Chapter 1

Introduction

1.1 Overview

Handwriting has undoubtedly been one of the oldest and most common means of communication that has advanced with the passage of time. Indeed the writing styles differ on the basis of geographical location, temporal, social and cultural situations but it has been proved that with time a person adopts his/her own writing style according to the personal preferences. Therefore it is said that in addition to the information in the text the writing is also the representation of a person's demographics like age, gender, handedness and personality. Although the relation of handwriting and personality has remained doubtful and still needs to be proved on scientific grounds but person recognition through handwriting is practically being witnessed in various fields like forensics, graphology, psychology etc. Handwriting analysis has gained a significant importance with respect to research since the initial decades of last century. Considering the handwriting analysis requirement the computer scientists thought to automate this process with the advanced techniques being used in image analysis and pattern classification.

There are a number of systems developed, serving this purpose with the objectives including; handwriting recognition, word spotting, writer identification and signature verification. However there are a very few significant and authentic contributions for personality profiling. Writer's demographics identification i.e. determining gender, age and handedness has been scrutinized in various documents.

Somaya and Abdelaali [1] highlighted a two step process in order to achieve this goal i.e. feature extraction and classification. They reported the classification rates reaching 74.05% for gender prediction, 55.76% for age range prediction, and 53.66% for nationality prediction when all writers produce the same handwritten text and 73.59% for gender prediction, 60.62% for age range prediction, and 47.98% for nationality prediction when each writer produces different handwritten text.

In another study [2] Somaya, Fethi, Samir and Ali proposed an approach for handedness detection from offline handwriting using fuzzy conceptual reduction. This approach gave importance to the most characterizing features of handwriting and discarded the features that contributed less and proved to provide a maximum reduction rate.

In [3], the authors provided a solution for demographics determination of a writer using gradient features. They proposed the use of two features that are Histogram of Oriented Gradients and Local Binary Patterns. The classification rates obtained using these features highlight the high accuracy and effectiveness of their approach. In another study [4], the authors provided a gender classification technique on the basis of discriminative attributes identified by the psychologists. The techniques involved use of neural networks and support vector machines.

Likewise, personality profiling from handwriting has also has also been one of the major concerns of the authors in [5]. They highlighted the psychological technique i.e. graphology and also various online technologies used for this purpose. In [6] the authors used segmentation to get features of a digitized handwriting and applied support vector machine approach to define the personality of the writer. The proposed system assures 94% of accuracy rate using the samples of 100 writers. Authors in [7] extracted features like margins, baseline, size, zones etc through image processing and mapped to the existing personality theories and results were the temperament and personality of the writer.

1.2 Objective

The objective of this project is to automate the personality and person demographics profiling through handwriting analysis using offline/scanned images of writing.

1.3 Problem Description

This project aims to not only automate the personality profiling process but it will also be helpful in providing the writer's demographics including gender and handedness by handwriting analysis. In the analysis of handwriting a psychologist considers different features of handwriting and knowing some predefined rules for those features profiles the behavior, feelings and thoughts that make a person unique. The features considered for personality profiling are size, spacing, baseline, margins, pressure and speed. On the other hand profiling the demographics is also done on the basis of some features of handwriting including slant, curvature and texture. Developing an automated version of handwriting analysis, will allow an instant profiling of a person's personality and demographics and will make the analysis easy and efficient for the clinical psychologists.

1.4 Methodology

The developed project is mainly dependent on the techniques of image processing. It takes scanned binarized image of a person's handwriting as input and extracts features including slant, curvature, character size, word and/or line spacing, margins and texture. These features are then be used by the two modules that are personality profiling and demographics respectively according to the suitability. The overall system is presented in Figure 1.1. For user demographics first the classifier will be made capable of recognizing the genders i.e. male or female and handedness i.e. right or left with the help of some training handwriting samples. Then for every query image the required features would be extracted from it and will be passed to both the modules for processing respectively and generate the writer's profile. Specifically for personality profiling the features that will be extracted will include slant, letter size and baseline. The database used in our study is a subset of the QUWI database. For personality profiling, Tables 1.1 to 1.4 show the relation between the handwriting features and personality traits proved by psychologists and mentioned in [8].

Table 1.1: Baseline and related personality traits

Type	**Personality Traits**
Straight	Stable outward behavior, realistic and disciplined

Ascending	Healthy mental energy, active, hopeful, cheerful, excitability and choleric behavior
Descending	Sinking feeling, state of depression, physical or mental tiredness

Table 1.2: Slant and related personality traits

Type	**Personality Traits**
Vertical	Real feelings, positive qualities of independence, Living in present, cool judgment, controlled emotions, laziness, apathy, coldness
Right	Demonstrative, passionate, affectionate, speaks of future, compliance, vision, and expressiveness
Left	Avoid emotional involvement in a situation, Materialistic, Oriented towards the past, Negative, fear of past, resistant, doubt and repression

Table 1.3: Letter Size and related personality traits

Type	**Personality Traits**
Large	Extrovert, Cannot concentrate, demand space in life, lavish, ceremonial, big planners, see things big, boisterous and loud

Medium	Social and have average ability to concentrate on things, Have need to conform in all areas
Small	Introversion, sense of economy, Deep concentration

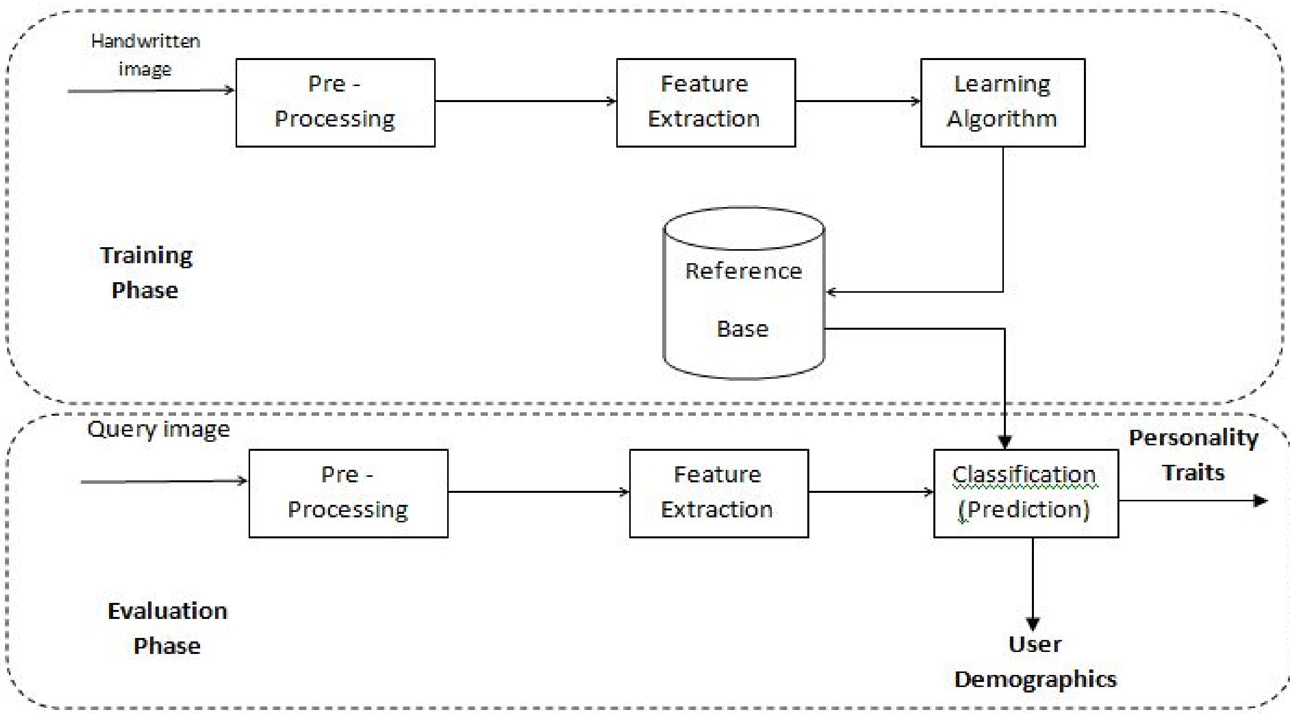


Figure 1.1: Working of the proposed system

1.5 Project Scope

The project will use samples of handwriting from the QUWI database to profile writer's demographics and personality. We will focus on offline image of handwriting only. The evaluation phase will receive the query sample that will be processed i.e. features will be extracted from the sample and will be classified on the basis of training samples' features in the reference database.

1.6 Solution Application Areas

The proposed automated handwriting analysis system can be applied to different fields for efficient profiling of a person's demographics and personality. User demographic classification can be employed as an effective module in handwriting based biometric systems to search for writers within a specific group of writers based on gender or handedness. Further in graphology it can be used by the psychologists for quick and efficient personality tests rather than doing it manually.

Chapter 2

Literature Review

In this chapter, we identify, evaluate and interpret any work related to the current project produced by others. It will provide the proposed layout for achieving our goals. As the project involves a lot of ground work this chapter holds a lot of significance. This chapter identifies the different feature extraction and classification algorithms.

2.1 Existing Work

Prediction of demographics from handwriting has received research attention over the last decade. In one of the well-known studies [3] on this subject, authors exploit gradient features to predict writer demographics. They employed two gradient features including Histogram of Oriented Gradients (HoG) and the Local Binary Patterns (LBP). The classification rates using Support Vector Machine (SVM) were comparable to already existing methods. HOG characterizes an object's shape on the basis of distribution of local intensity gradients or edge directions. In order to compute it the image is divided into rows and columns i.e. a grid and histogram is created for every cell. Moreover the histogram that has been created is normalized to fall between [0, 1]. It can be said that histogram calculated for full image using HOG features cannot highlight precise handwriting characteristics. Experiments were carried out on English and Arabic handwriting images taken from, IAM and KHATT datasets.

In another study [7] by Rashi and Hima, the authors highlighted automation of graphology and proposed to train a system that may replace a graphologist. For feature extraction they considered Margins, Baseline, Size, Zones, Spacing and Degree of Connection of an image sample as good features. They considered only the samples written on A4 paper in black ink and mapped the existing proven personality rules to generate a detailed report about personality of the person. They highlighted the applications of the system in various fields such as; graphology, forensic studies, biometrics and palaeography etc. The presented system consists of nine basic steps that are; scanning samples, thresholding them, binarizing images, applying thinning algorithm, performing segmentation, extracting features, analyzing, mapping with Briggs Dichotomies and displaying the results.

In another work [1], the authors proposed a method of predicting age, gender and nationality from offline handwriting automatically. They considered two main concepts of image processing for this purpose that are feature extraction and classification. They considered QUWI dataset for obtaining samples. The extracted features include direction, tortuosity, curvature, chain code and edge-based directional features. The authors combined two classifiers for classification; Random forest classifier and Kernel discriminant analysis using spectral regression (SR-KDA). Random forest classifier operates on decision tree approach and is combined with SR-KDA as it is good for complex computations. They used 70% of dataset for training and 30% for testing purpose, with the accuracy rate of 55.76% for age classification, 74.05% for gender classification, and 53.66% for nationality prediction when same text is produced by all writers whereas for different texts the results are 73.59%, 60.62%, and 47.98 respectively.

In [4] work of Nesrine, Hassiba and Youcef introduced two gradient features in order to predict the age, gender and handedness of the writer. They used Histogram of Gradient (HOG) feature that depicted the distribution of gradient orientation in an image. The second feature they used was the Local Binary Pattern (LBP) that is the improved gradient feature. For classification the authors used support vector machine classifier. They conducted the experiment over Arabic and English scripts obtained from IAM and KHATT database.

In [9], the authors predicted the profile of a person through handwritten signatures. They proposed method of determination of both biometric and non-biometric characteristics of an individual's handwritten signature sample. The samples they used were collected from Department of Electronics at the University of Kent UK. They extracted 16 common features for classification. To be more accurate they used multiple classifiers including Multi-Layer Perceptron (MLP), Fuzzy Multi-Layer Perceptron (FMLP), Radial Basis Function Neural Network (RBF), Optimised IREP (Incremental Reduced Error Pruning)(JRip), Support Vector Machines (SVM), Decision Trees (DT) and K-Nearest Neighbours (KNN). They compared the results and proved high accuracy in predicting age.

In [8], authors presented a system to predict a person's personality on the basis of handwriting. They discussed several features that were extracted and used for classification. They mentioned the existing rules of graphology i.e. what does certain feature mean for a particular personality. They proposed that GSC Algorithm may effectively be used for classification. According to their study graphology gives 90% result and their proposed system will give close enough result.

In another study [2] Somaya et al. proposed an approach for handedness detection from offline handwriting using fuzzy conceptual reduction. This approach gave importance to the most characterizing features of handwriting and discarded the features that contributed less and proved to provide a maximum reduction rate. They proposed to use K-Nearest Neighbor (KNN) classifier on a database of 121 writing samples. Then applying fuzzy conceptual reduction they achieved an accuracy of 83.43%.

The authors in another work [10] used segmentation to get features of a digitized handwriting and applied support vector machine approach to profile the personality of the writer. They considered six different handwriting features for classification that are size of letter, spacing between letters, slant , pen pressure, baseline and spacing between words. The proposed system realizes 94% of accuracy rate using the samples of 100 writers with the support vector machine classifier.

Chapter 3

REQUIREMENT SPECIFICATION

The goal of this chapter is to outline the project requirements in order to maintain a relation between the customer and the developer. The developer while following the software life-cycle phases requires software specifications, this document will act as a basis for the design phase by detailing all the requirements. This chapter describes the existing system and its limitations and how the proposed system will overcome those limitations. In addition, detailed analysis of all the requirements; functional and nonfunctional for the proposed system will be identified.

3.1 Purpose of Document

The purpose of the document is to explain in detail the profiling of a person's demographics and personality from offline images of handwriting. It will include the working, specification, functionalities, and constraints under which the system will operate and how it will behave with the external environment. Moreover it will also highlight all the functional and non-functional requirements of the system.

3.2 System Overview and Scope

This system will benefit all those areas where it is required to identify a person from handwriting. Mainly it will also be supporting the psychology field where a person's psychological behavior is to be assessed; as it has been proved earlier in the studies of this field.

3.3 General Description

3.3.1 Product Functions

The essential viewpoint is to give a .Net framework system which will be another development considering the fact that it is possible to determine a person's demographics and personality through handwriting.

In proposed system individual will interface with the GUI which will permit to transfer offline handwriting image to recognize gender, handedness and personality traits of the writer. The system will first convert the image into grayscale, apply binarization to converted image, will invert the binarized image, will convert the inverted image to unmanaged image, compute the GLCM matrix of unmanaged image and after that will extract Haralick features from it and using

SVM classifier will give the results. The classifier has already been trained using the available QUWI dataset for gender classification and handedness.

3.3.2 User Characteristics

The application does not require any uncommon skills for the users. The users must be acquainted with windows based desktop PCs. The application will give a straightforward and simple to use interface which would not require any particular knowledge or mastery.

3.4 Operating Environment

The following software and libraries needed to be installed.

- Windows XP or later
- .Net Framework
- Visual Studio 2013
- Accord.net
- AForge

3.5 Constraints, Assumptions and Dependencies

- **Visual:**
  - Blur handwriting image can't give the proper/correct results.
- **Criticality of system:**
  - In case of noisy image the system will not predict correctly.

3.6 Functional Requirements

The main function of our system is to profile person's demographics and personality.

3.6.1 Description

The developed system will serve the need to identify a person using demographics and personality traits. Developing an automated version of handwriting analysis, will allow an instant profiling of a person's personality and demographics and will make the analysis easy and efficient for the clinical psychologists as well.

3.6.2 Importance

This system is ideal for clinical psychologist and its demographics module can also be used in forensics.

3.6.3 Criticality

The feature extraction and matching extracted data with database is the most critical point of system.

3.6.4 Technical Issues

The system will not work properly and will give wrong/false results under following situations and conditions:

- Image of bad quality.

3.6.5 Risks

- Non-availability of samples.
- Poorly written samples i.e. overlapped lines
- Highly cursive writing

3.7 Non-functional Requirements

3.7.1 Performance Requirements

The desktop computer should match with the minimum specification requirements that are required by the application. The processor, memory and the type of the windows should be compatible with the application.

3.7.2 Software Quality Requirements

1. SQR-01

The system will be able to predict with an accuracy at par with existing systems.

2. SQR-02

System is capable to extend accordingly or to act as a module in other larger systems.

3. Availability

The system will be available only to the authorized person.

4. Flexibility

The system will be flexible enough for some later requirements change or features enhancement.

5. Usability

The interface and GUI design of the system will be user friendly with no training required to use the system.

3.8 Other Requirements

Some of the other particular non-functional attributes required by the system are listed in the following.

3.8.1 Accuracy

The application should be accurate enough to profile a person accurately.

3.8.2 Maintainability

The maintenance of the application will be carried out by the developer, if required.

3.8.3 Portability

The application is not portable since it is a desktop application.

Chapter 4

SYSTEM DESIGN

This chapter contains modular design of the proposed system. It has detailed description of the system's design and attempts to define the different methods, properties, classes and associations between them. Various UML diagrams are used to illustrate the architecture, components, modules, interfaces and data for the system to fulfill the specified requirements.

4.1 System Architecture

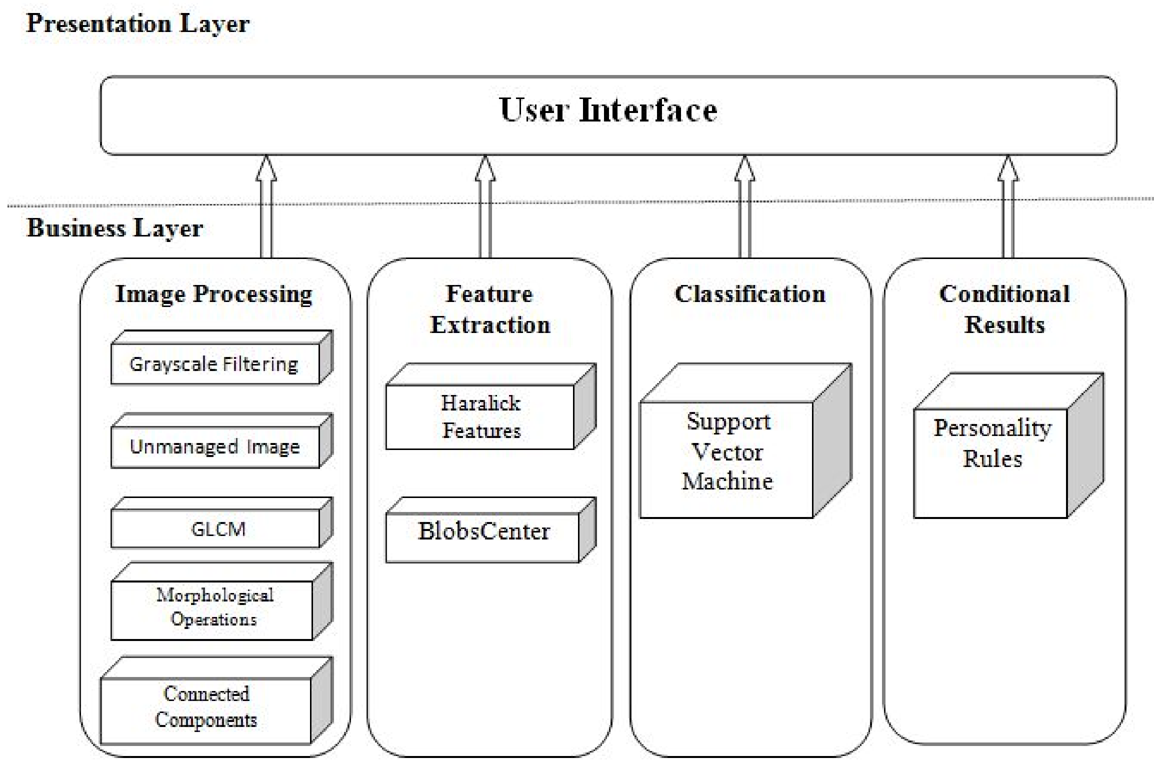


Figure 4.1: System Architecture Diagram

Figure 4.1 explains the system architecture in terms of two main layers that are Presentation Layer and Business Layer. The presentation layer incorporates the user interface whereas the business layer contains the internal working of the system i.e. image processing, feature extraction, classification and conditional results.

4.2 High Level Design

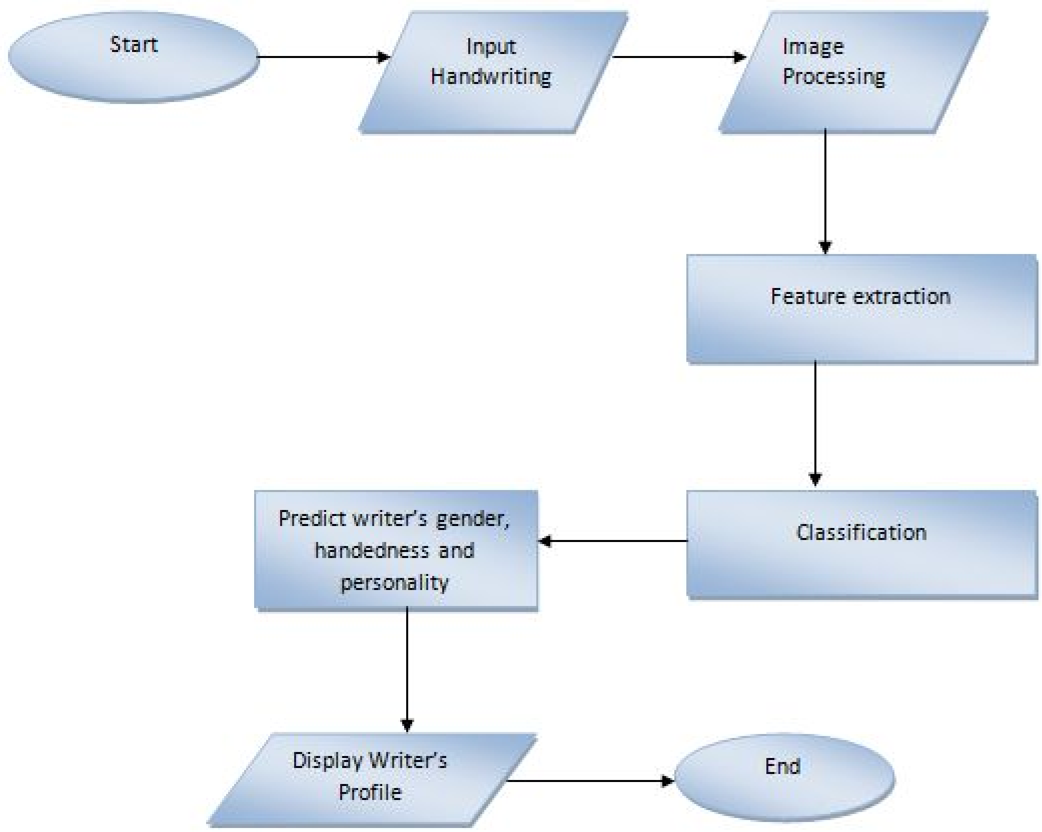


Figure 4.2: System Flow Diagram

Figure 4.2 describes the system flow in terms of implementation. As the system starts the user is prompted to upload the image. The image is then processed by the system, features are then extracted from the processed image and it is then classified. Considering the classification results system predicts the writer's gender, handedness and personality and displays the profile.

4.2.1 Collaboration Diagram

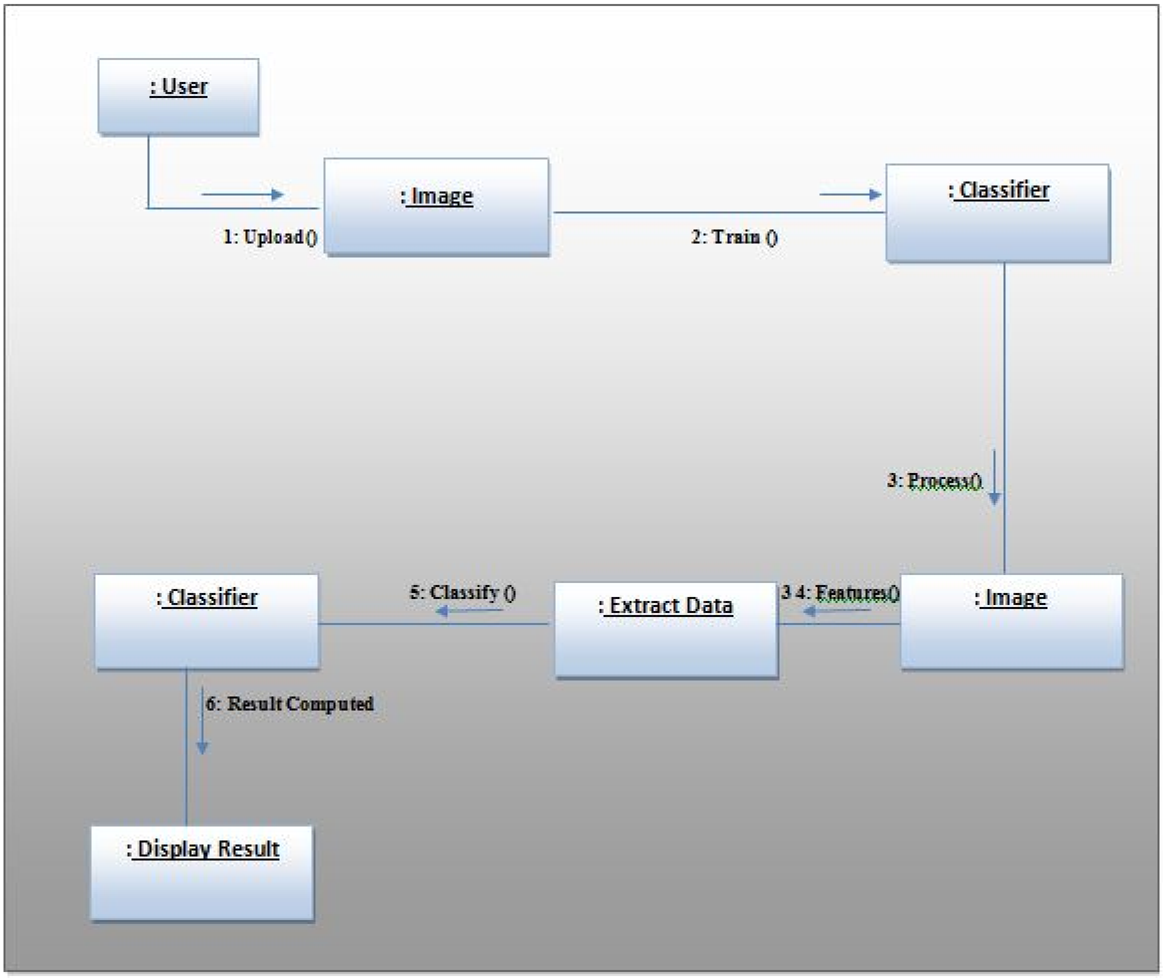


Figure 4.3: Collaboration Diagram

Figure 4.3 describes the organization of objects involved in the system and the flow of messages through them. Initial object is the image that is uploaded by the user, then the classifier that is trained by the user, again the image that is processed by the system, then data that are the features extracted by the system from processed image, classifier then classifies that data and computes the results that are then displayed.

4.2.2 Block Diagram

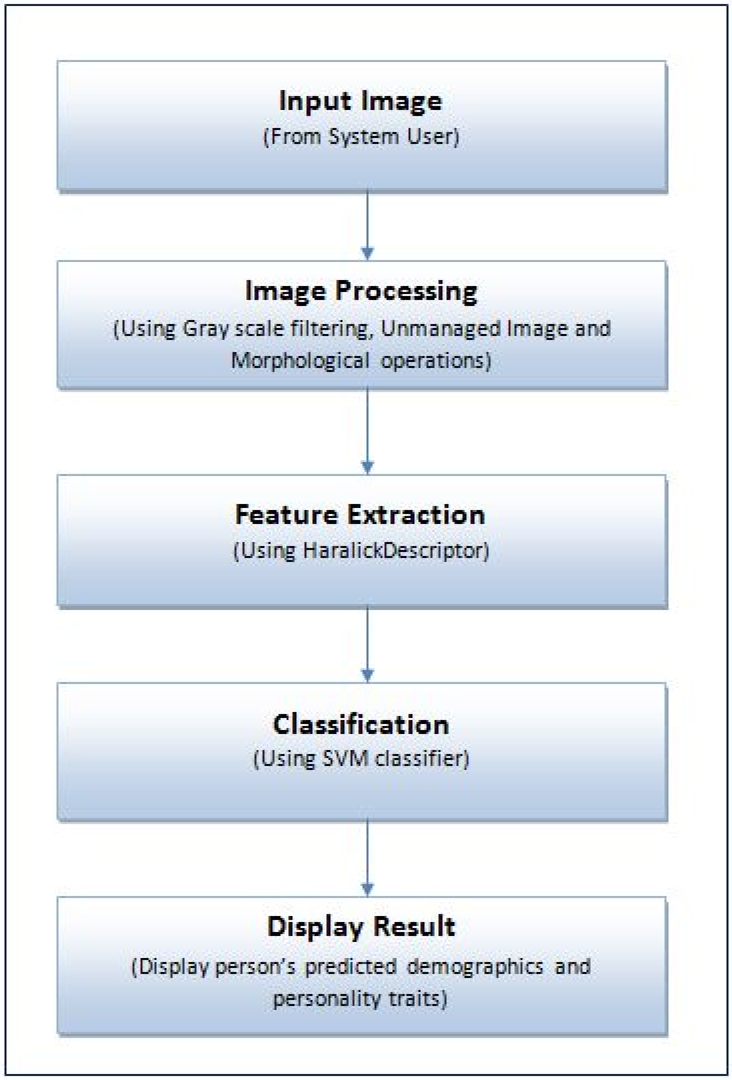


Figure 4.4: Block Diagram

Figure 4.4 shows the schema that highlights the arrangement of various components of the system. It defines the connection between the components of system that are Input Image, Image Processing, Feature Extraction, Classification and display result.

4.2.3 Package Diagram

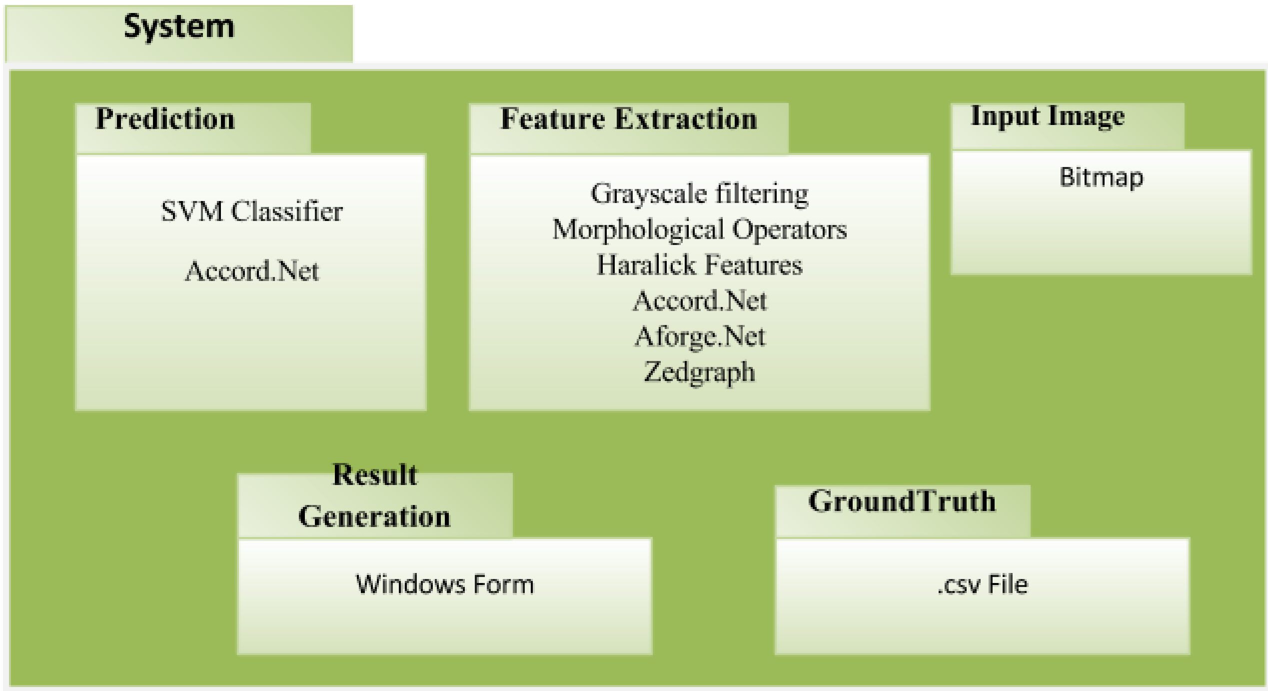


Figure 4.5: PackageDiagram

Figure 4.5 shows the dependencies between different packages of the system. It defines how a respective module is dependent on anything. Prediction in our system is dependent on the SVM Classifier from Accord.net Library, Feature extraction is dependent on Grayscale filtering, Morphological operators, Haralick features obtained from Accord.net and AForge.net library, Input image that is bitmap format dependent, Result Generation on Windows Form and ground truth on .csv file.

4.2.4 Deployment Diagram

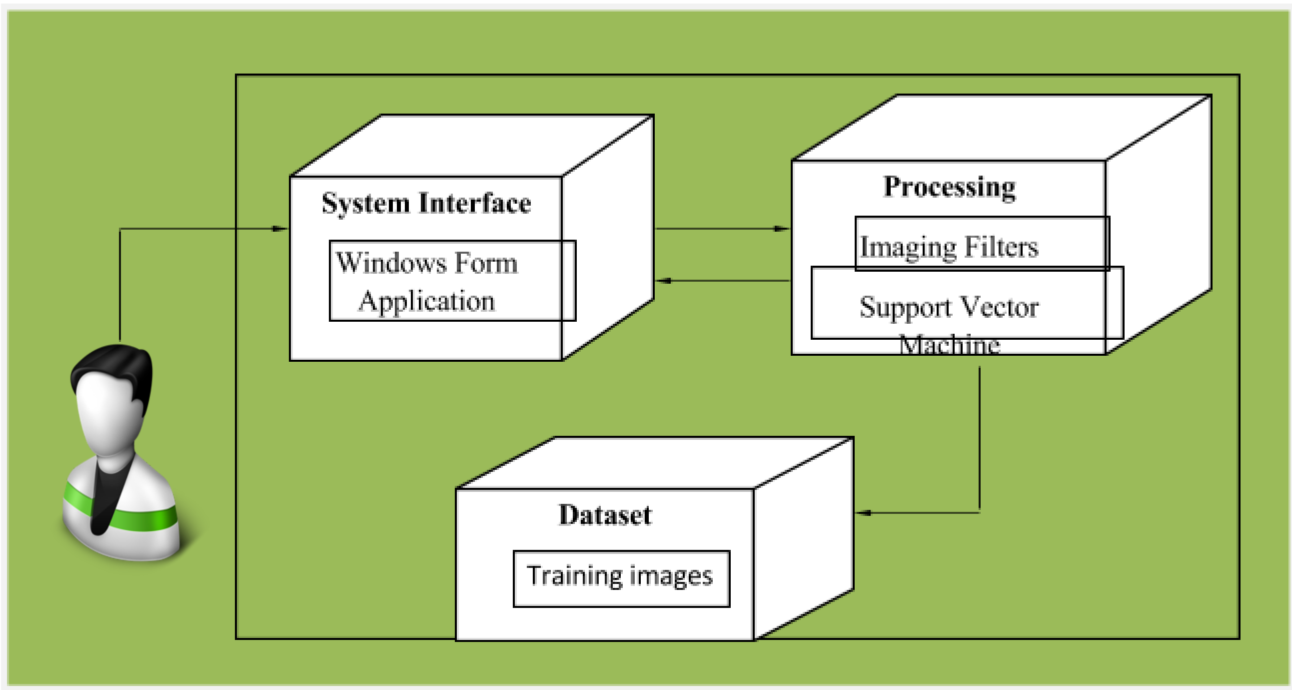


Figure 4.6 : Deployment Diagram

Figure 4.6 shows how the system objects will be deployed physically. System interface is linked with processing bidirectionally that is based on the dataset that contains training images.

4.2.5 Component Diagram

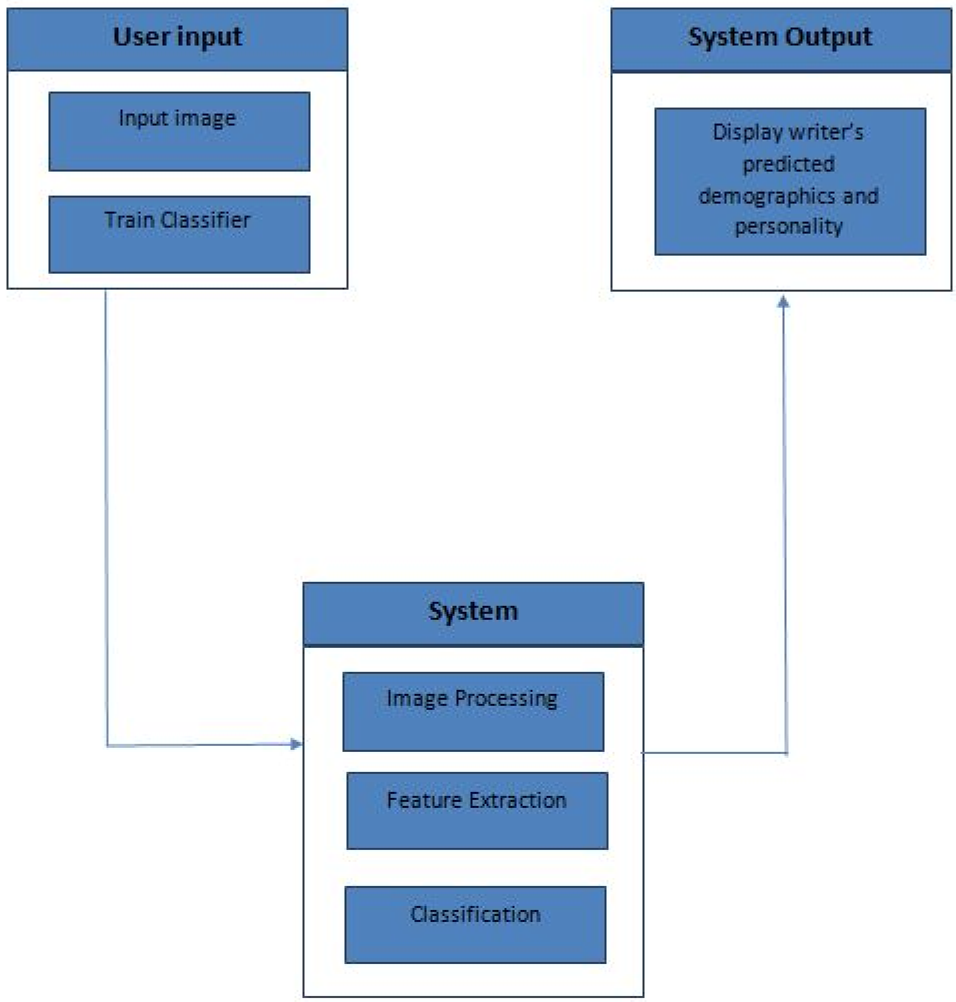


Figure 4.6: Component Diagram

Figure 4.6 defines how different components are used to make the functionalities of the system.

4.2 Low Level Design

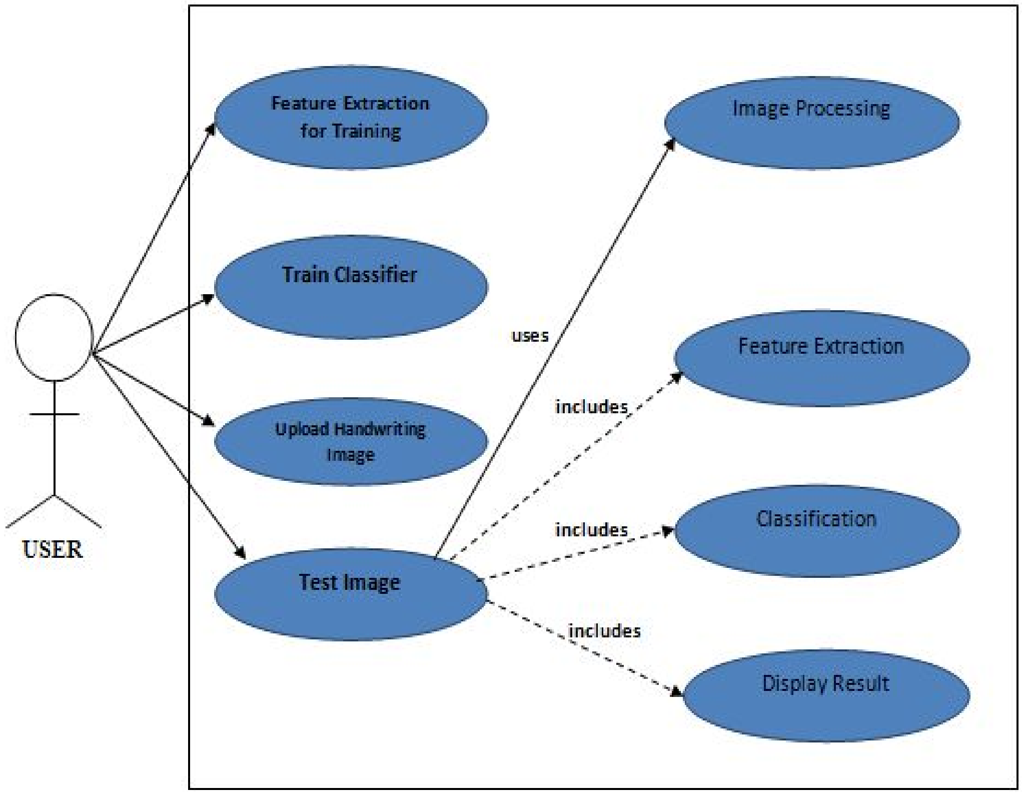


Figure 4.7 : Use Case Diagram

Actors

- User
- System

Use cases

- Upload Image
- Train Classifier
- Image Processing
- Feature Extraction
- Classification
- Display Result

Table:4.1: UC1: Upload Image

Use case Name	**UC1:** Upload Image
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	User Goal
Description	User Uploaded handwriting image from the source to system
Primary Actor	User
Pre-Condition	User should have handwriting image of the writer
Success Guarantee	Successfully uploaded image
Special Requirements	System should be responsive all the time and process fastly
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

Table:4.2: UC2: Train Classifier

Use case Name	**UC2:** Train Classifier
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	User Goal
Description	User trained the classifier in order to test the uploaded image
Primary Actor	User
Pre-Condition	Features from the training set should be extracted and saved in a file
Success Guarantee	Successfully trained classifier
Special Requirements	System should be responsive all the time and process fastly
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

Table:4.3: UC3: Image Processing

Use case Name	**UC3:** Image Processing
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	Sub-function
Description	In this system convert the input image into grayscale and then unmanaged image from which gray level concurrence matrix will be computed
Primary Actor	System
Pre-Condition	Image must be uploaded successfully
Success Guarantee	Perfectly transformed image
Special Requirements	None
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

Table:4.4: UC4: Feature Extraction

Use case Name	**UC4:** Feature Extraction
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	Sub-function

Description	In this system extract the image data from the computed GLCM Matrix. The features computed in this phase are Haralick features.
Primary Actor	System
Pre-Condition	Successfully transformed image
Success Guarantee	Image data must be ready for classification
Special Requirements	None
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

Table:4.5: UC5: Classification

Use case Name	**UC5:** Classification
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	Sub-function
Description	System passes the obtained feature vector to the classifier to compute results.
Primary Actor	System
Pre-Condition	We must have correctly computed feature vector
Success Guarantee	Data gives some output
Special Requirements	None
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

Table:4.6: UC6: Display Result

Use case Name	**UC6:** Display Result
Scope	Gender, Handedness and Personality Prediction from Offline Handwriting Images
Level	User Goal
Description	System will display writer's predicted gender, handedness and personality
Primary Actor	User
Pre-Condition	Classifier must be trained
Success Guarantee	System display result in windows form
Special Requirements	System should be responsive all the time and display results fast
Frequency Of Occurrence	Anytime(user dependent)
Miscellaneous	None

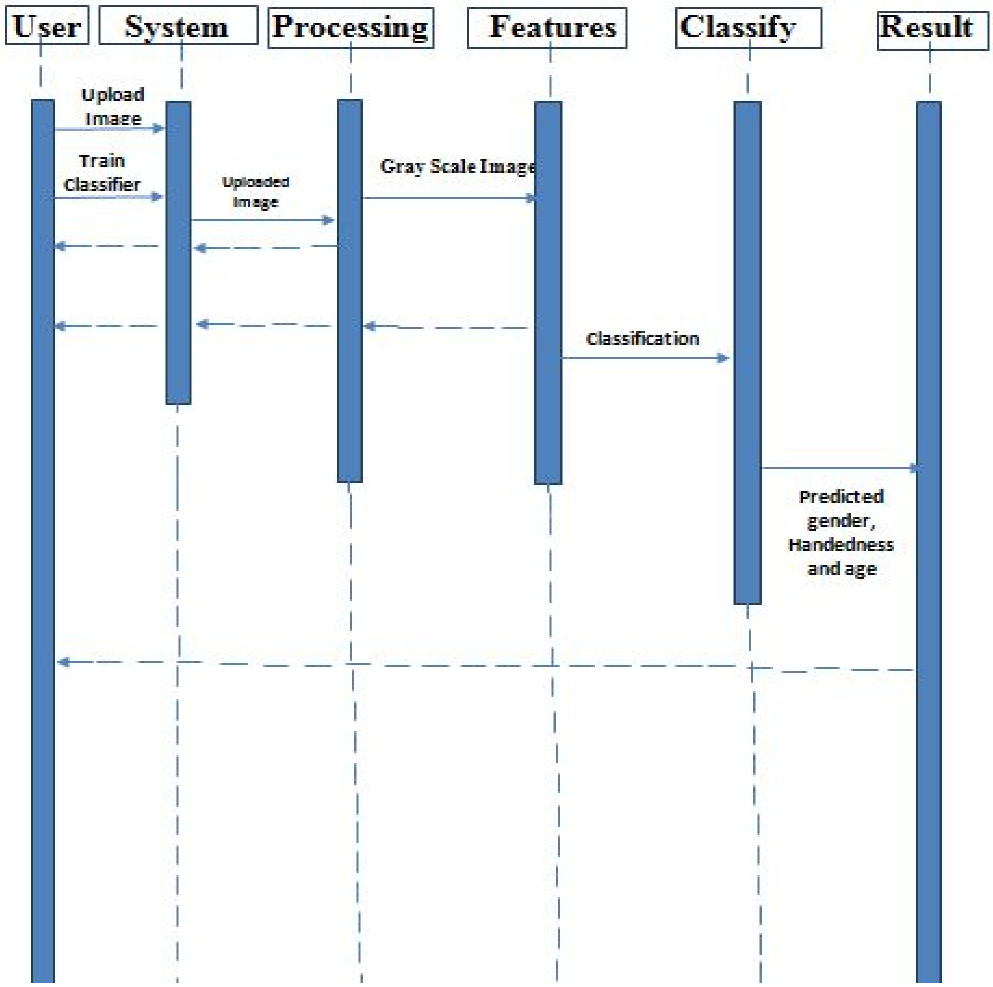


Figure 4.8: Sequence Diagram

Chapter 5

SYSTEM IMPLEMENTATION

This chapter provides a detailed description for all algorithms, tools used and implementation details of the system.

5.1 Tools and Technology

5.1.1 Visual Studio 2013

Microsoft Visual Studio is an incorporated improvement environment (IDE) from Microsoft. It is utilized to create PC programs for Microsoft Windows, and additionally sites, web applications and web administrations. Visual Studio utilizes Microsoft programming improvement stages, for example, Windows Programming interface, Windows Frames, Windows Presentation Establishment, Windows Store and Microsoft Silverlight. It can create both local code and oversaw code.

Visual Studio underpins distinctive programming. Worked in languages incorporate C, C++ and C++/CLI (by means of Visual C++), VB.NET (by means of Visual Essential .NET), C# (by means of Visual C#), and F#. Support for different languages, for example, Python, Ruby, Node.js, and M among others is accessible by means of dialect administrations introduced independently. It additionally supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were bolstered before.

5.1.2 Accord.Net, AForge.Net

These are the libraries used in the project in order to carry out different image processing tasks. These libraries can easily be configured with the Visual Studio.

5.2 Languages

This project has been developed in C# windows form. It gives a good user interface that is efficient and user friendly.

5.3 Methodology and Algorithmic Development

Following are the different algorithms details which are involved in implementation.

5.3.1 Gender and Handedness Classification

The following steps are carried out for classification of gender and handedness.

1. Grayscale Conversion

The image is first converted to grayscale. AForge.net library has a class named Grayscale that is used to convert a colored image to grayscale. It takes RGB coefficients used for color image conversion to grayscale.

2. Unmanaged Image

This allocates new image in unmanaged memory, as to just wrap provided pointer to unmanaged memory, where an image is stored. Using unmanaged images is mostly beneficial when it is required to apply multiple image processing operations to a single image. Unmanaged images are represented internally using unmanaged memory buffer therefore the processing is fast. Both Accord and AForge have classes to convert managed image to unmanaged but the class from Accord is used in the project for this purpose.

3. Gray Level Co-occurrence Matrix

The GLCM calculates how often different combinations of pixel brightness values (gray levels) occur in an image. Figure 5.1 is an example of computing gray level co-occurrence matrix for an image with four gray levels. using the displacement (1,0) vector. There exist four displacements corresponding to four directions i.e. (0,1), (1,-1), (0,-1) and (-1,-1) correspond to 0°, 45°, 90° and 135°. These displacements define the direction of the neighbouring element. The size of GLCM matrix is equal to the number of gray levels in the image for e.g. if we have the image with four gray levels the GLCM matrix will be of size 4x4. GLCM computation takes into account two neighbouring pixels and looks for the co-occurrence of that pair throughout the image. It then gives the value of co-occurrence to the location of of that pair in the GLCM matrix.

0	0	1	1
0	0	1	1
0	2	2	2
2	3	3	3

(a)

	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

(b)

Figure 5.1: GLCM Example (a) 4x4 Image (b) GLCM matrix

Accord provides a class that converts an unmanaged image to gray level co-occurrence matrix.

GrayLevelCooccurrenceMatrix gl = new GrayLevelCooccurrenceMatrix();

double[,] glcm = gl.Compute(uIi);

4. Haralick Feature Extraction

Using the GLCM matrix as input 13 Haralick features; Angular Second Momentum, Contrast, Correlation, Sum of Squares: Variance, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Entropy, Difference Variance, Difference Entropy, First Information Measure, Second Information Measure.

In Accord; HaralickDescriptor class extracts these 13 textural features from the image taking the GLCM of the image as input;

HaralickDescriptor ho = new HaralickDescriptor(glcm);

5. Training and Classification

The proposed system is using 300 images from QUWI database in order to train for gender classification and around 50 images from the same database for handedness classification. Haralick features are extracted for training images and these features are stored in the csv file to train the classifier.

i. Support Vector Machine Classifier

Support Vector Machine (SVM) classifier is used in the proposed system for classification purpose. It is a classifier defined by a separating hyperplane. Being more descriptive, it takes labeled training data and outputs an optimal hyperplane which is then used to classify new examples. There are different types of support vector machines but the proposed system works on Kernel Support Vector Machines.

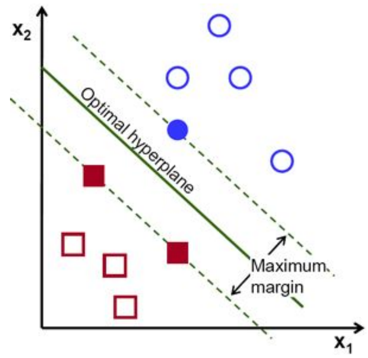


Figure 5.2: Support Vector Machine Optimal Hyperplane

a. Kernel Support Vector Machine

This type of SVM is used for efficient optimization and more accurate classification.

```
KernelSupportVectorMachine  machine = new KernelSupportVectorMachine(new Gaussian(), 2);
```

b. Sequential Minimal Optimization

It is a learning algorithm which trains the kernel support vector machine on the basis of inputs i.e. features and the outputs i.e. the labeled data corresponding to the inputs.

```
// Instantiate a new learning algorithm for SVMs
SequentialMinimalOptimization smo = new SequentialMinimalOptimization(machine, input, output);
```

5.3.2 Personality Traits:

1. Grayscale

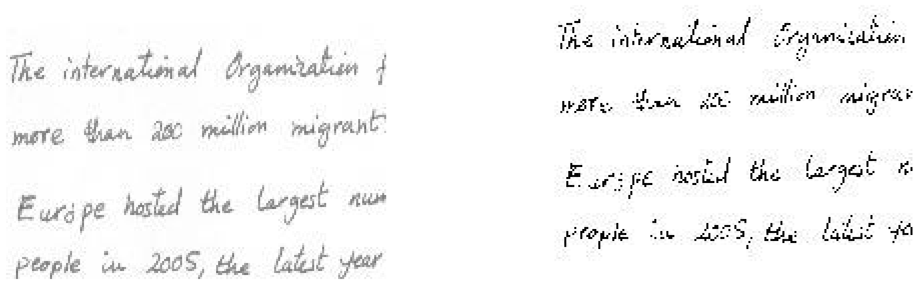
In order to make the image processing fast and independent of colors the image is converted to grayscale. AForge.net library has a class named Grayscale that is used for this conversion. It takes RGB coefficients to convert color image to grayscale.

```
Grayscale filter = new Grayscale( 0.2125, 0.7154, 0.0721 );
```

2. Binarization

The grayscale image obtained from the previous conversion is then binarized. AForge.NET system gives diverse binarization channels, which might be utilized to accomplish the task. The most simple binariazation technique that is used in the project is standard thresholding. The channel takes a threshold value and changes all pixels values equivalent or higher than threshold to white pixels and all other pixels with values below the threshold are changed to dark pixels.

```
Threshold filter = new Threshold( 100 );
```



(a) (b)

Figure 5.3: (a) Gray Scale Image (b) Binarized Gray Scale Image

3. Image Inverting

The channel rearranges hued and grayscale images i.e. converts dark pixels of the image to light and vice versa. Our system after binarizing the image inverts it.

```
Invert filter = new Invert( );
```

Figure 5.4 shows the inverted image of the above binarized image.

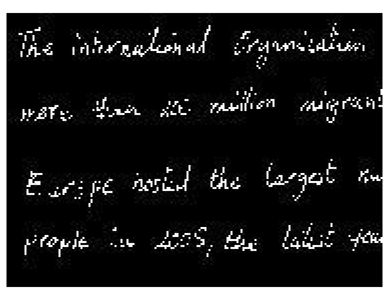


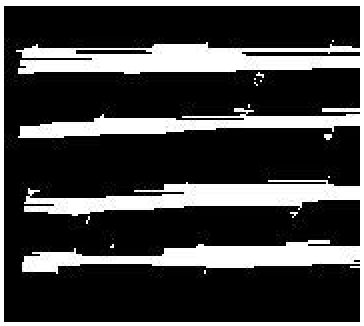
Figure 5.4: Inverted Image

4. Horizontal Run Length Smoothing Algorithm

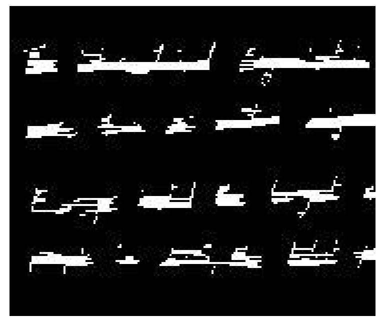
In addition to the above processes in the next step system applies Horizontal run length smoothing algorithm to the image. This algorithm merges all the white pixels leaving the isolated components separate. This merging results in the conversion of characters to words and sentences. It can be said that its usage fills flat dark holes between white pixels.

```
HorizontalRunLengthSmoothing hrls = new HorizontalRunLengthSmoothing( 32 );
```

Figure 5.5 shows the implication of Horizontal Run Length Algorithms (a) is for lines and (b) is for words.



(a)



(b)

Figure 5.5: (a) Horizontal Run Length Smoothing Algorithm (a) Line Detection (b) Words Detection

5. Image Erosion

After the implication of run length smoothing algorithm the system erodes the image that is brings about the shrinking effect on the image. It is applies to the image with black background. A structuring element is defined in order to remove any kind of extrusion in the image and separate the incorrectly joined objects. AForge.net has an Erosion class that performs this task.

```
Erosion filter = new Erosion( );
```

Figure 5.6 shows the eroded image of the above image on which HRLSA has been applied for lines.

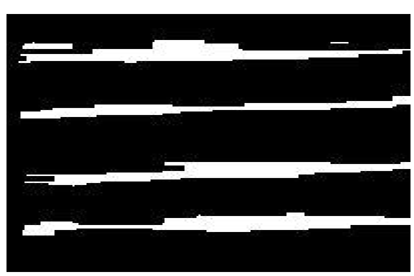


Figure 5.6: Erosion

6. Image Dilation

The successful erosion of the image is then followed by another morphological operation in the proposed system; that is Dilation. It brings about growing effect in the image. Being more descriptive it expands the objects in the foreground. This operator is also responsible for smoothing the object boundaries and filling the gaps between the components of the image.

AForge.net provides a class named Dilation in order to serve the purpose.

```
Dilatation filter = new Dilatation( );
```

Figure 5.7 shows the dilated image of the above eroded image.

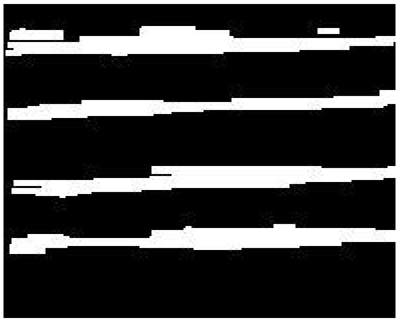


Figure 5.7: Dilation

7. Image Closing

Closing morphology operator is then applied to the image after being dilated. It is a compound operator that carries out dilatation followed by erosion. It is used to connect or fill objects. Since dilatation is used first, it fills the object areas, then erosion restores the objects. As dilatation connects object areas at first hand, erosion does not remove after that connection because of the formed connection.

AForge.net provide a class named Closing to serve this purpose.

```
Closing filter = new Closing( );
```

Figure 5.8 shows the closing operation on the above dilated image.

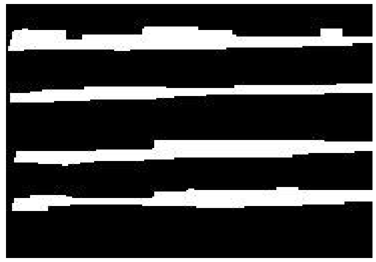


Figure 5.8: Closing

8. Connected Components Labeling

The system then performs labeling of objects in the source image. It colors each separate object using different color. It treats all non black pixels as objects' pixels and all black pixel as background.

AForge.net provide a class named ConnectedComponentLabeling to fulfil this task.

```
ConnectedComponentsLabeling filter = new ConnectedComponentsLabeling( );
```

Figure 5.9 shows the Connected Component Labelling on the above binarized image.

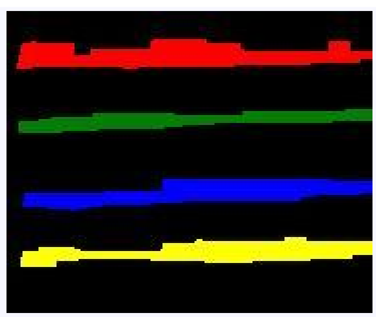


Figure 5.9: Connected Component Labelling

9. Classification

Classification for the personality in the proposed system is unsupervised that is it does not have any training data. The system uses three handwriting features for personality classification including size, slope and spacing between words. Aforge.Net has a class named BlobCenter that processes the image, calculates the blobs and stores informaion about the objects in those blobs. Properties of every blob provide data for computation of features.

i. Spacing

Rectangle property of Blob class gives the height, width, x-coordinate and y-coordinate of the object. The system by performing computations using these values stores the distance between two neighbouring objects in an array. It then computes the average of all those stored values and gives the result on the basis of a threshold value above which the spacing is considered to be more and below that value the spacing is less.

Figure 5.10 shows how spacing classifies a person’s personality into two classes.

Figure 5.10: Spacing and related Personality Traits

ii. Size

Blob class also has an area property, it is basically the number of pixels utilized by an object in the image. In order to see if the letter size is small, average or large the system stores the the area of all the letters and takes the average. For this feature we take two threshold values such that above one value size is considered to be average and above the other value size is considered to be large and vice versa.

Figure 5.11 shows how size classifies a person’s personality into three classes.

Figure 5.11: Size and related Personality Traits

iii. Slope

AForge.net has a class named DocumentSkewChecker which is used to determine the slope of the written text all that is needed is to input the image of the document. GetSkewAngle function of this class serves the purpose and provides orientation for overall image.

Figure 5.12 shows how slope or orientation classifies a person's personality into three classes.

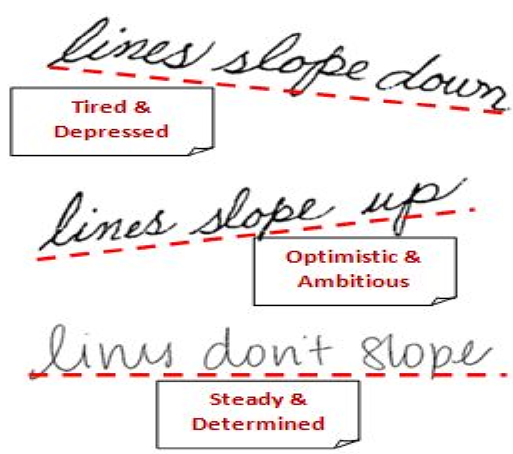


Figure 5.12: Slope and related Personality Traits

Therefore, the chapter explains the process of the system implementation. The main steps and the substeps have been explained in detail.

Chapter 6

SYSTEM TESTING AND EVALUATION

Testing is a practice utilized to finish the product advancement handle and plan the application/item for organization. Testing guarantees that the item created is free of mistakes and works in a worthy way. This section highlights the distinctive sorts of testing strategies used to check the framework's execution. Alongside testing, the framework is assessed for its consistence with the prerequisites indicated before. In this chapter, we introduce diverse test cases and the framework's conduct under every test to choose if the activity is satisfactory or not.

6.1 Software Testing Techniques

There are various techniques that can be utilized to check for a framework's precision and any bugs. The testing methods used in this project are said underneath with a detail of the testing strategy and the system's reaction under each.

6.1.1 Usability Testing

Keeping in mind the end goal to make the product easy to understand, ease of use testing is performed. We checked for the distinctive ease of use standards on the off chance that they are found in the application or not. End clients utilize the product and we observed their reactions concerning the perceivability, mapping of controls, affordance and consistency of catches, symbols, names and different tools utilized.

6.1.2 Software Performance Testing

Quality Attributes of a product, for example, its unwavering quality i.e. reliability, accessibility and soundness are tested through execution estimations. Therefore, it can be said that non-functional elements of the framework are tested.

6.1.3 Compatibility Testing

Compatibility tests are performed to check if the application made can keep running on various computing equipment and operating systems. This project is intended to keep running on any platform however it is tested on various windows operating system.

6.1.4 Functionality Testing

The principal point of any system is to provide complete functionality that it was at first proposed for, and full all the client necessities. Testing for the diverse elements and usefulness of a system is an ongoing procedure that is done all through the improvement procedure; it is a quality confirmation movement. Distinctive types of testing strategies used to check the usefulness are mentioned point by point beneath;

i. Unit Testing

Distinctive functionalities, techniques and classes utilized as a part of the system are tested through unit testing. Unit tests are performed by the coder to check if the required yield is created by each strategy or not.

ii. Test Cases

Table 6.1: TC-01: Upload Image

Test Case ID	TC-01
Test Priority (Low/Medium/High)	Medium
Module Name	Upload Image
Test Title	Load Image from source to system
Description	See if the required handwriting image is correctly loaded in the system or not
Precondition	The image path must be valid
Dependencies	Accord.Imaging.dll (Assembly)

Step	Test Step	Test Data	Expected Result	Actual Result	Status

1.	Click Browse button.	Collection of handwriting images.	Open File dialog should appear.	Open File dialog appears.	Pass
2.	Select the image to be tested.	Selected image's path	Selected image's path and the image should appear on screen.	Image and image path appeared on the screen.	Pass

Table 6.2: TC-02: Feature Extraction

Test Case ID	TC-02
Test Priority (Low/Medium/High)	High
Module Name	Feature Extraction
Test Title	Extract Features
Description	Check if the features are extracted and successfully stored in the .csv file.
Precondition	Training set must be there.
Dependencies	Accord.MachineLearning.dll (Assembly)

Step	Test Step	Test Data	Expected Result	Actual Result	Status
1.	Click Extract Features button.	Collection of handwriting images of training set.	Message box should appear saying "Features Extracted".	Message box appeared saying "Features Extracted".	Pass
2.	Open the csv file to check if the feature values are correctly written on it or not.	.csv file.	Feature values should be written onto the FeatureFile.cs v	Feature values were written onto the FeatureFile.cs v	Pass

Table 6.3: TC-03: Train Classifier

Test Case ID	TC-03
Test Priority (Low/Medium/High)	High
Module Name	Classifier Training
Test Title	Train Classifier
Description	Check if the classifier is successfully being trained on the basis of features extracted from the training set.
Precondition	Feature values of the training set must be stored in the file.
Dependencies	Accord.MachineLearning.dll (Assembly)

Step	Test Step	Test Data	Expected Result	Actual Result	Status
1.	Click Train button.	Collection of handwriting images of training set.	Message box should appear saying “Classifier Trained”.	Message box appeared saying “Classifier Trained”.	Pass

Table 6.4: TC-04: Display Result

Test Case ID	TC-04
Test Priority (Low/Medium/High)	High
Module Name	Displaying Results
Test Title	Test Image
Description	Check if the results appearing on the GUI are accurate or not.
Precondition	Image must be correctly processed.
Dependencies	Accord.MachineLearning.dll (Assembly)

Step	Test Step	Test Data	Expected Result	Actual Result	Status

1.	Click Test button.	Collection of handwriting images of testing set.	Results should appear on the GUI.	Results appeared on the GUI.	Pass
2.	Validate Results and find accuracy rate.	Excel file containing the results of the test data.	Results should be similar.	Results were 58% and 70% similar.	Pass

iii. Integration Testing

Integration testing is a phase of software testing where individual modules are combined and tested as a group. Individual modules of my project include;

- Gender prediction
- Handedness prediction
- Personality prediction

These modules are first tested separately in unit testing and here they are integrated and tested as one.

iv. System Testing

It checks the complete behavior of the system developed on the basis of SRS document. Fundamental concentration of system testing is to assess practical prerequisites of the system. We have performed the usability tests to check the usability of the system, recuperation testing to confirm the unwavering quality of the system i.e. reliability and functional testing to check the fulfillment as far as necessities of the client are concerned.

v. White Box Testing

It tests the internal structure of the component or system developed. We have tested our system's internal structure and grouping of proclamations along with the limit conditions assessment.

vi. Black Box Testing

It tests the functionalities of the system without considering the internal structure and implementation of the system. Accuracy of system yields is guaranteed by applying diverse inputs to it.

6.1.5 Acceptance Testing

It determines if the system has met the required particulars or not. It is performed by the user or system manager. My system has been tests according to the criteria of acceptance testing and has given positive results.

6.2 Results

As discussed earlier the system was evaluated on a subset of the QUWI database. 300 writing samples were used in the training set. For testing, 100 samples were used for the gender tasks and 34 for the gender tasks. It should be noted that we used the same experimental protocol as employed in the ICFRH 2016 International competition on these tasks. Table 6.5 shows the results obtained by the participants of the competition [11] and the proposed system. It can be seen that our results are comparable with most of the participating systems.

Table 6.5: Comparative Results

Method	**Gender Classification Rates**
MCS NUST 1	56.00
MCS NUST 2	57.00
Nuremberg 1	54.00
Nuremberg 2	74.00
CVC 1	44.40
CVC 2	44.40
Our Results	58.00
Method	**Handedness Classification Rates**
Nuremberg 1	62.50
Nuremberg 2	84.37
CVC 1	43.75
CVC 2	43.75
Our Results	71.00

For personality we have used the images from the same QUWI database for both setting threshold values and testing and received almost 80% results which is a considerable success of the system.

Chapter 7

CONCLUSION

A system to predict gender, handedness and personality of individuals from offline handwriting images is presented. Harlick features computed from handwriting images are used to train a support vector machine classifier for gender and handedness detection. Evaluations are carried out on a subset of the standard QUWI database. The system realized a gender classification rate of 58% and handedness classification rate of 70%. These results are comparable to those reported in the ICFHR 2016 competition on these tasks. For personality profiling, we employed an unsupervised approach and computed slope of lines, line spacing and word spacing to identify the personality of writer.

The present project can be extended to include prediction of other demographics of writers as well. For instance prediction of age groups and race can also be investigated. Moreover, the present study is based on offline image of handwriting. Online handwriting can also be analyzed that has additional information in terms of writing speed, pressure, number of strokes etc.

References

1. Al Maadeed, Somaya, and Abdelaali Hassaine. "Automatic prediction of age, gender, and nationality in offline handwriting." *EURASIP Journal on Image and Video Processing* 2014.1 (2014): 1-10.
2. Al-Maadeed, Somaya, et al. "A novel approach for handedness detection from off-line handwriting using fuzzy conceptual reduction." *EURASIP Journal on Image and Video Processing* 2016.1 (2016): 1-14.
3. Bouadjenek, Nesrine, Hassiba Nemmour, and Youcef Chibani. "Histogram of Oriented Gradients for writer's gender, handedness and age prediction."*Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on*. IEEE, 2015.
4. Bouadjenek, Nesrine, Hassiba Nemmour, and Youcef Chibani. "Age, gender and handedness prediction from handwriting using gradient features."*Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015.
5. Fisher, Janet, et al. "Identifying Personality Traits, and Especially Traits Resulting in Violent Behaviour through Automatic Handwriting Analysis."*Proceedings of Student-Faculty Research Day, CSIS, Pace University, USA, May* 4 (2012).
6. Prasad, Shitala, Vivek Kumar Singh, and Akshay Sapre. "Handwriting Analysis based on Segmentation Method for Prediction of Human Personality using Support Vector Machine." *International Journal of Computer Applications* 8.12 (2010): 25-29.
7. Kacker, Rashi, and Hima Bindu Maringanti. "Personality analysis through handwriting." *GSTF Journal on Computing (JoC)* 2.1 (2014).
8. Prof. Seema Kedar, Ms. Vaishnavi Nair, Ms. Shweta Kulkarni. " Personality Identification through Handwriting Analysis: A Review*".* *International Journal of Advanced Research in Computer Science and Software Engineering. Volume 5, Issue 1*, January 2015
9. Michael Fairhurst, and M´arjory Abreu. "An Investigation of Predictive Profiling from Handwritten Signature Data". *2009 10th International Conference on Document Analysis*
10. Prasad, Shitala, Vivek Kumar Singh, and Akshay Sapre. "Handwriting Analysis based on Segmentation Method for Prediction of Human Personality using Support Vector Machine." *International Journal of Computer Applications* 8.12 (2010): 25-29.
11. Chawki Djeddi, Somaya Al-Maadeed, Abdeljalil Gattal, Imran Siddiqi, Abdellatif Ennaji, Haikal El Abed. "Multi-script Writer Demographics Classification Using "QUWI" Database". *ICFHR2016 Competition*