NaYaNa
A Phonetic Script to Make Communication Easy

Submitted in partial fulfilment of the requirements

of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

By

Group No: 43

Roll No.	Name
1404090	Rahurkar Prachi Shriram
1404091	Ramrakhiani Deepa Ram
1404092	Rao Smriti Ganesh

Supervisor

PROF. VAISHALI SURYAWANSHI

(Assistant Professor, Department of Computer Engineering, TSEC)

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Computer Engineering Department

Thadomal Shahani Engineering College

University of Mumbai

2017-2018

CERTIFICATE

This is to certify that the project entitled **"NaYaNa – A Phonetic Script To Make Communication Easy"** is a bonafide work of

Roll No.	Name
1404090	Rahurkar Prachi Shriram
1404091	Ramrakhiani Deepa Ram
1404092	Rao Smriti Ganesh

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"BACHELOR OF ENGINEERING"** in **"COMPUTER ENGINEERING"**.

Prof. Vaishali Suryawanshi

(Supervisor/ Guide)

Dr. Tanuja Sarode

(Head of Department)

Dr. G. T. Thampi

(Principal)

Project Report Approval for B.E.

Project report entitled **NaYaNa –A Phonetic Script To Make Communication Easy** by

Roll No.	Name
1404090	Rahurkar Prachi Shriram
1404091	Ramrakhiani Deepa Ram
1404092	Rao Smriti Ganesh

is approved for the degree of ***"BACHELOR OF ENGINEERING"*** in ***"COMPUTER ENGINEERING"***.

Examiners

1.---

2.---

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be the cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1)

____

(Prachi Rahurkar)

1404090

2)

____

(Deepa Ramrakhiani)

1404091

3)

____

(Smriti Rao)

1404092

Date:

Abstract

NaYaNa is a new phonetic script, created by Dr. Nagarjuna G. and Dr. Vickram Crishna, Homi Bhabha Centre for Science Education, Tata Institute of Fundamental Research, Mumbai, which is based on the complete range of vocalisations currently accepted internationally as an ability of all human beings. Z@Z is read as NaYaNa, same as N@N. This is a script designed with some principles to make the script easy to read, although it potentially matches the full capabilities researched and documented as the IPA. IPA is not a script for regular use in communication (reading and writing), but rather a complete catalogue of all distinct vocalisations that we can produce. The main motivation for designing this script is to make writing dyslexia friendly. We intend to popularise the use of Z@Z for regular communication.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Z@Z is a new phonetic script based on the complete range of vocalizations currently accepted internationally as an ability of all human beings. Z@Z is read as NaYaNa. Z@Z is same as N@N.

This is a script designed with some principles to make the script easy to read although it potentially matches the full capabilities researched and documented as the IPA (International Phonetic Alphabet). IPA is not a script for regular use in communication (reading and writing), but rather a complete catalogue of all distinct vocalizations we can produce.[1] Phonetic transcription (also known as phonetic script or phonetic notation) is the visual representation of speech sounds (or phones). The most common type of phonetic transcription uses a phonetic alphabet, such as the International Phonetic Alphabet.

1.2 Aim & Objectives

The main motivation for designing this script is to make writing dyslexia friendly. Dyslexia is a reading disorder. It causes difficulty with reading, spelling, writing and sometimes speaking. In people with dyslexia, the brain has trouble recognizing or processing certain types of information. This can include matching letter sounds and symbols (such as the letter b making the buh sound) and blending them together to make words.

Some people with dyslexia don't have trouble sounding out or "decoding" words. But they may struggle to understand what they read. It can be very hard for people with dyslexia to read in a way that's automatic, or seemingly without effort. Like other types of learning and attention issues, dyslexia is a lifelong condition. Children don't outgrow it.

Characteristics of dyslexia often include:

- Difficulty associating sounds with letters and letters with sounds

- Confusion when pronouncing words and phrases, such as saying “mawn lower” instead of “lawn mower”
- Difficulty reading aloud with the proper tone and grouping words and phrases together appropriately
- Difficulty “sounding out” unfamiliar words
- Trouble writing or copying letters, numbers and symbols in the correct order
- Trouble rhyming

We also intend to popularize the use of Z@Z for regular communication. The same script can be used by multiple languages. This in turn will help people learn multiple languages easily and thus improve communication.

1.3 Scope

NaYaNa will help in improving the dyslexic students spell words correctly. Many words in English have silent letters. Silent letters are letters that you can't hear when you say the word, but that are there when you write the word. This will not be the case as far as writing in NaYaNa is concerned.

It will also help in the use of HOMONYMS, which area words that sound alike but have the same spelling with different meanings (for example, pole – a long slender rod, pole – North or South Pole). Homophones are a type of homonym that also sound alike and have different meanings, but have different spellings (for example, new and knew). It might not help dyslexic students much with HOMOGRAPHS (words that are spelled the same but have different meanings and may or may not be pronounced the same) though (for example, bow and bow).

It will provide an input method on the Android and Macintosh platforms to communicate in NaYaNa. It also has a provision for 'text-to-speech' on the Macintosh platform. Along with this, the system also has an optical character recognition module on a small scale (at the word level) to convert NaYaNa to English.

Chapter 2

Review of Literature

2.1 Domain Explanation

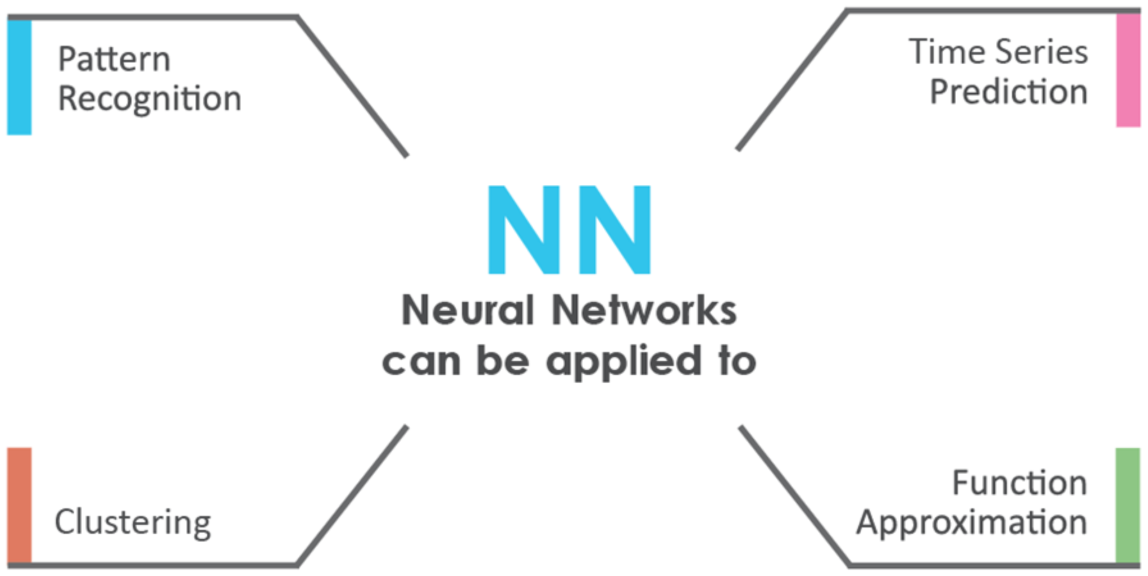


Figure 2.1 Neural Network Applications

In information technology, a neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. Neural networks -- also called artificial neural networks -- are a variety of deep learning technologies. Our project comes under the domain of neural networks. The machine is trained to understand different orientations of an alphabet and still be able to recognise the alphabet. We are also doing the OCR (Optical Character Recognition) for our script.

The most popular and simple approach to OCR problem is based on feed forward neural network with backpropagation learning. The main idea is that we should first prepare a training set and then train a neural network to recognize patterns from the training set. In the training step we teach the network to respond with desired output for a specified input. For this purpose each training sample is represented by two components: possible input and the desired network's output for the input. After the training step is done, we can give an arbitrary input to the network and the network will form an output, from which we can resolve a pattern type presented to the network.

2.2 Existing Solution

The International Phonetic Alphabet (IPA) is an alphabetic system of phonetic notation based primarily on the Latin alphabet. It was devised by the International Phonetic Association in the late 19th century as a standardized representation of the sounds of spoken language. The IPA is used by lexicographers, foreign language students and teachers, linguists, speech-language pathologists, singers, actors, constructed language creators and translators.[3]

The IPA is designed to represent only those qualities of speech that are part of oral language: phones, phonemes, intonation and the separation of words and syllables. The International Phonetic Alphabet (IPA) is an alphabetic system of phonetic notation based primarily on the Latin alphabet.
The IPA is to provide one letter for each distinctive sound (speech segment), although this practice is not followed if the sound itself is complex.[3]

The phonetics and phonology of the English language differ from one dialect to another, usually without interfering with mutual communication. Phonological variation affects the inventory of phonemes (i.e. speech sounds that distinguish meaning), and phonetic variation is differences in pronunciation of the phonemes. This mainly describes the standard pronunciations of the United Kingdom and the United States.

Stress plays an important role in English. Certain syllables are stressed, while others are unstressed. Stress is a combination of duration, intensity, vowel quality, and sometimes changes in pitch. Stressed syllables are pronounced longer and louder than unstressed syllables, and vowels in unstressed syllables are frequently reduced while vowels in stressed syllables are not.

In terms of rhythm, English is generally described as a stress-timed language, meaning that the amount of time between stressed syllables tends to be equal. Stressed syllables are pronounced longer, but unstressed syllables (syllables between stresses) are shortened. Vowels in unstressed syllables are shortened as well.

The International Phonetic Alphabet is occasionally modified by the IPA Association. After each modification, the Association provides an updated simplified presentation of the alphabet in the form of a chart.

Disadvantage of using IPA for general writing:

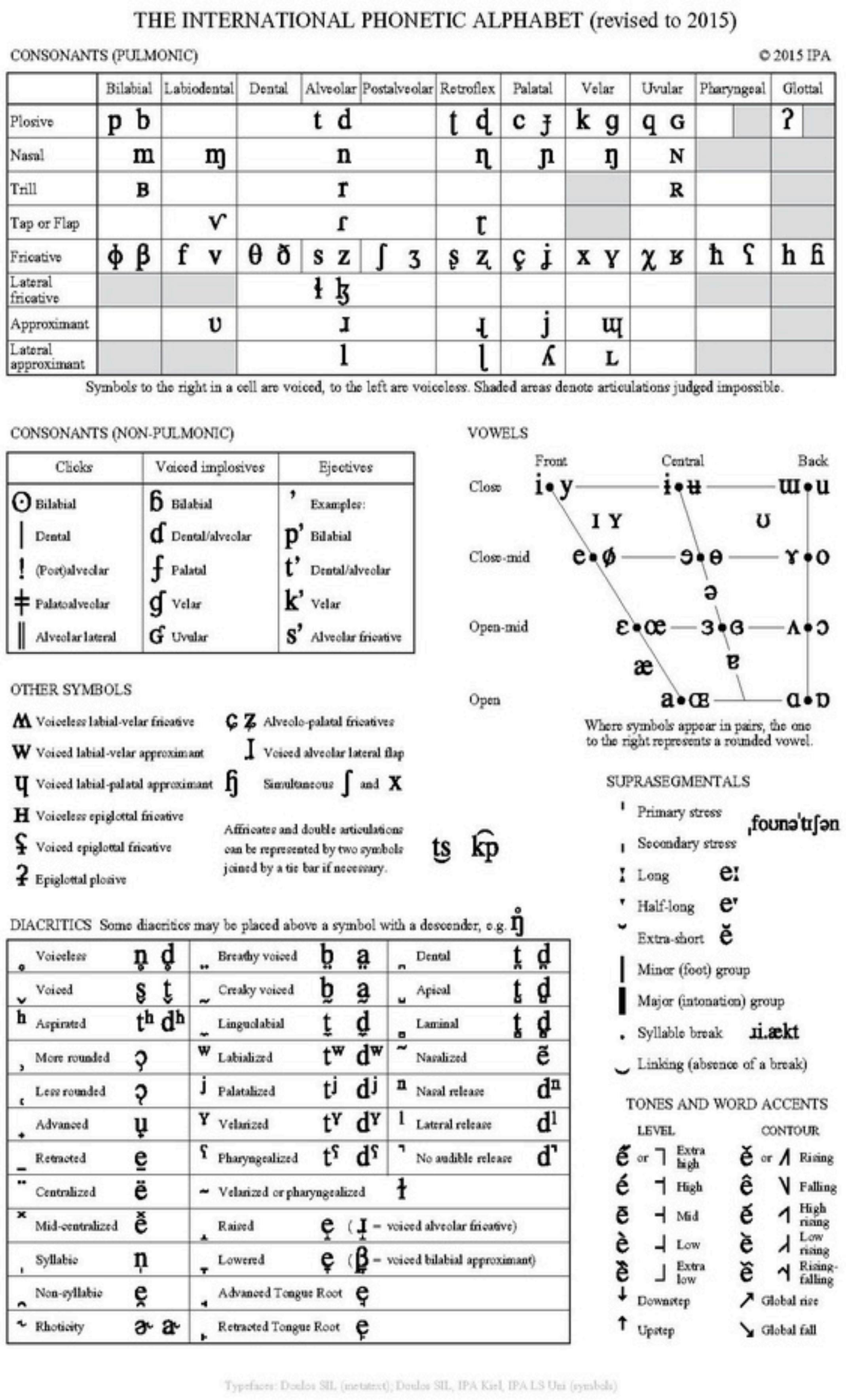
- Although the IPA offers over 160 symbols for transcribing speech, only a relatively small subset of these will be used to transcribe any one language.
- Not all aspects of the alphabet can be accommodated in a chart of the size published by the IPA.

Another Script that works on the similar lines of NaYaNa is Bharati.

Bharati is being proposed a common script for India. The Roman script is used as a common script for many European languages (English, French, German, Italian etc.), which facilitates communication across nations that speak and write those languages. Likewise, a common script for the entire country is hoped to bring down many communication barriers in India. Bharti can be used for 22 Indian languages.

NaYaNa's Approach V/S Bharti's Approach:

- NaYaNa is aimed at making writing dyslexia friendly whereas Bharati does not focus on this aspect.
- NaYaNa is relatively simple to learn since it has a shorter set of characters.



THE INTERNATIONAL PHONETIC ALPHABET (revised to 2015)

CONSONANTS (PULMONIC) © 2015 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
ʘ Bilabial	ɓ Bilabial	ʼ Examples:
ǀ Dental	ɗ Dental/alveolar	pʼ Bilabial
ǃ (Post)alveolar	ʄ Palatal	tʼ Dental/alveolar
ǂ Palatoalveolar	ɠ Velar	kʼ Velar
ǁ Alveolar lateral	ʛ Uvular	sʼ Alveolar fricative

VOWELS

	Front	Central	Back
Close	i • y	ɨ • ʉ	ɯ • u
	ɪ ʏ		ʊ
Close-mid	e • ø	ɘ • ɵ	ɤ • o
		ə	
Open-mid	ɛ • œ	ɜ • ɞ	ʌ • ɔ
	æ	ɐ	
Open	a • ɶ		ɑ • ɒ

Where symbols appear in pairs, the one to the right represents a rounded vowel.

OTHER SYMBOLS

ʍ Voiceless labial-velar fricative

w Voiced labial-velar approximant

ɥ Voiced labial-palatal approximant

ʜ Voiceless epiglottal fricative

ʢ Voiced epiglottal fricative

ʡ Epiglottal plosive

ɕ ʑ Alveolo-palatal fricatives

ɺ Voiced alveolar lateral flap

ɧ Simultaneous ʃ and x

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary. t͡s k͡p

DIACRITICS Some diacritics may be placed above a symbol with a descender, e.g. ŋ̊

Voiceless	n̥ d̥	Breathy voiced	b̤ a̤	Dental	t̪ d̪
Voiced	s̬ t̬	Creaky voiced	b̰ a̰	Apical	t̺ d̺
ʰ Aspirated	tʰ dʰ	Linguolabial	t̼ d̼	Laminal	t̻ d̻
More rounded	ɔ̹	ʷ Labialized	tʷ dʷ	Nasalized	ẽ
Less rounded	ɔ̜	ʲ Palatalized	tʲ dʲ	ⁿ Nasal release	dⁿ
Advanced	u̟	ˠ Velarized	tˠ dˠ	ˡ Lateral release	dˡ
Retracted	e̠	ˤ Pharyngealized	tˤ dˤ	No audible release	d̚
Centralized	ë	Velarized or pharyngealized	ɫ		
Mid-centralized	e̽	Raised	e̝ (ɹ̝ = voiced alveolar fricative)		
Syllabic	n̩	Lowered	e̞ (β̞ = voiced bilabial approximant)		
Non-syllabic	e̯	Advanced Tongue Root	e̘		
˞ Rhoticity	ɚ a˞	Retracted Tongue Root	e̙		

SUPRASEGMENTALS

ˈ Primary stress ˌfoʊnəˈtɪʃən

ˌ Secondary stress

ː Long eː

ˑ Half-long eˑ

˘ Extra-short ĕ

| Minor (foot) group

‖ Major (intonation) group

. Syllable break ɹi.ækt

‿ Linking (absence of a break)

TONES AND WORD ACCENTS

LEVEL		CONTOUR	
e̋ or ˥	Extra high	ě or ˩˥	Rising
é ˦	High	ê ˥˩	Falling
ē ˧	Mid	e᷄ ˦˥	High rising
è ˨	Low	e᷅ ˩˨	Low rising
ȅ ˩	Extra low	e᷈ ˧˦˧	Rising-falling
ꜜ	Downstep	↗	Global rise
ꜛ	Upstep	↘	Global fall

Typefaces: Doulos SIL (metatext); Doulos SIL, IPA Kiel, IPA LS Uni (symbols)

Figure 2.2 IPA Chart

2.3 Hardware & Software Requirements

2.3.1 Hardware and Software Requirements

Table No. 2.1: Minimum Requirements

	Windows
MATLAB V.13 (R2013a)	Windows7
Processor	Dual core, core2duo, Intel I3
RAM	2GB RAM
Disk Space	Disk Space varies depending on size of partition and installation of online help files. The MathWorks Installer will inform you of the hard disk space requirement for your particular partition.
Graphics Adapter	8-bit graphics adapter and display (for 256 simultaneous colours)
CD-ROM drive	For installation from CD

2.3.2 High Level Specifications

- MATLAB V.13(R2013a)
- Intel Dual core or core2duo, Intel I3XP based personal computer
- 2GB RAM recommended
- 8-bit graphics adapter and display (for 256 simultaneous colours). A 32-bit or 64bit OpenGL capable graphics adapter is strongly recommended.

2.3.3 Low Level Specifications

- Microsoft Windows supported graphics accelerator card, printer, and sound card.
- Microsoft Word 8.0 (Office 97), Office 2000.
- TCP/IP is required on all platforms when using a license server.

- Some license types require a license server running FLEXlm 8.0d, which is provided by the Mathworks installer.

Chapter 3

Analysis

3.1 Functional Requirements

- The system should process the input given by the user only if it is an image file (jpg, png etc.).
- System shall show the error message to the user when the input given is not in the required format.
- System should detect characters present in the image.
- System should retrieve characters present in the image and display them to the user.

3.2 Non-Functional Requirements

3.2.1 Requirements

- Performance: Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
- Functionality: This software will deliver on the functional requirements mentioned in this document.
- Availability: This system will retrieve the handwritten text regions only if the image contains written text in it.
- Flexibility: It provides the users to load the image easily.
- Learn ability: The software is very easy to use and reduces the learning work.
- Reliability: This software will work reliably for low resolution images and not for graphical images.

3.2.2 Other Constraints

- This script is valid currently only for English and Hindi languages.
- It does not help with homographs.

3.3 Proposed System

Our system uses the NaYaNa (N@N) script given below.

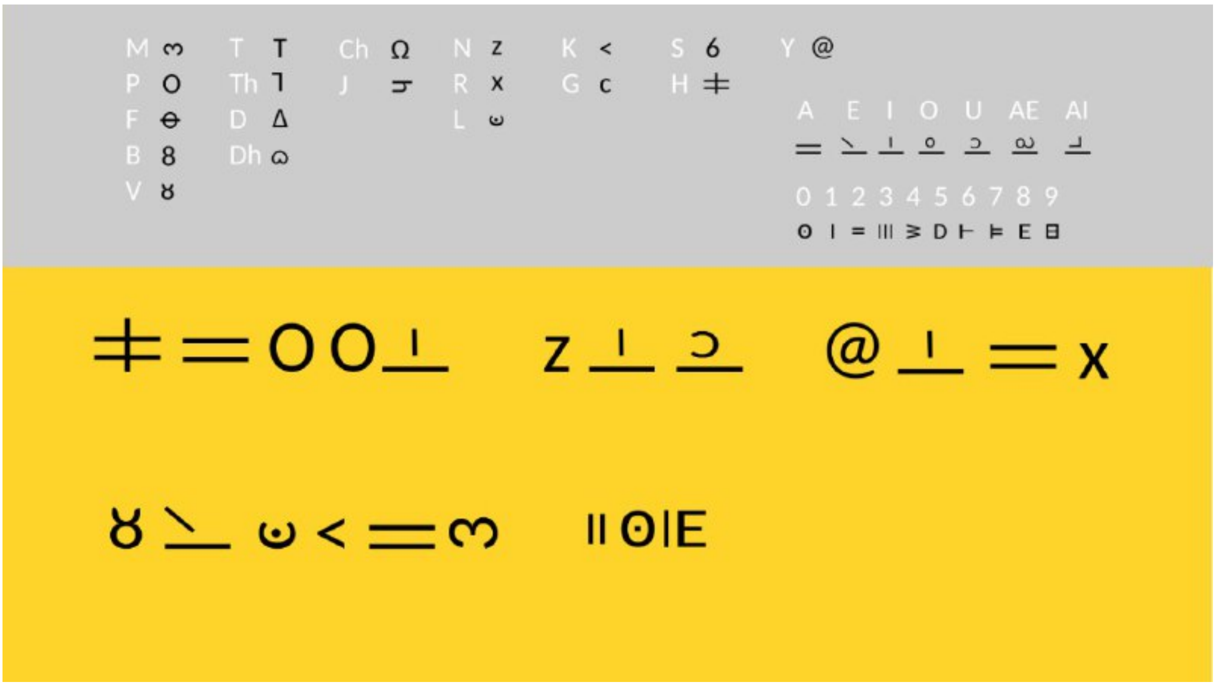


Figure 3.1 NaYaNa Script

Alphabets of N@N script are as shown above.

Vowel sounds are produced by a free passage of breath through the oral cavity. The oral cavity has a static upper palate, let us call it roof, and a motile tongue, the floor. The alphabets representing vowels thus show a gap, to let breath flow.

Consonant sounds are produced by modulated contact between the floor and roof (using tongue, palate, teeth, lips, etc.) with different degrees of passage of breath through oral cavity. Accordingly we have chosen the alphabets for consonants. Consonants alphabets without vowel modifiers (on the roof of the consonants) will be considered as the first default sound of =. Sharp sounds have sharp edges in the shape, while softer ones have round edges.

They are produced by a combination of a consonant with ‘h’. Instead of using a new alphabet for ‘f’ we use Φ a combination of o and ≠. The strikethrough can be horizontal or vertical depending on the shape of the character.

In the current version of the alphabet, we do not have any more rules. Using the above, we can write most of the vocalisations. We know that human variations in speech are very rich. So, by following the basic principles of N@N script, we can extend it as and when required.

Our proposed system uses optical character recognition (OCR) that supports recognition of characters of the N@N script. Along with OCR, our system will also use neural networks in order to train our system to recognise the different orientations of a letter. This system takes as input text in N@N script and produces the corresponding text in English.

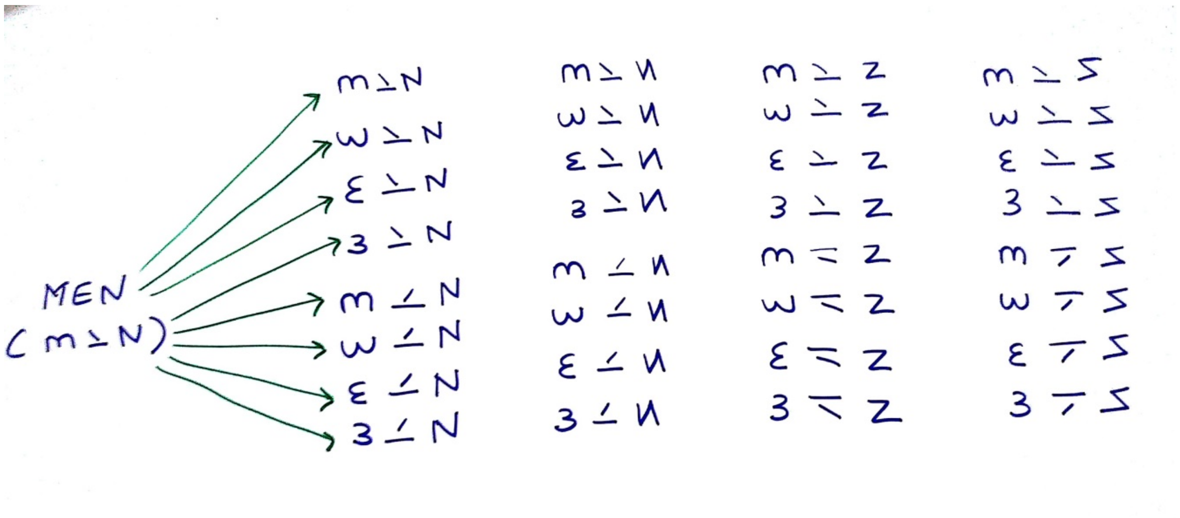


Figure No. 3.2: Multiple orientations of alphabets in a word.

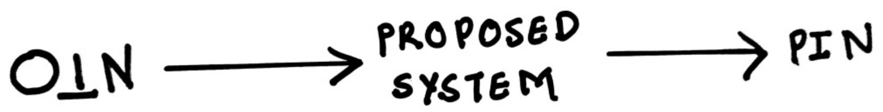


Figure 3.3 Input (image) and Output of the proposed system

Keyboard mapping will be used here in order to be able to type this language onto a computer. A well designed keyboard UI is an important aspect of this system as the end user must be able to type the N@N script successfully in order to get the correct output.

Chapter 4

Design

4.1 Design Consideration

This is a script designed with some principles to make the script easy to read. The main motivation for designing this script is to make writing dyslexic friendly.

Design Principles

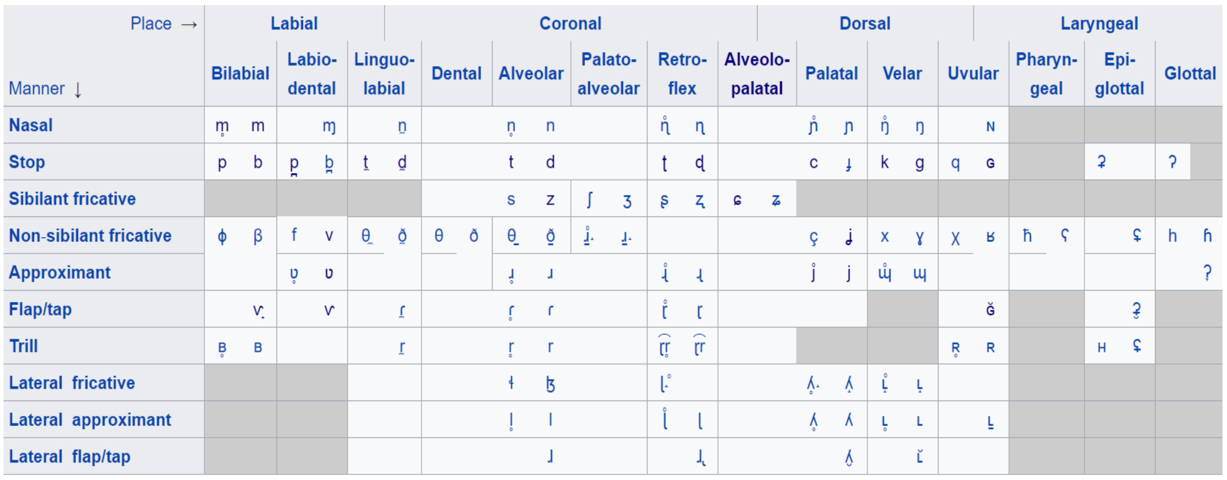
1. Easy to learn.
2. Easy to read and write.
3. Characters designed are chosen on the basis of unique rotational and mirror symmetry. This makes them accessible to persons with this form of dyslexia. That is, if 'd' is used, 'p' and 'b' will not be used for any other phoneme. Similarly each of the following pairs of characters, 'N' and 'Z', 'M' and 'W', 'n' and 'u' etc. would represent the same phonemes if they were used in Z@Z.
4. The shape of the character has some metaphorical similarity to the shape and form of the vocal apparatus (lips, tongue, teeth, palate, cheeks and buccal cavity).
5. The shape also corresponds with the quality of sound of the phoneme such as sharp sounds having sharp edged characters, open sounds having open characters etc.
6. A one to one mapping of the characters is made with IPA as per the need. Since IPA has a defined Unicode space, this ensures that any designed font for Z@Z can be used on any Unicode compliant computer.
7. It is designed to be visually accessible to human beings primarily. However it is elegantly accessible for machines for text to speech (due to IPA compatibility) and OCR due to the unique and unambiguous shapes.
8. The shapes may resemble some commonly used characters from other languages, but do not necessarily represent similar sounds.
9. This script does not differentiate upper case and lower case.[4]

We shall apply the above design principles and generate the script.

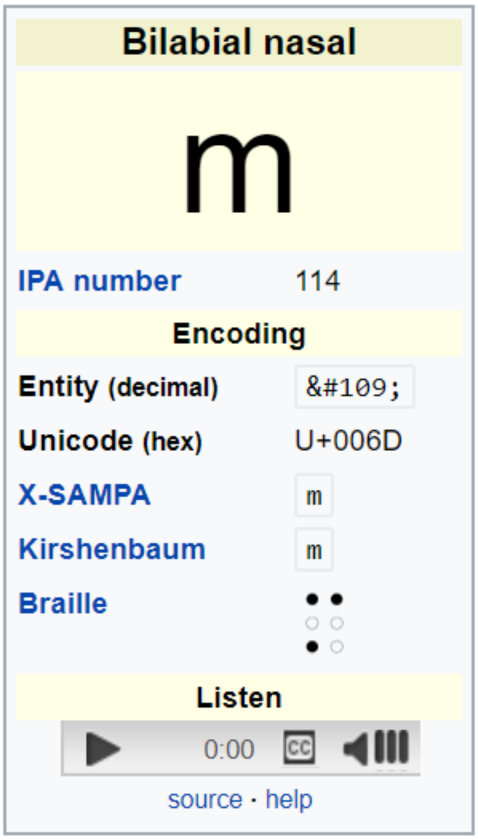
4.2 Design Details

We will click on a particular symbol, say nasal bilabial ‘m’, from the IPA with sound vocalisations. Then, on the page which opens, listen to the sound by clicking on play (as indicated in the picture below).

Table No. 4.1: Wikipedia Vocalisations



Place →	Labial			Coronal					Dorsal			Laryngeal		
Manner ↓	Bilabial	Labio-dental	Linguo-labial	Dental	Alveolar	Palato-alveolar	Retro-flex	Alveolo-palatal	Palatal	Velar	Uvular	Pharyn-geal	Epi-glottal	Glottal
Nasal	m̥ m	ɱ	n̼		n̥ n		ɳ̊ ɳ		ɲ̊ ɲ	ŋ̊ ŋ	ɴ			
Stop	p b	p̪ b̪	t̼ d̼		t d		ʈ ɖ		c ɟ	k ɡ	q ɢ		ʡ	ʔ
Sibilant fricative					s z	ʃ ʒ	ʂ ʐ	ɕ ʑ						
Non-sibilant fricative	ɸ β	f v	θ̼ ð̼	θ ð	θ̠ ð̠	ɹ̠̊˔ ɹ̠˔			ç ʝ	x ɣ	χ ʁ	ħ ʕ	ʢ	h ɦ
Approximant		ʋ̥ ʋ			ɹ̥ ɹ		ɻ̊ ɻ		j̊ j	ɰ̊ ɰ				ʔ̞
Flap/tap	ⱱ̟	ⱱ	ɾ̼		ɾ̥ ɾ		ɽ̊ ɽ				ɢ̆		ʡ̆	
Trill	ʙ̥ ʙ		r̼		r̥ r		ɽ͡r̥ ɽ͡r				ʀ̥ ʀ		ʜ ʢ	
Lateral fricative					ɬ ɮ		ɭ̊˔		ʎ̝̊ ʎ̝	ʟ̝̊ ʟ̝				
Lateral approximant					l̥ l		ɭ̊ ɭ		ʎ̥ ʎ	ʟ̥ ʟ	ʟ̠			
Lateral flap/tap					ɺ		ɺ̢		ʎ̆	ʟ̆				



Bilabial nasal

m

IPA number 114

Encoding

Entity (decimal) &#109;

Unicode (hex) U+006D

X-SAMPA m

Kirshenbaum m

Braille ⠍

Listen

Figure 4.1 Bilabial Nasal m

If the sound of this consonant matches with the vocalisation ‘muh’, we take this Unicode. Then, in FontForge, in the cube which has the same Unicode as this, we insert the NaYaNa letter. This is done for every NaYaNa vowel and consonant.[3]

Table No. 4.2: N@N letters with unicodes

Unicode (hex)	IPA number	Appearance in n@n	Description
U+006D	114	m	m in Monday
U+006E	116	Z	n in n@n(naa)
U+0070	101	o	p in puppy
U+0288	105	T	t in tongue
U+0072	122	x	r in run
U+006B	109	<	c in cut,cup
U+0261	110	c	g in gun
U+0062	102	8	b in number, boy
U+0077	170	alpha inverted	w in water, wonder
U+0074, U+0361,U+0283,U+02A8	130,134,215	omega	ch in church
U+0064	104	heart	d in deepa
U+0256	106	delta	d dust,dine
U+029D	139	@	ya in NaYaNa
U+0073	132	Six(6)	s in sun,sand
U+0073, U+033A		sh	sh in shack,shine
U+0278	126	phi	phi in file
U+03B8	130	theta	th in teeth,thin
U+006C	155	la	l in let,luck
U+0068	146	ha	h in hut
U+0061	304	a	a in sugar
U+0069	301	i	i in pin
U+0075	308	u	u in put
U+0065	302	e	e in pen or pet
U+006F	307	o	o in no

For drawing the letter ‘m’, we use the software Inkscape. Similarly, we draw all our NaYaNa vowels and consonants on Inkscape first. For every letter, (.svg) format is created on Inkscape. This (.svg) file is then imported in the FontForge cube which has the same Unicode as the letter.

After all (.svg) files, for every vowel and every consonant of NaYaNa, are imported in FontForge, the (.sfd) file is created. This file is then converted to (.ttf) format for keyboard mapping. We shall use this .ttf file for keyboard mapping.

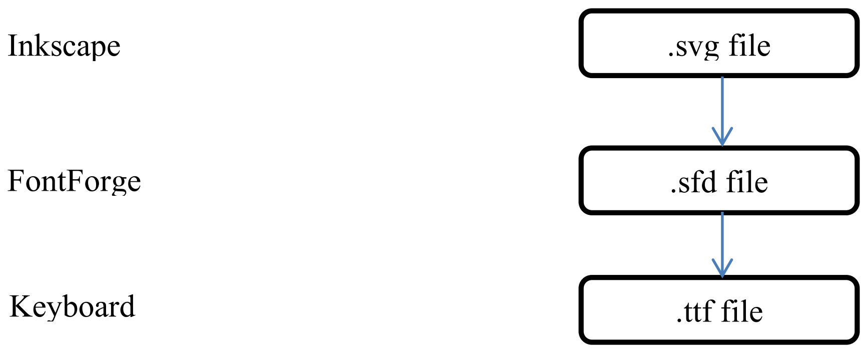


Figure 4.2 Task Flow of creating input method

4.2.1 Gradient Feature Extraction

The gradient measures the magnitude and direction of the greatest change in intensity in a small neighbourhood of each pixel. (In what follows, "gradient" refers to both the gradient magnitude and direction). Gradients are computed by means of the Sobel operator. The Sobel templates used to compute the horizontal (X) & vertical (Y) components of the gradient are shown in Fig.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Horizontal Component and Vertical Component

Figure 4.3 Sobel Masks for gradient

Given an input image of size D1×D2, each pixel neighbourhood is convolved with these templates to determine these X and Y components, Sx and Sy, respectively. Eq. (1) and (2) represents their mathematical representation:

1) $S(i, j) = I(i-1, j+1) + 2 * I(i, j+1) + I(i+1, j+1) - I(i-1,j-1) - 2*I(i,j-1) - I(i+1,j-1).$
2) $S(i, j) = I(i-1, j-1) + 2* I(i-1, j) + I(i-1, j+1)\ y\ - I(i+1,j-1) - 2* I(i+1, j) - I(i+1, +1)$

Here, (i, j) range over the image rows (D1) and columns (D2), respectively. The gradient strength and direction can be computed from the gradient vector [Sx, Sy]. After obtaining gradient vector of each pixel, the gradient image is decomposed into four orientation planes or eight direction planes (chain code directions) as shown in the figure below.[5]

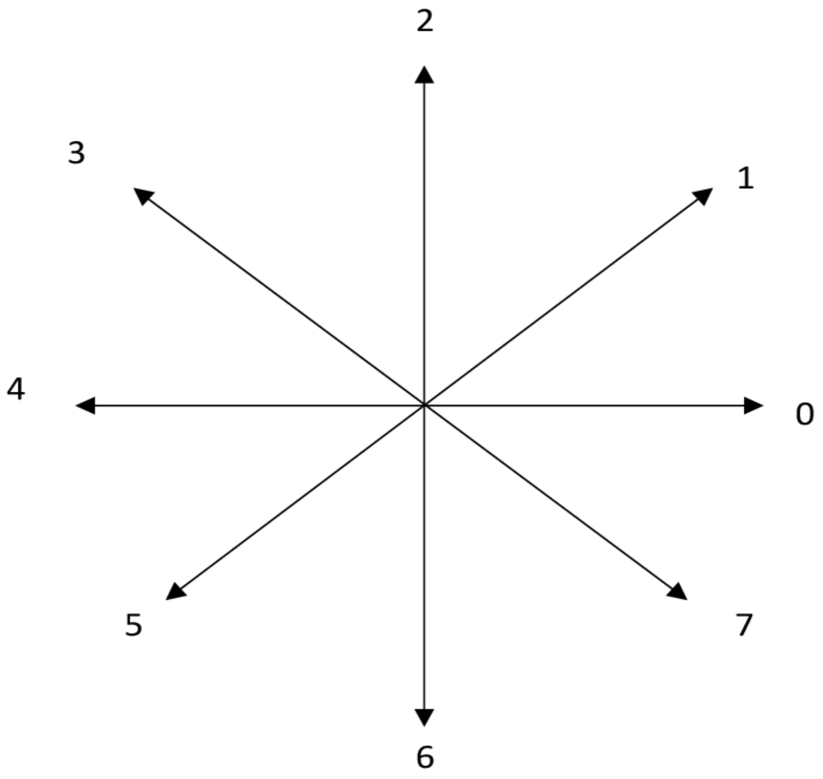


Figure 4.4 Direction of chain codes

<u>Generation of Gradient Feature Vector:</u>

A gradient feature vector is composed of the strength of gradient accumulated separately in different directions as described below:

(1) The direction of gradient detected as above is decomposed along 8 chain code directions.
(2) The character image is divided into 81(9 horizontal × 9 vertical) blocks. The strength of the gradient is accumulated separately in each of 8 directions, in each block, to produce 81 local spectra of direction.

(3) The spatial resolution is reduced from 9×9 to 5×5 by down sampling every two horizontal and every two vertical blocks with 5×5 Gaussian Filter to produce a feature vector of size 200 (5 horizontal, 5 vertical, 8 directional resolution).

(4) The variable transformation (y = x0.4) is applied to make the distribution of the features Gaussian-like. The 5 × 5 Gaussian Filter used is the high cut filter to reduce the aliasing due to the down sampling.[5]

4.2.2 Classification

Artificial Neural Network

Animals recognize various objects and make sense out of large amount of visual information, apparently requiring very little effort. Simulating the task performed by animals to recognize to the extent allowed by physical limitations will be enormously profitable for the system. This necessitates study and simulation of Artificial Neural Network. In Neural Network, each node perform some simple computation and each connection conveys a signal from one node to another labelled by a number called the “connection strength” or weight indicating the extent to which signal is amplified or diminished by the connection.

Different choices for weight results in different functions are being evaluated by the network. If in a given network whose weight are initial random and given that we know the task to be accomplished by the network , a learning algorithm must be used to determine the values of the weight that will achieve the desired task. Learning Algorithm qualifies the computing system to be called Artificial Neural Network. The node function was predetermined to apply specific function on inputs imposing a fundamental limitation on the capabilities of the network. Typical pattern recognition systems are designed using two pass. The first pass is a feature extractor that finds features within the data which are specific to the task being solved (e.g. finding bars of pixels within an image for character recognition). The second pass is the classifier, which is more general purpose and can be trained using a neural network and sample data sets. Clearly, the feature extractor typically requires the most design effort, since it usually must be hand-crafted based on what the application is trying to achieve. Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Once the network is trained, the match pattern is obtained to generate the associated character. Output will be the beautified version of the uploaded image and will be saved in a .doc or in text file.[7]

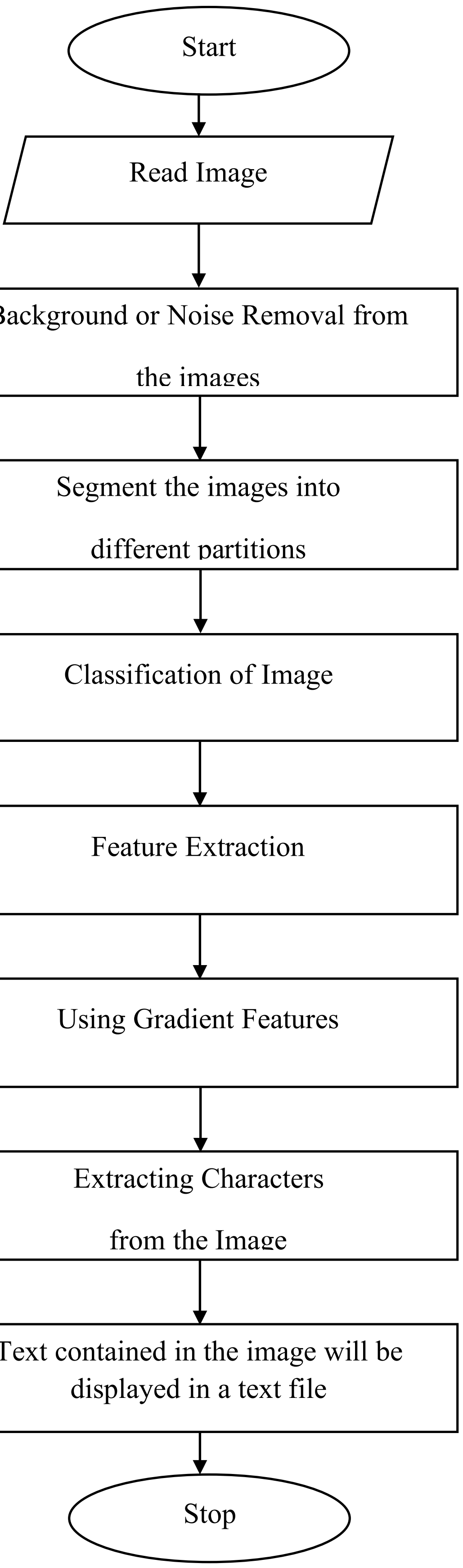

Start
Read Image
Background or Noise Removal from the images
Segment the images into different partitions
Classification of Image
Feature Extraction
Using Gradient Features
Extracting Characters from the Image
Text contained in the image will be displayed in a text file
Stop

Figure 4.5 Flowchart

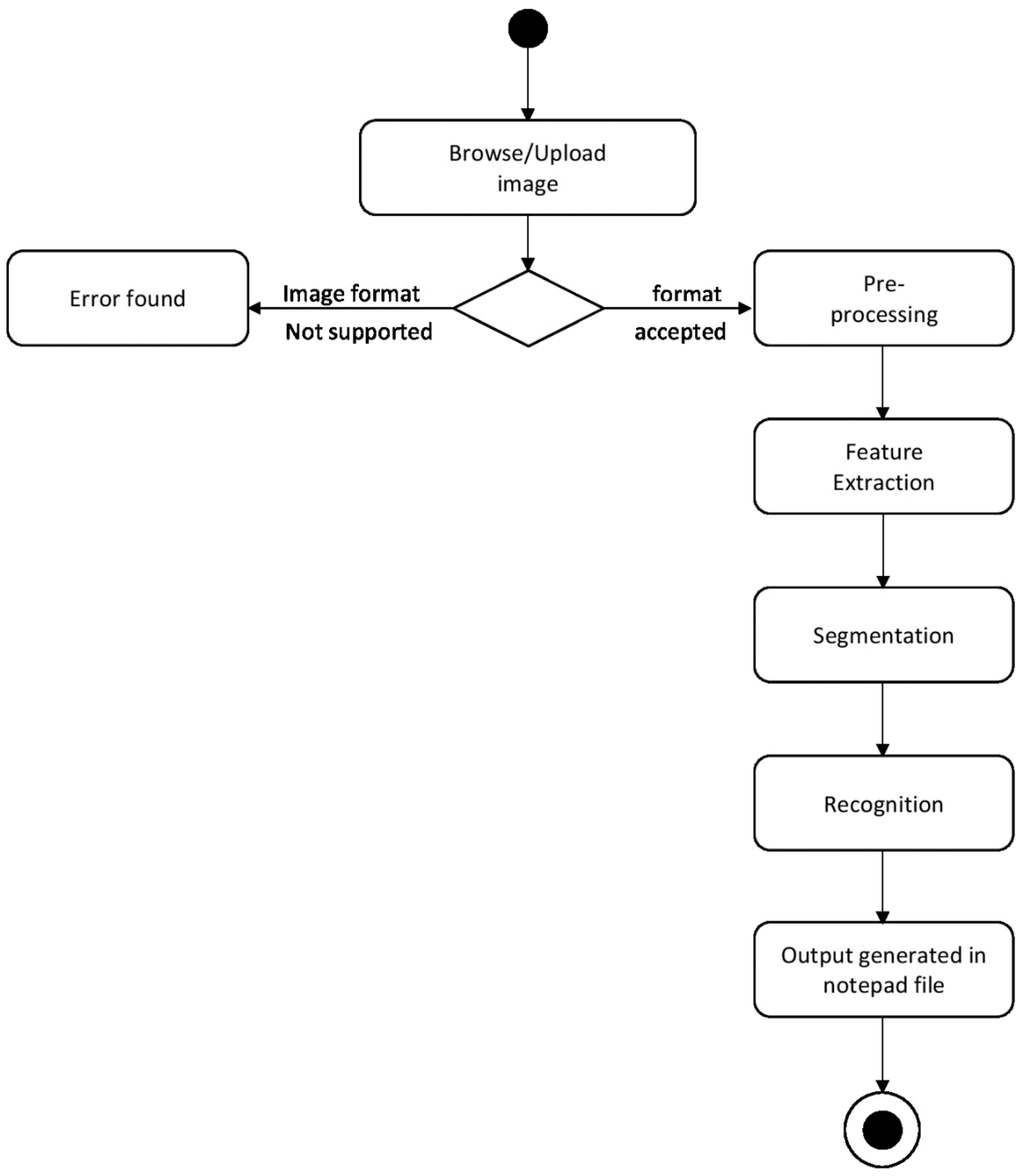


Figure 4.6 Activity Diagram

4.3 Design

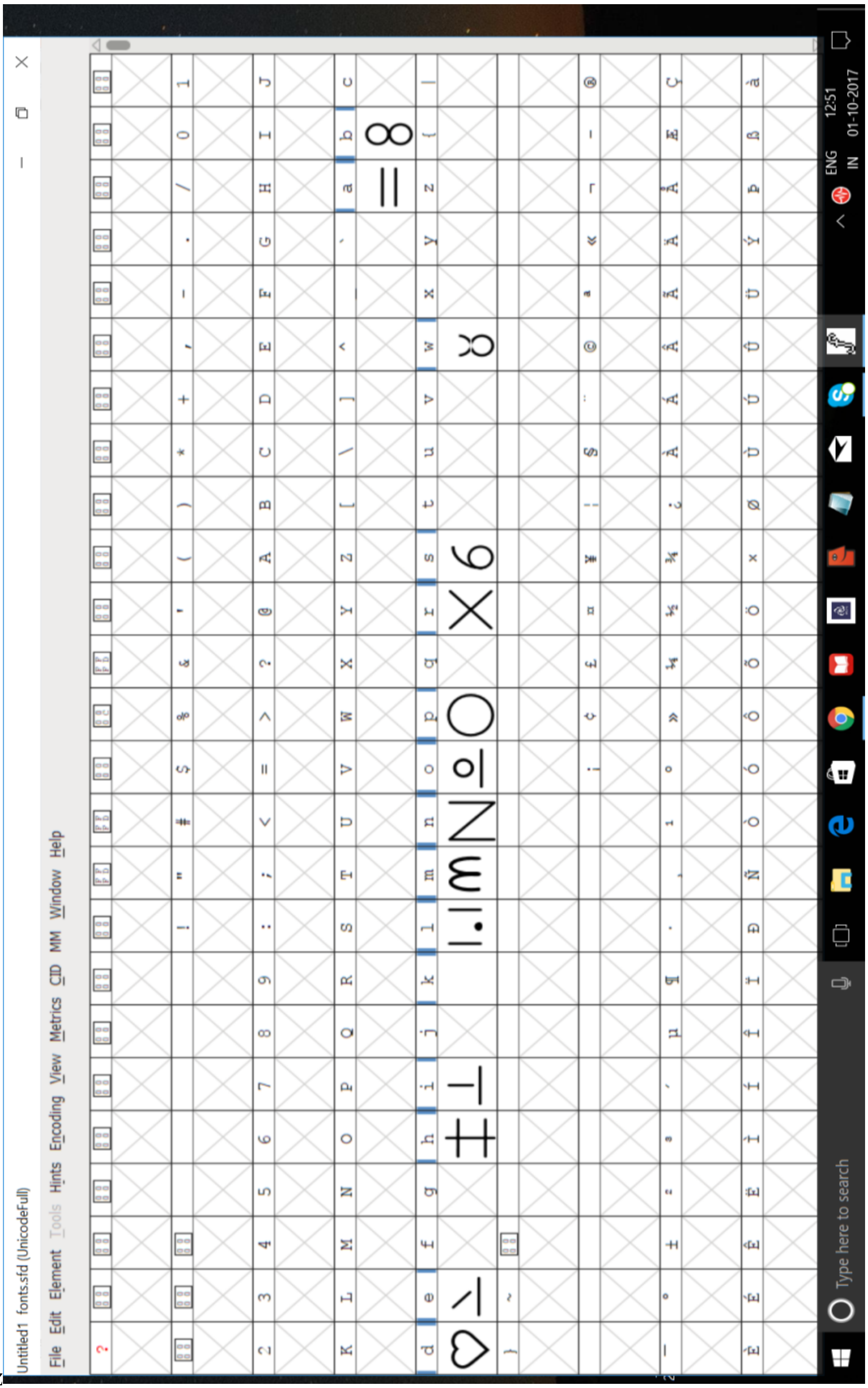

Untitled1 fonts.sfd (UnicodeFull)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help
Type here to search
ENG IN
12:51 01-10-2017

Figure 4.7 Fontforge 1

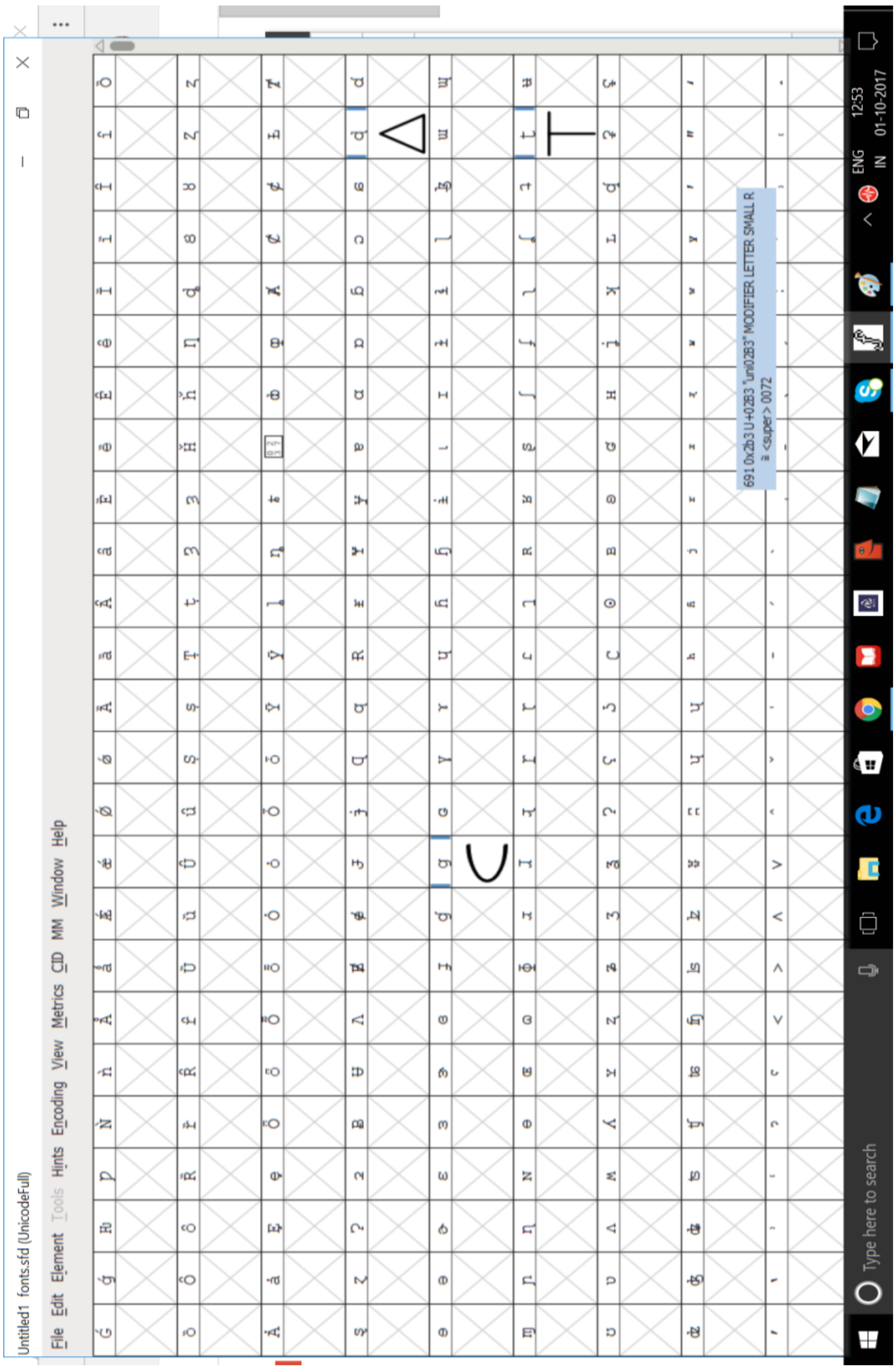
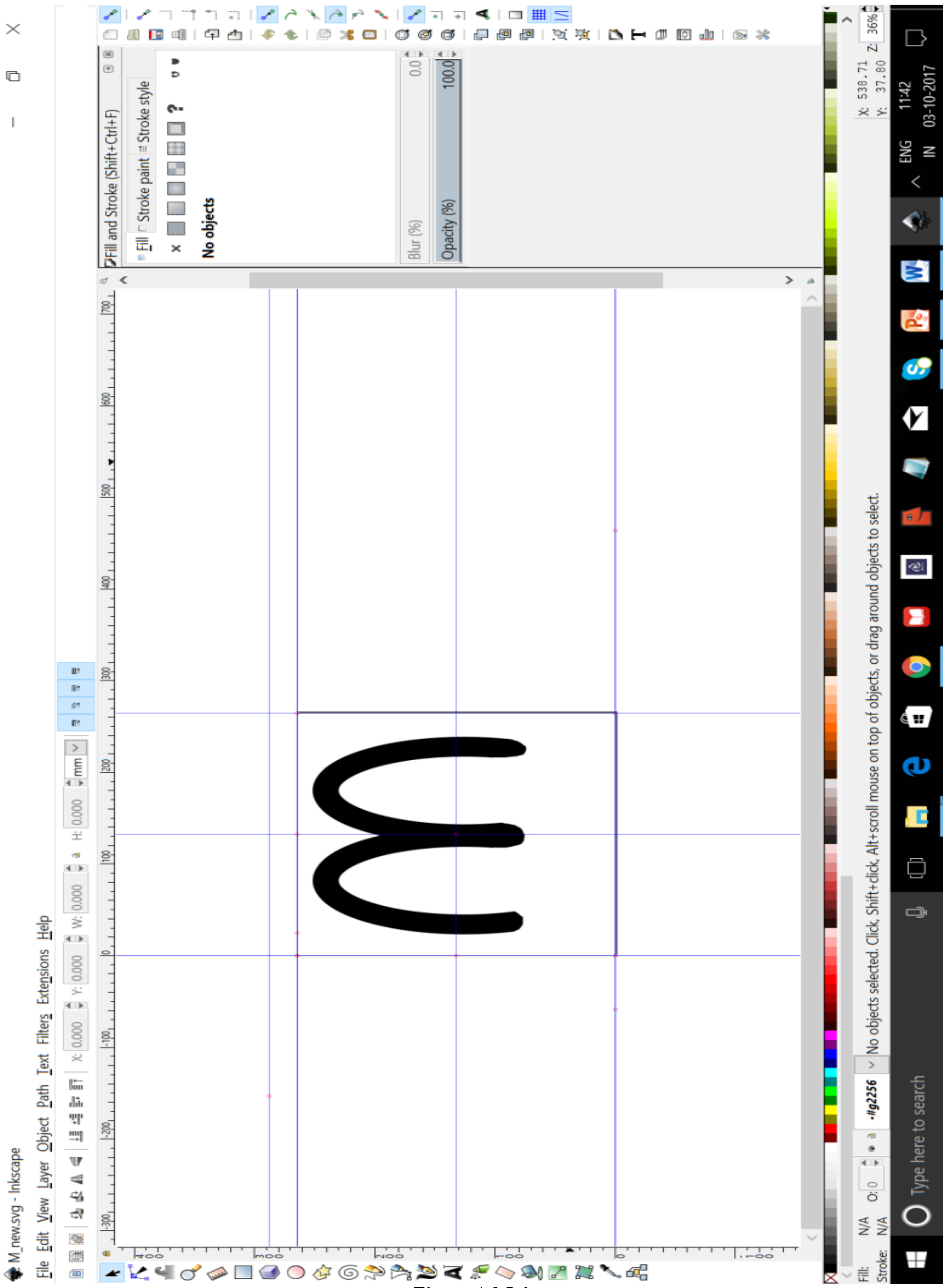

Untitled1 fonts.sfd (UnicodeFull)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help
691 0x2b3 U+02B3 "uni02B3" MODIFIER LETTER SMALL R
ř <super> 0072
Type here to search
ENG IN
12:53 01-10-2017

Figure 4.8 Fontforge 2


M_new.svg - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X: 0.000 Y: 0.000 W: 0.000 H: 0.000 mm
Fill and Stroke (Shift+Ctrl+F)
Fill Stroke paint Stroke style
No objects
Blur (%) 0.0
Opacity (%) 100.0
Fill: N/A Stroke: N/A O: 0
#g2256
X: 538.71 Y: 37.80 Z: 36%
Type here to search
ENG IN 11:42 03-10-2017

No objects selected. Click, Shift+click, Alt+scroll mouse on top of objects, or drag around objects to select.

Figure 4.9 Inkscape

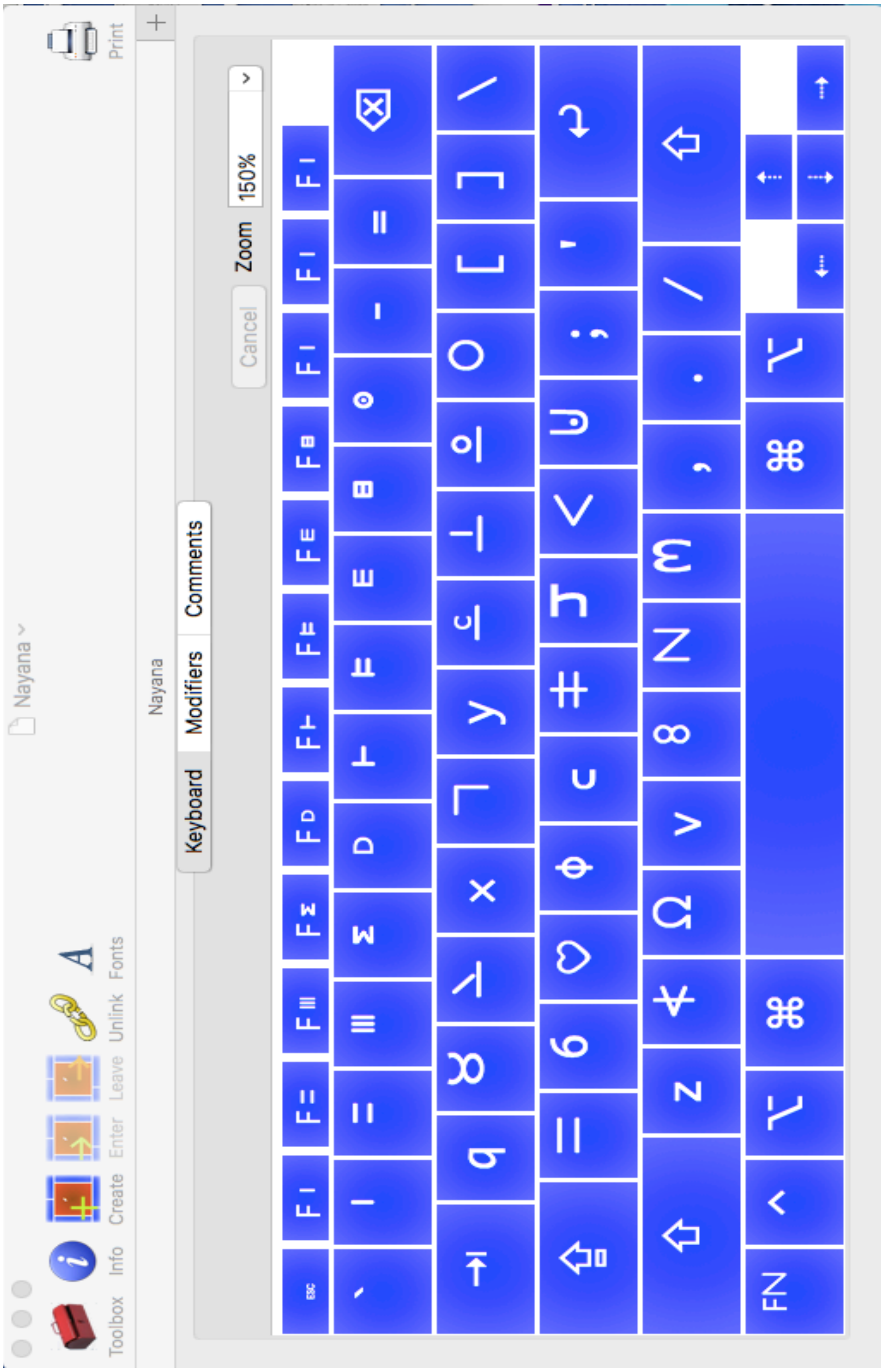


Figure 4.10 Keyboard

Chapter 5

Implementation

5.1 Software Development Platform

5.1.1 Neural Network Structure

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

MATLAB Image Processing Toolbox

We have used MATLAB Image Processing Toolbox for the development of this software. Image processing involves changing the nature of an image in order to improve pictorial information of the image for human interpretation for autonomous human perception. The Image Processing Toolbox is a collection of functions that extend the capability of the

MATLAB numeric computing environment. The toolbox supports a wide range of operations on the image.

Key Features

- Image enhancement, including filtering, filters design, deblurring and contrast enhancement.
- Image analysis including features detection, morphology, segmentation, and measurement.
- Spatial transformations and image registration.
- Support for multidimensional image processing.
- Support for ICC version 4 colour management system.
- Modular interactive tools including ROI selection, histograms and distance measurements
- Interactive image and video display.
- DICOM import and export.

<u>MATLAB Neural Network Toolbox</u>

Key Features

- Supervised networks, including multilayer, radial basis, learning vector quantization (LVQ), time-delay, nonlinear autoregressive (NARX), and layer- recurrent
- Unsupervised networks, including self-organizing maps and competitive layers
- Apps for data-fitting, pattern recognition, and clustering
- Parallel computing and GPU support for accelerating training (using Parallel Computing Toolbox)
- Pre-processing and post processing for improving the efficiency of network training and assessing network performance
- Modular network representation for managing and visualizing networks of arbitrary size
- Simulink® blocks for building and evaluating neural networks and for control systems applications[5]

<u>Working of the Front End</u>

When you save your GUI layout, GUIDE automatically generates an M-file that you can use to control how the GUI works. This M-file provides code to initialize the GUI and contains a framework for the GUI call-backs the routines that execute in response to user-generated events such as a mouse click. Using the M-file editor, you can add code to the callbacks to perform the functions you want.

Home Page

1. The system displays the Home page with some options.

2. The user will browse for the input image.

3. User clicks on LOAD IMAGE Button to upload the image.

Processing Module

1. This GUI will receive the query image.

2. Displays the noiseless image of the uploaded image.

3. The characters are extracted from the image and displayed. 4. Output will be displayed in a .txt /.doc file.

5.2 Screen Shot of the Front End

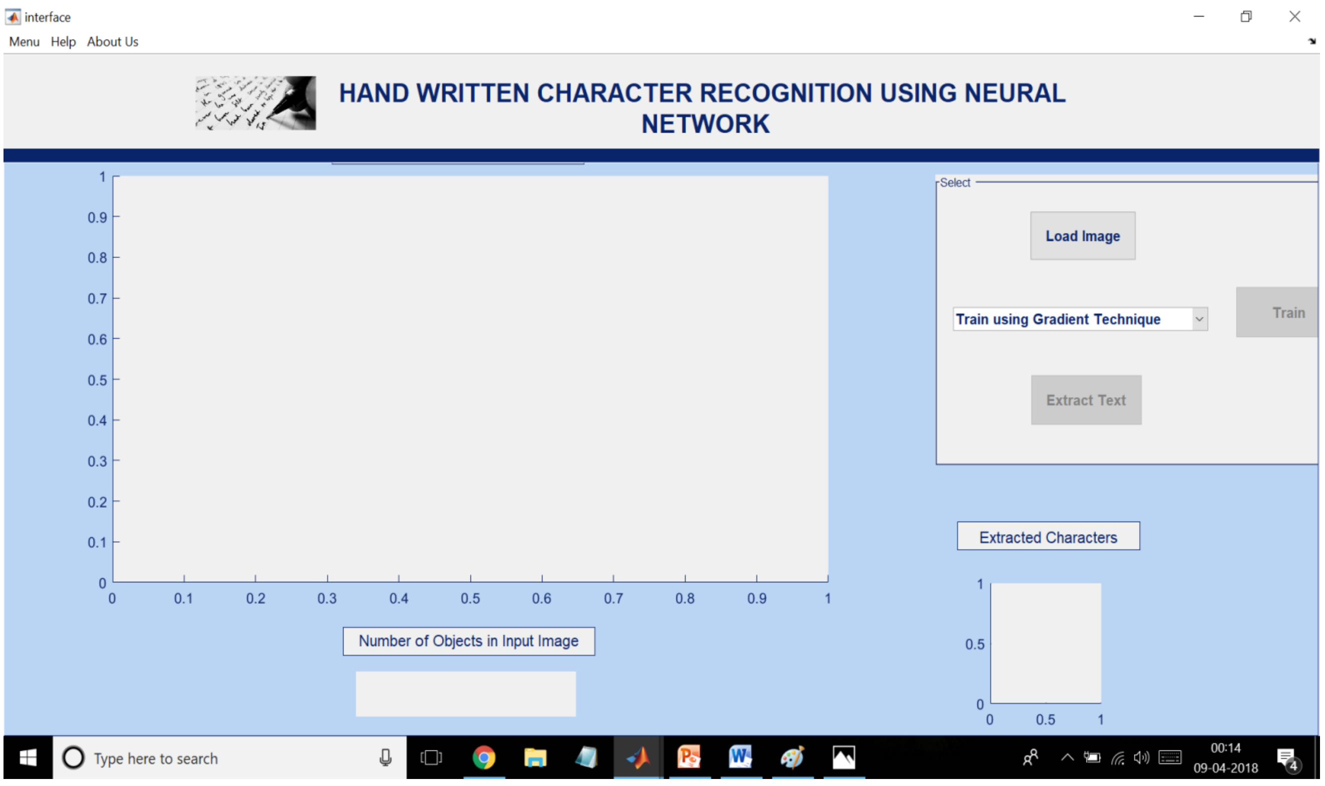

interface
Menu Help About Us
HAND WRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK
Select
Load Image
Train using Gradient Technique
Train
Extract Text
Extracted Characters
Number of Objects in Input Image
Type here to search
00:14
09-04-2018

Figure 5.1 GUI Working Step 1

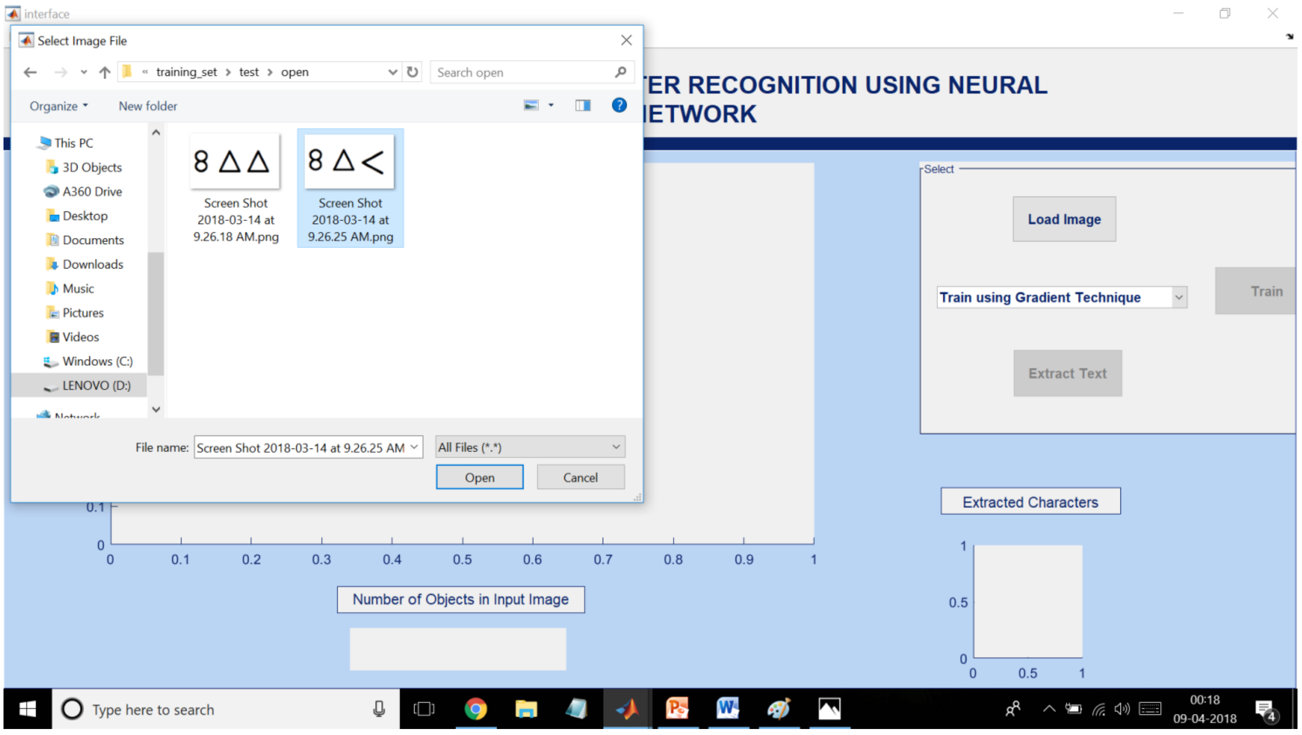


Figure 5.2 GUI Working Step 2

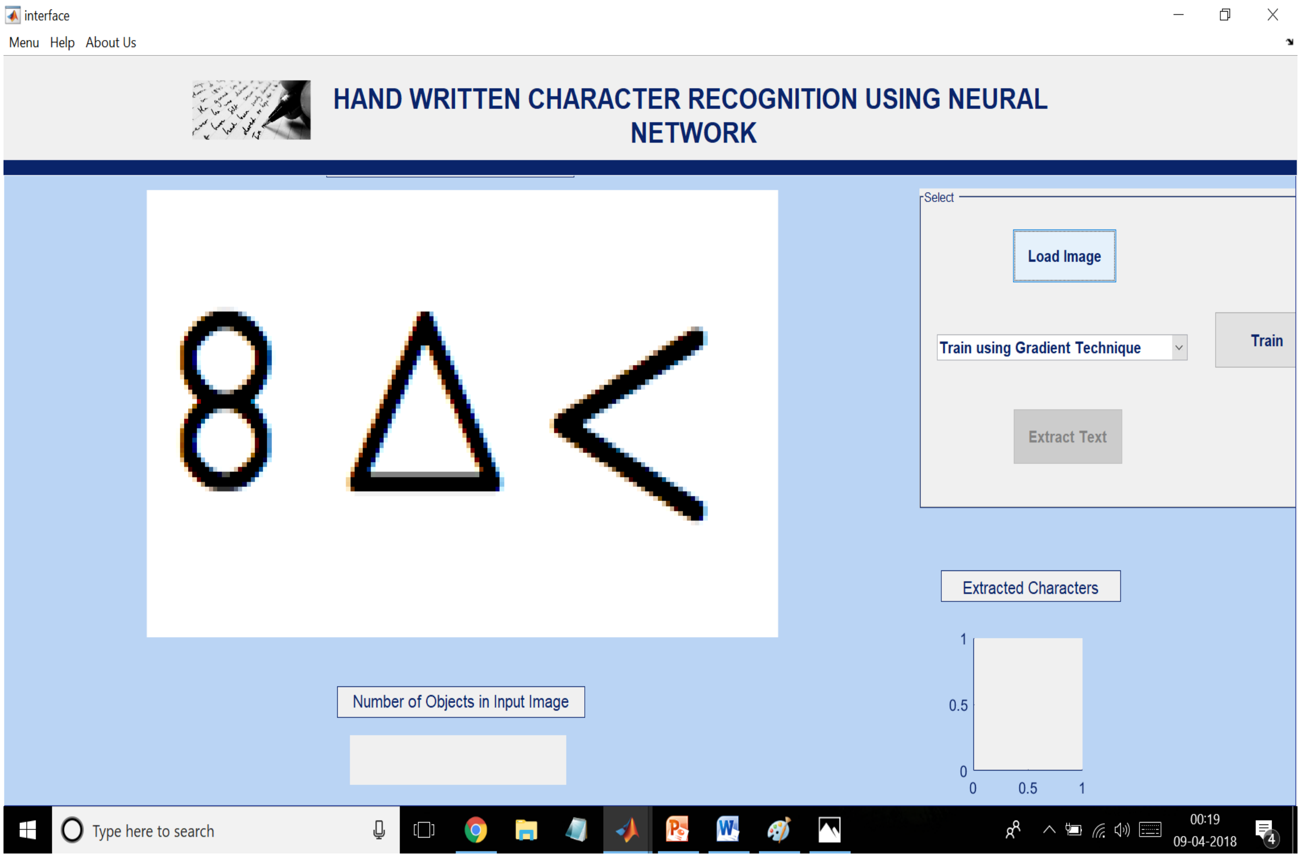

interface
Menu Help About Us
HAND WRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK
Select
Load Image
Train using Gradient Technique
Train
Extract Text
Extracted Characters
Number of Objects in Input Image
Type here to search
00:19
09-04-2018

Figure 5.3 GUI Working Step 3

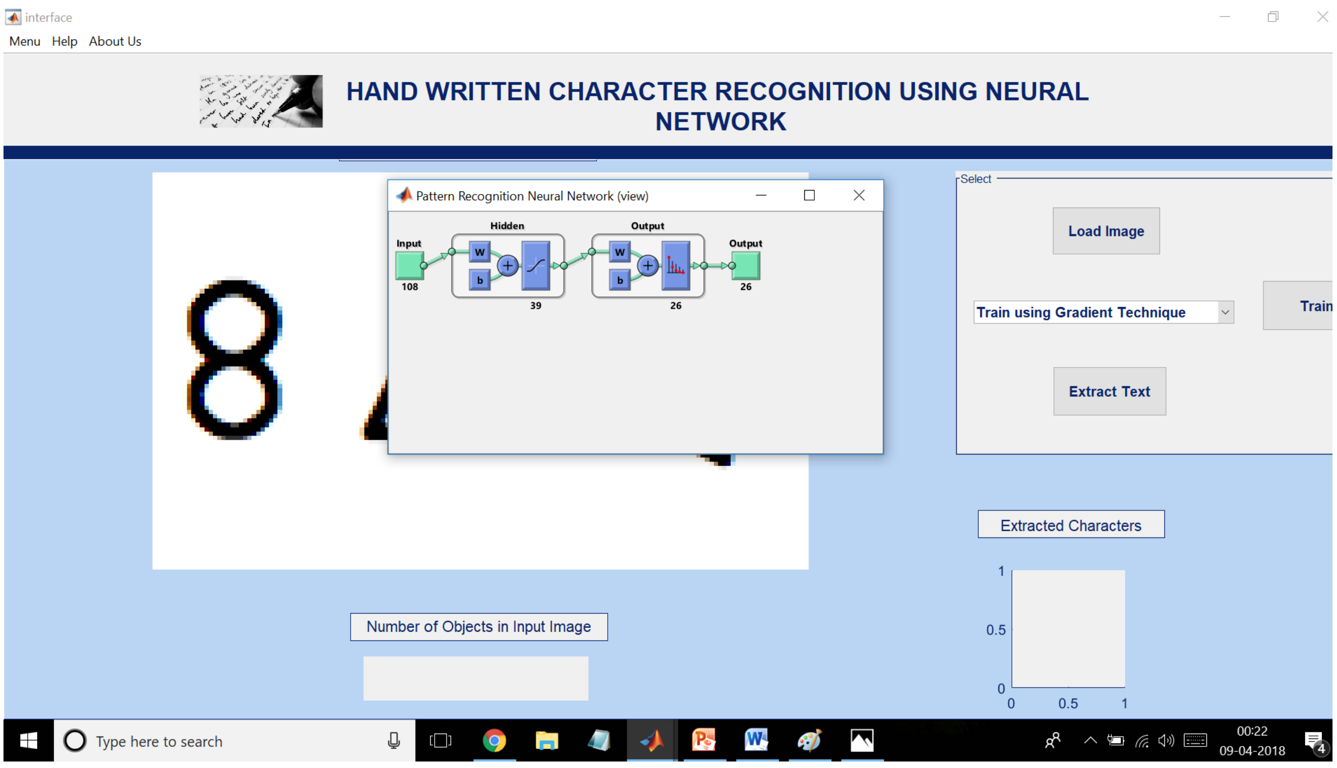

interface
Menu Help About Us
HAND WRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK
Pattern Recognition Neural Network (view)
Hidden
Output
Input
108
39
26
Output
26
Select
Load Image
Train using Gradient Technique
Train
Extract Text
Extracted Characters
Number of Objects in Input Image
Type here to search
00:22
09-04-2018

Figure 5.4 GUI Working Step 4

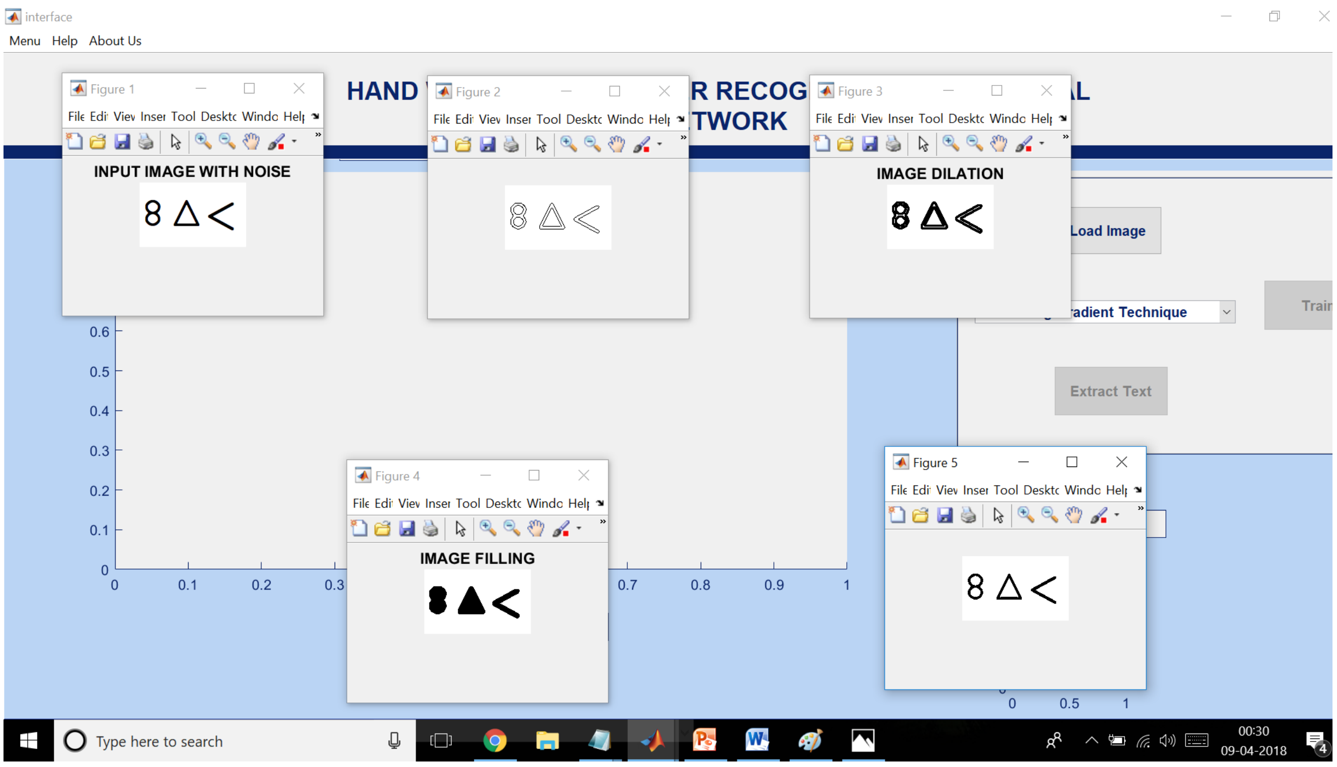


Figure 5.5 GUI Working Step 5

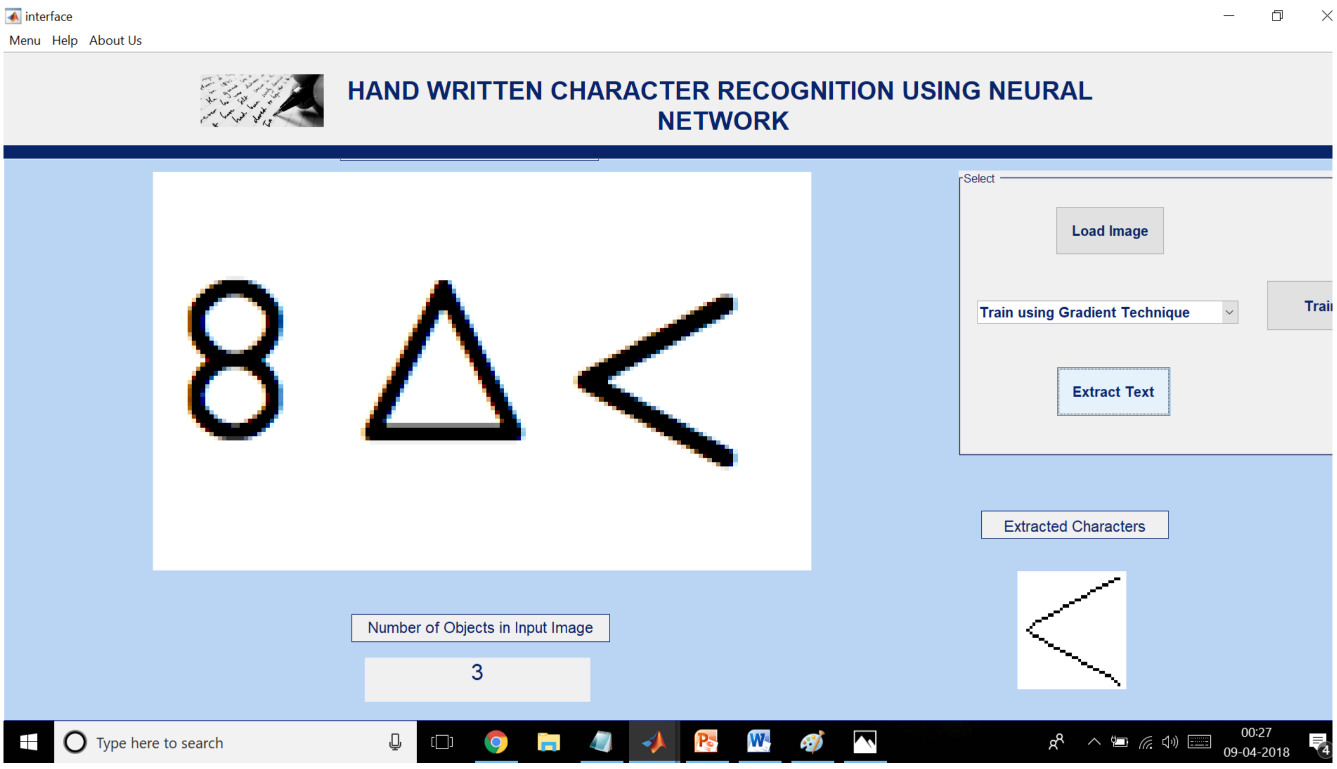
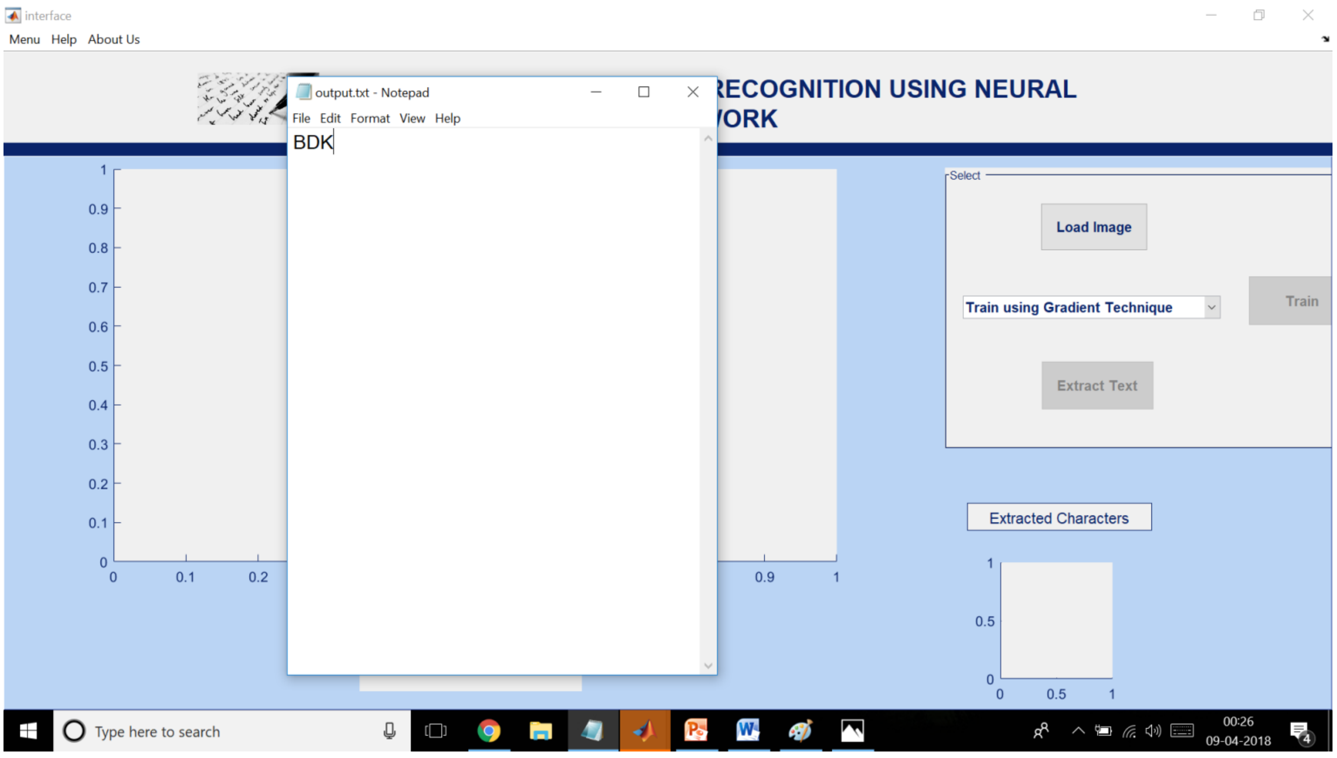

interface
Menu Help About Us
HAND WRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK
Select
Load Image
Train using Gradient Technique
Extract Text
Extracted Characters
Number of Objects in Input Image
3
Type here to search
00:27
09-04-2018

Figure 5.6 GUI Working Step 6



```
BDK
```

Figure 5.7 GUI Working Step 7

5.3 Testing

5.3.1 Verification

The set of Test Cases are used to test the functionality of each module; if that module works properly then that Test Cases marked as Pass or else Fail.

Table No. 5.1: Test cases

Test Id.	**Test Case**	**Input Description**	**Expected Output**	**Test Status**
1	Uploading Image	When user clicks on open button, open field box will be opened to select image file	Image file should be selected and uploaded	Pass
2	To pre-process images	Image will be taken for pre-processing	Conversion from RGB to B/W image (binarisation)	Pass
3	Feature Extraction	A Grey Scale Image	Character features should be extracted	Pass
4	Output File	Normalised character to the neural network	File containing only text	Pass

5.3.2 Validation

The below table is used to determine whether or not a system satisfies the acceptance criteria and to determine whether or not to accept the system.

Table No. 5.2: Functions performed

SI no.	Functions	Required Output	Actual Output
1	Upload the image with valid format	Image should be uploaded if supported	Valid image is uploaded successfully
2	Invalid image format	Error message should be displayed	Error message is displayed if the image format is not supported
3	Pre-processing of the uploaded image	Image should be pre-processed in order to convert to Gray scale	Image is pre-processed
4	Extraction of features	Character features such as edges and curves are calculated	Image features are extracted
5	Displaying result	Text of the file displayed	Text contained in the file is displayed

5.3.3 Failure modes and action on failure

Table No. 5.3: Possible events

SI No.	Event	Action
1.	Wrong input file uploaded	Error message should be displayed to the user and home screen must be displayed
2.	System shut down	Tasks should be cancelled, and process should be restarted again

5.3.4 Evaluation

The Handwritten Character Recognition system was tested on several different scanned images containing handwritten text with different styles and the results were highly encouraging.

- The proposed method performs pre-processing on the image for removing the noise and further uses feature extraction using gradient technique which gives relatively good classification compared to OCR.
- The method is advantageous as it uses twelve features to train the neural network using gradient technique. The advantage lies in less computation involved in feature extraction, training and classification phases of the method.
- The proposed methodology has produced good results for images containing handwritten text written in different styles, different size and alignment with varying background. It classifies most of the handwritten characters correctly if the image contains less noise in the characters and also in the background. Characters written with legible handwriting are classified more accurately.

5.4 Perceptron Model

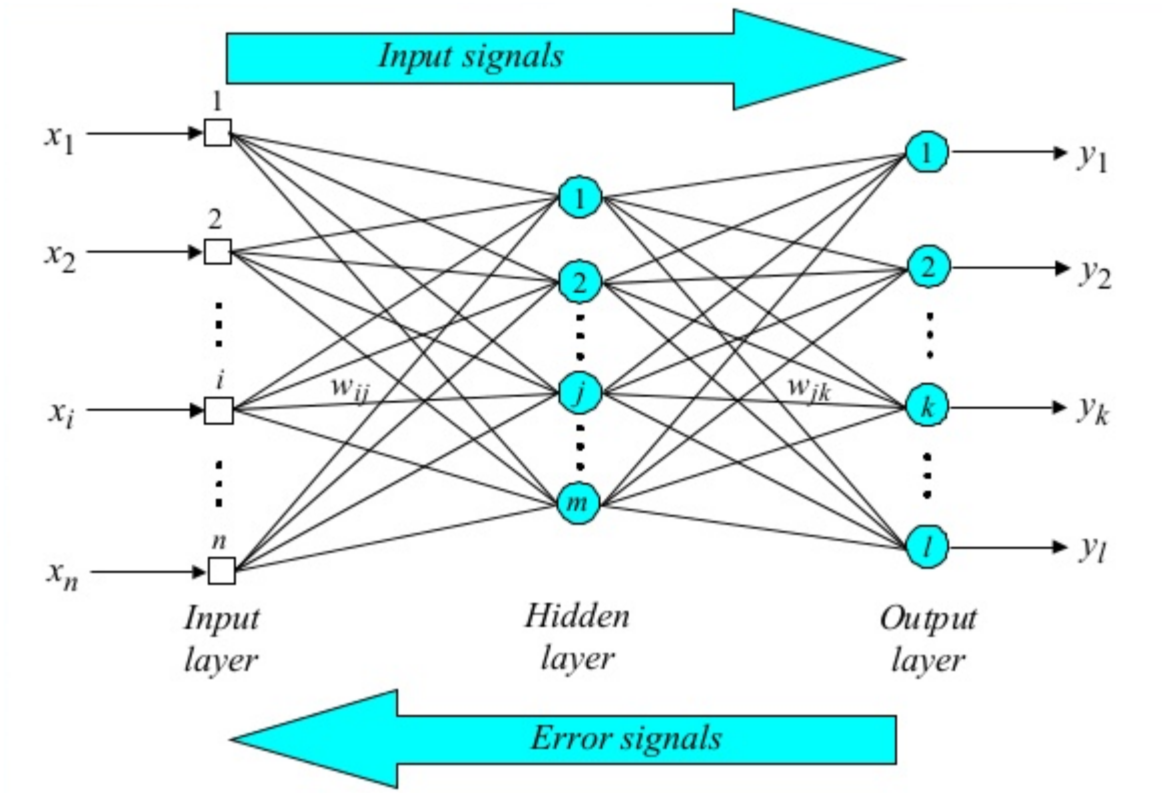


Figure 5.8 Generic Multi-Layer Perceptron

The ANN has 108 input layer, 39 hidden layer and 26 output layer neurons.

5.4.1 Flow of Character Recognition

There are many different approaches to solving the optical character recognition problem. One of the most common and popular approaches is based on neural networks, which can be applied to different tasks, such as pattern recognition, time series prediction, function approximation, clustering, etc.[8]

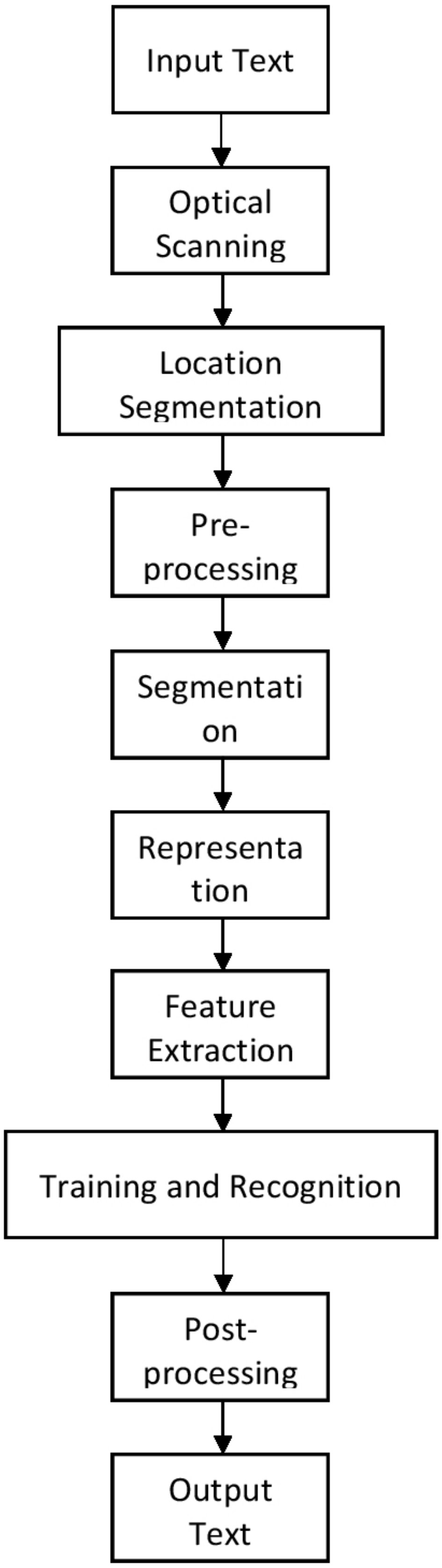


Figure 5.9 OCR Flowchart

Steps involved in OCR:

Step 1: Take Input

- The image is scanned

Step 2: Optical Scanning

- It is stored as a bit mapped file in TIFF format

Step 3: Location Segmentation

- Isolation of words or characters so that they can be recognized individually

Step 4: Pre-Processing

- Eliminate noise (i.e. breaks or gaps or holes)
- Smoothening and digitalization of image

Step 5: Segmentation

- Cutting up character image into meaningful components

Step 6: Representation

- Adjusts gray levels
- Zoning(Statistical method of feature extraction) is involved

Step 7: Feature Extraction

- Finding out character specific traits that allow them to be distinguished from one another
- One of the most difficult aspects of pattern recognition

Step 8: Training and Recognition

- Assign samples to predefined classes

Step 9: Post Processing

- Error detection and correction

Step 10: Output[8]

Chapter 6

Conclusion

6.1 Summary of work done

- The effectiveness of the method that uses feature extraction using gradient technique from scanned images containing handwritten characters is presented.
- The feature extraction methods have performed well in classification when fed to the neural network and pre-processing of image using edge detection and normalization are the ideal choice for degraded noisy images.
- The method of training neural network with extracted features from sample images of each character has detection accuracy to a greater extent.
- The proposed methodology has produced good results for images containing handwritten text written in different styles, different size and alignment with varying background
- The system is developed in MATLAB and evaluated for a set of sample images containing handwritten text on Intel dual core computer.
- The method is advantageous as it uses nine features to train the neural network using character geometry and twelve features using gradient technique.

6.2 Future Prospects

This language, when developed on a larger scale in Matlab or Python, might have a monumental impact on written and spoken communication.

References

1. "metaStudio - File - Reference Table of NaYaNa Script with examples." Internet: https://metastudio.org/5305f8f97d9d337b3f0f2203/file/58515e656f3a72022dcd6ab8, Oct 25, 2017
2. Bharat Kalla. "Project Report of OCR."Internet: https://www.slideshare.net/nikbharat/project-report-of-ocr-recognition, Sept 3, 2014
3. "International Phonetic Alphabet."Internet:https://en.wikipedia.org/wiki/International_Phonetic_Alphabet, Oct 25, 2017
4. "OCR, Neural Networks and other Machine Learning Techniques."Internet:http://www.cvisiontech.com/resources/ocr-primer/ocr-neural-networks-and-other-machine-learning-techniques.html, Oct 25, 2017
5. "Handwritten Character Recognition Using Neural Networks". Internet: https://github.com/sachinkariyattin/HWCR, May 31,2014
6. Jürgen Schmidhuber, "Deep Learning in neural networks," in ANN: Issue, Vol. 61, Neural Networks, C. T. Leondes, Ed. San Diego: Academic Press, 2015, pp. 85-117.
7. T. Dash, T. Nayak, "English Character Recognition using Artificial Neural Network," Physical Review, vol.134, pp. A635-A646, Dec. 2010.
8. C. R. Hogervorst, "Handwritten character recognition using neural networks," M.S. thesis, University of Waterloo, Waterloo, ON, Canada, 2008.
9. J. Pradeep, E. Srinivasan, S. Himavathi, " Neural network based handwritten character recognition system without feature extraction," M.E. thesis, Pondicherry Engineering College, Pondicherry, India, 2008.

Acknowledgement

We would like to express our gratitude to Dr.Nagarjuna G. and Dr. Vickram Crishna from Homi Bhabha Centre for Science Education for allowing us to take their script forward and taking into account our inputs to the script..

We take this opportunity to express our sincere thanks and gratitude to Mrs. Vaishali Suryawanshi, Assistant Professor, Department of Computer Engineering, Thadomal Shahani Engineering College for her constant guidance, support and critical review of the project throughout the project development life cycle.

We would also like to express our gratitude to Dr.G.T.Thampi, Principal, Thadomal Shahani Engineering College and Dr. Tanuja Sarode, Professor, Department of Computer Engineering, Thadomal Shahani Engineering College. We thank Mr. Ram Tejwani for his guidance throughout the project.

Rahurkar Prachi Shriram

Ramrakhiani Deepa Ram

Rao Smriti Ganesh