A dissertation on

A Hybrid Approach for Handwritten Devanagari Numerals Recognition Using HOG Algorithm and K-NN Classifier

submitted in partial fulfillment of the requirements

for the award of degree of

Masters of Technology

In

Computer Science and Applications

Submitted By

Bhavesh Kumar Shukla

Roll No. 601303009

Under the supervision of

Dr. Rajiv Kumar

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

JULY 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **" A Hybrid Approach for Handwritten Devanagari Numerals Recognition Using HOG Algorithm and K-NN Classifier"**, in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Rajiv Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Bhavesh Kumar Shukla

(Bhavesh Kumar Shukla)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Rajiv Kumar)

Assistant Professor, CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

Nothing big can be achieved without the guidance, assistance and co-operation of people and one must be indebted to them by words and deeds. A coordinated effort of others bridges the gap between where you are and where you want to be. I am highly thankful to my guide Dr. Rajiv Kumar, Assistant Professor, Computer Science & Engineering Department, Thapar University, Patiala, for his guidance, advice, efforts, moral support, motivation and the attitude with which he solved all my queries in making this thesis possible. It has been a great honor to work under him.

I am also thankful to Dr. Deepak Garg, Associate Professor and Head, Computer Science and Engineering Department, Thapar University, Patiala, for providing us with adequate infrastructure in carrying the research work.

I am also thankful to Dr. Sanmeet Bhatia (PG Coordinator) for her contribution in making this thesis possible. I am also thankful to all the faculty and staff members of the Computer Science and Engineering Department for their full cooperation and help. I am also grateful to Mr. Sharad Tiwari who helped me in understanding the mathematical foundations needed for this thesis work. Besides the above dignitaries, I am also thankful to my friends Prashant Singh and Ashwani Kumar for their kind and noble support. I am forever indebted to my parents for their constant support and encouragement throughout life. This thesis is devoted to them. Most importantly, I pay my reverence to the almighty God for giving me inner peace and strength and not letting me down at the time of crisis.

Bhavesh Kumar Shukla

Bhavesh Kumar Shukla

(601303009)

ABSTRACT

Character Recognition has become fascinating research area over a last few decades. Reading handwritten characters is easy task for human but it is difficult task for machines. Optical Character Recognition (OCR) System is used to recognize handwritten and printed characters. A lot of work has been done on English characters but these days Indian scripts have become interesting research area. Devanagari is one of them. In the present study, an effort is given for handwritten Devanagari numerals recognition. Many approaches have been developed in the field of character recognition but still it remains challenging work for researchers. In Recognition Process, Neural network is used as basic technique in most of the approaches. A lot of training and large computations are required in these approaches. So, an effort is done by author to make an easy hybrid approach for handwritten Devanagari numerals recognition (HDNR). This comes under the category of Offline recognition process. In proposed approach, various techniques such as binarization, filtering, smoothing, normalization and thinning are used in preprocessing stage. Bounding box is used for segmentation. Feature extraction is main portion of OCR system because accuracy is mostly based on extracted features. Here Histogram of Oriented Gradients (HOG) algorithm is used for feature extraction. There are a lot of feature extraction methods but we preferred HOG because of its better feature evaluation capacity. After feature extraction, they are classified into ten classes such as zero, one, two, three, four, five, six, seven, eight and nine. K-NN Classifier has been trained with these features. There are various classifiers but we selected K-NN Classifier because its accuracy is better as comparison to other classifiers. There are a lot of applications of handwritten Devanagari numerals recognition such as reading bank cheques, passport readers, postal code readers, commercial forms reader, bill processing systems etc. Proposed Hybrid approach has been applied to many documents and author obtained satisfying results.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Overview

Now-a-days computers are capable of processing huge amount of data. Capacity of computer is increasing day by day at tremendous rate. Information exchange between computer and human is a big and serious problem. In order to fully utilize the capability of computer, an interface is required which should be efficient and useful to the user. User interface should be simple so that there is no difficulty for the user to communicate with the computer. The very first method of data communication between user and computer was done through conventional input devices like mouse and keyboard. But these devices have few limitations when we compare them to input through handwriting. Chinese and Japanese scripts have large alphabet sets and because of complicated typing nature of the scripts, it becomes complicated to give input to the computer using conventional input devices. Handwriting is the easiest way to interchange information between computer and human. So, handwriting recognition has a big potential to improve communication between computer and user. Handwriting recognition is the capability of a computer to explain the data which is basically handwritten and input is taken from many sources like touch screens, documents and other devices. Main purpose of handwriting recognition is to read the data and then produce data in any desired editable format. Many researchers have worked on handwriting recognition all over world. Now-a-days handwriting recognition has become a fascinating research area.

Now-a-days all things are becoming automated to do any work easily and quickly. Digitization is becoming an upcoming trend in today's fast life. We are using computer in each part of life but the main problem is to make computer familiar with real world situations. One such type of process used for human machine interface is OCR. Making a machine which has human like capability is a challenging work and this is known as Artificial Intelligence.

1.2 Character Recognition

Basically character recognition process belongs to objects (symbols, alphabets and numbers) which are drawn on any image and their recognition. On the basis of data acquisition, character recognition is divided into under-mentioned categories:-

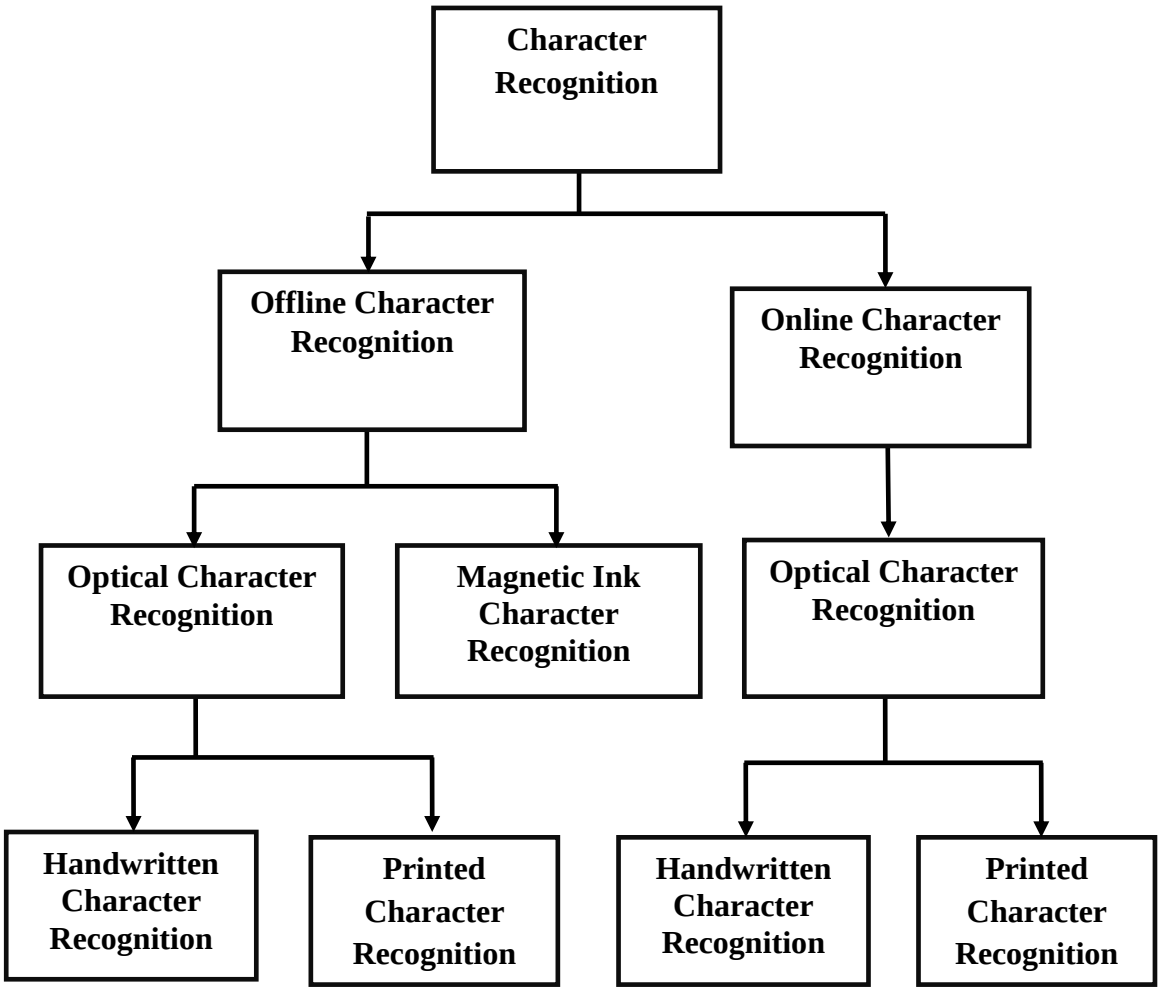

Figure 1.1: Classification of Character Recognition

Character recognition technique is classified into two parts:-

- Offline Character Recognition
- Online Character Recognition

1.2.1 Offline Character Recognition

In Offline recognition, characters are not recognized at the time of writing. Here data is written on any surface such as paper document and recognition process is performed by computer by scanning that surface. Firstly, scanned paper is stored in the gray scale format i.e. bitmap image and after that, recognition process is performed to find good recognition accuracy. Recognition features are extracted and enhanced from bitmap images with the help of digital image processing. This is also called Optical Character Recognition (OCR). Machine printed character recognition is also a part of OCR. In offline, no real time interaction exists. So, offline recognition methods are not so suitable for human-machine communication. Here we proposed method for Offline handwritten recognition. Offline character recognition is also divided into two categories:

- Optical Character Recognition (OCR)
- Magnetic Ink Character Recognition (MICR)

In MICR, magnetic ink is used to print the characters. Reading device identifies each character on the basis of each character's specific magnetic field. MICR is used for bank cheque authentication. In OCR system input documents are obtained optically with the help of camera or scanners then these documents are recognized by applying some algorithm. Characters may be handwritten or printed, of any shape and size. OCR is further divided into printed recognition or handwritten recognition. Implementing Handwritten Character Recognition System is a difficult job.

1.2.2 Online Character Recognition

In online recognition system characters are recognized at the same time when characters are written at media. Digital tablet is used as a surface for writing and digital pen is used for writing on that surface. As we write on the surface, two dimensional co-ordinates of related points are gathered and stored along with the time component [6]. Online recognition is shown in figure 1.2. Main difference between online recognition and offline recognition is that online recognition works in real time environment and offline does not [8].

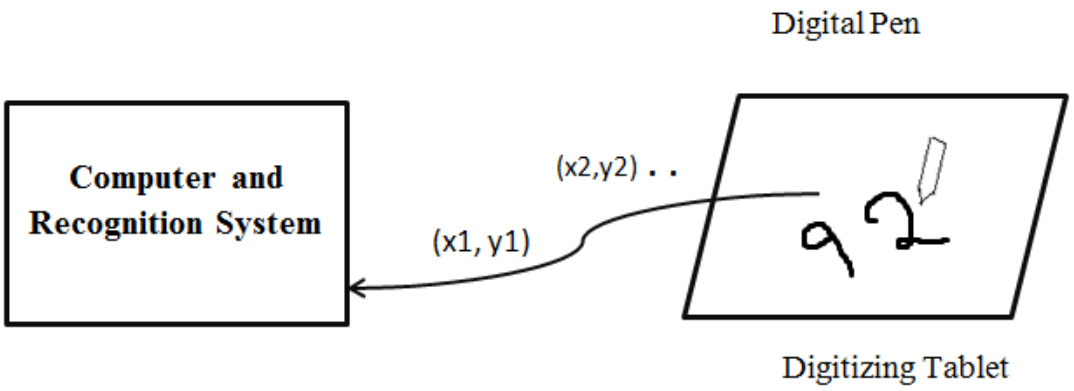

Figure 1.2: A tablet digitizer, input sampling and communication to the Computer

1.3 Optical Character Recognition

OCR systems provide facility to convert documents such as PDF files, scanned paper or image taken by camera into computer editable format. In other words, OCR is an electronic or mechanical way to convert any images of written or printed data into machine editable format. It works as a translator. OCR systems enable us to take any book and directly convert that into an editable computer file. An OCR method associates symbolic recognition with character's image [1]. Main objective of OCR is to copy the human's reading ability at a faster rate. The area of document recognition and analysis is very big and it has many applications. Character recognition is categorized into handwritten and printed character recognition. Further recognition of handwritten character is divided into two parts i.e. online and offline [6]. In offline recognition, handwritten document are scanned and stored in digital gray scale form and then recognition process is performed. In online recognition, a special pen is used to write on surface and pen stroke is stored in x and y two dimensional coordinate. Practically it is clear that online handwritten character recognition gives more accurate and efficient result as compared to offline handwritten character recognition. Figure 1.3 shows the OCR phases.

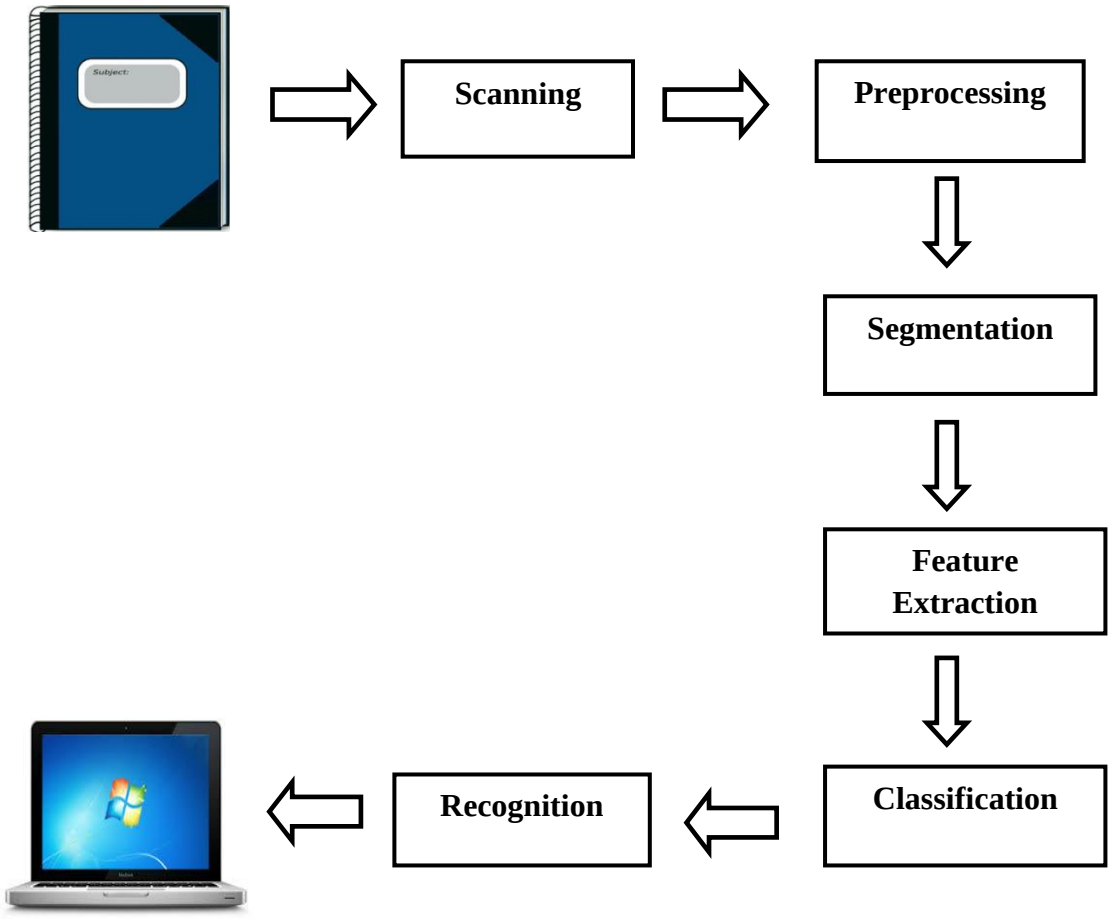

Figure 1.3: Phases of OCR

1.4 Phases of OCR System

There are many phases in OCR systems performed one by one to carry out the whole task. These phases are as follows:-

1.4.1 Data Acquisition

The way in which input image is acquired (online or offline) is known as data acquisition [23]. Any OCR system take data as input into two way either online or offline. When we scan any handwritten document, it is converted into digital image. This is a method of Offline data acquisition and such type of recognition is offline handwriting recognition. In online handwriting recognition, when we write with special pen on digital tablet, that image is also stored in digital format. Difference between offline and online handwriting recognition is described in Table 1.1.

Serial No	Compared area	Offline Character	Online Character
1	Real time interactivity	No	Yes
2	Pre-processing	Take more time	Take less time
3	Segmentation	Difficult	easy
4	Recognition rate	Lower	Higher
5	Availability of pen stroke	No	Yes
6	Input	Scanning paper document	With special pen

Table 1.1: Offline vs. Online handwritten recognition

1.4.2 Pre-processing

Preprocessing plays an important role in OCR. Binarization, skew removal, noise removal, skeletonization, inversion, morphological operations etc. are many operations which are done in pre-processing stage. Most of noises present in the input image are removed in pre-processing stage which helps to improve the accuracy rate of recognition system. Various operations to be done during pre-processing phase are discussed in the following sections:-

1.4.2.1 Noise Removal

A lot of random variations of color or brightness is known as image noise and considered as electronic noise. Mostly noise is produced by scanner or camera. Noise generates undesirable bits. Noise removal is the major objective of pre-processing. Noise affects the outputs. Figure 1.4 shows noise removal.

Figure 1.4 (a) Noisy image

Figure 1.4 (b) Image with removed noise

1.4.2.2 Inversion

Converting black pixels into white pixels and white pixels into black pixels is the process of inversion. We know that in images, values of black pixels are 0 and values of white pixels are 1. Shape or written matter in any image is represented by black pixel which is pointed by bit 0. In image processing all operations are applied on binary value 1. So any operation can't be performed on written shape or printed matter. Hence, inversion of image is necessary to perform all operation and to make processing fast. Figure 1.5 shows inversion process. This inverted image is passed to next steps.

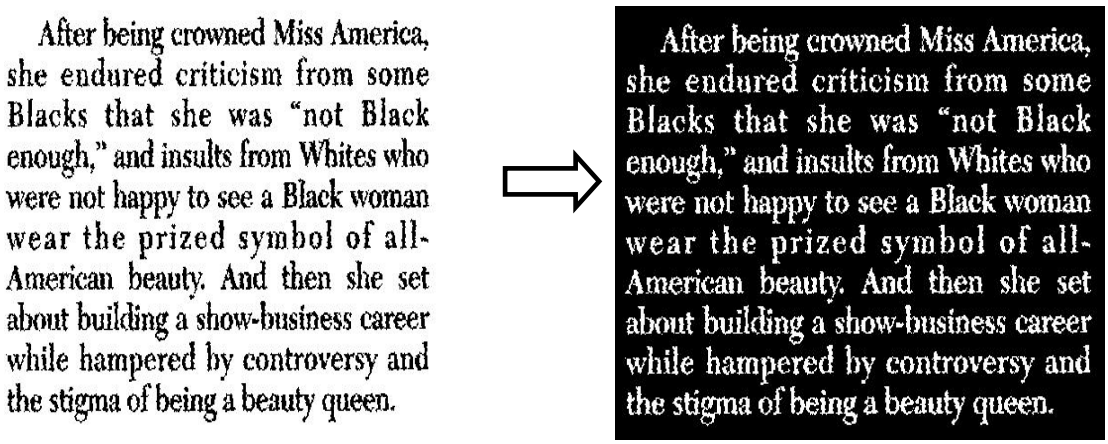

Figure 1.5: Image of inversion

1.4.2.3 Binarization

Generally input is given in RGB format. We convert RGB image into gray scale image. Then this converted gray scale image is transferred to binary image with the help of a threshold value. Converting RGB image to gray scale to binary image is known as binarization. For a gray scale image, threshold value is pre-determined or calculated by histogram of that image. When we convert a gray scale image into binary then we take each pixel of gray scale image. If pixels of that image which have values greater than or equal to threshold value then give 1 to that pixel otherwise give 0 to that pixel. Threshold may be local or global. If we take threshold value for total image then that is global. If we divide an image into many parts and calculate different- different threshold values for each part, then that is local thresholding

RGB image

Gray scale image

Binary image

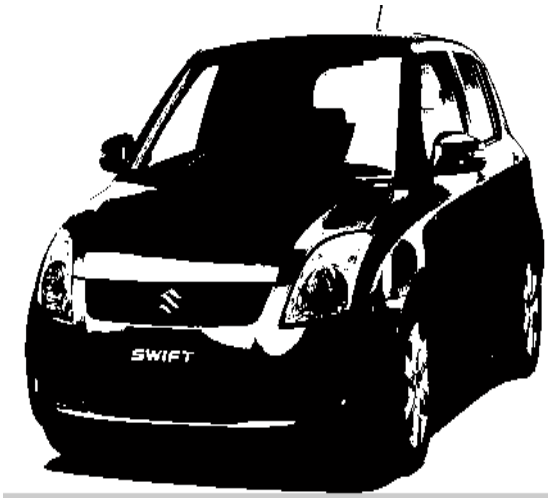

Figure 1.6: Image of Binarization

1.4.2.4 Skeletonization

Reducing the width of any line is known as Skeletonization for example, reduction of many pixels of an object into single line of pixel. This will help in removal of irregularities in the letters and help to make a simpler algorithm for recognition. It reduces the required memory space and processing time. Thin version of any shape is called skeleton.

1.4.2.5 Normalization

Removing variations of handwriting and obtaining a standard size is the main goal of Normalization. Methods of normalization are given as follows:-

(a) Skew normalization: Skew means tilt in any scanned bitmap image. The main reason of skew is document not fed properly into scanner or handwritten style. Without removing skew of any document, segmentation cannot be done in proper way. So it is required to develop an algorithm which automatically detects the skew of scanned bitmapped image and corrects them [25].

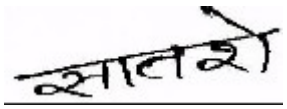

Figure 1.7: Skewed image

(b) Slant Normalization: All characters are normalized in a standard form with the help of slant normalization. Calculating average angle of each character is slant estimation method. Slant angle is calculated with the help of starting and ending coordinates of line element. Dividing an image into horizontal and vertical windows is a method for slant detection. Slant calculation depends on Center of gravity for the lower and upper of every window [5].

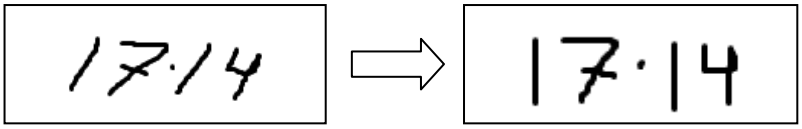

Figure 1.8: Slant Correction

(c) Contour Smoothing: Removal of errors produced because of inconsistent handwriting is known as contour smoothing. It improves the efficiency of the successive steps in preprocessing as it reduces the sample points required in order to represent the document.

(d) Size Normalization: Adjusting the size of all characters to a standard size is known as size normalization. Each character is partitioned into many zones and each zone is scaled separately. Hence normalization can be performed vertically and horizontally both [7].

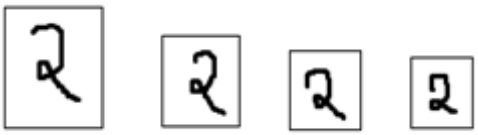

Figure 1.9: Size normalization

1.4.2.6 Compression

Reducing unnecessary bits from the image is called as image compression. It reduces the size of image. Thinning and Thresholding is two famous compression techniques which are as follows:-

(**a) Thinning:** Extract the information about the character's shape is called as thinning. It is the conversion of offline data to almost online like data. There are two types of thinning named as pixel wise thinning and non pixel wise thinning. If global information of shape is taken then it is called as non pixel wise thinning. If pixel wise information of shape is taken then it is called as pixel wise thinning.

(b) Thresholding: Thresholding is required for reducing the storage space and increasing the processing speed. In thresholding, grayscale image is converted into binary image based on selected threshold value. There are two types of thresholding named as local thresholding and global thresholding. If a threshold value is selected for total image then it is called global thresholding. If more than one threshold values are selected for different-different portions of image then it is called as local thresholding [3].

1.4.2.7 Morphological Operations

We perform these operations to replace the convolution operations by logical operations in order to filter the document image. Morphological operations are designed for connecting the broken stokes and decompose the strokes which are connected, contour smoothing, thinning of characters and boundary extraction. So, we can say that morphological operations can be used for removal of noise in the document.

1.4.3 Segmentation

Breaking the image into many parts is known as segmentation. Segmentation makes analysis work easier. If we give handwritten image to recognition system, then for recognizing each character, it is necessary to break whole image line by line, word by word and character by character [4].Segmentation process is performed by many methods like edge based segmentation, threshold based segmentation, region based segmentation, clustering technique and bounding box methods.

(a) Edge based segmentation: Here edges of objects are detected as objects' boundaries that are used in segmentation process. Edge decides the boundaries of image so that segmentation becomes easy. There are many methods to detect the edge such as gray histogram technique, gradient based method etc. In Gray Histogram Technique, measurement of histogram is totally dependent on color and intensity of pixels of image. This technique gives better efficiency as compared to other. This method is not used when valley and detected edge are very large. While Gradient based segmentation is best for segmenting the image when intensity is changed immediately. Sobel, canny, laplacian of gaussian (log) are good edge detection operators in the gradient based segmentation. Canny is the best edge detector but it takes more time as compared to sobel.

(b) Region based segmentation (Bounding box): This technique starts from the middle of object and go outwards until it encounters the complete object's boundaries and draws a line around the object's boundary. Mainly region based segmentation is used to break an image into many regions on basis of some similarity. There are few techniques such as Thresholding, region splitting and merging etc. that belong to region based segmentation. Thresholding of histogram is used for image

segmentation. Thresholding is used to reduce the memory storage and to increase the speed of processing. Grayscale image is converted into binary image on the basis of some threshold value. Calculate a threshold value to apply Thresholding. The pixels which have intensity greater than threshold value are selected and stored. In such a way, information is separated from background. If a threshold value is calculated for total image area, then it is known as Global Thresholding and when many threshold values are calculated for all region of image that is known as local Thresholding.

(c) Clustering techniques: Cluster techniques make group of those patterns together which have similar attributes in some sense. On that basis segmentation is done. Clustering is performed on the basis of some attribute like color, size, texture etc. There are some clustering techniques like Fuzzy c means clustering and k-mean algorithm. In fuzzy c mean clustering, grouping of dataset is done in 'n' clusters where each point of dataset belongs to each cluster on a certain degree. In K-means algorithm, data vectors are grouped in some predefined clusters. Centroid of each predefined clusters is randomly initialized. Euclidian distance is measured to decide pixels proximity and then assigned into cluster.

1.4.4 Feature Extraction

In OCR system, input images are represented in the compressed format for processing. At that time, features are extracted for every class to differentiate with other classes. Magnitude and direction [12] of maximum variation of intensity in a small zone are measured by gradient. All characters have some features which are used in the recognition of characters. Feature extraction explores information about shape of characters which exist in pattern so that classification work becomes easy. In this stage, a text segment is analyzed and then stable features are selected. Feature extraction is the heart of any recognition system because classification process is totally based on the result of Feature extraction process. If the features are extracted in an effective way, then classification result will be good and accurate. If more number of features are extracted then recognition process will be more accurate.

1.4.5 Classification and Recognition

Mapping features of training sets to a feature vector is the main purpose of classifier. Numerous approaches are used for classification. Sometimes combinations of different-different algorithms are checked for getting more accuracy. There are many

methods for classification. Each classifier has some advantage and also some disadvantage. There are various classifiers which are used for classification like K-NN classifier, SVM, Artificial Neural network etc. Classification is known as decision making stage of OCR system and takes the features extracted by feature extraction as input. Classifier decides the class membership of a pattern [1]. Recognition task should be simple and easy to take decision so that easily reduces the misclassification probability rate. Features of a pattern or object is stored in a feature vector Y= [y_1, y_2, y_3, ----y_n]. Space of vector is known as feature space. Classification creates a problem when a pattern does not exist in vector feature space.

1.5 Applications of Offline Handwritten recognition

Offline handwritten recognition system has many applications as follows:-

- **Postal code recognition**

  OCR has the capability of reading postal forms. It reads handwritten digits of postal code. Working in post office, has become easy because of character recognition system.

- **Health Care**

  OCR technology is also used in Hospital. Healthcare professionals deal with large no. of forms of patients. To keep up all information they use OCR system. OCR system extracts the information from application form and stored in the database, so that patient's information is recorded.

- **Form Processing:**

  HDCR system is used to read any application form and collect the information. Recognition system is used to conduct any exam successfully because it is used to feed all applicants records in the computer.

- **Banking**

  In banks, offline handwritten recognition is used for reading the cheque.

  OCR has the capability to process bank cheques without human intervention. Bank cheques can be easily acquired into an image by a phone camera followed by scanning the text on it and then transferring the amount of money successfully. This mechanism gives almost cent percent accuracy in printed

checks while it is fairly accurate for hand written checks. It is implemented in hand written cheques by occasional human confirmation before transfer of money. This mechanism speeds up the banking process.

- **Legal Industry**

  Legal industries are also digitizing papers and documents now-a-days. This allows them to save spaces required to store documents. It also makes it simple to search any file once the documents are scanned and a database is maintained. Quick and easy access is therefore available to legal professionals and a huge database which is stored in digital format and can look up for them easily as they are text-searchable.

1.6 Motivation behind Thesis

OCR plays an important role in today's life. OCR makes human's work easy because it solves various complicated issues. Handwritten text or scanned digital documents are given as input to OCR system. In post office and other offices, OCR is used to sort the mail. Developing an efficient Offline handwritten Devanagari recognition system is the goal of many researchers. OCR helps to reduce people's work load. Numerous researches have been performed on English alphabets. A big part of population in India uses Devanagari script but very few researches on Devanagari is going on. So various techniques are analyzed to Devanagari for feature extraction and classification; one best combination of feature descriptor and classifier is taken. Basically author develops system for recognition of handwritten Devanagari numerals.

1.7 Objective of Thesis

Objectives of thesis are as follows:-

- Studying the previous researches in the field of character recognition.
- Finding drawbacks of previous researches.
- Thinking broadly for removing those drawbacks.
- Developing new techniques to eliminate the shortcomings.
- Making such systems which easily remove problems like noise, skewness and slant of images and do correct segmentation.

- Implementing new proposed algorithm.
- Finding the result of new proposed recognition system.
- Comparing the result of new technique with the results of previous techniques.
- Improving accuracy rate and efficiency of recognition system as compared to previous researches.
- There are many features extraction algorithms and many classifiers. Combination of feature extraction algorithm and classifier makes recognition system. So checking which combination is giving best result.

1.8 Organization of Thesis

Thesis is properly organized in a sequence. It starts with Introduction and historical background of character Recognition. Chapter 2 which is titled as literature review includes the description of many previous researches related with present study. Chapter 3 which is titled as Problem statement gives the problem statement belonging to recognition system. This chapter defines the problem. Chapter 4 shows the implementation of proposed methods and all methodologies which are used. Results of proposed work are shown, discussed and compared to previous works in chapter 5.Conclusion and future scope of research is discussed in Chapter 6.

Chapter 2

Literature Review

There are many handwritten recognition techniques that have been proposed in the past. Initially only template matching and various other simple algorithms had been introduced which were degrading the accuracy rate of recognition, so later many feature descriptors and classifiers were invented. The various recognition techniques have been described in this section.

A. Negi *et al.* [2001] proposed an OCR system for Telugu. Telugu is a south Indian language. There is a critical orthography with so many number of unique character shapes composed of compound and simple characters made up from 36 consonants(*known as hallus*) and 16 vowels(*known as achchus*) in Telugu language. They proposed a practical and efficient technique for Telugu OCR systems which reduced the total number of templates that can be recognized to only 370, instead of designing classifier for thousands of shapes. OCR accuracy rate of Compositional approach for connected component is about 92%. Many experiments for resolutions and varying fonts reported that this approach is good [8].

A. S. Britto *et al.* [2003] have proposed recognition process based on two-stage Hidden Markov Model (HMM). Implicit segmentation is performed in the first stage. Implicit segmentation compromises with difficulties of strings. A feature set is introduced into first stage. This stage performs right segmentation. Hypotheses is re-ranked and verified with the help of isolated digit classifier in the second stage of HMM. Verification is used for complementing String Contextual Based (SCB) stage. Given numerals are preprocessed in the SCB stage and measure the bounding box of string. HMM classifier is trained on separated digits [10].

U. Pal and B.B. Chaudhari [2004] gave a survey paper on character recognition of Indian scripts. They analyzed 12 Indian scripts. They gave review of OCR work performed on various Indian scripts and also shown the future work. There are 22 official languages in India. Mostly Indian scripts are originated from Brahmi by various modifications. One script is used to write many languages i.e. Marathi. Hindi, Rajasthani, Nepali and Sanskrit have been written using Devanagari scripts. In the

Indian languages, generally a line is broken into three zones i.e. upper zone, lower zone and middle zone. The lower/ upper case concept is not present in Indian scripts. Mostly Indian languages are written from left side to right side. Urdu is written from right side to left side. Urdu belongs to Arabic script. Pattern recognition is classified into feature based and template based approach. Initially, template based approach was used but in modern OCR, feature based approach has been used [11].

N. Dalal and B. Triggs [2005] did experiment on HOG features descriptor for human detection. They made dataset of 1800 human images with huge range of variations. Detecting humans in any image is a difficult task due to large range of poses. HOG representation has various advantages. It captures gradient structure and edges of local shape. It does with controllable degree of invariance for photometric and local geometric transformations. Before presenting performance and implementation analysis, they compared the performance of HOG detector with other existing methods. Detectors depended on R-HOG or C-HOG and kernel or linear SVM [12].

U. Bhattacharya *et al.* [2006] have proposed an algorithm which has two stages for classification in the recognition process of handwritten Devanagari numerals. They used ANN and HMM classifiers. Vertical and horizontal strokes are extracted from numerals and then shape feature vectors are made by these strokes. Shape feature is passed to HMM classifier as an input. This proposed method gives 92.83% accuracy rate [13].

K. Huang *et al.* [2007] proposed a SVM based method into two stage framework with the help of some features. In the first stage, Training data set is used to train regular SVM. Some data is neglected by SVM and it can't be classified. Only misclassified samples of first stage are considered in the second stage. So performance can be improved. Due to good Performance of SVM, number of misclassifications is very few. Training of SVM has become complicated because of misclassification. Further multi-way to binary approach is implemented with the help of different features. This approach is used to convert the multi-way classification into binary classification. An experiment is performed on samples of Devanagari numerals to evaluate this approach. Good accuracy is acquired by this approach [14].

M. Hanmandlu and O.V. R. Murthy [2007] proposed an algorithm for recognizing Hindi and English numerals using Fuzzy model. They used exponential membership functions. Recognition is performed by modifying these functions. Two structural parameters are used to modify membership function. These membership functions are fitted to Fuzzy sets. Fuzzy logic is used to recognize numerals. Fuzzy membership functions are made with the help of normalized vector distance in the form of features. They explained the making process of Fuzzy set with the help of a feature. They described the concept of fuzzy set in the following way. First, take 'm' samples of numeral and each numeral has 'n' possible features. Thus, specific feature of every sample makes the fuzzy sets. There is a little bit chance that all samples have same feature value. Variations of feature value increase the fuzziness. If all samples have same value for a particular feature then it makes a crisp set [15].

U. Pal *et al.* [2007] presented a recognition system for recognition of offline handwritten Devanagari characters. They used the features for recognition which basically depend upon the directional information. Directional information is measured with the help of arc tangent of gradient. Mean filtering is performed four times on grayscale image for feature extraction. Non linear normalization is applied on image. Segmentation has been applied on normalized image and then Robert filter was applied on segmented blocks. Modified quadratic classifier is used for recognition process. For finding accuracy, 5 fold cross validation technique was applied [16].

B. Shaw *et al.* [2008] have proposed handwritten Devanagari recognition system based on segmentation. Pseudo characters are found by segmentation of word image. Hidden Markov Models (HMM) technique is used for recognition of pseudo characters. String edit distance is used for word recognition. Accuracy rate is 81.63%. at pseudo characters level. Accuracy rate is 84.31% at word level [17].

U.Pal *et al.* [2009] discussed about the results of various classifiers and give an idea for future research. They provided comparative analysis of results by different classifier for recognition of offline handwritten Devanagari characters. Results of twelve different classifiers like Support vector machine, Mirror image learning, K-nearest neighbor, Projection Distance, Euclidean distance, Compound projection distance, Subspace method, nearest neighbor etc. are compared. After comparison, it

is found that Mirror image learning gave highest accurate results (95.19%) with gray scale curvature features [18].

A. N. Holambe *et al.* [2010] proposed an approach in which curvature and gradient based feature descriptor is used. Performance of K- Nearest neighbor, Reduced nearest neighbor, Euclidian distance based K-NN, Farthest like neighbor are compared. A 2×2 mean filtering is used to find gradient features of gray level image. Many classes are defined for two or more dissimilar points of different class labels. If no. of classes is two or more then K-NN is used. In K-NN, more than one neighbor can decide the class of a point x. Euclidian distance based K-NN is also implemented [20].

R. Jayadevan *et al.* [2011] gave a survey paper on recognition of Offline Devanagari Script. From the survey, it is clear that errors are generated in recognition of printed Devanagari characters due to wrong segmentation. While in the handwritten Devanagari characters, error is generated because of handwriting variations. Segmentation of touching characters is difficult. For proper recognition, proper segmentation is needed. Some researchers made efforts to build benchmark database to improve quality of OCR systems. A Devanagari script is used in writing various official languages such as Marathi, Nepali, Hindi, Sanskrit, Sindhi and Konkani. Various other languages such as Punjabi, Gujarati use scripts very much similar to Devanagari scripts. They tried to describe all advanced methods applied in recognition of handwritten as well as printed Devanagari script till 2010 [22].

R.S. Hegadi [2012] has proposed a method based on zoning concept to recognize printed Kannada numerals. Zoning is a famous technique for recognition in OCR. Tiny dots present in the document are removed. Row segmentation is followed by column segmentation so that each numeral can be segmented. Regions of segmented numerals are obtained, which are used for recognition process. Thinning operation is performed to thin the numerals. Total end points and their coordinate's values are obtained. The zone in which end points are present and the regions that every numeral generates, are used in the recognition process of numeral [24].

K. Bansal and R. Kumar [2013] proposed an algorithm which integrates water reservoir similarity based features and structural features to classify Devanagari numerals. To recognize any numerals, maximum four times checking is needed. They

applied hybrid scheme for character segmentation. Presences of features such as left side bar, right sidebar, loops, water reservoir have been used for classification. Features have two categories i.e. structural and statistical. Left sidebar, right side bar and loops belong to structural category while water reservoir belongs to statistical category. Further water reservoir is also divided into four categories i.e. left reservoir, right reservoir, top reservoir, bottom reservoir. When water is poured on one side of object then the part of object where water is stored, is known as reservoir. Left reservoir exists in Devanagari numerals  and  . Right reservoir exists inDevanagari numerals  ,  and  . Top reservoir exist in  ,  and  . Loop exists in  and  . Sidebar exist in and  . On the basis of these features classification of Devanagari numerals is done [26].

S. Iamsa-at and P. Horata [2013] presented a framework to investigate and compare the capability of two classifiers: Extream Learning Machine (ELM) and Deep Learning Feedforward Backpropagation Neural network (DFBNN). Datasets of Handwritten Bangla numerals, Devanagari numerals and Thai characters are prepared. Each dataset is categorized into two parts: Extracted and non extracted by HOG algorithm. Experimental results show that HOG feature extraction algorithm can increase the recognition accuracy rate of both classifiers ELM and DFBNN. Recognition rate of DFBNN is better than ELM classifier. Recognition rate of ELM and DFBNN with or without HOG is studied [27].

R. S. Hegadi and P.M. Kamble [2014] have presented a technique in which multilayer neural network is used to recognize handwritten Marathi numerals. Preprocessing is used to remove the noises of scanned document. Every numeral of document is segmented. Many morphological operations are applied such as dilation operation. All numerals are resized into same size. Each resized numeral is translated into vector which is passed to neural network. This technique gives less recognition rate for numeral 3. Overall Recognition rate is good [28].

P. M. Kamble and R. S. Hegadi [2015] applied R-HOG features descriptor for recognition of handwritten Marathi characters. They used Feed forward Artificial Neural Network (FFANN) and SVM for classification. They obtained high accuracy rate with FFANN. Some simple arithmetic operations are performed on each image pixel in this algorithm to make them appropriate for the real time applications. They

normalized all handwritten samples of Marathi characters into 20 ×20 pixel size. At first the training sets of Marathi character are collected. For training, R-HOG features of each Marathi character are taken and values of each character are computed. Again, all these features of testing samples were also calculated and measured to know how many features of training samples and testing samples are matching. These classifications are done by FFANN and SVM. Performance of FFANN is better as compared to SVM [29].

From this vast study of literature survey, it is found that many different types of algorithm were given by many researchers to develop an efficient and accurate recognition system. But there are implementation gaps and none of the algorithm produces satisfactorily results for all kinds of recognition systems. Hence, it is required to fill these gaps. Therefore, a study has been proposed where an efficient algorithm was developed and implemented which can recognize handwritten Devanagari numerals.

Chapter 3

Problem Statement

In this chapter, gaps which are existing in the current work, problem statement and method for filling these gaps or obtaining the objectives of research have been discussed**.**

3.1 Gap Analysis

In the literature review, we studied and discussed previous research works and applied methods for character recognition process. Many researchers applied various approaches to improve the accuracy rate. They used many combinations of classifiers and feature descriptors for recognition process. Many gaps were found in the previous works which are as follows:

1) Preprocessing and segmentation are two important phases of recognition system. If there is any problem in these phases then recognition would be reduced. So a method is required which performs preprocessing and segmentation work properly.
2) Now handwritten documents having same contents but written by different owners will be different because of the variations in the size and shape of characters present in the document. So there is a need of method which handles the variation problem.
3) A lot of research work is available for English alphabets as well as numerals but very few articles are available for Devanagari numerals present in handwritten documents. So a method is required to recognize Devanagari numerals in efficient and accurate way.
4) Features are not extracted properly some times. So there is a need to develop a system to extract the features of handwritten Devanagari numerals present in handwritten document which gives Devanagari numerals in very efficient and accurate way.

Thus, the above gaps defined the objectives of this thesis which are:

- To attain a training dataset of handwritten Devanagari numerals, extract their features and classify them into ten classes.

- To develop an algorithm to remove variation problem in handwritten numerals and improve the accuracy rate of recognition system.
- To test the algorithm on the testing dataset and analyze if the results are desirable or not.
- To compare the algorithm proposed and compare the results with the existing algorithms.

3.2 Problem Statement

According to literature Review of offline handwritten character recognition, only one question arises: which method is best for character Recognition process? Which combination of feature descriptor and classifier should be taken for best result? These questions lead us to work in the field of character recognition process. We took many available algorithms to characterize so that best algorithms can be marked for recognition. Experimental results are needed to compare the performance of different algorithms. On the base of literature review, It can be concluded that there is no researches on the combination of HOG feature extraction algorithm with K-NN Classifier for recognition of Devanagari or Hindi numerals. K-NN classifier as a recognizer achieves more popularity and obtained results are fantastic. Thus, in the current study, using HOG algorithm as a feature descriptor and K- Nearest Neighbor (K-NN) as Classifier is good idea. A hybrid approach, using these two, is designed. Recognition of Devanagari numerals of various shapes is the main target of present research. Preprocessing, segmentation and feature extraction are added with recognition process. Recognition rate totally depends on these stages and present study has concentrated on these stages along with recognition. If more features are extracted, then recognition will be easy and accurate.

Chapter 4

Problem Solution

"Handwriting variations" is one the most common problems that occurs in handwritten documents with time. It affects the readability of the document and the recognition of text by OCR. Therefore, the need arises to develop an algorithm that removes any interference caused by the ink which seeps through the paper and attain readability and recognition of text by OCR. The proposed solution to improve recognition rate has been described in four sections below.

4.1 Overview of Problem Solution

Today, many researchers are working on recognition of Devanagari characters. There is a lot of work done in recognition field but still researches are going on for improving efficiency and accuracy. Developing an efficient handwritten Devanagari character recognition (HDCR) system is the goal of many researches. Basically author has developed a hybrid approach for recognition of handwritten Devanagari numeral recognition (HDNR). Thus find out the idea by which human identify the Hindi numerals in general. Making a recognition system is a tough job because of variations in different people's handwriting and imperfections like noise, angle, loops, and alignment in handwritten Devanagari numerals. Removing these imperfections is not an easy task. So divide this problem into sub problems. This means that entire process cannot be done properly as a single process. This work is performed by many sub-processes and their combination gives required result. A series of sub-processes like document scanning, preprocessing, segmentation, feature extraction etc. are done for recognition system. Output of first process is used as input to second process; output of second process is used as input to third process, and so on.

For example, document scanning is the first sub process and output of this is used as input in the next sub process i.e. pre-processing which performs noise removal, skew detection, normalization etc. Output of pre-processing is used as input in segmentation and then output of segmentation is used in feature extraction process. Output of feature extraction is used by K-NN Classifier in the classification phase and finally produces the final result means recognize the numerals. So Recognition system is a combination of a series of processes. HOG is very simple algorithm. Here HOG is

used for extracting features of training set images. These features are stored into a database file and grouped into ten classes. These features are used to train K-NN Classifier. Proposed HDNR system is based on a learning process in which proposed technique takes input from the user. Training and testing of Devanagari numerals is performed in the learning process.

4.2 Handwritten Devanagari Numeral Recognition (HDNR)

Handwritten Devanagari Character Recognition (HDCR) is a part of Pattern Recognition that has become an interesting field of research from last few decades but very few researchers gave concentration on Devanagari numerals. A lot of applications of Hindi language are in government offices like income-tax, railway, banks etc. Many application forms have options for filling the form either in English or in Hindi because many people don't know English and those forms are scanned by scanner but that images can't be edited without HDCR system. We proposed handwritten Devanagari Numerals Recognition (HDNR). The main goal of this HDNR system is to identify Devanagari numerals from given digital images, without human interruption. Offline HDNR system accepts input image from user and performs pre-processing, feature extraction and then classifies the Devanagari numerals.

4.3 Architecture of HDNR

An HDNR system take an image as input, performs pre-processing on image, segmenting all numerals which are present in that image, extracts the features of each numeral, classifies the numerals based on comparison of extracted features and stored features of database file(library image model) . HDNR system recognizes the numerals on behalf of similarities of features between library image models and loaded image. Noises of image are removed in pre-processing so that recognition work becomes easy. Pre-processing stage also includes morphological operation, normalization, gap filling, skew detection and so on, which makes loaded image suitable for further computation. Figure 4.1 show the architecture of HDNR system and figure 4.2 shows the recognition process.

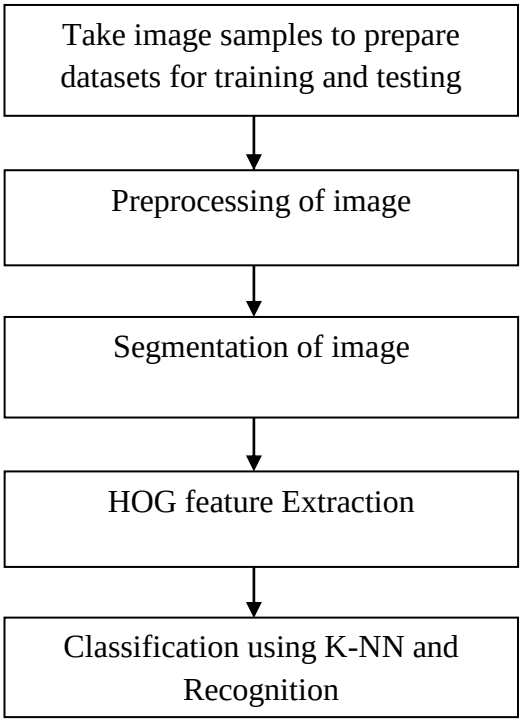

Figure 4.1: Architecture of proposed algorithm

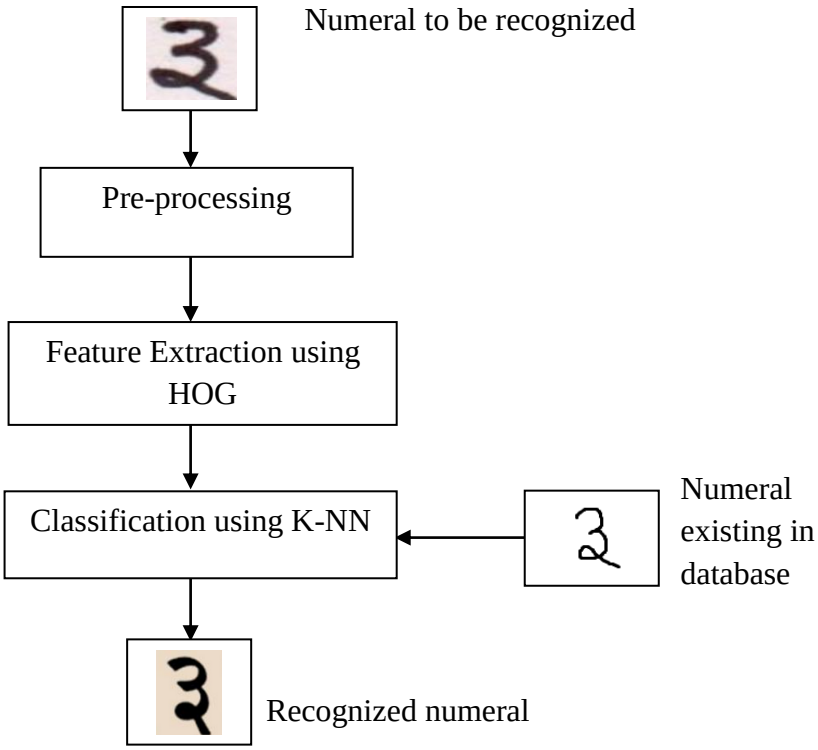

Figure 4.2: Recognition of single Devanagari Numeral

The architecture of proposed algorithm has been described in four sections given as follows:-

4.3.1 Preprocessing

Preprocessing phase is mainly used to filter the given input image along with other processes. This phase gives suitable result for further phases like segmentation. In the proposed algorithm many operations are performed in the Preprocessing such as RGB to grayscale conversion, morphological operations, normalization, edge detection, Gray to binary conversion, inversion etc.

4.3.2 Segmentation

Segmentation which is used to break the whole page into paragraphs and lines that is known as external segmentation. When each line is broken into words and characters, that is known as internal segmentation [2]. In proposed solution, all numerals present in given input image are separated on the basis of bounding box of image and then simply all numerals are cropped from image. Only the area which comes under the bounding box of each numeral is cropped and each segmented numeral is considered as a separate object and stored in an array.

4.3.3 Feature Extraction

Classification works become easy if suitable features are extracted. There are two types of features named as low level features and high level features. Some neighboring points make low-level features. High level features provide the information about loops, straight line, crossings, dots etc. There are many methods for feature extraction. In proposed algorithm, HOG feature extraction algorithm is used to extract the features of each numeral present in images of training sets and testing sets. Features of training set's images are stored in the database. Features are very important for classification. HOG creates HOG feature vector for a given image which is further used to detect any particular object. Stored features are grouped into ten categories: zero, one, two, three, four, five, six, seven, eight, and nine. A training data set of 3000 images in which 300 images for each Devanagari Hindi numeral is made.

HOG Algorithm

HOG stands for Histogram of Oriented Gradients. It is a feature descriptor used in image processing and computer vision for object detection. This method counts occurrences of the gradient orientation in the nearest portions of the image. It makes classification work easier. The basics of algorithm used for feature extraction. At First, compute the gradient values. Commonly used technique is to apply 1D centered point discrete derivative mask in vertical and horizontal direction.

$D_x = [-1 \;\; 0 \;\; 1]$ and $D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$

An image I is given. We calculate x and y derivatives using convolution operations:

$I_x = I * D_x$ and $I_y = I * D_y$

Magnitude of gradient, $|G| = \sqrt{I_x^2 + I_y^2}$

Orientation of gradient is $\theta = \arctan \frac{I_y}{I_x}$

Where * is a convolution operator.

For orientation, use the function atan2 that gives value in the interval (-π, π]. For a pixel, orientation of gradient is θ = atan2 (I_y , I_x) radians.

Transformed angle into degree is α= θ*180/ π, that give values in range (-180, 180] degree. For signed gradient, we translate the range of gradient from (-180,180] to [0,360) degree. It is done using formula:

$$\alpha_{signed} = \begin{cases} \alpha, & if\ \alpha \geq 0 \\ \alpha + 360, & if\ \alpha < 0 \end{cases}$$

4.3.4 Classification and Recognition

For classification, author used K-NN classifier. K-NN is trained with the stored features of database file and each image of single numeral is labeled into one of the 10

categories: zero, one, two, three, four, five, six, seven, eight, and nine. Author has taken some sample images for each Devanagari numeral to make testing set. Features of testing samples are extracted and matched with stored features. Output of the K-NN classifier is one of the classes among ten classes as mentioned above.

K-NN Classifier

KNN is a classifier which works like human brain. K-NN algorithm is simple in comparison to other machine learning algorithms. K-NN algorithm is a type of instance based learning. Output is a class membership in K-NN algorithm. Any object is classified by the voting of its neighbor. K-NN algorithm gives higher accuracy. K-NN Classifier stores all available cases and classifies new cases based on distance function.

If k=1, then the case is simply allotted to its nearest neighbor.

Distance functions:

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhaltan $\sum_{i=1}^{k}|x_i - y_i|$

Hamming distance $D_H = \sum_{i=1}^{k}|x_i - y_i|$

$x = y \rightarrow D = 0$

$x \neq y \rightarrow D = 1$

Taking large value of k gives more precise result because it reduces the overall noises. Mostly we take values of k between 3 to 10 for datasets which produces more accurate result for 1 nearest neighbor.

4.4 Algorithm

Training module is used first before running the main program. Training module takes training sets of ‘m’ images. Extract the features of training sets and store them into SF database file. K-NN classifier is trained with these stored features in the Training module. Database file SF is loaded into main program. Main program uses file SF in Check_class module.

Main Program: Proposed Algorithm

/* SF is stored feature */

1. Take an image and store in I_m.
2. M_1= Clear_image (I_m).
3. A_1= Segment_numeral (M_1, I_m).

   /*A_1 is array in which all numerals are stored in the binary image format.*/
4. For each image (T_i) in array A_1.

   Result=Check_class (T_i , Stored feature SF)

   If Result = 0

   Then display '0'

   Else if result = 1

   Then display '1'

   Else if result = 2

   Then display '2'

   Else if result = 3

   Then display '3'

   Else if result = 4

   Then display '4'

   Else if result = 5

   Then display '5'

   Else if result = 6

   Then display '6'

   Else if result = 7

   Then display '7'

   Else if result = 8

   Then display '8'

   Else display = '9'

   End if.

   End for.

Module 1: Clear_image (Input image: I_m)

1. Check that input image I_m is RGB or Gray scale. If image is RGB then convert into grayscale.
2. Detecting edge of characters using 'canny' edge detector with automatic threshold calculation.
3. Morphological dilation operation has been applied.
4. Filling of image is performed if necessary.
5. Remove unnecessary element.
6. Labeling all connected components in the image.
7. Image properties are measured such as area, centroid and bounding box and store in M_1.
8. Return M_1.

Above algorithm takes image as input and convert that image into gray scale image. Perform morphological operation on that image to remove imperfection. Remove all noises and unnecessary things. All connected components in image are labelled. Image properties are measured and stored in M_1 and return M_1 which will be used for segmentation.

Module 2: Segment_numeral (M_1, I_m)

1. Crop all present numerals in image I_m using M_1 and store each cropped images in array A.
2. For each image in array A
   $A_1[i]$= convert into binary image and then invert the image
   End for
3. Return array A_1

Above segmentation algorithm take image properties such as Area, bounding box and centroid as input. Crop each object (numerals) from image with the help of these properties and store each cropped image into array A. All images of array A are converted into binary image and then inverted. All these inverted images are stored in array A_1 and return A_1 which are further used for Feature Extraction and classification phase in the module Check_class.

Module 3: Training

1. Take a training set TS of ‘m’ images.
2. Extract the features of this training set using HOG feature extraction algorithm and store them in database file SF.
3. Group the images of numerals into ten categories.
4. Trained K-NN with these stored features.

Above algorithm takes ‘m’ images for training set and then extract features of those images. Features are stored into database file SF. All features are grouped into ten categories manually. K-NN is trained with these features.

Module 4: Check_class (Segmented numeral image T_i, Stored feature SF)

/* T_i is a segmented numeral of sample image and SF is matrix of extracted feature which are stored in database */

1. Store T_i in a temporary variable T.
2. Extract the features (say F_i) of T_i.
3. Compare F_i with SF.
4. A class, whose maximum features match with testing samples, is assigned to it.
5. Return class.

Above algorithm takes stored features and sample image as input. Extract the features of sample image and store in F_i. Compare F_i with SF and the class to which maximum features belong is return as output.

Chapter 5
Result and Discussion

5.1 Results of Proposed Algorithm

The proposed hybrid algorithm which is described in chapter 4 is implemented in MATLAB R2013a version. Author has used HOG feature extraction algorithm and K-NN Classifier to make a hybrid approach for handwritten Devanagari numerals recognition (HDNR). Author has taken 3000 images of Hindi numerals to make a training set. HOG feature extraction algorithm is used to extract the features of training set images. These extracted features are stored in database file. These features are grouped into ten classes: zero, one two, three, four, five, six, seven, eight and nine. K-NN classifier has been trained with the help of these extracted features. Author takes 1000 sample images to make testing set. When author gives any testing image to HDNR system, then features of that image are extracted. K-NN classifier takes that features and matches with the stored features. The class to which maximum features belong is returned as output.

Characters	Number of Samples	Train/Test	Accuracy %
	400	300/100	99
	400	300/100	99
	400	300/100	98
	400	300/100	97
	400	300/100	96
	400	300/100	80
	400	300/100	97
	400	300/100	99
	400	300/100	85
	400	300/100	89

Table 5.1: Accuracy rate of Devanagari numerals

Proposed algorithm gives good accuracy rate for each numeral which is shown in table 5.1.

Results

Module ‘training’ takes 3000 images for training set and then extracts features of those images. Features are stored into database file SF. All features are grouped into ten categories: zero, one two, three, four, five, six, seven, eight and nine. K-NN has been trained with these features. Database file SF is loaded in the implementation of main program and it is used by module ‘Check_class’.

When the program is run then an input window is opened to select the image. Author selects an input image for recognition process. Figure 5.1 shows selection process. After selecting the image program is executed for recognition.

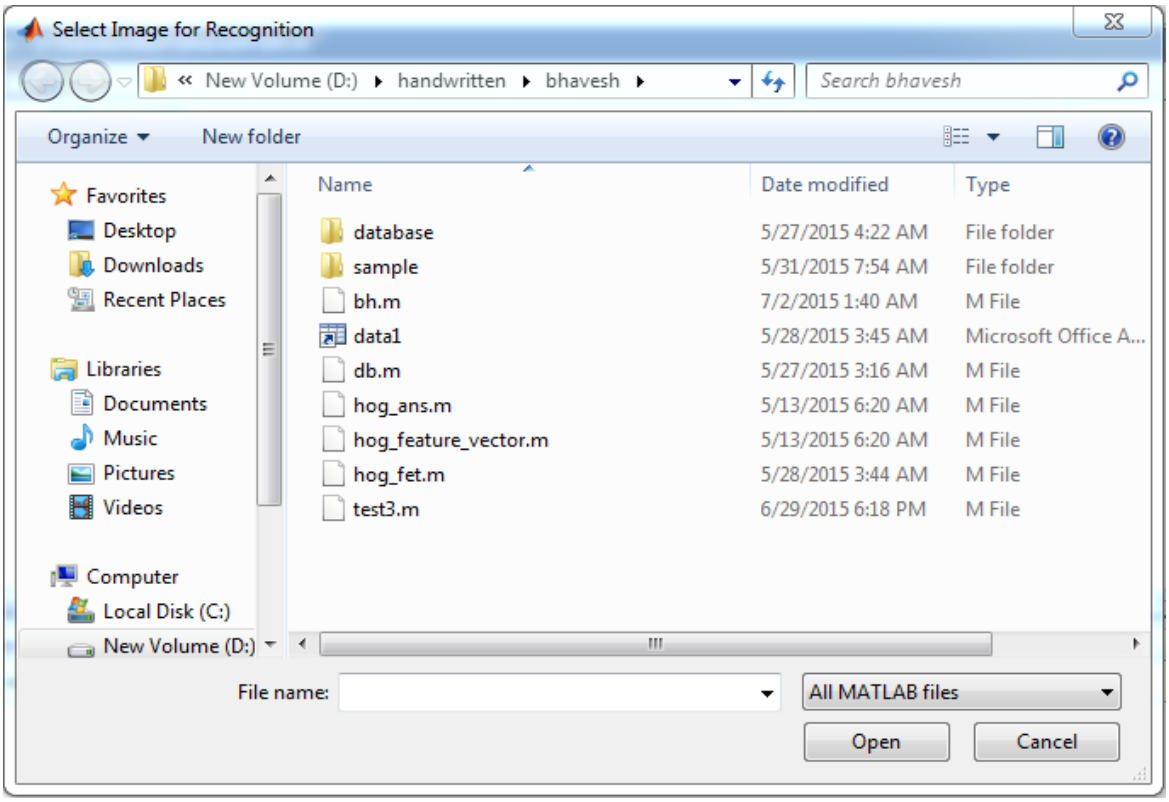

Figure 5.1: input window for selecting the input image

Figure 5.2 shows the selected original image which is read by main program. All recognition steps are performed on input image step by step. When main program is executed, all modules perform their work in a proper sequence. Input image is passed to first module. First module performs its work on input image. Then output of first module is passed as input to second module. Output of second module is passed as input to third module and so on.

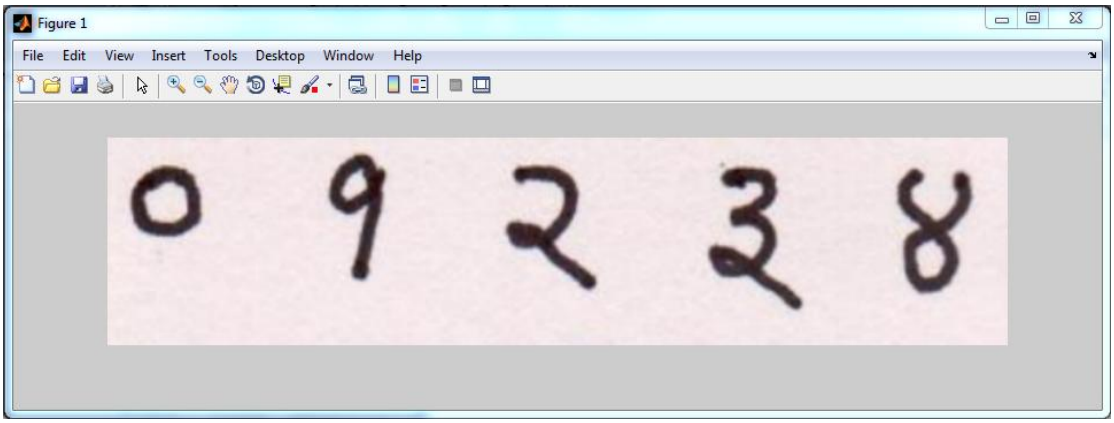

Figure 5.2: Original input image

First module ‘clear_image’ checks that input image is RGB or Grayscale. If image is RGB, then it is converted into grayscale. Input image is converted into gray scale image. Figure 5.3 shows gray scale image. Further this gray scale image is used in all morphological operations.

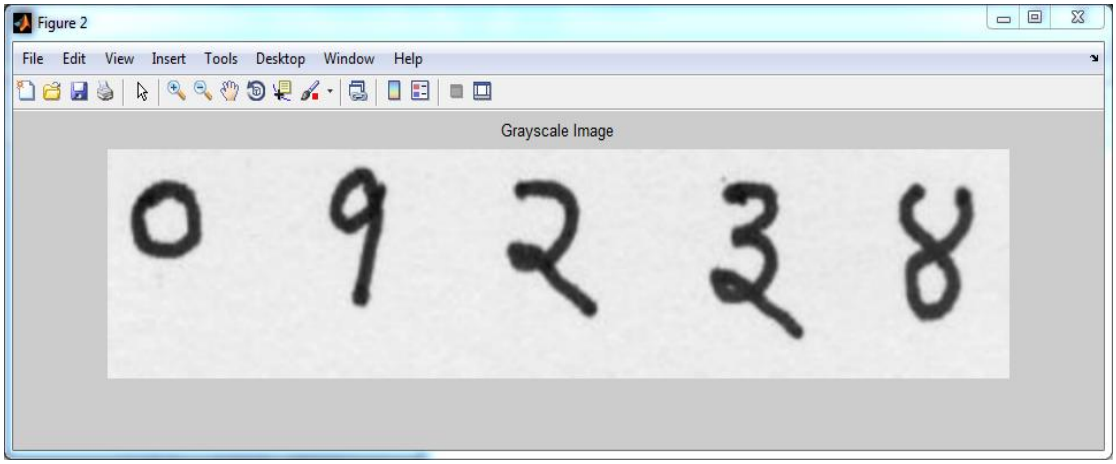

Figure 5.3: Gray scale image

Module ‘clear_image’ detects the Edges of all numerals which are present in gray scale image. Figure 5.4 shows edge detected image. Further morphological operations will be performed on this edge detected image.

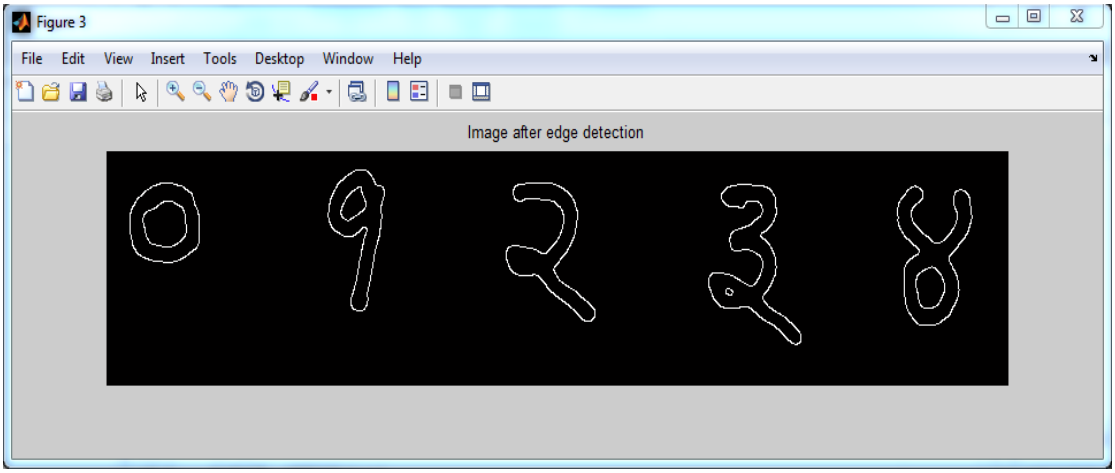

Figure 5.4: Image after edge detection

For finding good accuracy of recognition, morphological operations are performed to remove noises and other imperfections. If pixel intensity is thin, then that will be correct in morphological operation. Module ‘clear_image’ performs morphological operation which is shown in figure 5.5. This image is used as input for further operations like filling operation.

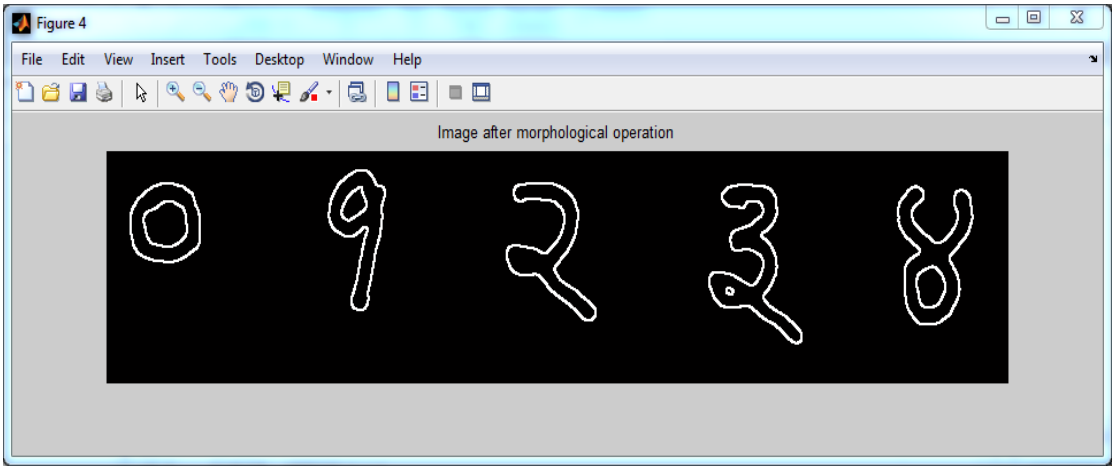

Figure 5.5: Image after morphological operation

If holes are present in the image, then that area are filled by filling operation. To make the recognition process more efficient, filling operation is done. Module ‘clear_image’ performs it. It is shown in figure 5.6.

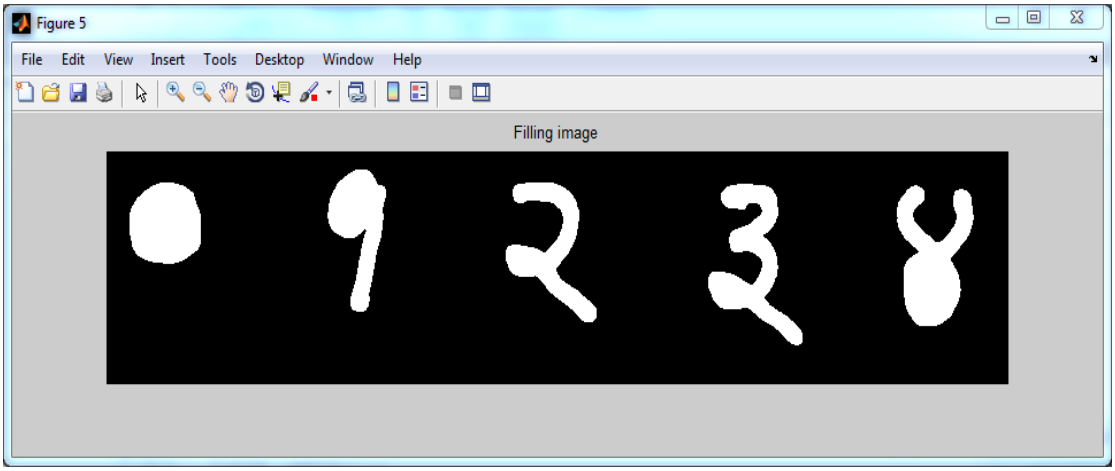

Figure 5.6: Filling image

Output of first module ’clear_image’ is passed as input to second module ‘segment_numeral’ and module ‘segment_numeral’ performs segmentation process. All numerals are considered as an object and area of these numerals is bounded using bounding box properties and these bounded areas are cropped in segmentation stage. Bounding box is shown in Figure 5.7.

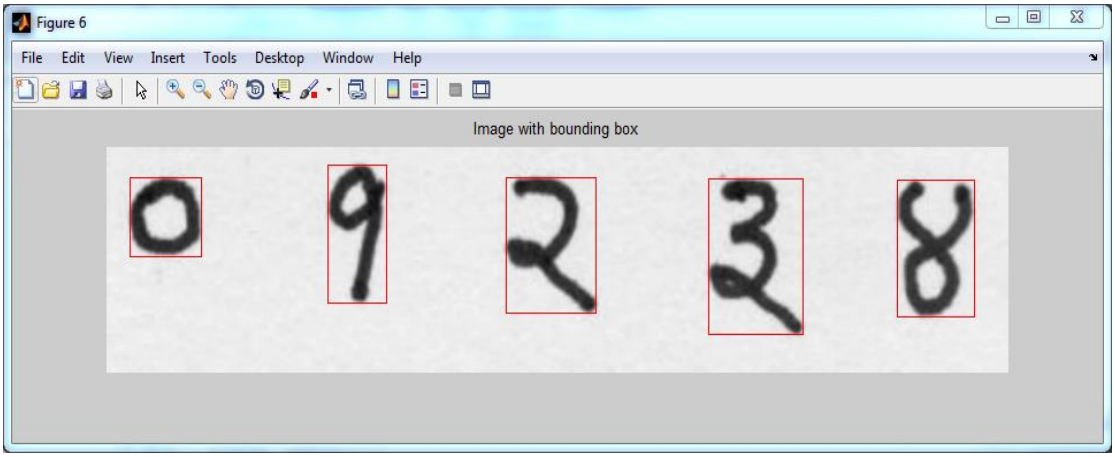

Figure 5.7: Image showing bounding box

Figure 5.8 shows all separated numerals. These separated numerals are stored in an array. This array is passed as input to the module 'Check_class'. These separated numerals are taken one by one and features of these numerals are extracted and compared with stored features of database file to decide the class of numerals.

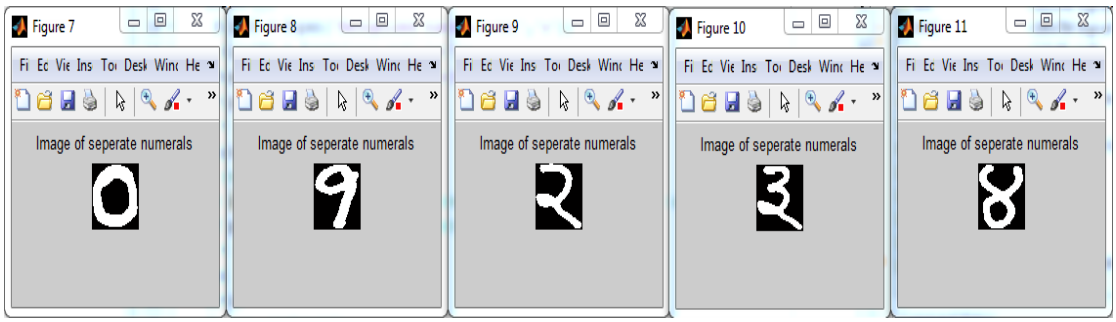

Figure 5.8: Segmented numerals

In module 'Check_class' features of separated numerals are extracted using HOG algorithm. K-NN classifier is used to match these features with stored feature and finally decides the class of each numeral. All recognized numerals are shown in a notepad file as shown as fig 5.9.

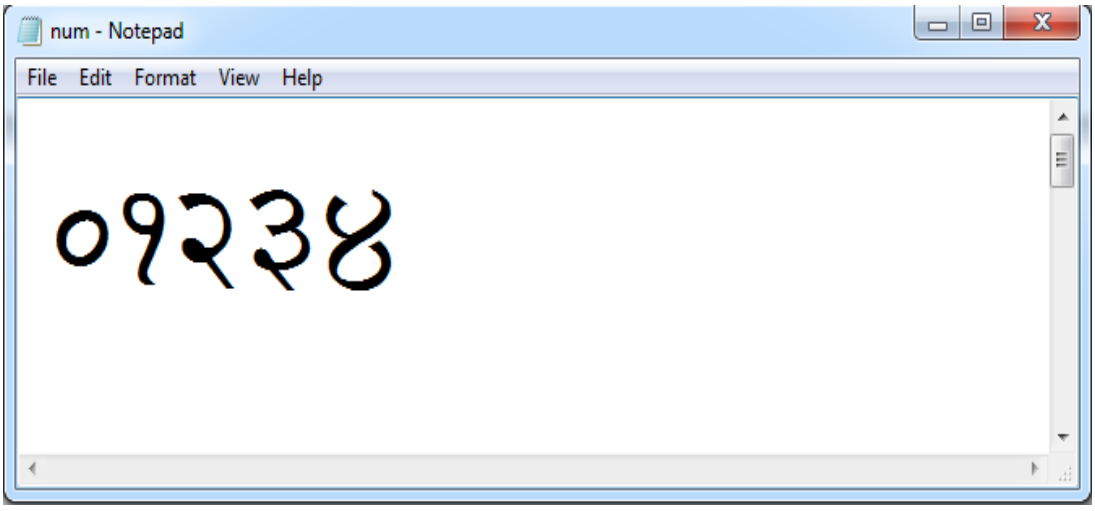

Figure 5.9: Recognized numerals

5.2 Results on other images

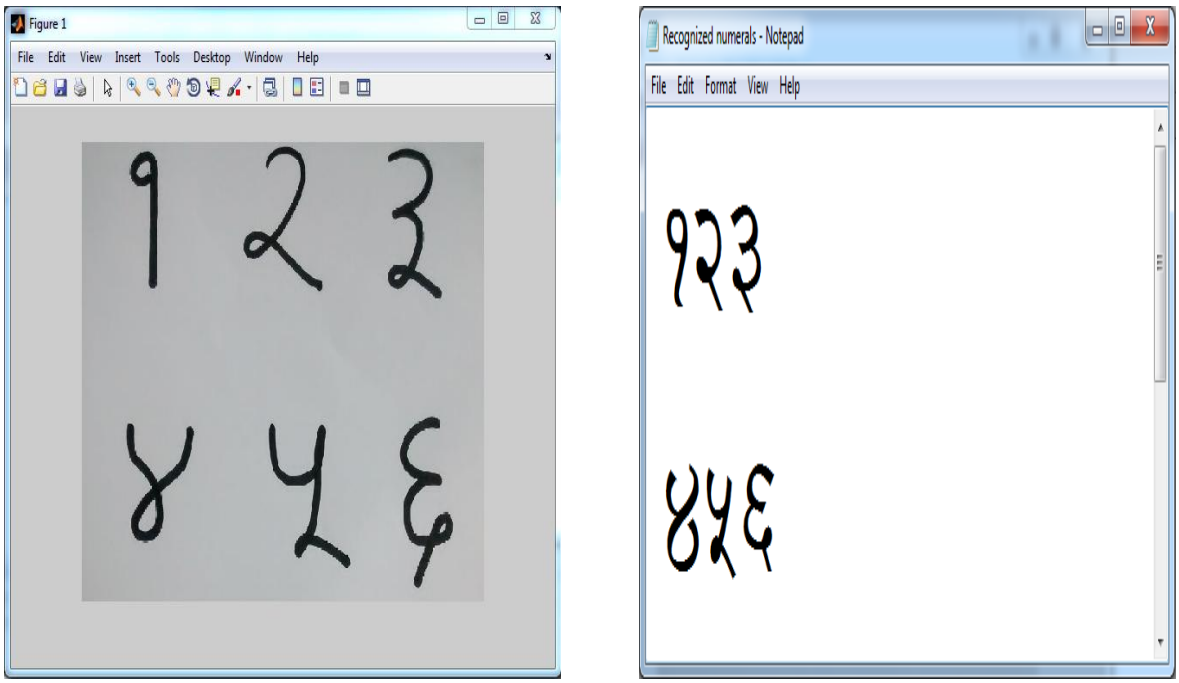

Figure 5.10: (a) Original image **(b) Result from proposed approach**

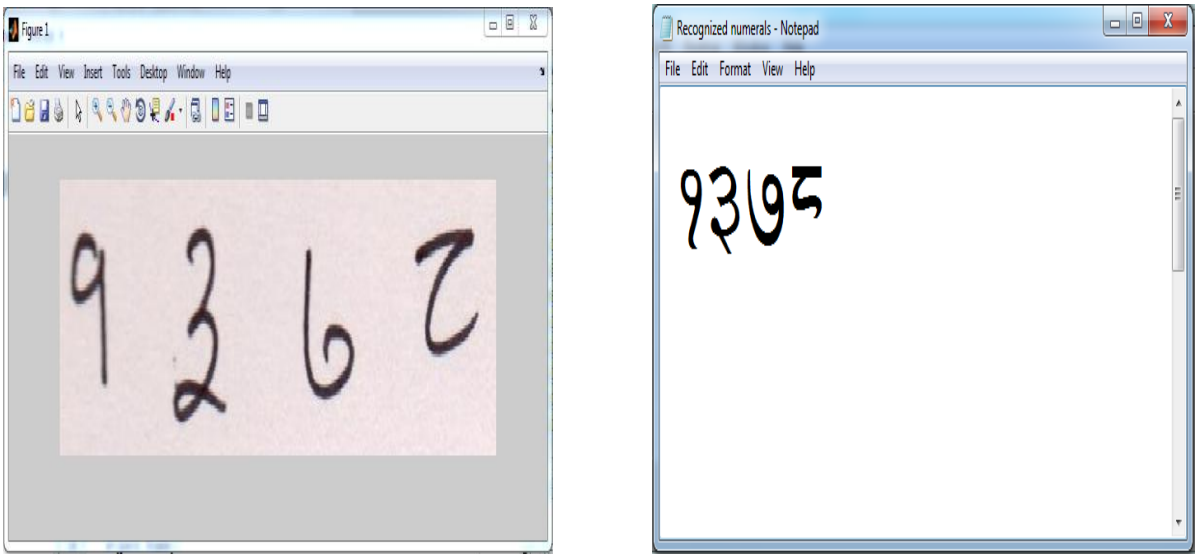

Figure 5.11: (a) Original image **(b) Result from proposed approach**

Here, proposed hybrid approach gives better accuracy rate when it is compared to other previous approaches discussed in the literature review. Generally, many researchers have used HOG algorithm with SVM, neural network and other

classifiers. But here, author has used HOG algorithm with K-NN classifier which gives better results as compared to others. Proposed algorithm gives 93.9% accuracy rate.

Chapter6

Conclusion and Future Scope

6.1 Conclusion

OCR has become an interesting research field since last few years. Developing an efficient recognition system for handwritten Devanagari numerals is the main motive of this research. Offline handwritten Devanagari numeral recognition system using HOG feature descriptor and K-NN classifier is discussed in this research. Recognition process of handwritten Devanagari numerals is a difficult job because of variations in handwriting. In Devanagari, some numerals' shapes are very much similar. Writing of different writers or even same writers may change at different time. Variations in writing depend on many things such as pencil or pen, quality of papers, stress and mood of writer, width of line etc. It is possible that same writer's writing style and font size may change when he or she writes at many places. Different author or even same author may write differently at different time. Other researchers used HOG and SVM combination but accuracy rate was low. Some researchers used many feature descriptors with neural network but accuracy was improved a bit. In the present study, author used combination of HOG feature descriptor with K-NN classifier, results are improved in a better way. Total number of 3000 images of numerals is taken to make training set and 1000 images are taken to make testing set. Finally it is concluded that if feature is extracted by HOG and used to train the K-NN classifier, then recognition accuracy rate will improve surely. Here mostly misclassification is obtained in case of the recognition of 5, 8 and 9.

6.2 Future scope

Many methods to recognize characters are invented and described by many researchers since last three decades. These approaches have been implemented and tested by the researchers. No recognition system in the world has accuracy 100% till date. Accuracy rate of proposed hybrid approach can be improved in future. Field of character recognition is very big. This approach is developed only for recognition of Devanagari numerals. Further it can be expanded for recognition of all Devanagari characters and words also. This research also can be further expanded in the field of

many other regional languages such as Gurumukhi, Marathi, and Bengali etc. Here proposed hybrid approach is developed only for offline character recognition.

This research also can be done on touching characters recognition or touching numerals recognition. So, there is a lot of works which can be further added in this research work.

References

[1] S. Mori, C. Y. Suen and K. Yamamoto, "Historical review of OCR research and development," *In the Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029-1058, 1992.

[2] Y. Lu, "Machine Printed Character Segmentation- An Overview," *Pattern recognition*, vol. 28, pp. 67-80, 1995.

[3] J. C. Yen, F. J. Chang and S. Chang, "A New Criterion for Automatic Multilevel Thresholding," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370-378, 1995.

[4] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690-706, July 1996.

[5] A. W. Senior and A. J. Robinson, "An Off-line Cursive Handwriting Recognition System," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 309-312, 1998.

[6] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition : a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63-84, 2000.

[7] N. Arica and F. Yarman, "An Overview of Character Recognition focused on Off-line Handwriting," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 31, no. 2, pp. 216-233, 2001.

[8] A. Negi, C. Bhagvati and B. Krishna, "An OCR system for Telugu," *In the Proceedings of the 6th International Conference on Document Processing*, pp. 1110-1114, 2001.

[9] R. Bajaj, L. Dey and S. Chaudhury, " Devnagari numeral recognition by combining decision of multiple connectionist classifiers, " *Sadhana*, vol. 27, no. 1, pp. 59-72, 2002.

[10] A. S. Britto, R. Sabourin, F. Bortolozzi, C. Y. Suen, " The recognition of handwritten numerals strings using a two-stage HMM based method," *International Journal on Document Analysis and Recognition*, pp. 102-117, 2003.

[11] U. Pal and B. B. Chaudhuri, "Indian script character recognition: A Survey," *Pattern Recognition*, vol. 37, pp. 1887-1899, 2004.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection, " *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. 886-893, 2005.

[13] U. Bhattacharya, S.K. Parui, B. Shaw and K. Bhattacharya, "Neural combination of ANN and HMM for handwritten Devnagari Numeral Recognition," *In 10th International Workshop on Frontiers in Handwriting Recognition,* pp. 613-618, 2006.

[14] K. Huang, J. Sun, Y. Hotta, K. Fujimoto and S. Naoi, "An SVM-Based High-accurate Recognition Approach for Handwritten Numerals by Using Difference Features," *9th International Conference on Document Analysis and Recognition(ICDAR 2007),* vol. 2, pp. 589-593, Sept. 2007.

[15] M. Hanmandlu and O. V. R. Murthy, "Fuzzy Model Based Recognition of Handwritten Hindi Numerals," *Pattern Recognition,* vol. 40, no. 6, pp. 1840-1854, June 2007.

[16] U. Pal, N. Sharma, T. Wakabayashi and F. Kimura, "Off-Line Handwritten Character Recognition of Devnagari Script," *9th International Conference on Document Analysis and Recognition (ICDAR 2007),* vol. 1, pp. 496-500, Sept. 2007.

[17] B. Shaw, S. K. Parui, and M. Shridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," *In Proc. IEEE International Conference on Information Technology*, pp. 256-257, 2008.

[18] U. Pal, T. Wakabayashi and F. Kimura, "Comparative Study of Devnagari Handwritten Character Recognition Using Different Feature and Classifiers," *10th International Conference on Document Analysis and Recognition (ICDAR2009),* pp. 1111-1115, July 2009.

[19] S. Kumar, "Performance comparison of features on Devanagari handprinted dataset," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 33-37, 2009.

[20] A. N. Holambe, S. N. Holambe and R. C. Thool, "Comparative study of Devanagari handwritten and printed character & numerals recognition using

Nearest-Neighbor classifiers," *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, vol. 1, pp. 426-430, July 2010.

[21] A Goyal, K. Khandelwal, and P. Keshri, "Optical Character Recognition for Handwritten Hindi," *CS229 Machine Learning*, pp. 1-5, 2010.

[22] R. Jayadevan, S.R. Kolhe, P. M. Patil, and U. Pal, "Offline Recognition of Devanagari Script: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 6, pp. 782-796, Nov. 2011.

[23] G. Katiyar and S. Mehfuz, "Evolutionary Computing Techniques in Off-Line Handwritten Character Recognition: A Review," *UACEE International Journal of Computer Science and its Applications*, vol. 2, pp. 133-137, 2012.

[24] R. S. Hegadi, "Recognition of printed Kannada numerals based on zoning method," *International Journal of Computer Application*, pp. 5–8, 2012.

[25] S. Panwar and N. Nain, "A Novel Approach of Skew Normalization Text Lines and Words," *8th International Conference on Signal Image Technology and Internet Based Systems*, pp. 296-299, Nov. 2012.

[26] K. Bansal, R. Kumar, "Devnagari numerals classification andRecognition using an integrated approach", *International Journal in Foundations of Computer Science & Technology (IJFCST)*, vol. 3, no.4, pp. 61-66, July 2013.

[27] S. Iamsa-at and P. Horata, "Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network," *International Conference on IT Convergence and Security (ICITCS)*, , pp.1-5, 2013.

[28] R. S. Hegadi and P. M. Kamble, "Recognition of marathi handwritten numerals using multi-layer feed-forward neural network," *World Congress on Computing and Communication Technologies (WCCCT), IEEE*, pp. 21–24, 2014.

[29] P. M. Kamble and R. S. Hegadi, "Handwritten Marathi character recognition using R-HOG Feature," *International Conference on Advanced Computing Technologies and Applications (ICACTA-2015)*, vol. 45, pp. 266-274, 2015.

Video Presentation

YouTube Link : https://www.youtube.com/watch?v=LpI2zUm0Cvc