89.68% highest accuracy among all these three methods.

U. Bhattacharya et al. [6] they have used two classification methods HMM and ANN on Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition. They have got 91.28% of highest accuracy.

3. Data Collection

The Gurmukhi alphabet developed from the Landa alphabet and was standardised during the 16th century by Guru Angad Dev Ji the second Sikh guru. The name Gurmukhi means "from the mouth of the Guru" and comes from the Old Punjabi word *Gurmukhi*. Punjabi is an Indo-Aryan language spoken by about 105 million people mainly in West Punjab in Pakistan and in East Punjab in India. Punjabi descended from the Shauraseni language of medieval northern India and became a distinct language during the 11th century. In India Punjabi is written with the Gurmukhi alphabet, while in Pakistan it is written with a version of the Urdu alphabet known as Shahmukhi. The written standard for Punjabi in both India and Pakistan is known as Majhi, which is named after the Majha region of Punjab. Punjabi is one of India's 22 official languages and it is the first official language in East Punjab. In Pakistan Punjabi is the second most widely-spoken language but has no official status. In figure 1.1 the Gurmukhi script alphabet is shown in fig-1.1

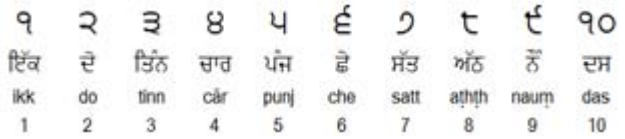


Figure 1.1: Gurmukhi Numeral

Devnagari descended from the Brahmi script sometime around the 11th century AD. Its original form was developed to write Sanskrit but was later adapted to write many other languages such as Hindi, Marathi and Nepali. The ideal (printed) Devnagari numerals are shown in figure 1.2. From this figure it is shows that there are variations in the shapes of numerals ५, ८ and ९ in their printed forms. However, from the samples in figure 1.2 it can be observed that there exist wide variations in the handwritten forms of Devnagari numerals.

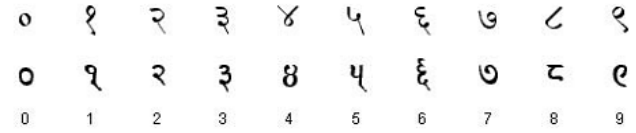


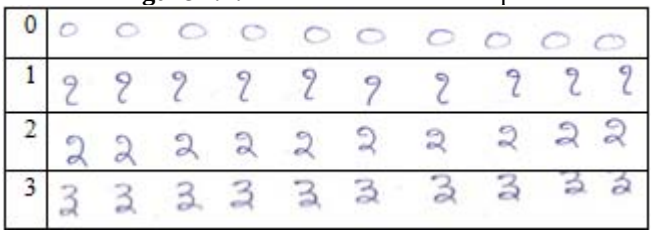
Figure 1.2: Devanagari Numerals shapes

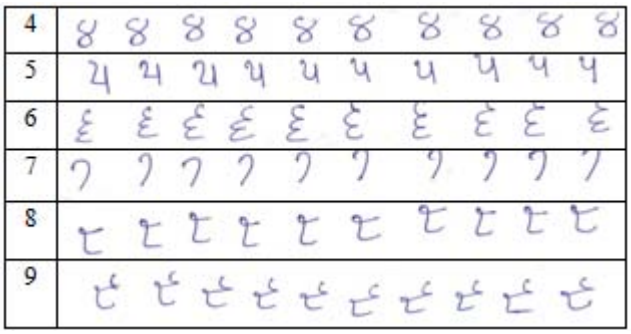
Our database of isolated handwritten Devnagari numerals consists of 22546 samples from 1049 persons. The whole set of available data have been kept into a training set.

Figure 1.3: Handwritten Devanagari Numerals samples

For recognition of Gurumukhi numerals data set of 1500 samples have been collected .Gurmukhi numeral has been collected from 15 different writers of different age, professions and qualification. Gurmukhi numeral samples Shown in fig.-1.4

Figure 1.4: Gurmukhi numeral samples





4. Pre-Processing

Numerical recognition is nothing but the conversion of handwritten or typed numerical imaget into machine understandable form. For this the document is first scanned using a regular scanner. Before performing the feature extraction directly, first the scanned data/image is passed through some pre-processing steps (shown in the flow chart below), these steps are as follows:

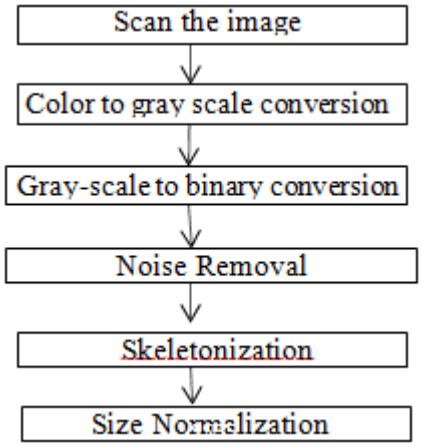


Figure 1.5: pre-processing steps

If the scanned image is a color image it is converted into a gray scale image.

- The gray scale image is still a raw image; it may carry some unwanted information. This information is termed as noise. The noise/distortion may introduce while scanning the image. The noise is of various types like salt and pepper noise, shot noise etc. Median filter is used to remove such noise.
- After filtering the image it is converted into binary form. The process is also called as Image Segmentation.
  Image segmentation is performed by assigning a variable to a threshold value, if the value of pixel in gray scale image is equal or above the threshold value then the pixel is replaced by 1 else 0. At the end the image is inverted for easy processing.
- The image obtained after binarization is sent for thinning / skeletonization process. Skeletonization reduces the width of numerals from many pixels to one.
- Finally the image is restored to a standard dimension 32*32. (It is not necessary to use/take these dimensions only. Our aim is to create a database with images having equal fixed size. Any dimensions can be taken to achieve feasibility in operation.)

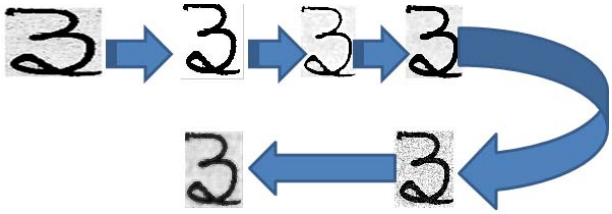


Figure 1.6 (A)RGB image (B) binary image (C) dilation of image (D) perform erosion on image (E) add the noise into image (F) removing noise from image

5. Feature Extraction

This is the most important step in character/numeral recognition, since classification is done on the basis of the set of features extracted during this step. Feature extraction simply means the acquisition or measurement of those parameters of input image that are most useful for classification purpose. Many of such features are invented and used by scientists for pattern classification.

Following listed features have been used for current experiment. Two types of features namely image centroid zone, and zone centroid zone. 200 feature vectors have been formed using combinations of both basic features. These methods provide the ease of implementation and good quality recognition. Step-by-step algorithm has been defined in the next section. In the next section, these algorithms have been defined. The following paragraph explains the details about feature extraction method.

5.1 Image Centroid Zone

The centroid of image (numeral/character) has been computed. The given image has been further divided into 100×100 equal zones where size of each zone is (10×10). Then, the average distance from image centroid to each pixel present in the zones/block has been computed. 100 feature vectors of each image are thus obtained. Zones which are empty are assumed to be zero. This procedure is repeated for all zones present in image (numeral/character).

Figure 1.7 shows example of character image of size 32×32. First, centroid of image is computed. Then, image is divided into 16 equal zones each of size 8×8. Later, average distance from image centroid to each pixel present in the image is computed.

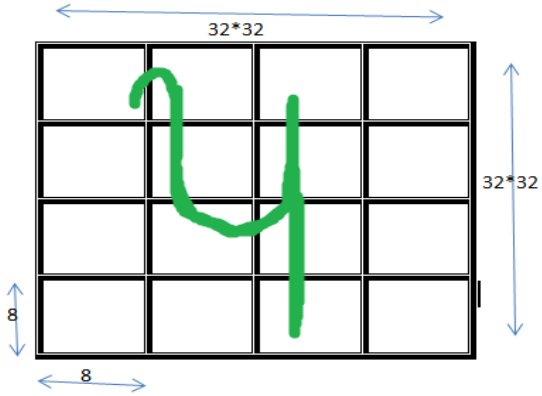


Figure 1.7 (ICZ) Image 32×32 and block 8×8.of handwritten Gurumukhi Numeral "**five**"

4.2. Zone Centroid Zone

In ZCZ, image is divided into 100×100 equal zones and centroid of each zone is calculated. Followed by computation of average distance of zone centroid to each pixel present in zone. Zones which are empty are assumed to be zero. This procedure is repeated for all pixels present in each zone.

Efficient zone based feature extraction algorithm has been used for handwritten numeral recognition of four popular south Indian scripts as defined in [7]. Here, same method has been applied on few north Indian scripts. Algorithm 1 provides Image centroid zone (ICZ) based distance metric feature extraction system, while Algorithm 2 provides Zone Centroid Zone (ZCZ) based Distance metric feature extraction system. Further, Algorithm 3 provides the combination of both (ICZ+ZCZ) feature extraction systems. The following algorithms illustrate the working procedure of feature extraction methods as depicted in figure 3.3.

Figure 1.8 shows example of character image for size 32×32. In this figure image has been divided into 16 equal zones, each of size 8×8. Centroid of each zone in image has been computed. Then, average distance from image centroid to each pixel present in the zone is calculated.

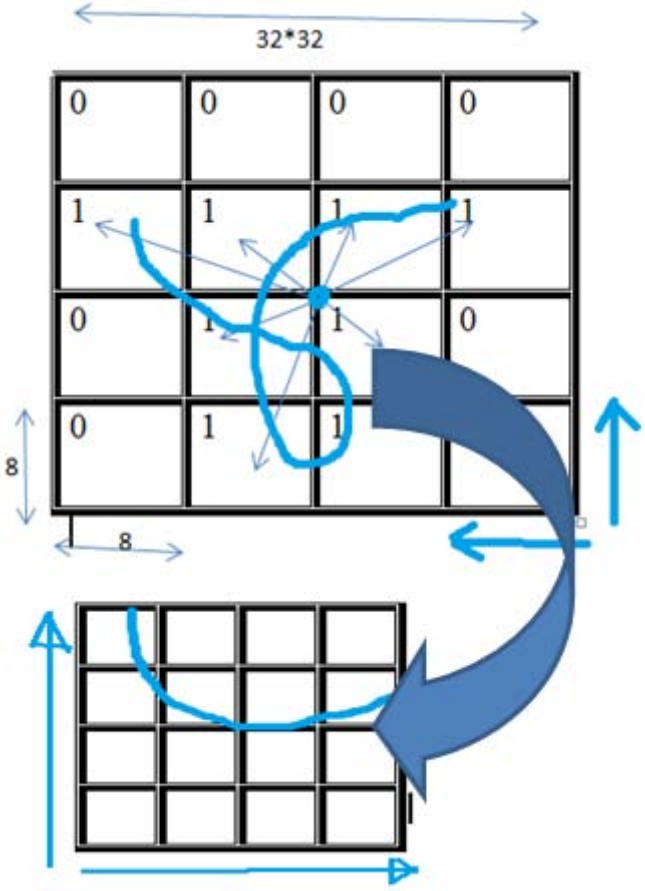


Figure 1.8: (ZCZ) Image 32×32 and block 8×8.of handwritten Devnagari Numeral "***four***"

Algorithm 1: Image Centroid Zone (ICZ) feature extraction method.
Input : Pre-processed Image (character/numeral)

Output: Extract the Features for Classification and Recognition

Method Begins
Step 1: Calculate centroid of input image.
Step 2: Division of input image in to 100 ×100 equal zones.
Step 3: Computation of the distance from the image Centroid to each pixel present in the zone.
Step 4: Repeats step 3 for the entire pixel present in the zone/boxes/grid.
Step 5: Average distance computed between these Points.
Step 6: Repeat this procedure sequentially for the entire zone present in the image.
Step 7: Obtaining 100 such feature for Classification and recognition process.
Ends.

Algorithm 2: Zone Centroid and Zone (ZCZ) based feature extraction system.

Method Begins
Step 1: Division of input image in to **n** equal zones.
Step 2: Compute centroid of each zones.
Step 3: Compute the distance between the zone centroid to each pixel present in the zones.
Step 4: Repeat step 3 for the entire pixel present in the zone/box/grid.
Step 5: Computation of average distance between these points present in image.
Step 6: This procedure are sequentially repeat for the entire zone.
Step 7: Obtaining, 100 such features for classification and recognition.
Ends.
Hybrid Algorithm 3: Hybrid feature extraction method is a combination of both of the algorithm **(**ICZ + ZCZ) defined above. This method provides 200 such features from each of the image.

6. Classification

Two types of classifiers have been used in this implementation namely SVM, and K-NN. Among these SVM is popular & efficient and also produce most efficient results in this implementation. In Gurmukhi character recognition, these two type of the classifiers have been used.

5.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) has been used for the purpose of Classification and Recognition. Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A Support Vector Machine (SVM) is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. It have capability of learning to achieve good generalization performance, which is objective of any machine, given a finite amount of training data. Striking a balance between goodness of fit obtained on a given training dataset and the ability of machine to achieve error free recognition on all the dataset. The standard SVM takes a set of input data and predicts, two possible classes of input. SVM training algorithm builds a model that assigns new examples into one category or the other. SVM utilized in pattern recognition is to construct a hyper-plane as the decision plane, which separates the positive and negative patterns with the largest margin.

SVM have proved to achieve good generalization performance by the use of concept of basis, without knowledge of the prior data [8].

5.2 K-Nearest Neighbour (KNN)

K Nearest Neighbour (KNN) is one of those algorithms that are very simple to understand but works incredibly well in practice. Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on. Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple.

The k-nearest 4eighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. K-NN is instance-based classifiers which operate on the premises that classify unknown instances by relating the unknown to the known according to some distance/similarity function. The intuition is that two instances far apart in the instance space defined by the appropriate distance function are less likely than two closely situated instances to belong to the same class.

1) The learning process
Unlike many artificial learners, instance-based learners do not abstract any information from the training data during the learning phase. Learning is merely a question of encapsulating the training data. The process of generalization is postponed until it is absolutely unavoidable, that is, at the time of classification. This property has lead to the referring to instance-based learners as lazy learners, whereas classifiers such as feed forward neural networks, where proper abstraction is done during the learning phase, often are entitled eager learners.

The k-nearest 4eighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor.

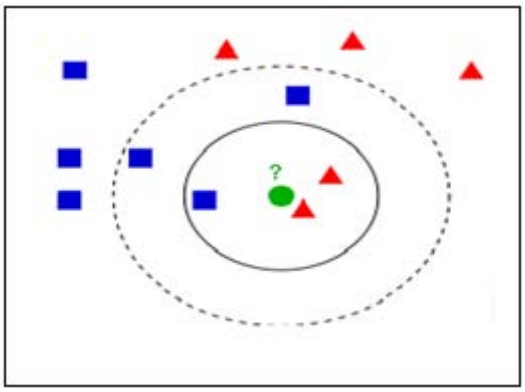


Figure 1.9: Classification of objects using K-NN

Figure 1.9 Show the Example of *k*-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ is assigned to the second class as there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

2) Parameter Selection

The best choice of *k* depends upon the data; generally, higher values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest 5eighbor algorithm. The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features. More description about K-NN classifier can be found at [9] and [10].

7. Result and Conclusion Gurmukhi Numeral Recognition with SVM

Details about the Gurmukhi script has been defined in above section. Five-fold cross validation has been used to validate result obtained. Zone based feature extraction techniques consisting 100 feature vectors and SVM classifier with RBF kernel has been used to achieve 99.73% accuracy which is the highest obtained result. Table 1 depicts various experiments on different size of image and block. The first method consisting of 100 feature vectors, second method also consists of 100 feature vectors and third method is combination of both of the method, so it consists of 200 feature vectors. Feature vector depends upon size of image and block. These all methods are described in above section.
Table 1 shows the experimental result on different sizes of image and block. In first and second method recognition, accuracy decrease when image size decrease despite of image size 32×32 and block size 8×8 and in second method accuracy is stable above 50×50 image size and block size 10×10. Feature vector fv2 and image size 100×100 provide the highest accuracy.

Table 1: Gurmukhi recognition accuracy on different size of image

S. No.	*Feature vector*	*Image size (block size)*	*γ*	*C*	*Recognition accuracy %*
1	Fv1	16×16 (4×4)	1	0.08	93.86
2	Fv2	16×16 (4×4)	2	0.08	98.06
3	Fv3	16×16 (4×4)	2	0.08	96.73
4	Fv1	16×16 (8×8)	4	0.08	74.60
5	Fv2	16×16 (8×8)	8	0.04	72.40
6	Fv3	16×16 (8×8)	16	0.08	90.40
7	Fv1	32×32 (8×8)	64	0.04	93.16
8	Fv2	32×32 (8×8)	64	0.04	98.13
9	Fv3	32×32 (8×8)	64	0.04	96.60
10	Fv1	50×50 (5×5)	8	0.04	38.60
11	Fv2	50×50 (5×5)	4	0.08	99.60
12	Fv3	50×50 (5×5)	8	0.04	53.53
13	Fv1	50×5010×10)	2	0.008	43.40
14	Fv2	50×50 (10×10)	32	8	99.66
15	Fv3	50×50 (10×10)	16	.008	68.13
16	Fv1	60×60 (10×10)	4	0.04	50.60
17	Fv2	60×60 (10×10)	4	0.04	99.33
18	Fv3	60×60 (10×10)	2	0.04	70.73
19	Fv1	60×60 (6×6)	16	0.001	35.46
20	Fv2	60×60 (6×6)	4	4	99.6
21	Fv3	60×60 (6×6)	4	0.001	44.33
22	Fv1	100×100 (10×10)	2	1	35.46
23	**Fv2**	**100×100 (10×10)**	**4**	**1**	**99.73**
24	Fv3	100×100 (10×10)	32	1	34.40

Recognition with SVM and observing the results at different values of parameter C. It has been analysed that increasing the value of C, increase the recognition rate, but after a certain increment normally after 2 the recognition rate becomes stable. In contrast, the recognition rate always changes with the change in C. The optimized results are obtained at C=2 and 4, γ value in range 2^{-5} to 2^{-1}.The trends of result variation with SVM classifier at $\gamma = .001$ and varying the value of C shown in figure 1.10

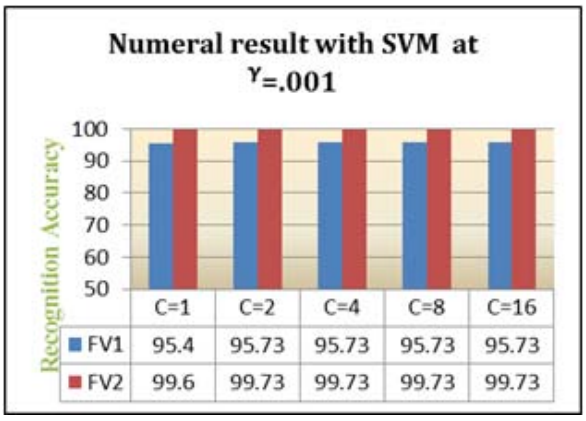


Figure 1.10: Results with SVM at γ =0 .001 and different values of.C

8. Devanagri Numeral Recognition with SVM

Current feature extraction techniques are experimented on Devanagri numeral. First, LIBSVM is trained with default option where the kernel type is RBF. In experimental results are tested on different value of C and γ. The value of c and γ is 1 and obtained the recognition accuracy is 99.11%. Now the value of c is chosen 500 randomly and the value of

gamma is started with 0.1 to 1 and the size of image is 60×60, the recognition accuracy thus obtained is shown in table 2. Brief discussion on Devanagri numeral has been done in above section

Table 2: Recognition accuracy at different-2 value of gamma (γ) and C

S. No.	Value of gamma(γ) for RBF	Value of C	Recognition Accuracy
1	0.001	3	96.07 %
2	0.001	4	96.36 %
3	0.001	8	96.79 %
4	0.08	8	96.79 %
5	0.08	32	97.37 %
6	0.08	500	97.39 %

Figure 1.11 shows the graphical representation of the accuracy at different value of the gamma (γ) and fixed value of C. We get maximum recognition accuracy 97.79 % which is highest among all the values.

The value of the C parameter is fixed in above experiment. Now we changed the value of γ. In our case there is no effect on the recognition rate by changing the value of gamma that can be seen by figure 1.11.

Table 3 shows accuracy at the value of gamma (γ) =0.1 and different value of C. The maximum recognition accuracyhas been obtained on it. The highest accuracy is 99.11 % at Handwritten Devanagri Numeral. The top three recognition accuracies are 98.45%, 98.95% and 99.11%.

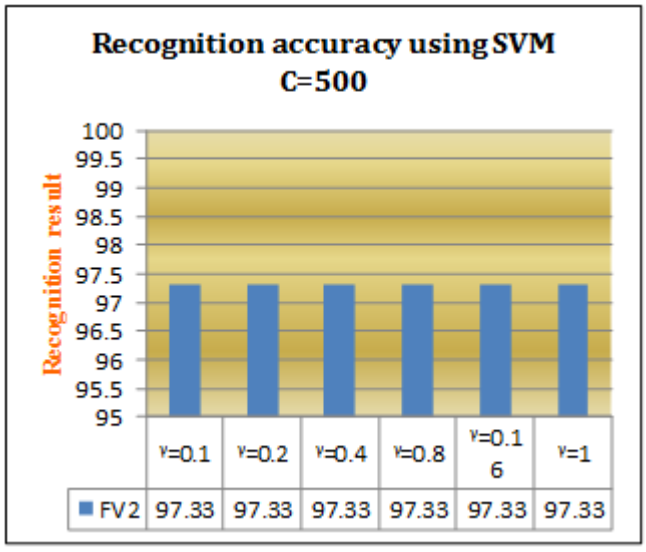


Figure 1.11: Result at different value of gamma and fixed value of C.

Table 3: Devanagri numeral recognition accuracy on image size 100×100

Sr. No.	C	Recognition accuracy
1	**1**	97.62%
2	2	98.11%
3	4	98.34%
4	8	98.45%
5	16	98.95%
6	500	99.11%

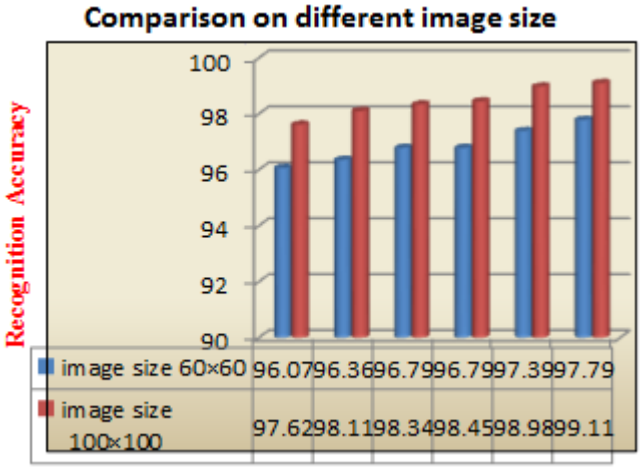


Figure 1.12: Comparison on different image size using SVM at Different value of gamma and C

In figure 1.12 compares the recognition accuracy on different size of images. Image size 60×60, and block size 6×6 on feature vector 100 yields highest accuracy of 97.79%. Then, image size 100×100, block size 10×10 for 100 feature vectors which produce 99.11% accuracy which is highest

References

[1] M. Hanmandlu, J. Grove , V. K. Madasu, S. Vasikarla, "Input Fuzzy Modeling for the Recognition of Handwritten Hindi Numerals," *Information Technology, 2007. ITNG '07. Fourth International Conference on*, vol., no., pp.208-213, 2-4 April 2007.

[2] Omid Rashnodi, Hedieh Sajedi, Mohammad Saniee, "Using Box Approach in Persia Handwritten Digits Recognition," *International Journal of Computer Applications (0975 –8887) Volume 32 No.3, October 2011.*

[3] Kartar Singh, Siddharth Renu Dhir, Rajneesh Rani, " Handwritten Gurmukhi Numeral Recognition using Different Feature Sets," *International Journal of Computer Applications (0975– 8887)Volume 28– No.2, August 2011.*

[4] S.L.Mhetre, Prof.M.M.Patil [4],"Comparative study of two methods for handwritten Devanagari and Gurmukhi numerals Recognition" IOSR Journal of Computer Engineering (IOSR-JCE)e-ISSN: 2278-0661,p- ISSN: 2278-8727Volume 15, Issue 6 (Nov. - Dec. 2013), PP 49-53 www.iosrjournals.org

[5] Reena bajaj et al., "Devnagari numeral recognition by combining decision of multiple connectionist classifiers" SadhanaVol. 27, Part 1, February 2002, pp. 59–72. © Printed in India

[6] U. Bhattacharya et al. "Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition" https://hal.inria.fr/inria-00104481.

[7] S.V. Rajashekararadhya, "efficient zone based feature extraction Algorithm for Hand written numeral Recognition of four popular south Indian Scripts," *Journal of theoretical and Applied Information Technology, 2008.*

[8] H. Swethalakshmi, Anita jayaraman, V. Srinivasa Chakravarthy , C. Chandra sekhar, "Modular Approach

to Recognition of Strokes in Telugu Script," *Document Analysis and Recognition, Ninth International Conference on* , vol.1, no., pp.501-505, 23-26 Sept. 2007.

[9] http://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed an introd- uction to k nearest-neighbour-knn-algorithm/K-nearest neighbour algorithm.htm.

[10] Dharamveer Sharma, Puneet Jhaji, "Recognition of Isolated Handwritten Characters in Gurmukhi Script," *International Journal of Computer Applications (0975 – 8887) Volume4–No.8, August 2010.*