# HANDWRITING RECOGNITION USING MACHINE LEARNING METHODS

Software Requirement Specifications

**Bachelor of Technology**

in

**Computer Science and Engineering**

of

**Cochin University of Science and Technology**

by


**Aniljith K**

**Aravind Sankar**

**Balu R Krishnan**

**Karthik K**

**Sangeeth Dev S**


**November, 2017**


Department of Computer Engineering

College of Engineering, Chengannur -689121

Phone: (0479) 2454125, 2451424; Fax: (0479) 2451424

College of Engineering, Chengannur

Kerala

# Contents

# List of Figures

# 1 INTRODUCTION

Handwritten character recognition is a field of research in artificial intelligence, computer vision, and pattern recognition. A computer performing handwriting recognition is said to be able to acquire and detect characters in paper documents, pictures, touch-screen devices and other sources and convert them into machine-encoded form. Its application is found in optical character recognition and more advanced intelligent character recognition systems. Most of these systems nowadays implement machine learning mechanisms such as neural networks.

Neural networks are learning models used in machine learning. Their aim is to simulate the learning process that occurs in an animal or human neural system. Being one of the most powerful learning models, they are useful in automation of tasks where the decision of a human being takes too long, or is imprecise. A neural network can be very fast at delivering results and may detect connections between seen instances of data that human cannot see

Despite the abundance of technological writing tools, many people still choose to take their notes traditionally: with pen and paper. However, there are drawbacks to handwriting text. Its difficult to store and access physical documents in an efficient manner, search through them effi- ciently and to share them with others. Thus, a lot of important knowledge gets lost or does not get reviewed because of the fact that documents never get transferred to digital format

# 2 OVERALL DESCRIPTION

In summary, our models take in an image of a word and output the content of the image.

It has already been stated that the primary goal of our project a System application is able to recognize handwritten characters based on user's input and image/camera input in an offline manner. Also, the application provides means of adding and learning a new character and learning interactively from user's feedback.''

## 2.1 MACHINE LEARNING

The next major upgrade in producing high OCR accu- racies was the use of a Hidden Markov Model for the task of OCR. This approach uses letters as a state, which then allows for the context of the character to be accounted for when determining the next hidden variable [8]. This lead to higher accuracy compared to both feature extraction tech- niques and the Naive Bayes approach [7]. The main draw- back was still the manual extraction features, which requires prior knowledge of the language and was not particularly robust to the diversity and complexity of handwriting.

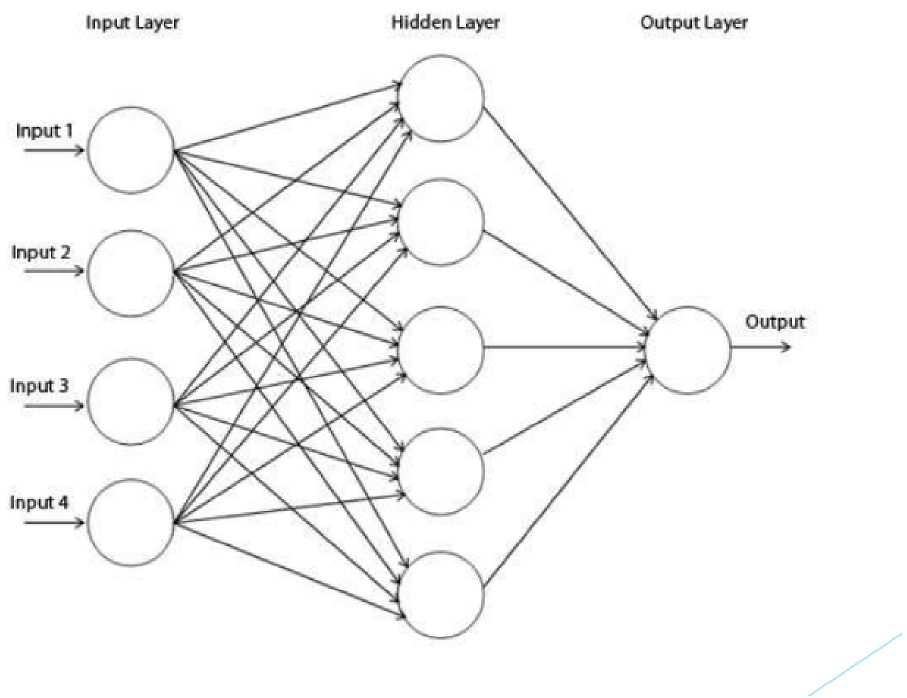



Figure 2.1: Architecture Of Neural Network

# 3 RELATED WORKS

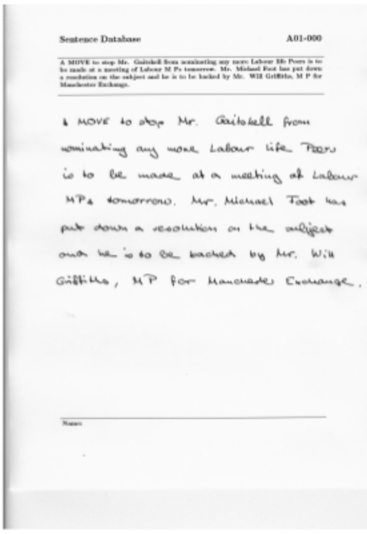
## 3.1 Early Scanners

The first driving force behind handwritten text classifi- cation was for digit classification for postal mail. Jacob Rabinows early postal readers incorporated scanning equip- ment and hardwired logic to recognize mono-spaced fonts [3]. Allum et. al improved this by making a sophisticated scanner which allowed for more variations in how the text was written as well as encoding the information onto a bar- code that was printed directly on the letter [4].

## 3.2 To the digital age

The first prominent piece of OCR software was invented by Ray Kurzweil in 1974 as the software allowed for recog- nition for any font [5]. This software used a more developed use of the matrix method (pattern matching). Essentially, this would compare bitmaps of the template character with the bitmaps of the read character and would compare them to determine which character it most closely matched with. The downside was this software was sensitive to variations in sizing and the distinctions between each individuals way of writing.

To improve on the templating, OCR software began us- ing feature extraction rather than templating. For each char- acter, software would look for features like projection his- tograms, zoning, and geometric moments [6].


Sentence Database A01-000

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.

Name:

Figure 1. An example form from the IAM Handwriting dataset. Word images in the dataset were extracted from such forms.

Figure 3.1: Dataset

# 4 DESIGN

## 4.1 DATA FLOW DIAGRAM

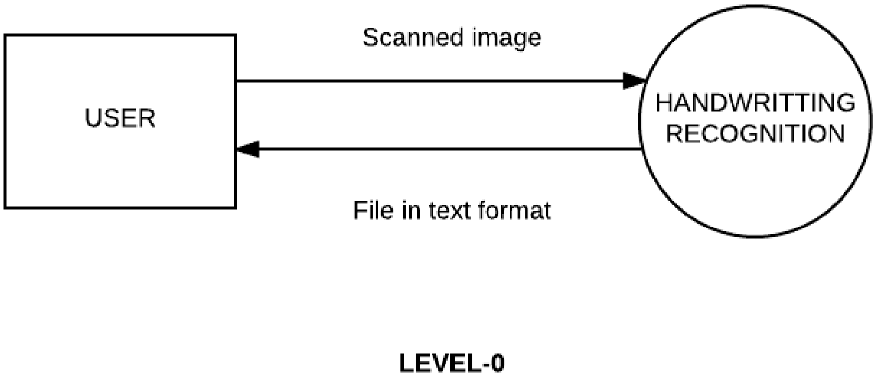



Figure 4.1: LEVEL-0 (Context Free Diagram)

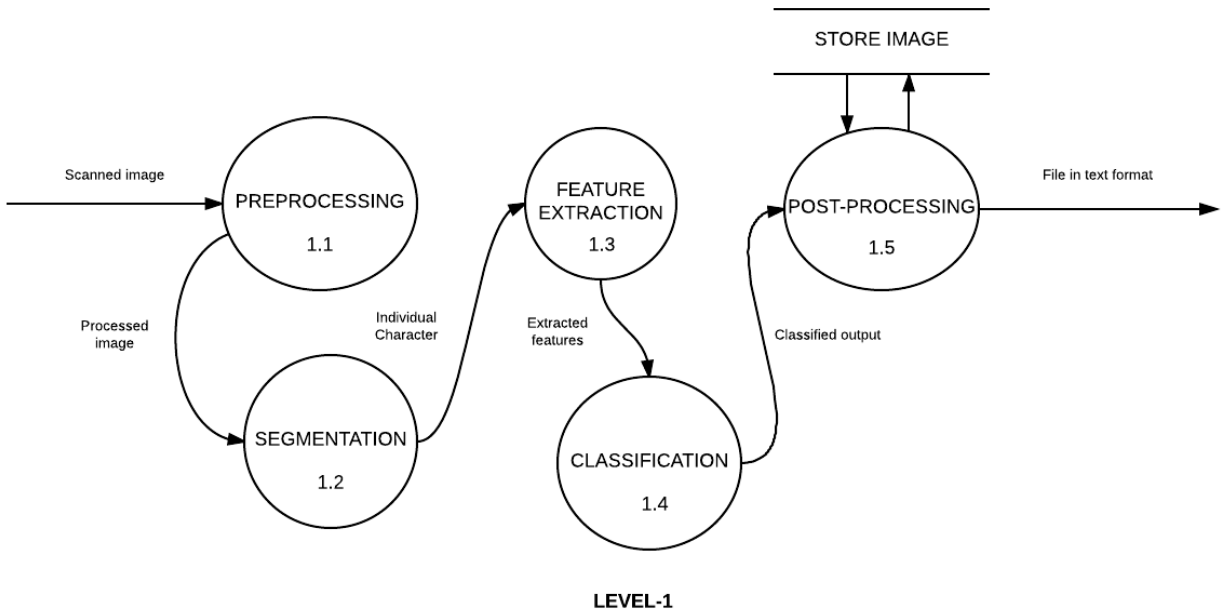



Figure 4.2: LEVEL-1

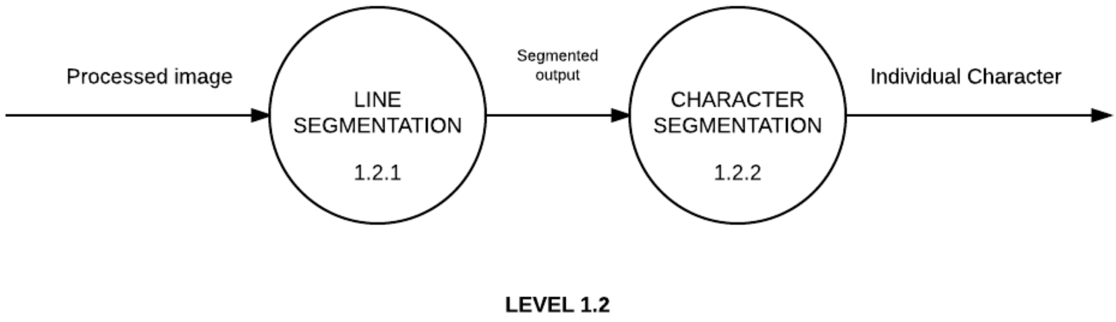



Figure 4.3: LEVEL-1.2

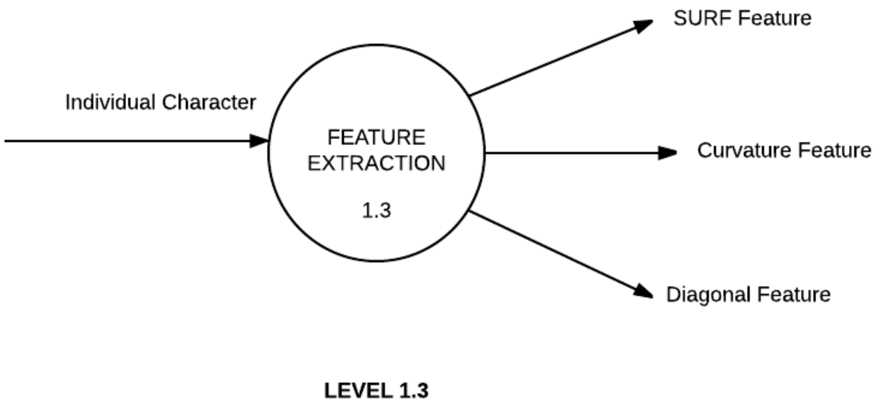



Figure 4.4: LEVEL-1.3

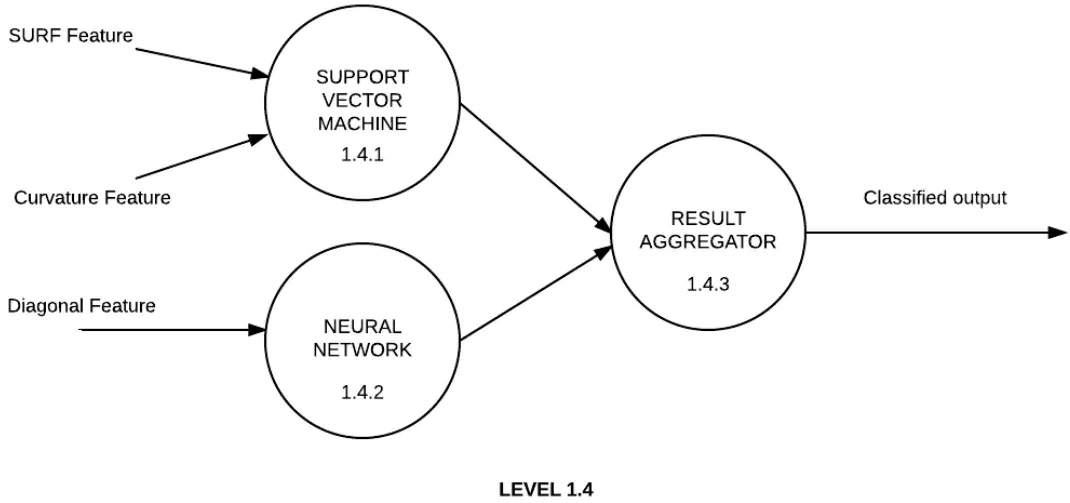



Figure 4.5: LEVEL-1.4

# 5 MODULAR DESCRIPTION

Design phase is mainly classified as follows:

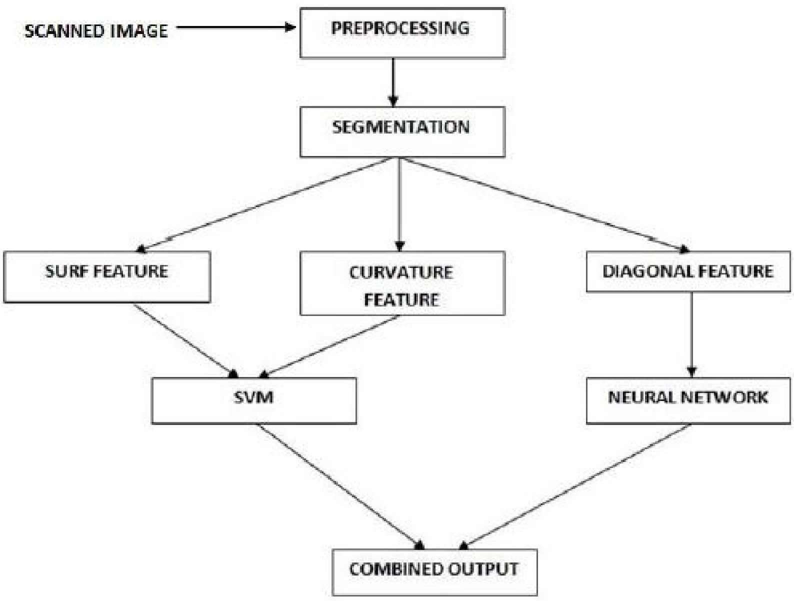



Figure 5.1: Methodology

## 5.1 IMAGE ACQUISITION

The images for the experiment are captured by using a scanner at 300 dpi resolution. The images can be in any format like JPEG, BMP, PNG etc. These images are given as input to the system for further steps. A sample scanned image is shown in figure

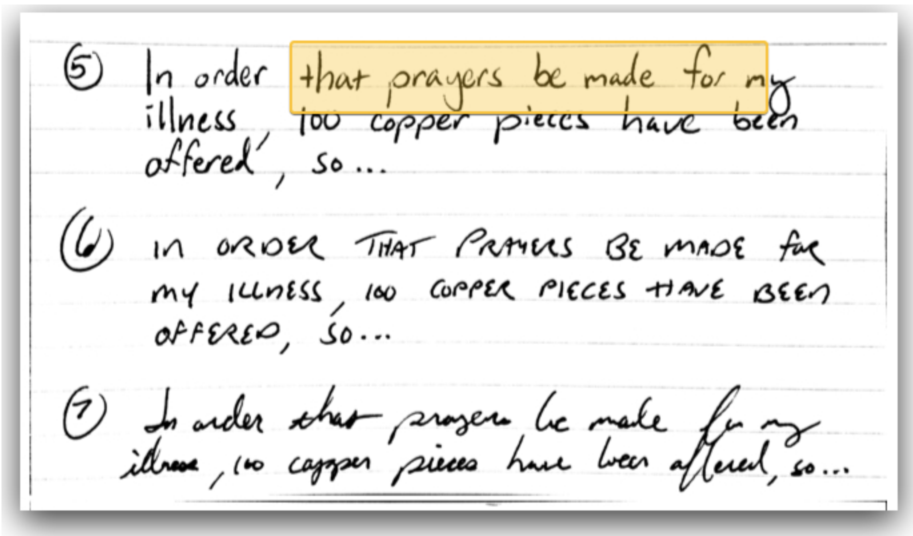



Figure 5.2: sample image

## 5.2 PRE-PROCESSING

Before training our models with the dataset, we have applied various preprocessing and data augmentation tech- niques on our dataset in order to make our data more com- patible with the models and to make our dataset more robust to real life situations.

The aim of preprocessing is to remove as much as distortions as possible from the scanned image. Degraded documents or poor quality of scanner are responsible for these distortions. At first, the scanned image is converted to grayscale if it is in RGB or any other format. After that the grayscale image is converted to binary using Otsus method of global thresholding. The advantage of using Otsu method is that no prior knowledge about the image is required for the conversion. In order to remove the salt and pepper noise present in the image, a 3x3 median filter is used. Thinning is performed using Hilditch algorithm to extract only the relevant features from the image.

## 5.3 SEGMENTATION

The purpose of segmentation is to isolate individual characters from the handwritten image. The task of segmentation from handwritten text is complex due to the existence of broken characters, touching characters and overlapped characters. Line segmentation from text is performed by using Horizontal projection profile. Character segmentation is done by Vertical Projection Profile Method.

For word-level classification, we suspected that our per- formance was suffering because of the large softmax layer output size (there were over 10,000 words in our training vocabulary and well over 100,000 words in the English language alone) and the difficulty of fine-grained recogni- tion of images of words. We decided that character-level classification may be a more promising approach because a fairly comprehensive character vocabulary is relatively much smaller than a similarly comprehensive word vocab- ulary (the characters A-Za-z0-9 are only 62 distinct sym- bols), significantly limiting the computational complexity of the softmax.

Furthermore, recognition of a character in an image is a simpler task than recognition of a word in an image because of the limited range of characters. How- ever, the first main challenge we would have to encounter in order to test this approach would be segmentation of word images into their component character images. The sec- ond main challenge would be recognizing word breaks in image and stringing together consecutive recognized char- acters in between these word breaks to form words. We will address the earlier of these challenges in this section. In order to implement this task, we employed the new Tesser- act 4.0 neural network-based CNN/LSTM engine[13]. This model is configured as a textline recognizer originally developed by HP and now maintained by Google that can rec- ognize more than 100 languages out of the box. For Latin- based languages, including English, the Tesseract model had been trained on about 400000 textlines spanning about 4500 fonts. We then finetuned the parameters of this pre- trained model on our IAM handwriting dataset. After we finetuned the model, we segmented the original word in- put images into their hypothesized component character images, feeding in these output segmented character images into our character-level classification.

## 5.4 FEATURE EXTRACTION

Features of input data are the measurable properties of observations, which one uses to analyze or classify these instances of data. The task of feature extraction is to choose relevant features that discriminate the instances well and are independent of each other. According to [3], selection of a feature extraction method is probably the single most important factor in achieving high recognition performance. There is a vast amount of methods for feature extraction from character images, each having different characteristics, invariance properties, and reconstructability of characters. [3] states that in order to answer to the question of which method is best suited for a given situation, an experimental evaluation must be performed. The methods explained in [3] are template matching, deformable templates, unitary image transforms, graph description, projection histograms, contour profiles, zoning, geometric moment invariants, Zernike moments, spline curve approximation, and Fourier descriptors. To describe the way feature extraction is sometimes done in handwriting recognition, we briefly explain one of them.

The idea behind performing feature extraction is to extract the salient characteristics of the image. The success of a character recognition system relies on effective feature classifier combination. Here we were extracted two features from the character. SURF and curvature are the extracted features. These extracted features are given as the input to two classifiers.

### 5.4.1 SURF FEATURE

It is called Speeded Up Robust Features. It is the speeded up version of SIFT. The main benefit of using SURF in the feature extraction stage is its capability to distinguish between points of interest in the image. SURF Algorithm involves 3 parts: Interest Point Detection, Local Neighborhood Description and Matching.

### 5.4.2 CURVATURE FEATURE

Majority of the Malayalam letters contains curves and loops. Since most of them are round and curved letters, curvature feature is suitable to identify and extract the features from them. Curvature is a feature that provides clearer structural sketch of the image.

### 5.4.3 DIAGONAL FEATURE

In diagonal feature extraction method, the input image is divided into zones of predefined sizes. Then features are computed for each of these zones. Zoning provides the local characteristics of an image. Here the image of size 90x60 is divided into 54 zones in which each zone is of size 10x10 pixels. Each zone provides 19 subfeatures which are averaged to form a single feature vector and stored on its corresponding zones. This process is repeated for each zone and finally we get 54 features for each character.

## 5.5 CLASSIFICATION AND RECOGNITION

The decision making part as well as the final stage of a character recognition system is classification. In this stage, unique labels are assigned to each character image based on the extracted features. Here we have used a pair of dissimilar classifiers. SVM and Neural Network are the classifiers used here.

### 5.5.1 SVM

It is a supervised learning model which performs linear classification. The advantage of using SVM is that it learns fast. But on the other side, it predicts slowly. SVM is used when your data has extremely 2 classes. SVM classifies data by finding the best hyperplane that separates all datapoints of one class from another.

### 5.5.2 NEURAL NETWORK

It is a network that learns from observed data. Neural Network learns slow but has fast prediction capacity. This can perform tasks which cannot be performed by a linear program. Neural Network for Malayalam character recognition consists of a set of input neurons that are activated by the pixels of an input image. The activation is then passed onto other set of neurons. The process is repeated until an output neuron is activated. This determines which character was read.

## 5.6 POST PROCESSING

The characters are mapped to the corresponding Unicode values. These values are used when comparing new data input . The Post-Processing results are stored in the database in the training phase. In the recognition phase the comparison takes place.

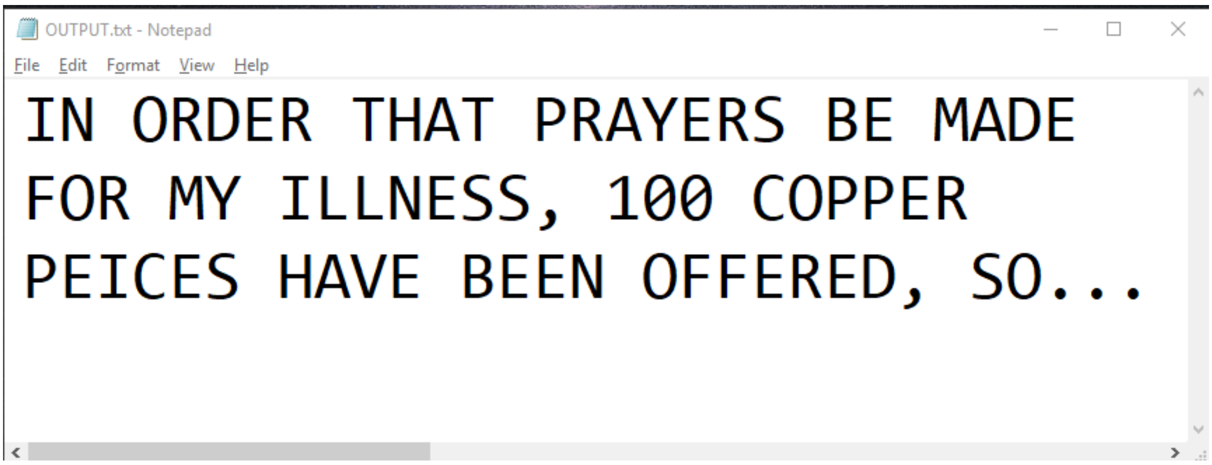



Figure 5.3: sample output

# 6 SCOPE

The aim of this project is to further explore the task of classifying handwritten text and to convert handwritten text into the digital format. Handwritten text is a very general term, and we wanted to narrow down the scope of the project by specifying the meaning of handwritten text for our purposes. In this project, we took on the challenge of classifying the image of any handwritten word, which might be of the form of cursive or block writing.

Now is the age of digitalization. Rather than storing the data on papers, it can be safely stored and easily accessed, then digitalized. Therefore Handwriting Recognition is of prime importance. Handwriting Recognition is an emerging as well as challenging area in the fields of pattern recognition and computer vision. Other applications include automatic number plate recognition, CTS scanning, preservation of degraded documents etc.

The aim of a handwriting recognition system is to convert human readable characters which are present in a photographed or digitized sheet of paper and convert it into a machine editable form.An upgradation to this project can result in the system to detect a particular persons handwriting. The future scope can be further studies to improve the current system.

# 7 SOFTWARE REQUIREMENTS

## 7.1 PYTHON

Python is a widely used high-level programming language for general-purpose programming.Python has a design philosophy that emphasizes code readabil- ity and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.Python features a dynamic type system and automatic memory management and supports multi- ple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name reso- lution, which binds method and variable names during program execution.

## 7.2 OPEN CV

OpenCV (Open Source Computer Vision) is a library of programming func- tions mainly aimed at real-time computer vision.OpenCV supports the Deep Learning frameworks TensorFlow, Torch/PyTorch and Caffe. There are bindings in Python. OpenCV runs on a variety of platforms. The features include,advance vision research by providing not only open but also optimized code for basic vision infrastructure, disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable,advance vision-based commercial appli- cations by making portable, performance-optimized code available for freewith a license that did not require code to be open or free itself.