**Chapter 1**

**Introduction**

* 1. **Background Study**

This project aims to propose work towards automatic recognition, extraction and translation of Kannada language natural sign board characters. In particular, it focuses on recognizing every single character in textual images. Figure 1, 2 and 3 apotheosizes the reason for this to be a flinty job. Even after ignoring the difficulty of text partitioning momentarily, the sources which brings problems are as follows:

1. font style and font size
2. graphics in background and foreground
3. camera alignment yielding to geometric distortion
4. illumination
5. resolution of image
6. removal of unwanted objects
7. identifying threshold value for different images and
8. edge detection

The above listed features give rise to the problem of object recognition. Henceforth, this technique cannot be used. Moreover, within the realm of possibility of such systems, OCR systems have been designed and developed for very less languages. Recognition of all languages of Indic origin is beyond their capability.

-----------------------------------------------------------------------------------------------------------

Various problems are to be solved so as to read Kannada texts from natural scene images in conjunction with finding text, identification of characters, partitioning of words. We try to highlight via this paper to recognize characters; documenting recognized characters into editable file format and lastly translating the phrase/ sentence recognized to another language (say English here). A standard database comprising of images/ characters of Kannada language is introduced. To measure the practicability of presenting the work as Kannada Character recognition, we establish a paradigm to measure the effectiveness of different attribute based on a standard data set as described above. The result delineates that indeed the confinement of character acknowledgment assignment is a dull work. The number of classes for Kannada characters is almost 619 counting consonants, vowels, digits and other images utilized with exceptionally small inter-class disparity as finished by Figures 2 and 3. This issue is eminently exceptionally ghastly for Kannada dialect where two typical characters in the letter set can contradict fair by substitution of a single speck, hyphen, bar, comma, accentuation like structure as appeared in Figure 4. The distinguished characters will be diverted to an editable record arrange. These writings can be afterward utilized for different purposes as required. The recognized characters can be archived in Latin script.

**1.2 Overview**

Right from the dawn of human civilization, people have started migrating from places to places for various reasons. Our Country; India being a nation with diverse culture and languages, people on moving away from their native may find it very much difficult to understand different languages. Let us say a non-native speaker of Kannada language comes to the state of Karnataka may not be able to read the Kannada script. So, we propose a system that reads the text encountered in natural scenes with the aim to provide assistance to the non-native speakers of Kannada language.

The area of scene text recognition focuses on the problem of recognizing arbitrary text in images of natural scenes. Examples of scene text may include street signs, name of shops, grocery item labels, and name plates etc. With the increased use of smart phones and digital cameras, the ability to accurately recognize text in images is becoming increasingly useful and many people will be benefited from advances in this area.

**1.3 Problem Statement**

This project presents work towards reading Kannada text from natural scenes. After recognizing text, it will be converted into editable format of Kannada text. After recognizing the text, it will be converted into any other desirable language. For now we focus to convert the text into English language.

**1.4 Introduction to OCR**

Optical character acknowledgment (moreover optical character recognition, OCR) is the mechanical or electronic transformation of pictures of written, manually written or printed content into machine-encoded content, whether from a checked report, a photo of a report, a scene-photo (for illustration the content on signs and bulletins in a scene photo) or from subtitle content superimposed on an picture (for case from a TV broadcast).

In OCR processing, the scanned-in picture or bitmap is analyzed for light and dim regions in arrange to recognize each alphabetic letter or numeric digit. When a character is recognized, it is changed over into an ASCII code. Extraordinary circuit sheets and computer chips planned explicitly for OCR are utilized to speed up the acknowledgment process.

OCR is being utilized by libraries to digitize and protect their property. OCR is moreover utilized to prepare checks and credit card slips and sort the mail. Billions of magazines and letters are sorted each day by OCR machines, impressively speeding up mail conveyance. The point of this venture is to recognize Kannada Normal Sign Board Characters. This can be valuable for both local speakers and non-native speakers as well for different reasons.

**1.4.1 Types of OCR**

* Optical character recognition (OCR) – targets typhoid content, one glyph or character at a time.
* Optical word recognition – targets typhoid content, one word at a time (for dialects that utilize a space as a word divider). (Ordinarily fair called "OCR".)
* Intelligent character recognition (ICR) – moreover targets transcribed print script or cursive content one glyph or character at a time, ordinarily including machine learning.
* Intelligent word recognition (IWR) – too targets manually written print script or cursive content, one word at a time. This is particularly valuable for dialects where glyphs are not isolated in cursive script.

OCR is for the most part an "offline" handle, which examinations an inactive report. Penmanship development examination can be utilized as input to penmanship acknowledgment. Instep of simply utilizing the shapes of glyphs and words, this procedure is able to capture movements, such as the arrangement in which fragments are drawn, the course, and the design of putting the write down and lifting it. This extra data can make the end-to-end handle more exact. This innovation is moreover known as "on-line character acknowledgment", "energetic character acknowledgment", "real-time character acknowledgment", and "cleverly character acknowledgment".

**1.5 Motivation for Work**

We have seen people migrating from place to place leaving their native place and going to different linguistic place. It often becomes difficult to read and understand another language. So, we came up with the idea to design a program to overcome the above problem. This can be beneficial not only to non-native speaker but also to native speaker as well.

**1.6 Problem with Existing System**

* OCR system is not present for all languages.
* No systems have been developed for Indic Languages. E.g. Kannada
* Identification and processing of character is difficult.
* Redirecting the character in the file is difficult.
* Translation is not available.

**1.7 Proposed Work**

Our objective is to develop a system that is able to perform various works which meets the requirement of user.

The Proposed system works in following manner.

* Segmentation - HOG

1. Browse any scene text image.
2. Convert RGB to GrayScale.
3. Calculate threshold from GrayScale.
4. Calculate Canny Edge Detection.
5. Removal of unwanted objects.
6. Feature extraction.

* Training a SVM model
* Testing a SVM model
* OCR

1. Browse text image.
2. Convert the recognized character into Latin Script.
3. Redirect into file.

**Chapter 2**

**LITERATURE SURVEY**

**2.1 Existing System**

Text extraction from the images has gained research interest. On this note OCR was invented. OCR is a kind of pattern recognition which contributes towards the recognition of text in the documents. Computer vision, artificial intelligence and pattern recognition contains OCR as part of research. When a document or paper is scanned by computer, it gets only image file. The text on the page is not understandable by the computer, so it is not possible to search or edit the page. OCR software can be used to produce editable file which is more flexible. The existing system of OCR works on grid infrastructure, i.e. works without a grid infrastructure. It deals with homogeneous character recognition of single language. Existing work focus on recognizing characters using bounding boxes. Three different classifiers are trained for this purpose i.e. k-nearest neighbor, random forest and neural networks. Further segmentation technique is used where text pixels are isolated from the background. Due to few drawbacks of OCR technique convco-HOG was proposed where it has more differentiating power by repeatedly examining all possible image pixels .This was improved to recognize characters in natural scenes. Since natural scene characters are type of hand written, their style varies recognizing each style is not easier task. Along with this there are few more features that affect the recognition process i.e. color texture of background and foreground ,geometric distortion which is caused by camera position ,illumination and resolution if the natural scene image This difficulty can be seen in Figure 2.1. This is done by introducing a database of images containing English and Kannada text.

**2.1.1 Related Work**

The work of Kannada characters acknowledgment in characteristic scenes is related to issues considered in camera arranged record examination. Larger part of the work in scene content acknowledgment is particularly based [10], [9], [4] and [3] on finding and adjusting the content zones and taking after the OCR application strategies. [8] Such approaches are in this manner limited to environment where OCR works well. From this time forward such approaches are controlled to environment where OCR works appropriately. In expansion to amendment prepare, it does not straightforwardly relate to our work, as it points on discovery of printed characters. The edge detection is carried out by the technique described by J Canny, called Canny Edge Detection Technique [24] and also by an improved edge detection technique [23].

The technique for static recognition of hand written characters have been efficiently solved by intra-class variation due to non-identical styles of writing [15], [14]. Such scenes prototypically assume only a finite number of appearance classes, unable to resolve differences in foreground/background color and texture, especially the graphics present. This is achieved by identification and removal of extraneous graphics in a commercial OCR operation [25]. For occurrence, [16] we have utilized cognizance from NLP and display a Markov chain system for parsing pictures. [5] Presentation of composition machines for developing probabilistic progressive picture models. This makes a difference in obliging relevant connections. This approach permits re-usability of parts among different substances and non-Markovian disseminations. [16] Proposed a strategy that amalgamates picture highlights and dialect data a single demonstrate and coordinating disparity data between character pictures.

Acknowledgment of digits utilizing pipelines based on crude pictures classifications have been broadly utilized [12]. [21] By shape coordinating procedure, this is too done [1]. The classification is carried forward by HOG technique, known as Histogram Oriented Gradient. In this line by line detection of characters and words is done.

**2.1.2 Data Sets**

We aim to recognize Kannada characters from natural scene images. To do so, we design a database containing images of natural scene having Kannada characters. These images have been gathered from the streets of Tumkur and Bangalore, India. The natural scene images comprises not only of street symbols but also of sign boards, hoardings, posters, pamphlets, banners, name plate, number plate etc. However collection and annotation of huge sample of images is a costly as well as costly job. So, we acquired a database of characters generated by computer fonts of size 72 as shown in Figure 5.

English language has characters separately in two cases namely upper case and lower case, but in case of Kannada Language it’s not the same. Kannada language alphabet does not have the system of upper case and lower case characters. It has 37 consonants and 16 vowels. By combining the vowels with consonants, it generates around 603 distinct classes. It has numerals from 0 to 9 which can further combined to generate infinite number of terms. Digits can be identified separately.

**2.1.3 Data Set of Natural Image**

We have captured 200 images from mobile phone and digital camera. Sample images are depicted in Figure2.2 We have applied two methods of segmentations: canny edge detection technique and rectangular bounding boxes as shown in Figure 6. Both techniques are equally likely to be good. The canny edge detection technique detects objects other than characters and digits.

Out of 20 images of natural scene in our database, 209 numbers of characters are present. The proposed system has identified 185 numbers of characters. The success of identified characters is tabulated separately with percentage ratio in Table 2.1.

**2.1.4 OCR (optical character recognition or optical character reader)**

OCR basically is recognition of printed or written text character by a computer; OCR is the process of converting an image of typed, handwritten or printed text into a machine encoded text with the help of mechanical or electronically conversion. OCR involves

1. Photo scanning of a text character by character
2. Analysis of the scanned text in the image
3. Translation of scanned character image into character code such as ASCII

In OCR processing, the scanned-in image is analyzed for light or dark areas in order to identify each alphabetic latter or numeric digit. When the character is recognized, it is converted into an ASCII code.

Types of OCR are listed below:-

1. Optical character recognition (OCR) - targets typewritten text, one glyph or character at a time. OCR can be subdivide into two types , soft based and machine based( or inline ).while core algorithm is similar for both , these two technologies are used on different types of text. Inline OCR are used at scan time and very often not used on documents rather on objects going down an assembly line. In-line OCR for documents is used primarily for mail-room processing on high-speed high volume scanner, or on manufacturing assembly line. The benefit of in-line OCR is that it’s that fastest OCR around. The downside of in-line OCR is that accuracy reduces when it comes to document scanning. In PC based OCR or software OCR we have the benefit of scalability. It can work on the widest range of document types; reason for the improvement is because it is using PC which has the latest technologies that work on degraded documents and complex documents. The drawback of software OCR is that it’s not as fast as in-line OCR.
2. Optical word recognition (OWR) - target typewritten text, one word at a time (for language that use a space as a word divider)
3. Intelligent character recognition (ICR) - target handwritten print script or cursive text one glyph or character at a time, usually involving machine learning.
4. Intelligent word recognition (IWR) - target handwritten, print script or cursive text, one word at a time, this is especially useful for language where glyphs are not separated in cursive script.

**2.1.5 Drawback of Existing System**

The drawback in the early OCR system is that they only have the capability to convert and recognize the textual image of English language to any desired language. That is, the older system is unilingual.

**2.2 Proposed System**

The goal of this project is to develop methods for improving natural scene text recognition. Here we have focused on recognizing characters in situations that would traditionally not be handled well by OCR (Optical Character Recognition) techniques. We will have an annotated database of images containing English and Kannada characters. The problem is addressed in an object categorization framework based on a bag-of-visual-words representation. We will assess the performance of various features based on nearest neighbor and SVM (Support Vector Machine) classification.

We will do this by including new types of information into models and by considering how to compose simple components into highly active systems. Mainly we will focus on three areas of scene text recognition, each with a decreasing number of prior conjectures. Firstly, we will introduce two techniques for character recognition, where word and character bounding boxes will be used. Next we will look at word recognition, where only word bounding boxes will be used. We want to develop a new technique for fragmenting text for these images called bilateral regression segmentation. Lastly, we will remove the assumption that words have been located and describe an end-to-end system that detects and recognizes text in any natural scene image.

**2.2.1 Segmentation**

Documents horizontal projection profile is found to separate the lines in the image. Histogram of the number of ON pixels along every row of the image is the horizontal projection profile. The letters in Kannada are composed by attaching to the glyph of a consonant, the glyphs of the vowel modifiers and the glyphs of the consonant conjuncts. If we considered all the combination, then building the classifier of these numbers of character is very difficult. So our strategy is that we will segment the word into its constituents, i.e. the base consonant, the vowel modifier and the consonant conjunct. It’s very difficult to achieve this. If we have good look on the Kannada word, we will see that for extracting glyph of a consonant the glyphs of the vowel modifiers and the glyphs of the consonant conjuncts we can divide the character into two zones. Top zone: Top zone mainly consist of main portion of the character. It includes base consonant or vowels or some vowel modifiers. Bottom zone: Bottom zone consists of glyphs for the consonant conjuncts. Here by using connected component method, we are first counting the number of consonants, vowels, vathu or vowel modifiers present in the text line. Connected component method is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. Connected component labeling is used in computer vision to detect connected regions in binary digital images, although color images and data with higher-dimensionality can also be processed. In connected component method connectivity checks are carried out by checking the labels of pixels that are North-East, North, North-West and West of the current pixel. Then we extracting that characters separately and send it for character recognition.

**2.2.2 Canny Edge Detector**

Edge detection is image processing technique for finding the boundaries of object within image. Edge detection is used for image segmentation and data extraction. The main criteria of canny edge detector is

1. Low error rate -good detection of only existent edge.
2. Good localization -distance between edge pixel detected and real edge pixel need to be minimized.
3. Minimal response –only one detector response per edge.

It includes five steps,

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of image
3. Taking magnitude and performing some local maxima i.e., non maximum suppression .here multiple pixel are thinned into single pixel
4. Linking and thresholding is performed on Gradient image. Here grouping of pixel is done based on low threshold and high threshold. High threshold is used to start curve and low threshold is used to continue curve
5. Finally edge map is obtained by considering only the edges that are connected to strong edges.

**2.3 Training a SVM Model**

SVM Model means Support Vector Machine; this model is used for training a dataset using supervised model for classifying and regression of data through learning algorithm that analyzes the data. This model is also used for mapping points in the space and also for linear and non-linear classification. As this model uses supervised learning in which a particular input is given for the required output which organizes different algorithm and training of data. In supervised learning we have to analyze different step as follows:

1. First is determining the different training dataset and analyzing the dataset whether it is handwritten data or typed.
2. Collecting of different dataset which includes set of input objects and output according to it.
3. Analyzing the feature representation of particular input and also the features on which input dataset depends like dimension, no of vectors and feature of vector.
4. Analyzing the structure of the function in model and algorithm like instead of SVM we can choose decision trees.
5. Running the algorithm for the featured function and completing the design with the certain parameters.
6. Last we have to calculate the accuracy for the given dataset as here we are giving the positive and negative dataset.

Training a model can be done once for the dataset and we can proceed for the testing of different dataset.

As here we are doing the training for the 251 positive and negative dataset. During training the dataset is mapped and particular accuracy is obtained.

**2.4 Testing a SVM Model**

After testing a SVM model then training part comes, where a particular dataset is mapped and checked with a regular dataset whether it is mapping or not. In this method we use bounding box technique for mapping of the data. In this data checked according to the no of lines and mapped for different lines in separate windows. If the data is mapped it will show the boundary for particular consonants and vowels and data segmented for particular line. If data doesn’t match the data will be blurred or unbounded, it happens in the case when the input sample is not clean or blurry image is there. So, clarity of the image is one of the important factors while training.

**Chapter 3**

**SYSTEM REQUIREMENTS SPECIFICATION**

**3.1 Overview**

This document provides the basic requirements of the system to be fully functional. Here we outline the minimal hardware and software requirements for setting up this project which can be used to demonstrate how to recognize Kannada character from the image and convert it into Latin Kannada script. For example - If we consider any image with contains Kannada character say ‘neer’, so our application will first recognize character ‘neer’ written in Kannada language and then convert it in the form of Latin script.

**3.2 System Requirements**

This document gives the necessary instructions for fulfilling the requirements of deploying the project.

**3.2.1 Hardware Requirements**

Hardware requirement specifications are technical descriptions of the hardware components and used in the project.

Following are the hardware required in this project:-

1. Computer supporting MATLAB R2017a (9.2.0.538062)

2. Camera or mobile phone to capture image of format supported by specified MATLAB version.

**3.2.2 Software Requirements**

A software requirement specification describes about the software that we have used. Along with it includes use case for user interactions and also functional and non functional requirements.

Following are the software requirements in this project:-

1. Operating system : Windows/Linux/MAC.

2. Coding language : MATLAB.

3. IDE : MATLAB R2017a (9.2.0.538062)

**3.2.3 Interface Requirements**

User: User should use a camera or a mobile phone to capture the image. User need to transfer the image in the system supporting MATLAB R2017a and are able to get Kannada character in the form of Latin script.

**3.2.4 Product Requirements**

**3.2.4.1. Usability Requirements**

The GUI of the application should be user friendly. There should be ease in understand the steps the need to be followed to get the Latin script. This should able to choose picture from the folder just by one click and further steps need to done easily without any difficulty.

**3.2.4.2 Reliability Requirements**

Application should be reliable enough so that user don’t have to face any unexpected result during the execution and also it should not put user in situation where user is unable to understand necessary steps to be taken to handle failure.

**3.2.4.3 Efficiency Requirements**

**3.2.4.3.1. Performance**

Performance of the application is measured using the percentage of character the application is able to recognize the time taken by it in the compilation of task. Performance can be measured in three aspects like Recognition rate: The portion of image that is been identified correctly. Rejected rate: The portion of image that is not been identified correctly.

**3.2.5 External Requirements**

**3.2.5.1. Privacy**

The application should not reveal user personal data and also should not fetch any information contained in the computer system. It should only perform the task of browsing the stored image and convert it into Latin script.

**3.3 Software Package:**

This refers to the individual resources that are been used to form complete software package to provide desired functionality.

**WHY MATLAB?**

It basically stands for MATrixLABorator which means that here everything is all about matrix operation. In this any input image is considered as a matrix and we apply various desired operation on the input image to do the analysis.

Taking image as an input which mainly deals in image processing which is a very vast topic. We can identify the design, feel, shade, picture, luminance etc of the image. With the help of MATLAB we can easily resolve the specialized calculation in easy and faster pace than high level programming language like C, C++, and JAVA etc.

**Chapter 4**

**SYSTEM DESING**

System design is the method of characterizing the engineering, components, modules and interfacing, and information for a framework to fulfill specified requirements. Frameworks plan may be seen as the application of frameworks hypothesis to item advancement. Actual Plan of the issue is depicted in this chapter as beneath-

**4.1 DFD – Data Flow Diagram**

It could be a graphical representation of the “flow” of information through a data framework, displaying it’s preparing aspects.

A Data Flow Diagram (DFD) is frequently utilized as preparatory step to make a diagram of the framework without going into awesome subtle elements, which can afterward be elaborated.

Figure 4.1: Data Flow Diagram

The above diagram illustrates the flow of data of this project. The input and output of every module is shown by the arrow head, depicting the flow of data.

**4.2 Unified Modeling Language (UML) – Diagrams**

UML could be a standardized displaying dialect empowering designers to indicate, visualize, construct and report relics of software. In this way, UML make these antiquities sealable, secure and strong in execution.

**4.2.1 Class diagram**

It may be a sort of inactive structure graph that depicts the structure of a framework by appearing the system’s classes, their traits, operations (or strategies) and the connections among objects.

**4.2.2 Object Diagram**

Object Diagram may be a chart of occurrence, counting objects and information esteem. An inactive question graphs an occurrence of a lesson graph; it appears a preview of the point by point state of a framework at a point in time.

Figure 4.3: Object Diagram

Figure drawn above shows the object diagram of this project.

A UML question chart speaks to a particular occurrence of a course chart at certain minute in time. When spoken to graphically, we see numerous likenesses to the course chart. An object graph centers on the properties of a set of objects and how these objects relate to each other.

**4.2.3 State Diagram**

State Graph portrays the conduct of the frameworks. It requires that the framework depicted is composed of a limited number of states. A state graph appears the conduct of classes in reaction to outside boosts. Particularly, a state chart portrays the conduct of a single protest in reaction to an arrangement of occasions in a framework.

Figure 4.4: State Diagram

**4.2.4 Activity Diagram**

Movement chart are graphical representations of workflows of step astute exercises and activities with back for choice, cycle and concurrency. In Bound together Modeling Dialect, movement charts are expecting to demonstrate both computational and organizational processes. Activity graph is UML conduct chart which appears stream of control with accentuation on arrangement and condition of the stream.

Figure 4.5: Activity Diagram

The above figure shows the activity diagram.

**4.2.5 Sequence Diagram**

A sequence diagram is an interaction diagram that appears how objects work with one another and in what arrange. It could be a build of a message arrangement chart. An arrangement graph appears how objects intelligent organized in time arrangement. Arrangement graphs are now and then eluded as occasion charts or occasion scenarios.

Figure 4.6: Sequence Diagram

UML Sequence diagrams are utilized to appear how objects connected in a given circumstance. A vital characteristic in grouping graph is that time passes from best to foot.

**4.2.6 Use Case Diagram**

A use case diagram is a graphical portrayal of the intelligent among the elements of a framework. A utilize case may be a technique utilized in framework examination to recognize, clarify and organize system’s necessities.

Figure 4.7: Use Case Diagram

Use case diagram is a behavioral UML diagram type and as often as possible utilized to analyze different frameworks. It empowers us to imagine the distinctive sorts of parts in a framework and how those parts associated with the framework.

**4.2.7 Collaboration Diagram**

A collaboration diagram, moreover called a communication or interaction graph, is an outline of connections and intuitive among software intuitive within the Unified Modeling (UML).

Figure 4.8: Collaboration Diagram

UML collaboration diagram outlines the relationship and interaction between computer program objects.

**4.2.8 Component Diagram**

In a Unified Modeling Language, a component graph delineates how components are wired together to create huge components or software systems. They are utilized to demonstrate structure of subjectively complex frameworks.

Figure 4.9: Component Diagram

Component diagram is a special kind of chart in UML. The reason is additionally diverse from all other graphs talked about so distant. It does not depict the functionality of the framework but it portrays the components utilized to form those functionalities.

**4.2.9 Deployment Diagram**

It is a structure diagram which appears design of the framework as arrangement of program relics to arrangement targets.

The title sending itself portrays the reason of the graph. Sending chart is utilized for portraying the equipment components where program components are conveyed.

Figure 4.10: Deployment Diagram

**Chapter 5**

**SYSTEM IMPLEMENTATION**

This chapter examines the overall plan of this project. The chapter too gives a portrayal of the architecture of the framework conjointly the detailed plan issues of the components of the system.

**5.1 System Architecture**

System architecture is a conceptual demonstration that characterizes the structure, behavior, and more views of a framework. A design portrayal could be a formal portrayal and representation of a framework, organized in a way that underpins thinking almost the structures and behaviors of the framework. A framework engineering can contain framework components that will work together to execute the generally system.

The following Figure shows the architecture of the system.

Figure 5.1: System Architecture

Our system architecture consists of following components.

* End user
* Graphical User Interface(GUI)
* Operating System

Inside the domain of the venture, we have four modules which are intuitively to the client to be specific Segmentation, Training, Testing and OCR. When the client interacts with one of these modules a yield is delivered. The division module is advance separated into six parts, which in turns gives particular yield.

The above figure shows the system architecture of this project. It contains all the essential module and tools which are required to develop the project. It gives us the blue print of the projects, thereby showing the flow of work.

**5.1.1 Flow Diagram**

This is the flow diagram of training and testing of standard dataset and to that of sample images taken from nature. It shows how each phase will work at different stages.

Figure 5.2: Flow Diagram

The above figure shows us the flow diagram of the project for training and testing part. The training part undergoes ‘feature extraction’ and then ‘classification’ which in turn goes to model generator. The testing part also undergoes the same process as training. The extension of testing after classification is result. After this phase the sample image is compared to standard annotated database. It gives the identified characters in notepad file.

**5.1.2 Sequential Flow Diagram**

This phase shows the linear flow of the project. Firstly, the input image is browsed and sent for further processing which is listed below.

Figure 5.3: Sequential Flow Diagram

The linear steps are as follows:-

1. Scene Capturing
2. Captured Image
3. Detect Image of Interest
4. RGB to GrayScale Image
5. Calculate Threshold
6. Calculate Canny Edge Detection
7. Removing Unwanted Objects
8. Feature Extraction- HOG
9. Training a Dataset
10. Training- Detection of Characters

**5.1.3 End User**

An end user is an individual who eventually employments or is planning to eventually utilize an item. The conclusion client stands in differentiate to clients who bolster or keep up the item such as sysops, framework directors, database directors, data innovation specialists, computer program professionals and computer specialists. Conclusion clients ordinarily don't have the technical understanding or aptitude of the item originators, a reality that's simple for originators to disregard or neglect, driving to highlights with which the client is disappointed.

**5.1.4 Operating System**

Operating System is low-level program that underpins a computer's essential capacities, such as planning errands and controlling peripherals. It acts as halfway between end user and the system created and introduced within the O.S.

**5.2 Features**

We have tried features like Shape Context and Geometric Blur, these are based on shape and edge detection. SVM classifier technique has also been used for its higher efficiency and faster processing unlike SIFT which is comparatively slow. The methods like canny edge detection, change of RGB image to Gray scale image, calculation of threshold for each image, highlights utilized for speaking to surface, such as channel reactions, patches and Spin Images have moreover been utilized [20]. We have also tried to remove the unwanted objects like graphics from foreground and background from the natural scene image. We have analyzed various commonly used parameters and feature detection technique used for each descriptor which have been described below. We have also tried to remove the unwanted objects like graphics from foreground and background from the natural scene image. We have analyzed different commonly utilized parameters and highlight location procedure utilized for each descriptor which has been depicted underneath.

**5.2.1 Convert RGB to Gray Scale** [22] To convert RGB to grayscale, the average of all the three i.e. R, G and B is computed. To do so we add R with G with B and then divide it by 3 to obtain the grayscale. For example: Figure 9 and Figure 10 describes the conversion of RGB to Grayscale respectively.

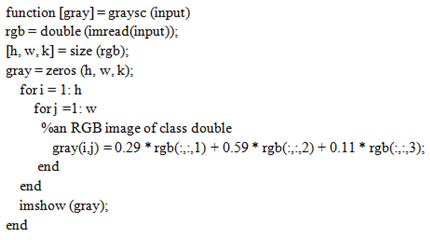
****

Figure 5.4: Algorithm for RGB to Gray Scale Conversion

**5.2.2 Geometric Blur** (GB) [2] Geometric blur is simply an average over geometric transformations of a signal. It is done by sampling method called feature extraction which is same as SC. It is isolated into distinctive locales and at that point the edge introductions are checked with distinctive obscure figure.

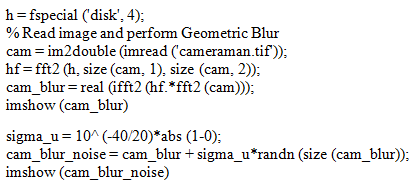


Figure 5.5: Algorithm for Geometric Blur

**5.2.3 Calculating Threshold** [28] Calculation of threshold of an image is done by separating a picture into closer view and foundation independently. This handle changes over the grayscale picture into binary picture. Example: Figure 11 shows how threshold image looks like.

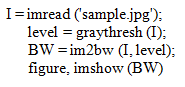


Figure 5.6: Algorithm for Threshold Calculation

**5.2.4 Shape Contexts** (SC) [1] SC is a feature descriptor. It is used for object recognition and description of shape that permits measuring shape similarity. We do it by Sobel edge detection technique using log-polar histogram. We use histogram of oriented gradients (HOG) technique as well.

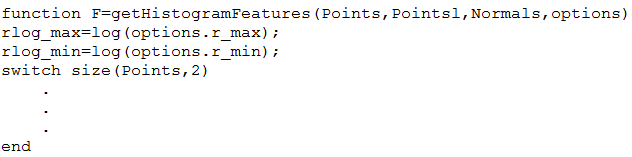


Figure 5.7: Pseudo Code for Shape Contexts

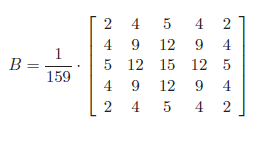
**5.2.5 Canny Edge Detection** [24], [23] By the help of Canny Edge Detection algorithm, we detect edges of objects in an image. It is a multi-stage algorithm. It is useful as it extracts structural information.

The algorithm runs in 5 separate steps:

1. Smoothing: Obscuring of the picture to evacuate clamor.
2. Finding gradients: The edges ought to be stamped where the slopes of the picture has expansive extents.
3. Non-maximum suppression: Only local maxima should be marked as edges.
4. Double thresholding: Potential edges are decided by thresholding.
5. Edge tracking by hysteresis: Last edges are decided by stifling all edges that are not associated to a really certain (solid) edge.

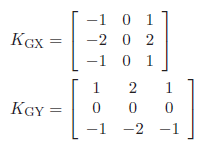
**5.2.5.1 Smoothing**

It is inescapable that all pictures taken from a camera will contain a few sum of commotion. To prevent that commotion is mixed up for edges, clamor must be diminished. In this manner the picture is first smoothed by applying a Gaussian channel. The bit of a Gaussian channel with a standard deviation of = 1.4 is appeared in Equation (1).

(1)

**5.2.5.2 Finding Gradients**

The Canny calculation essentially finds edges where the grayscale escalated of the picture changes the most. These ranges are found by deciding slopes of the picture. Slopes at each pixel in the smoothed picture are decided by applying what is known as the Sobel-operator. First step is to surmise the angle within the x- and y-direction individually by applying the kernels shown in Equation (2).

(2)

The angle extents (too known as the edge qualities) can at that point be decided as a Euclidean remove degree by applying the law of Pythagoras as appeared in Condition (3). It is in some cases disentangled by applying Manhattan separate degree as appeared in Equation (4) to reduce the computational complexity. The Euclidean separate degree has been connected to the test picture.

|G| = (3)

|G| = |Gx| +|Gy| (4)

Where: Gx and Gy are the angles within the x- and y-directions respectively. It is clear from Figure 3, that a picture of the slope sizes frequently demonstrate the edges quite clearly. However, the edges are regularly wide and thus don't show precisely where the edges are. To create it conceivable to decide this (see Section 2.3), the heading of the edges must be decided and put away as appeared in Equation (5).

Ɵ = arctan (|Gy| / |Gx|) (5)

**5.2.5.3 Non-maximum suppression**

The reason of this step is to change over the “blurred” edges within the picture of the slope magnitude to “sharp” edges. Typically it is done by protecting all neighborhood maxima within the angle image, and erasing everything else. The calculation is for each pixel within the slope picture:

1. Circular the slope heading ᶿ to closest 45◦, comparing to utilize of an 8-connected neighborhood.
2. Compare the edge quality of the current pixel with the edge quality of the pixel in the positive and negative angle course. I.e. in the event that the slope course is north (theta = 90◦), compare with the pixels to the north and south.
3. In case the edge quality of the current pixel is biggest; protect the esteem of the edge strength. If not, smother (i.e. evacuate) the esteem.

**5.2.5.4 Double thresholding**

The edge-pixels remaining after the non-maximum concealment step are (still) checked with their strength pixel-by-pixel. Numerous of these will likely be genuine edges within the picture, but a few may be caused by clamor or color varieties for occasion due to unpleasant surfaces. The only way to discern between these would be to utilize a limit, so that as it were edges more grounded that a certain value would be protected. The Canny edge calculation employs twofold thresholding. Edge pixels more grounded than the tall edge are checked as solid; edge pixels weaker than the low threshold are smothered and edge pixels between the two limits are checked as frail.

**5.2.5.5 Edge tracking by hysteresis**

Solid edges are translated as “certain edges”, and can promptly be included within the final edge picture. Powerless edges are included in case and as it were in case they are associated to solid edges. The logic is of course that clamor and other little varieties are improbable to result in a solid edge (with appropriate alteration of the limit levels). Hence solid edges will (nearly) as it were be due to true edges within the unique picture. The frail edges can either be due to genuine edges or noise/color variations. The last mentioned sort will likely be dispersed freely of edges on the entire image, and hence as it were a little sum will be found adjoining to solid edges. Powerless edges due to genuine edges are much more likely to be associated specifically to solid edges.

**5.2.6 Removal of Unwanted Objects** [25] the subject of removing unwanted objects from natural scene images without generating any possible distortion has been handled.

**5.2.7 Histogram of Oriented Gradients (HOG)** [26], [27] It is a highlight descriptor utilized in computer vision and picture handling for the reason of protest location. The procedure tallies events of slope introduction in localized parcels of a picture. Features = extractHOGFeatures (I) It returns extricated Hoard highlights from a truecolor or grayscale input picture, I. The highlights are returned in a 1-by-N vector, where N is the Hoard highlight length. The returned highlights encode neighborhood shape data from locales inside a picture.

**5.2.8 Spin Image** [11], [6] It is a two dimensional histogram encoding method for conveyance of picture brightness. The two dimensional of the histogram is d, separate from the center point, and i the escalated esteem. We have utilized d=11 and i=5 for concentrated esteem, coming about in 55-dimensional descriptors.

**5.2.9 Maximum Response of Filters** (MR8) [18] It is a surface descriptor based giving 8D vectors, on a set of 38 channels but as it were 8 responses.

**5.2.10 Patch Descriptor** (PCH) [19] It is the least complex thick include extraction strategy. For each position, the crude n × n pixel values are vectorized, producing an n2 descriptor. We utilized 5×5 patches.

**Chapter 6**

**TESTING**

**6.1 Test workflow**

Software testing is an examination directed to give partners data about the nature of the product item or administration under test. Software testing can likewise give a target, autonomous perspective of the product to enable the business to acknowledge and comprehend the dangers of programming usage. Test strategies incorporate the way toward executing a program or application with the aim of discovering programming bugs (blunders or different deformities), and checking that the product item is fit for utilize.

Software testing includes the execution of a product part or framework segment to assess at least one properties of intrigue. When all is said in done, these properties show the degree to which the part or framework under test

* Meets the prerequisites that guided its outline and improvement,
* Reacts effectively to a wide range of sources of info,
* Performs out its capacities inside an adequate time,
* Is adequately usable,
* Can be introduced and keep running in its proposed surroundings, and
* Accomplishes the general outcome its partners want.

**6.1.1 White-box testing**

White-box testing (otherwise called clear box testing, glass box testing, straightforward box testing and auxiliary testing, by observing the source code) tests inward structures or workings of a program, instead of the usefulness presented to the end-client. In white-box testing, an inside point of view of the framework, and in addition programming abilities, are utilized to configuration test cases. The analyzer picks contributions to practice ways through the code and decide the proper yields. This is practically equivalent to testing hubs in a circuit, e.g. in-circuit testing (ICT).

Same time white-box trying can be a chance to be connected at the unit, integration Also system levels of the product trying process, it may be as a rule carried out at the unit level. It could test ways inside a unit, ways between units throughout integration, furthermore between subsystems throughout a system–level test. However this system for test configuration could uncover a lot of people errors or problems, it might not recognize unimplemented parts of the detail or absent necessities.

### 6.1.2 Unit testing

Unit testing may be a product improvement transform that includes a synchronized provision of a wide range for abandon counteractive action Also identification methodologies in place to decrease programming advancement risks, time, What's more costs. It may be performed toward those programming designer alternately specialist throughout those development stage of the product improvement lifecycle. Unit trying means on dispense with development errors preceding code may be pushed with extra testing; this methodology will be planned on increment the caliber of the coming about product and in addition those effectiveness of the generally improvement methodology.

### 6.1.3 Integration testing

Integration testing will be at whatever kind of product trying that tries on check the interfaces between parts against a programming plan. Programming parts might be coordinated previously, an iterative approach alternately all together ("big bang"). Typically those previous may be recognized An preferred act since it permits interface issues should make spotted that's only the tip of the iceberg rapidly What's more altered. Joining trying meets expectations on uncover defects in the interfaces and communication between coordinated circuit parts (modules). Progressively bigger aggregations of tried product segments comparing on components of the structural plan would incorporated What's more tried until those programming meets expectations likewise an arrangement.

**6.1.4 System testing**

System testing tests a totally incorporated system to confirm that the system meets its requirements. For instance, a system test may include testing a logon interface, at that point making and altering a section, in addition to sending or printing comes about, trailed by synopsis preparing or cancellation (or chronicling) of passages, at that point logoff.

### 6.1.5 Compatibility testing

A normal reason for product disappointment (real or perceived) may be an absence from claiming its similarity for other requisition software, working frameworks (or working framework versions, of age or new), or target situations that contrast incredibly starting with those first (such Likewise a terminal alternately GUI provision planned should a chance to be run on those desktop Right away constantly obliged should get to be a Web application, which must render to a Web browser). For example, on account of a absence of retrograde compatibility, this cam wood happen a result the programmers create What's more test product just on the most recent adaptation of the focus environment, which not the sum clients might be running.

Testing is one of the essential eliminate which is conveyed at each progression of advancement of an application or an item. Software testing can be characterized as a procedure of executing a program or application with the aim of finding the software bugs. Testing is performed with the goal that an application or item can be approved and confirmed to check the application or item is working the route expected by the engineer or the end client.

**6.1.6 Black-box Testing**

Black-box testing is a technique for software testing that looks at the usefulness of an application without peering into its inner structures or workings. This strategy for test can be connected for all intents and purposes to each level of software testing: unit, integration, system and acceptance.

Test cases contains diverse conditions or factors like test case id to recognize the test case, test case name which shows the segment of a framework under test, input given to every segment, expected consequences of the specific test case, real outcomes to contrast and the outcomes expected and the status which demonstrates pass or come up short aftereffect of a test case.

**Chapter 7**

**Result and Discussion**

Here we describe the experiments under different schemes as mentioned above. The four classification schemes used are:

1. multiple kernels learning (MKL)
2. support vector machines (SVM)
3. closest neighbor (NN) classification utilizing c2 measurement as a closeness degree and
4. Histogram of Oriented Gradients (HOG).

The following outputs have been obtained using a sample natural scene image. The pictures of the output are given in sequential order.

**7.1 Scene Captured from Nature**

Figure 7.1: Original Image from Sample Database

**7.2 Conversion of RGB to GrayScale Image**

Figure 7.2: Converting RGB to Gray scale Image

**7.3 Calculating Threshold of Image**

Figure 7.3: Calculating Threshold of Image

**7.4 Canny Edge Detection**

Figure 7.4: Canny Edge Detection

**7.5 Removal of Unwanted Objects with Lesser Pixels**

Figure 7.5: Removing Unwanted Objects less than 10 pixels

**7.6 Feature Extraction**

Figure 7.6: Feature Extraction

**7.6.1 Feature Extraction Using - HOG**

Figure 7.6: Feature Extraction – HOG

**7.7 Training**

Figure 7.7: Training

**7.8 Testing**

Figure 7.8: Testing

Figure 7.9: Step wise process of Testing

**CONCLUSION AND FUTURE WORK**

In this project, we have handled the issue of recognizing characters in pictures of natural scenes. The database of test common scene pictures containing Kannada characters have been captured in Bangalore and Tumkur, India. Clarifying of normal pictures for training purposes can be costly and time expending. The techniques listed in neural network is not that effective when it comes to efficiency, so we have used SVM whose efficiency and computational time id more.

By distinguishing the downsides of the existing framework, we have handled the issue of recognizing Kannada characters in normal scene. The benefits of this experiments are- It increases the efficiency and effectiveness of work there it saves time. Documents can be text searchable and editable. It can help non native speaker learn Kannada language and communicating with native people will be easy. It has got social benefits as well.

In future this project will incorporate the feature of audio and video signaling. The identified text will be audible for the illiterate users who cannot read.