International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
OPEN ACCESS
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST52310636

Comparing the Performance of Handwritten Number Recognition in Devanagari and Gurmukhi Script

Rakesh Kumar Roshan[1], Anand Kumar[2]
[1]Department of CSE, Asst. Professor, RRSDCE, Begusarai, Bihar, India.
[2]Department of CSE, Asst. Professor, SCE, Sasaram, Bihar, India.

ARTICLEINFO

Article History:

Accepted: 05 Jan 2024
Published: 22 Jan 2024

Publication Issue :
Volume 11, Issue 1
January-February-2024
Page Number :
286-294

ABSTRACT

In this work, we compared the effectiveness of two distinct approaches for numerical recognition. This work aims to offer a dependable and effective technique for handwritten numeral recognition. The Image Centroid Zone feature extraction and recognition algorithm is used in the first method. This method involves extracting the image's features, which are then compared to the feature set of a database image for categorization. In contrast, the second approach uses Zone Centroid Zone methods to extract features, which are then used to train a support vector machine (SVM) to recognize the input image. The study field of Handwritten Optical Numeral Recognition (HONR) is significant due to its extensive applicability in various fields such as bank cheque reading, postcode reading, form processing, post offices and hospitals.

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

I. INTRODUCTION

OCR - The area of computer science known as optical character recognition reads text off of paper and converts the images into a format that can be used by the computer (like ASCII codes). With the aid of an OCR system, you may scan an article from a magazine or book and input it straight into an electronic computer file, which you can then edit with a word processor. An optical scanner for text reading and advanced software for image analysis are included in every OCR system. While some low-cost systems recognize characters only using software, the majority of OCR systems employ a combination of hardware (specialized circuit boards) and software. Modern OCR systems still struggle to read handwritten writing, although they can read text in a wide range of fonts.

II. LITERATURE REVIEW

M. Hanmandlu et.al. [1] Suggested a method based on the modified exponential membership function for handwritten Hindi numeral recognition. 3500 datasets were used where 95% recognition rate is found overall. Omid Rashnodi, et.al. [2]. To improve recognition accuracy and reduce recognition time for Persian numerals, they suggested the box approach method. Support vector machines (SVMs) with linear kernels have been used as the classifiers in the classification phase. A correct recognition rate of 98.945% was attained. Kartar Singh, et.al. [3]. They have made use of Background Directional Distribution (BDD) characteristics, zonal density, and projection histograms. For classification, the SVM classifier with an RBF (Radial Basis Function) kernel is employed. Their accuracy scores are 99.2%, 99.13%, and 98%. S.L.Mhetre, et.al. [4] The Image Centroid Zone and Zone Centroid Zone methods were utilised to extract features, which were then fed into an artificial neural network to recognise the input image. After using ANN and Grid approaches to 500 data points, the results showed an accuracy of 86.4% and 83.6%, respectively.

Reena bajaj et al. [5] Handwritten Devnagari numeral recognition is the focus of this paper. Among these three methods, he obtained the maximum accuracy of 89.68% by using density features, descriptive component features, and moment features of the right, left, upper, and lower profile curves.

U. Bhattacharya et al. [6] For the Neural Combination of ANN and HMM for Handwritten Devanagari Number Recognition, they have employed the two classification techniques, HMM and ANN. They achieved the highest accuracy of 91.28%.

1. Collection of Data

The Gurmukhi script originated from the Landa alphabet and was established in the 16th century by the second Sikh leader, leader Angad Dev Ji. Derived from the Old Punjabi word Gurmukhi, the name Gurmukhi means "from the mouth of the Guru". Spoken primarily in East Punjab, India, and West Punjab, Pakistan, Punjabi is an Indo-Aryan language that is spoken by around 105 million people. Punjabi originated in the Shauraseni language of northern mediaeval India and emerged as a separate language in the eleventh century. Punjabi is written in India using the Gurmukhi alphabet, but in Pakistan, Shahmukhi, an Urdu alphabet variant, is used. In both India and Pakistan, the written standard for Punjabi is called Majhi.

Figure 1.1: Gurmukhi Numeral

Sometime in the eleventh century AD, Devnagari sprang from the Brahmi script. Originally designed to write Sanskrit, it was eventually modified to write a wide range of other languages, including Hindi, Marathi, and Nepali. Figure 1.2 displays the ideal Devnagari numerals (printed). This image illustrates how the shapes of several ५, ८, and ९ in their printed versions vary. The handwritten forms of Devnagari numbers, however, vary greatly, as can be seen from the samples in figure 1.2.

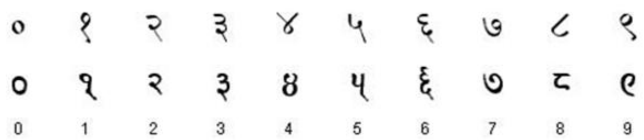

Figure 1.2 : Devanagari Numerals shapes

We have 22546 samples of isolated handwritten Devnagari numerals from 1049 individuals in our collection. A training set has been created from the entirety of the accessible data.

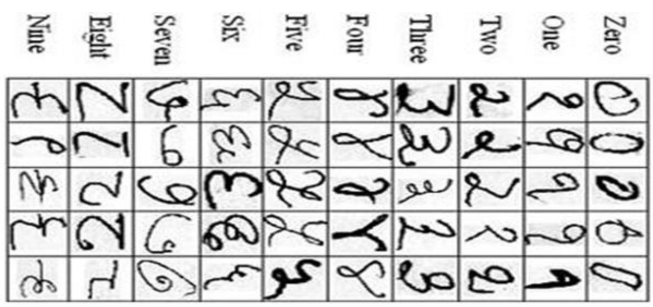

Figure 1.3 : Handwritten Devanagari Numerals samples

1500 samples of data have been gathered in order to recognize the Gurumukhi numerals.15 distinct authors with a range of backgrounds, occupations, and ages have contributed Gurmukhi numerals. Samples of Gurmukhi counting displayed in Figure 1.4

Figure 1.4: Gurmukhi numeral samples

2. Prior to Processing

Encoding handwritten or typed numerical images into a machine-readable format is the essence of numerical recognition. To do this, a standard scanner is used to scan the page first. Pre-processing procedures (shown in the flow chart below) are applied to the scanned data/image before the feature extraction is carried out directly. These steps are as follows:

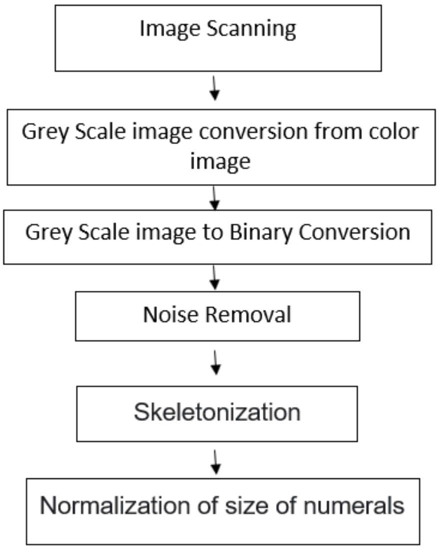

Figure 1.5 : Steps of pre-processing

Grayscale image conversion takes place if the scanned image was originally colour image.

- Although grayscale, an image retains its raw quality and could contain undesired information. We call this information "noise." When scanning the image, noise or distortion could be introduced. There are other kinds of noise, such shot noise, salt and pepper noise, and so on. To get rid of such noise, apply a median filter.
- After the picture has been filtered, binary form is created, that is an alternative term for the procedure is image segmentation.Using a variable to set a threshold value, picture segmentation is carried out. If a grayscale image pixel's value is greater than or equal to the threshold, the pixel is replaced with 1; otherwise, it is set to 0. To make processing easier, the image is flipped at the end.
- After the binarization process image is transferred for the skeletonization and thinning procedure. Numerical widths are reduced from many pixels to one using skeletonization.
- The picture is finally resized to its typical 32 by 32 dimensions. The use or taking of these dimensions alone is not required. Making a database of photos with a set, uniform size is our goal. Achieving

operational feasibility can be done in any dimension.

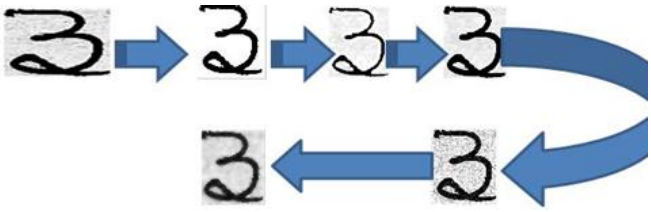

Figure 1.6 RGB image, binary image, dilation of image, perform erosion on image, add the noise into image, removing noise from image

III.Extraction of Features

Since classification is based on the set of features retrieved during this step, it is the most crucial step in character/numeral recognition. The process of obtaining or measuring the input image parameters that are most helpful for classification is known as feature extraction. Scientists have created and employed a large number of these features for pattern classification. The present experiment makes use of the features described below. There are two different Kinds of features: zone centroid zone and image centroid zone.

Combinations of the two fundamental features were used to create 200 feature vectors. These techniques offer high-quality recognition at an easy-to-use level. The following section defines the step-by-step algorithm. These algorithms have been defined in the upcoming section. The specifics of the feature extraction technique are explained in the paragraph that follows.

2.1 Image Centroid Zone

After determining the image centroid, the given image is divided into 100 x 100 equal zones, with each zone's size being 10 x 10. Next, the average distance between the image centroid and each pixel in the zones/block is calculated, yielding 100 feature vectors for each image. Zero is assumed for any empty zones. This process is repeated for all the zones in the image.

A sample 32x32 character image is shown in Figure 1.7. The image's centroid is calculated first. Next, the image is separated into 16 identical zones, each measuring 8 by 8. Subsequently, the average distance between each pixel in the image and the centroid is calculated.

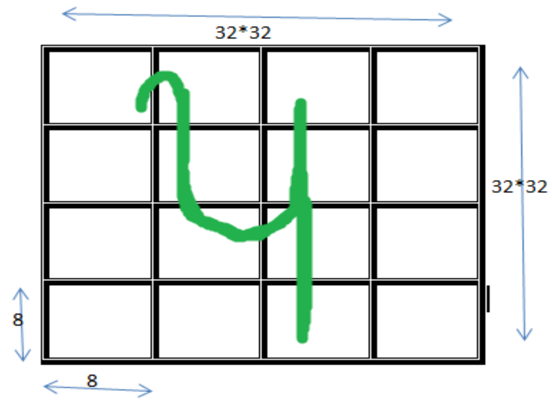

Figure 1.7 (ICZ) Image 32×32 and block 8×8.of handwritten Gurumukhi Numeral “**five**”

2.2 Zone Centroid Zone

ZCZ divides an image into 100 x 100 equal zones, and calculates the centroid of each zone. then comes the calculation of the average distance between each zone pixel and the zone centroid. Zero is assumed for vacant zones. Every pixel in every zone goes through this process once more.

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 1.3.

An illustration of a character image in size 32 × 32 is shown in Figure 1.8. The image in this figure has been divided into sixteen equal zones, each measuring eight by eight. The centroid of every zone in the picture has been calculated. Next, the average distance between each pixel in the zone and the image centroid is computed.

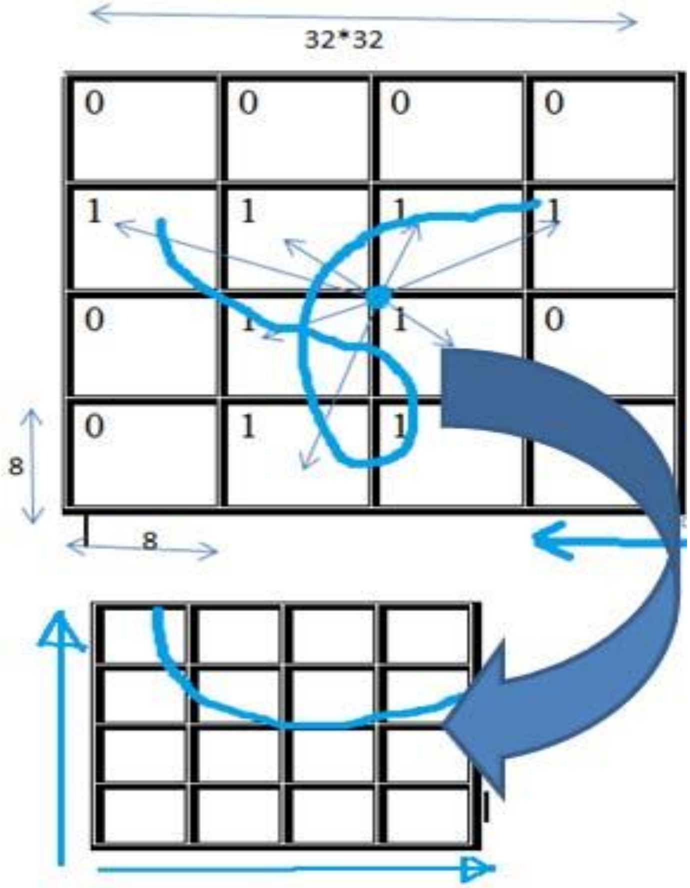

Figure 1.8: (ZCZ) Image 32×32 and block 8×8.of handwritten Devnagari Numeral "four"

Algorithm 1: Feature extraction method using Image Centroid Zone (ICZ).

Input: Pre-processed Image (a binary Number/ a character)

Output: Extraction of the Features for Classification and Recognition

Step 1: Determine the input image's centroid.

Step 2: Split the input image into 100 equal-sized zones.

Step 3: Calculating the separation between each zone pixel and the centroid of the image.

Step 4: For each pixel in the zone, boxes, or grid, repeat step 3 once.

Step 5: Calculate the average distance between these points.

Step 6: Completely repeat this process for every zone in the picture.

Step 7: Getting 100 of these features for the classification and recognition procedure.

Algorithm 2: feature extraction method using zone Centroid zone (ZCZ).

Step 1: Splitting the input image into n equal sections.

Step 2: Determine each zone's centroid.

Step 3: Determine the separation between each zone pixel and the zone centroid.

Step 4: Carry out step 3 again for every pixel that is present in the zone, box, or grid.

Step 5: Calculating the average distance between each of these image-present sites.

Step 6: Repeat these steps in order for the entire zone.

Step 7: Collecting 100 of these attributes for recognition and classification.

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.

IV. Classification

In this implementation, two classifier types SVM and K-NN have been used. SVM is one of them that is widely used, effective, and yields the best results in this implementation. These two categories of classifiers have been applied to the recognition of Gurmukhi characters.

2.3Support Vector Machines (SVM)

The usage of Support Vector Machines (SVM) for Recognition and Classification has become widespread. The idea of decision planes—which establish decision boundaries—is the foundation of support vector machines. One that distinguishes between a group of objects with various class memberships is known as a decision plane. In computer science and statistics, the term "support vector machine" (SVM) refers to a group of linked supervised learning techniques that examine data and identify patterns for use in regression analysis and classification. Given a limited quantity of training data, it can learn to attain good generalization performance, which is the goal of any machine.

Balancing the machine's capacity to produce error-free recognition across all datasets with the goodness of fit found on a particular training dataset. With a set of input data, the conventional Support Vector Machine (SVM) predicts two alternative input classes. A model that classifies fresh instances into one or more categories is created by the SVM training algorithm. When using SVM for pattern recognition, the decision plane is constructed as a hyper-plane that has the highest margin of separation between the positive and negative patterns.

By using the concept of basis, SVM has demonstrated high generalization performance without requiring knowledge of past data [8].

2.4 K-Nearest Neighbor (KNN)

One of those really easy-to-understand algorithms that performs amazingly well in practice is K Nearest Neighbour (KNN). Its applications span a wide range, including graphs, proteins, computational geometry, vision, and more. It is also remarkably adaptable. Although most people study the technique and never apply it, it's unfortunate because a well-executed KNN can simplify many complex situations.

K-NN is an object classification technique that uses the closest training examples in the feature space. K-NN classifiers are instance-based and work under the assumption that unknown instances can be classified by comparing them to known examples using a distance or similarity function. Two examples that are far apart in the instance space defined by the proper distance function are less likely to belong to the same class as two instances that are near together, according to intuition.

The learning processes

During the learning phase, instance-based learners do not extract any information from the training data, in contrast to many artificial learners. Encapsulating the training data is all that is required for learning. Generalization is not done until it is absolutely necessary, which is when classification occurs. Due to this characteristic, classifiers like feed forward neural networks—where appropriate abstraction is completed during the learning phase—often refer to eager learners, whereas instance-based learners are referred to as lazy learners. Among all machine learning methods, the k-nearest neighbor algorithm is one of the simplest. It classifies an item based on the majority vote of its neighbors, assigning it to the class that has the most members among its k nearest neighbors (k is a positive integer, usually small). The object is simply put into the class of its closest neighbor if k = 1. The k-NN classification example is shown in Figure 1.9. Either the first class of blue squares or the second class of red triangles should be assigned to the test sample (green circle). Given that there are two triangles and only one square inside the inner circle, the second class is allocated if k = 3. Assuming that the first class has k = 5 (3 squares versus 2 triangles inside the outer circle).

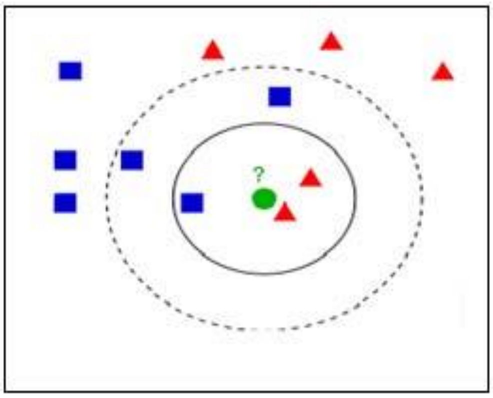

Figure 1.9: Classification of objects using K-NN

- **Parameter Selection**

While higher values of k often lessen the impact of noise on the classification, they also blur the borders between classes. The optimal value of k depends on the data. Many heuristic methods, such as cross-validation, can be used to choose a good k. The nearest 5eighbor approach is applied in the exceptional scenario (k = 1) when the class is predicted to be the class of the closest training sample. The existence of irrelevant or noisy features can significantly reduce the accuracy of the K-NN algorithm. Further information regarding the K-NN classifier is available at [9] and [10].

V. Finding and Recommendation SVM-Based Gurmukhi Number Recognition

Information on the Gurmukhi script has been defined in the section above. The outcome has been validated by the use of five-fold cross validation. The greatest result obtained was 99.73% accuracy using zone-based feature extraction techniques with 100 feature vectors and an SVM classifier with an RBF kernel. A variety of trials with varying picture and block sizes are shown in Table 1. The first approach uses 100 feature vectors; the second uses 100 feature vectors as well; the third uses a combination of the two methods, using 200 feature vectors in total. The size of the image and block determines the feature vector. All of these techniques are explained in the section above. The experimental results for various picture and block sizes are displayed in Table 1. Accuracy decreases with decreasing image size in both the first and second methods of recognition, even for images that are 32 × 32 and block sizes of 8 × 8. In the second approach, accuracy remains constant above 50 by 50 image size and block size of 10 × 10. The maximum accuracy is obtained with feature vector fv2 and image size 100×100.

Table 1: Gurmukhi recognition accuracy on different size of Image

S. No.	*Feature*	*Image size (block*	γ	*C*	*Recognition*
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	32	8	99.66
15	Fv3	50×50	16	.008	68.13
16	Fv1	60×60	4	0.04	50.60
17	Fv2	60×60	4	0.04	99.33
18	Fv3	60×60	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	2	1	35.46
23	**Fv2**	**100×100**	**4**	**1**	**99.73**
24	Fv3	100×100	32	1	34.40

Utilising SVM for recognition and tracking the outcomes at various parameter C values. Analysis has shown that raising the value of C increases the recognition rate; however, the recognition rate often stabilises after a certain increment, usually around 2. On the other hand, as C varies, the recognition rate does as well. The optimal outcomes are attained at C=2 and 4, with η falling between 2 and 5 and 2-1.Figure 1.10 displays the trends of result variation with SVM classifier at γ =.001 and varying the value of C.

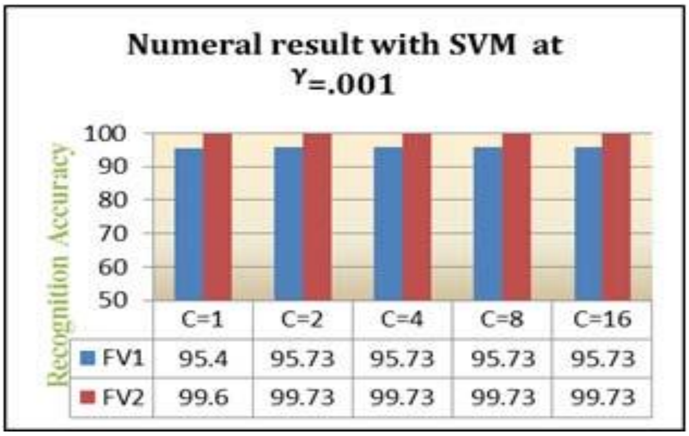

Figure 1.10: Results with SVM at γ =0 .001 and different values of C

VI. Numeral Recognition in Devanagari Using SVM

The Devanagari numeral is used to test current feature extraction techniques. First, RBF is used as the default kernel type while training LIBSVM. Tests are conducted on various values of C and γ in the experimental data. With a recognition accuracy of 99.11%, the values of c and γ are both 1. Table 2 displays the recognition accuracy that was achieved by randomly selecting 500 for the value of c and starting the gamma value at 0.1 and increasing it to 1. The image size is 60 by 60. In the section above, there was a brief discussion of Devanagari numerals.

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

S.	Value of	Value	Recogniti
1	0.00	3	96.07 %
2	0.00	4	96.36 %
3	0.00	8	96.79 %
4	0.0	8	96.79 %
5	0.0	32	97.37 %
6	0.0	500	97.39 %

The accuracy at various gamma (γ) values and fixed values of C is graphically represented in Figure 1.11. Out of all the values, the maximum recognition accuracy we obtain is 96.79%. In the above experiment, the value of the C parameter is fixed. We have now adjusted γ's value. In our scenario, altering the gamma value has no influence on the recognition rate, as figure 1.11 illustrates.

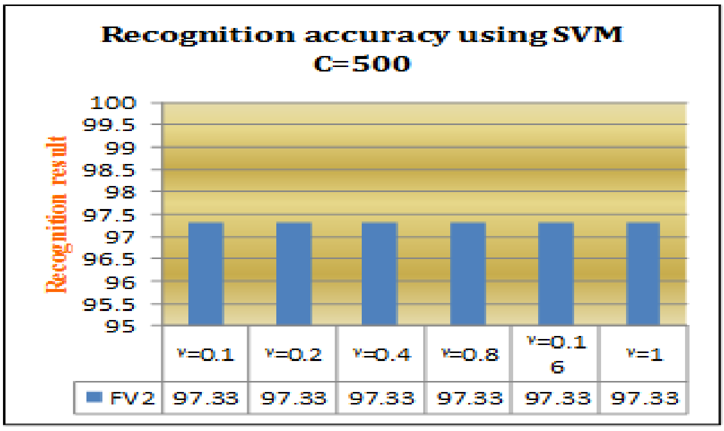

Figure 1.11: Result at different value of gamma and fixed value of C.

Accuracy at gamma (γ) = 0.1 and various values of C are displayed Table 3. It has achieved the highest level of recognition accuracy. When using handwritten Devanagari numerals, the highest accuracy is 99.11%. At 99.11%, 98.95%, and 98.45%, respectively, the top three recognition accuracies.

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

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

The recognition accuracy for various image sizes is compared in figure 1.12. The greatest accuracy of 97.79% is obtained with image size 60×60 and block size 6×6 on feature vector 100. Then, for 100 feature vectors at an image size of 100 by 100 and a block size of 10 by 10, the highest accuracy of 99.11% is produced.

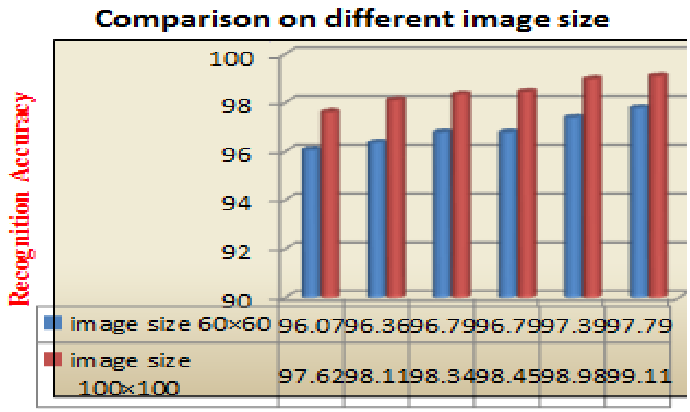

Fig. 1.12: Comparison on different image size using SVM at Different value of gamma and C

VII. 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 introduction 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.

Cite this article as :

Rakesh Kumar Roshan, Anand Kumar, "Comparing the Performance of Handwritten Number Recognition in Devanagari and Gurmukhi Script", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 1, pp. 286-294, January-February 2024. Available at doi : https://doi.org/10.32628/IJSRST52310636

Journal URL : https://ijsrst.com/IJSRST52310636