DESIGN DOCUMENTATION

for

Personalised Malayalam Font Generation

Prepared by
MDL16CS011 CSU16105 Adithya Nair
MDL16CS018 CSU16109 Allen Iype Sibi
MDL16CS053 CSU16127 Hariraj K
VJC16CS012 CSU16163 Akhil Seshan

November 25, 2019

Contents

List of Figures

1 Introduction

Personal handwriting can add colours to human communication. Handwriting, however, takes more time and is less favoured than typing in the digital age. Since a complete Malayalam font has typically more than thousand unique characters and symbols, and most of them are much more complicated than English alphabets, it takes a lot of time and effort for even professional font engineers to create a Malayalam font. This system presents an easy and fast solution for an ordinary user to create a Malayalam font of his or her handwriting style. The system adopts the approach: to synthesise Malayalam characters using components extracted from the user's handwriting. The existing Malayalam fonts created by Swathanthra Malayalam Computing will be used as reference. It uses data from the existing fonts, to identify characters and symbols from a user's handwriting and convert these symbols into a font.

1.1 Purpose

This document serves as the Software Design Documentation for the project "Personalised Handwritten Font Generation". The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to be built. This Software Design Document provides information necessary to provide description of the details for the software and system to be built. The selection of the format of input data-set, layout of output in the GUI,the format of the final output, the various diagrams explaining the scenarios are included in this document. The final project would be based on the design specified in this documentation

1.2 Overview

The Software Design Document is divided into the the following sections.

1. System Architecture
2. Data Description
3. Algorithms

These sections includes the following:

- Section (1) explains the System architecture of the proposed system from preprocessing to font generation.

- Section (2) gives the information about the static and dynamic aspects of the system. It includes the following:
  - Data flow diagram(DFD)
  - Use case diagram
  - Activity diagram
  - Dataset design
- Section (3) contains the algorithms that are used while implementing the proposed system.

2 System Architecture

The basic architectural diagram of the proposed system is shown in Figure 2.1. The entire system architecture can be divided into the following parts:

1. Training Data
2. Model Generation
3. Input Preprocessing
4. Identification and Mapping
5. Font Generation

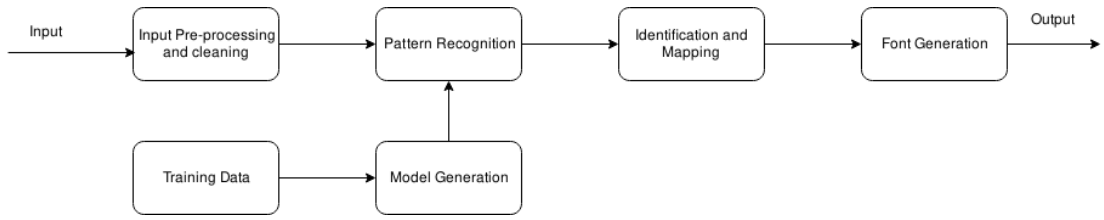

Figure 2.1: Basic Architecture

2.1 Training Data

This phase of the system architecture includes the data-set that is required for training the model for font generation. Here in this system, handwritten texts are used as data sets. Other fonts are also used for better and optimised prediction model.

2.2 Model Generation

In this step, Hilbert Transformation (a specific linear operator that takes a function, $u(t)$ of a real variable and produces another function of a real variable $H(u)(t)$) and Image Segmentation (process of partitioning a digital image into multiple segments) are applied on the training data to break them down into texts and build a training model.

2.3 Input Prepossessing

There are certain steps that needs to be applied on the image that has been accepted as input, those steps include:

- Resize Image
- Remove Noise (Denoise)
- Segmentation
- Morphology (Smoothening edges)

2.4 Identification and Mapping

The preprocessed input data is mapped with the respective alphabets in the character set. The output from this stage is used to optimise the training data.

2.5 Font Generation

In this stage, TrueType (ttf) fonts will be generated that maybe exported by the user. The generated font includes the following values in its metadata:

- Element, Font Info
- Version Number
- Family Name

3 Data Description

This section explains how the information domain of the proposed system is converted into data structures. This section contains the data-flow diagram, use-case, activity and class diagrams of the proposed system.

3.1 Data Flow Diagram

This section contains the data-flow-diagrams, a way of representing a flow of data of a process or a system. The DFD shows what kind of information will be input to and output from the system, the advancement of data through the system, etc.

3.1.1 Level 0 DFD

Level 0 DFD or the context diagram gives an overview as a whole of the proposed system. Here we have a pictorial input given to the system to get the personalised font as the output.

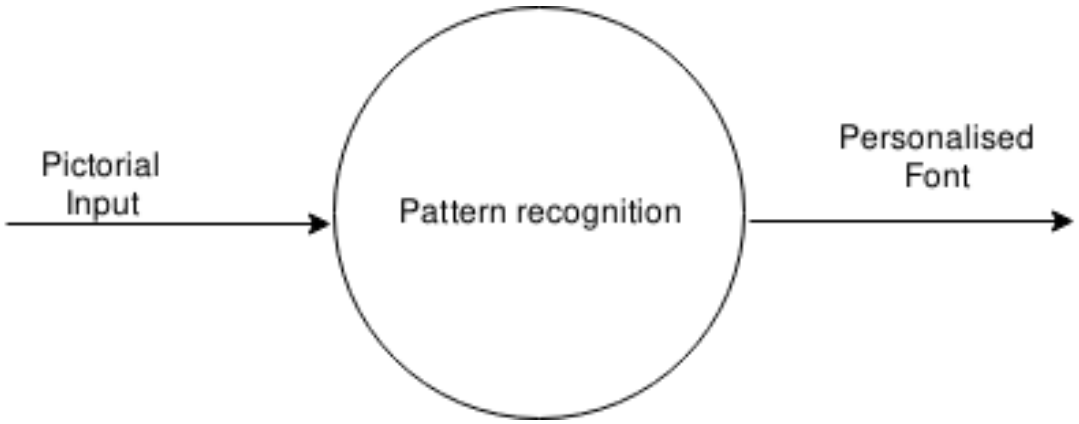

Figure 3.1: Data Flow Diagram: Level 0

3.1.2 Level 1 DFD

Level 1 DFD provides a more detailed breakout of the modules of the level 0 DFD. The main functions of this system include image preprocessing, alphabet detection, feature extraction and font generation.

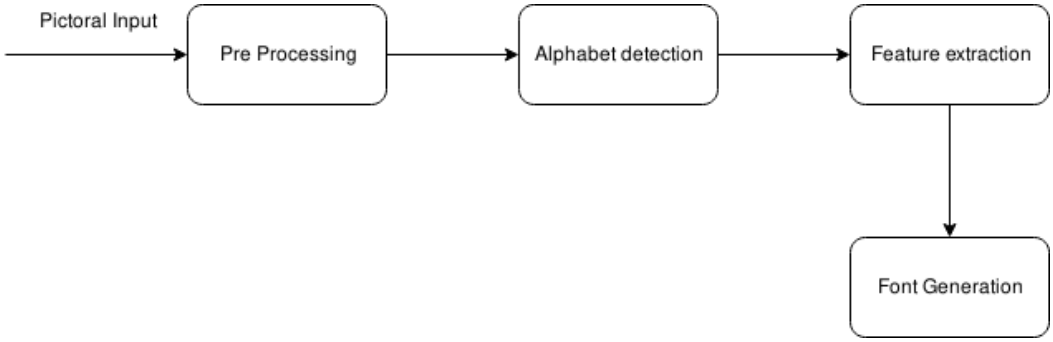

Figure 3.2: Data Flow Diagram: Level 1

3.1.3 Level 2 DFD

Level 2 DFD provides a more detailed look at the processes that make up the system in level 1 DFD. Here, the system consists of grey-scale converter and image sharpening tool in the preprocessing stage which is followed by feature extraction stage that consists of image segmentation and Hilbert transformation algorithms. The system also consists of font generation system that consists of alphabet mapping and re-segmentation processes.

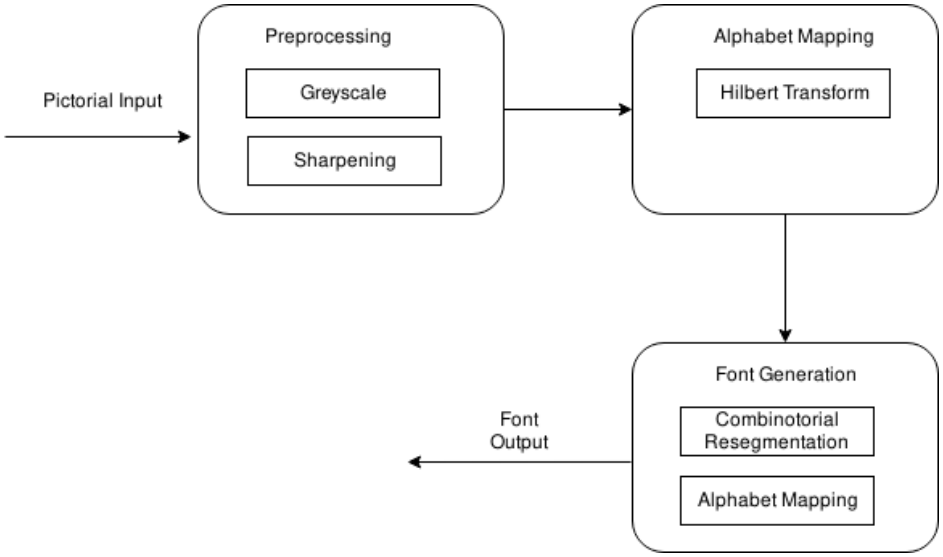

Figure 3.3: Data Flow Diagram: Level 2

3.2 Use case diagram

This section contains the use case diagram of the proposed system. The user or the customer has given the permission or access to input the image, which is then processed to generate personalised font by the model that is being trained using data sets. The user or the customer will also be able to export the generated font.

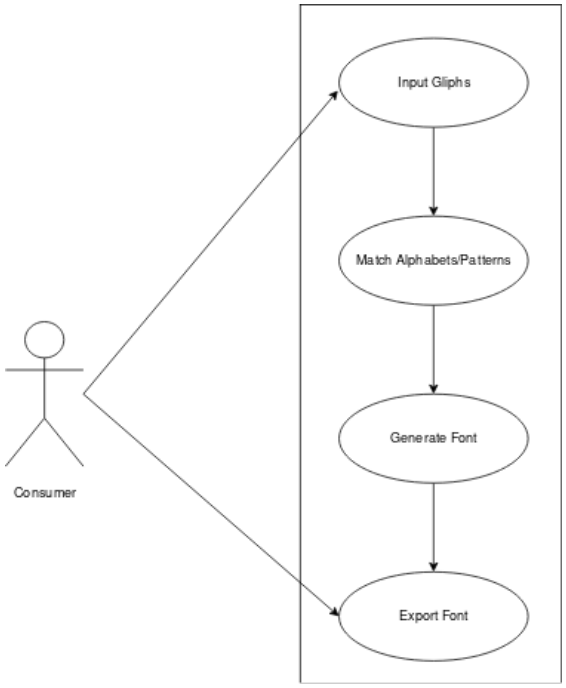

Figure 3.4: Use case diagram

3.3 Activity diagram

This section contains the activity diagram of the proposed system.

Figure 3.5: Activity diagram

3.4 Dataset Design

For Malayalam inputs, the training data is obtained from the handwritten text repository provided by Swatantra Malayalam Computing.Existing Malayalam fonts such as Rachna are used a reference for labelling. The data-set is preprocessed and the required training data set is obtained.

4 Algorithms

Algorithm 1 Hilbert Transformation

```
function ROTATE(n, x, y, rx, ry) if ry = 0 then
    if rx = 1 then
        x = n − 1 − x
        y = n − 1 − y
    return y, x, n =0
function XY2D(n, x, y)
    rx = 0
    ry = 0
    s = n//2
    d = 0 while s > 0 do
    rx = (xs) > 0
    ry = (ys) > 0
    d+ = s ∗ s ∗ ((3 ∗ rx)^ry)
    x, y = ROTATE(s, x, y, rx, ry)
    s = s//2
    return d
function UNROLLMATRIXINTOHILBERTCURVE(imageMatrix, imageHeight, imageWidth)
    hilbertArray = [−1] ∗ imageHeight ∗ imageWidth
    k = 0
for i ← 0 to imageHeight do
    for j ← 0 to imageWidth do
        k = imageWidth − i − 1
        pos = XY2D(imageWidth, j, k)
        hilbertArray[pos] = imageMatrix[i][j]
    return hilbertArray
```

Bibliography

[1] Jomy, John, K. V. Pramod, and Balakrishnan Kannan. "Handwritten character recognition of south Indian scripts: a review." arXiv preprint arXiv:1106.0107 (2011).

[2] Ahrary, Alireza, and Sei-ichro Kamata. "A new on-line signature verification algorithm using Hilbert Scanning patterns." 2009 IEEE 13th International Symposium on Consumer Electronics. IEEE, 2009.

[3] Moon, Bongki, et al. "Analysis of the clustering properties of the hilbert space-filling curve." IEEE Transactions on knowledge and data engineering 13.1 (2001): 124-141.

[4] Lin, Xue-Hui, and Li-Dong Cai. "Scrambling research of digital image based on Hilbert curve." Chinese Journal of Stereology and Image Analysis 9.4 (2004): 224-227.

[5] Jain, Anil K., Friederike D. Griess, and Scott D. Connell. "On-line signature verification." Pattern recognition 35.12 (2002): 2963-2972.