

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**Automatic Analysis of handwriting Prediction Demographics**

***By***

**Ghadeer Hussain 16-ARID-1167**

**Shabab Hussain 16-ARID-1110**

***Supervisor*Mr. Zeeshan Javed**

***Bachelor of Science in Information Technology (2016-2020)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Ghadeer Hussain Shabab Hussain

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

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (IT) **“Automatic Analysis of handwriting Prediction Demographics”** was developed by “**Ghadeer Hussain, 16-ARID-1167”**, “and “**Shabab Hussain turi, 16-ARID-1110”** under the supervision of “**Mr. Zeeshan javed**” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Information Technology.

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

**Supervisor**

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

**External Examiner (If any)**

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

**Administrator UIIT**

**Executive Summary**

Handwriting is one of the oldest modes allowing individuals to communicate with one another. In old ages, different shapes and symbols were used for communications and later proper alphabets were developed. Despite tremendous growth of digital media, the importance of handwriting has retained its place. A handwritten piece of text carries information about the person who has produced the text as compared to the printed text. Handwriting analysis has been successfully applied to problems like character/handwriting recognition, signature verification and identification and verification of writers.

A more established and validated problem is the prediction of user demographics from his or her handwriting. Correlation is known to exist between handwriting and the gender, age and handedness (left or right) etc. of writers. Although a very mature research area in psychological sciences, prediction of user demographics from handwriting is a very less investigated area by the computer scientists. This project will focus on development of an automated system to predict user demographics from handwriting.

The main hypothesis of this project is to study whether a correlation exists between handwriting of individuals and their different demographic attributes. For this particular project, we will restrict our study to the prediction of gender, age and handedness. The developed system will take scanned images of handwriting as input and predict the gender, age and handedness of the author by extracting the set of features from male and female writers and then to train the classifiers to learn to discriminate between the two. The scope of this study is limited to digitized (offline) images of handwriting only. The proposed system for demographics classification will rely on extracting a set of features from binarized images of handwriting. These images will aim to capture the orientation and textural information in the writings

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Zeeshan Javed” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Ghadeer Hussain Shabab Hussain

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

**Abbreviations**

|  |  |
| --- | --- |
| **oBIFs** | Oriented Basic Image Features |
| **QUWI** | Qatar University Writer Identification Dataset |
| **SVM** | Support Vector Machine |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 2

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Project Background](#_Toc268523780) 3

[1.4 Literature Review](#_Toc268523780) 3

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 5

**Problem Definition** 6

[2.1 Purpose](#_Toc268523795) 7

[2.2 Product Functions](#_Toc268523796) 7

[2.3 Proposed Architecture](#_Toc268523804) 7

[2.4 Project Deliverables](#_Toc268523806) 8

[2.5 Operating Environment](#_Toc268523807) 8

[2.6 Assumptions and Dependencies](#_Toc268523808) 8

**Requirement Analysis** 9

[3.1 Functional Requirments](#_Toc268523823) 10

[3.2 Non – Functional Requirments](#_Toc268523825) 11

[3.2.1 Usability](#_Toc268523787) 11

[3.2.2 Reliability](#_Toc268523787) 11

[3.2.3 Performance](#_Toc268523787) 11

[3.2.4 Supportability](#_Toc268523787) 11

[3.2.5 Design Constraints](#_Toc268523787) 11

[3.2.6 Licensing Requirements](#_Toc268523787) 11

[3.3 Use case Model](#_Toc268523823) 12

[3.3.1 Use Case Diagarm](#_Toc268523787) 12

[3.3.2 Actors Discription](#_Toc268523787) 16

[3.3.3 Use Case Discription](#_Toc268523787) 17

**The Design** 21

[4.1 UML Structural Diagrams](#_Toc268523830) 22

[4.1.1 Component Diagram](#_Toc268523787) 22

[4.1.2 System Component Diagram](#_Toc268523787) 24

[4.1.3 Package Diagram](#_Toc268523787) 25

[4.1.4 Deployment Diagram](#_Toc268523787) 26

[4.2 UML Behavioral Diagrams](#_Toc268523830) 27

[4.2.1 Activity Diagrams](#_Toc268523787) 27

[4.2.2 State Machine Diagrams](#_Toc268523787) 29

[4.3 UML Interaction Diagrams](#_Toc268523830) 30

[4.3.1 Sequence Diagrams](#_Toc268523787) 30

[4.4 Node Structure](#_Toc268523830) 31

[4.5 Communication Design Protocol](#_Toc268523830) 32

**Implementation** 33

[5.1 Communication Protocol Implementation](#_Toc268523830) 34

[5.2 PC Application Implementation](#_Toc268523830) 36

[5.3 Embedded Application Implementation](#_Toc268523830) 38

[5.4 Wireless Sensor Application Implementation](#_Toc268523830) 51

**Testing and Evaluation** 52

[6.1 Verification](#_Toc268523830) 53

[6.1.1 Functional Testing](#_Toc268523787) 53

[6.1.2 Static Testing](#_Toc268523787) 57

[6.2 Validation](#_Toc268523830) 57

[6.3 Usability Testing](#_Toc268523830) 57

[6.4 Unit Testing](#_Toc268523830) 57

* 1. [Integration Testing](#_Toc268523830) 57

6.6 [System Testing](#_Toc268523830) 57

**GUI** 58

**Future Work** 70

**References** 72

**List of Figures**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

**Chapter 01: introduction**

This chapter is all about introduction of our project which include a brief introduction of project, literature background of our project and methodology we will use in our system.

**Brief:**

A system for automatic analysis of handwriting to predict gender and handedness of writers is developed in this study. Although a number of studies have been carried out on this problem by researchers in psychological studies, computerized analysis of handwriting for this task is a relatively recent area of research. The main hypothesis of this study was to investigate the features that could be exploited for prediction of writer demographics from handwriting samples. The developed system relies on binarizing the handwriting images and extracting a set of textural features namely histogram of oriented gradients and Harlick features. These features are used to train an SVM to classify a given writing on the basis of gender and handedness. The system was evaluated on the QUWI database and the realized results validated the ideas proposed in this study. The developed system can be effectively employed in forensic document analysis systems where the search space can be reduced by focusing on a set of writers with certain demographic attributes.

# **1.2. Relevance to the course model:**

Following Courses are related to our project that we learnt in our degree.

* Software Engineering-I and Software Engineering-II help us to design UML diagrams and Use case model for our system.
* Database systems and Distributed Database system help us in designing and maintaining database for our system.

# **1.3. Project background:**

Over the past few years, researchers have carried out efforts to enhance the computerized analysis of handwriting to identify writers and to predict demographic attributes such as the gender, age and handedness from hand- writing.

# **1.4 Literature review:**

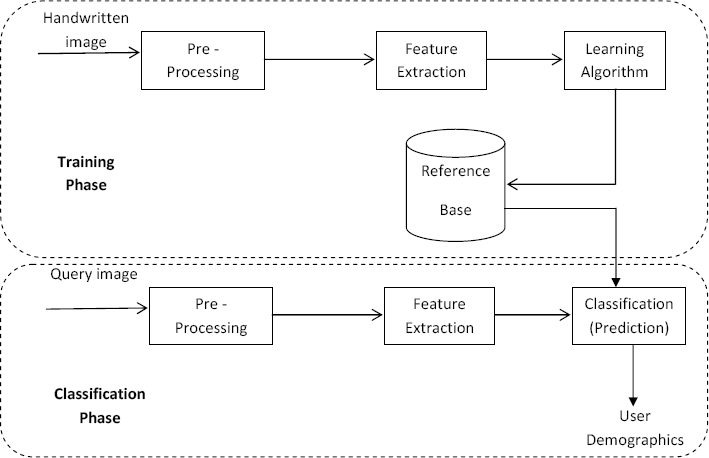
Gender Classification from offline Multi-script handwriting images using Oriented basic Image Features[. [1]](https://www.sciencedirect.com/journal/expert-systems-with-applications/vol/99/suppl/C) is a system which is working on hand writing based gender recognition that exploits texture as the discriminative attribute between male and female handwriting. The textural information in handwriting is captured using combinations of different configurations of oriented Basic Image Features (oBIFs) oBIFs histograms and oBIFs columns histograms extracted from writing samples of male and female handwriting are used to train a Support Vector Machine classifier (SVM). The system is evaluated on three subsets of the QUWI database of Arabic and English.

Improving Handwriting based gender classification using ensemble classifiers [[2]](https://www.sciencedirect.com/journal/expert-systems-with-applications/vol/85/suppl/C) is a running system which predict gender from images if handwriting using textural descriptors. Multiple classifier to discriminate male and female writings. The features include local binary patterns (LBP), histogram of oriented gradients (HOG), statistics computed from gray-level co-occurrence matrices (GLCM) and features extracted through segmentation-based fractal texture analysis (SFTA). For classification, we employ artificial neural networks (ANN), support vector machine (SVM), nearest neighbor classifier (NN), decision trees (DT) and random forests (RF). Classifiers are then combined using bagging, voting and stacking techniques to enhance the overall system performance.

Writer identification using texture descriptors of handwritten fragments [[3]](https://www.sciencedirect.com/journal/expert-systems-with-applications/vol/47/suppl/C). A texture based approach for offline, text-independent, writer identification. The proposed technique divides a given handwriting into small fragments and considers each fragment as a texture. Texture descriptors including histograms of Local Binary Patterns (LBP), Local Ternary Patterns (LTP) and Local Phase Quantization (LPQ) are then computed from these fragments. The writer of a document is characterized by the set of histograms calculated from all the fragments in the writing. Two writings are compared by computing the distance between the descriptors of their writing fragments. The technique evaluated on IFN/ENIT and IAM databases comprising handwritten text in Arabic and English, respectively, realized high identification rates.

# **1.5 Methodology and software lifecycle for this project**

This section presents in detail the methodology of our system. Like any pattern classification task, the proposed system comprises two traditional phases of training and classification. During training phase, features ex- tracked from a set of writing samples are used to train a classifier. Once trained, the features extracted from a query writing sample are fed to the classifier which outputs the required class labels. In our case, the feature set comprises histogram of oriented gradients and Harlick textural descriptors. For classification we employ Support Vector Machine (SVM) and the target classes in our system include male/female and left/right-handed.



# ***1.5.1. Rational behind selected methodology***

* Pre-Processing
* Pre-processing is the first step in any image analysis system that prepares the images into a form that is appropriate for further processing. Depending upon the image acquisition module and the processing required, preprocessing for document or handwriting images may include segmentation of text from graphics, skew correction, noise removal, Binarization or extraction of different regions of interest. In our case, we work on scanned images of handwriting and the features we calculate are global textural measures which do not require segmentation of text into lines or words. Consequently, the preprocessing in our case comprises segmenting the text and the background, i.e. Binarization.
* Depending upon the images under study, Binarization can be based on global or local thresholding. In global thresholding, a single threshold value computed from the histogram of the gray scale image is used to binarize the image. Local thresholding, on the other hand, comprises sliding a window over the image and computing a threshold for every pixel of the image as a function of its neighboring pixels. Since we deal with scanned images of handwriting, the histogram of the images comprises a bi-modal distribution with two distinct peaks (Figure 3.2), one for the text and the other for the background. Consequently, we employ global thresholding and binarize the image as follows.

g(x,y) =

Where f (x, y) represents the original gray scale image, g (x, y) is the binarized image and T represents the threshold computed from the image. Figure 3.3 represents a grayscale image and the respective binarized version of the image.



Figure 3.2: Bimodal intensity histogram in handwriting images

Once the images are binarzied we proceed to the extraction of features as detailed in the following.

* Feature Extraction

To characterize writer demographics from handwriting, we have chosen to employ textural features which are known to work well for this problem. Texture cannot be defined for a particular point as it is described by the spatial distribution of grey levels in a neighborhood. The scale for texture is defined by the resolution at with the image is observed. For instance, if we observe image of a tiled floor from large distance, the texture formed by the tiles is well defined but the pattern within the tiles is not clear. But as the distance is reduced the pattern formed by the tiles seems very clear.

Texture classification involves distinguishing the textured region from a set of texture classes. The algorithms for texture analysis extract unique features from each region to assist the classification process of such patterns and statistical methods are widely used in texture classification.

The textural features considered in our study include.

. Histogram of Oriented Gradients

. Haralick Features

# **Chapter 2: Problem definition**

Problem statement of our project and the solution to that problem is discuss in this chapter.

Our project deliverables, project environment and assumption and dependencies of our project is also part of this chapter.

## **2.1. Problem statement:**

The main hypothesis of this project is to study whether a correlation exists between handwriting of individuals and their different demographic attributes. For this particular project, we will restrict our study to the prediction of gender and handedness. The developed system will take scanned images of handwriting as input and predict the gender and handedness of the author. The system mainly relies on extracting a set of features from male and female writers and then training a classifier to learn to discriminate between the two. The scope of this study is limited to digitized (offline) images of handwriting only.

**2.2 Purpose**

Purpose of our system is to predict and demographic of handwriting to the male and female. So it is easier and faster to predict the handedness of writer.

**2.3 Product function**

The developed system will take scanned images of handwriting as input and predict the gender and handedness of the author. The system mainly relies on extracting a set of features from male and female writers and then training a classifier to learn to discriminate between the two.

.

**2.4 Purposed architecture**

Our proposed system will work with handwriting prediction and demographic. Prediction and demographic process would be done, when a user tried to upload any picture of hand writing.

## **2.5. Deliverables and development requirements**

### ***Deliverables***

A system for automatic analysis of handwriting to predict gender and handedness of writers is developed in this study. Although a number of studies have been carried out on this problem by researchers in psychological studies, computerized analysis of handwriting for this task is a relatively recent area of research.

### ***Development requirements***

* Core I 7 laptop
* 4gb RAM
* 150 GB HHD
* Window 10

## **2.6. System creation**

The preprocessing, feature extraction and classiﬁcation steps are implemented in python while the algorithmic support is provided by the machine learning libraries.

The menu bar contains a list of menus each performing a speciﬁc action. These menus include the ‘Samples’, ‘Machine creation’, ‘Feature importance’, ‘Model testing’ and ‘Image Upload’ buttons. The GUI panel also contains four buttons including ‘Upload’, ‘Resize’, ‘Binarize’ and ‘Exit’. A brief description a important ones from the aforementioned is presented in the following.

* Upload: The ‘Upload’ button is used to select an image from a given directory. All images of the directory are then uploaded to the system.
* Resize: The images in the QUWI dataset are all of different sizes. Prior to feature extraction, all images need to be at the same size so that the features computed from different images have same length.
* Binarize: Converts the 8-bit gray scale image to binary image. • Exit: The ‘Exit’ button is used to quit using the system.
* Sample: ‘Sample ‘menu displays the feature values of all the uploaded images.
* Machine Creation: The machine creation menu initiates the SVM and prompts user for the training parameters. The features extracted from training images along with the class label of each are fed as input to train the SVM.
* Model Testing: Uploads the testing dataset, feeds features extracted from the test data to the trained machine and outputs the confusion matrix and the overall classiﬁcation rate.
* After having presented a brief review of the implementation and system design, we now present the experimental results of our system in the next.

# **Chapter 3: Requirement Analysis**

# Use Cases

Use cases are a widely used and highly regarded format for capturing requirements. Before writing functional requirement use cases can help you to understand the requirements in the way user expect. Following table presents you not only the template to write use case(s) as well as guides you to write each section with example.

**User mode**:

|  |  |
| --- | --- |
| Use case | Description |
| Actor | User |
| Precondition | Input image should be available |
| Main scenario | User upload image |
| Extraction scenario | If the image is not compatible not possible to upload file. |
| Post condition | Image successfully uploaded |

**User module**

**Pre –processing module**

|  |  |
| --- | --- |
| Use case | Description |
| Actor | system |
| Precondition | Uploaded input image |
| Main scenario | Pre-processing is carried out by converting the image from RGB format to binary format |
| Post condition | Extract characters before segmentation |

**Pre –processing module**

**Segmentation module**

|  |  |
| --- | --- |
| Use case | Description |
| Actor | system |
| Precondition | Preprocessed image should be available |
| Main scenario | The preprocessed input image is segmented into isolated characters by assigning a number to each character using a labeling process. this labeling provides information about number of character in the image each individual character is uniformly resigned into pixels. |
| Extraction scenario | If the image is not compatible not possible to upload file |
| Post condition | Image successfully uploaded |

**Segmentation module**

# **3.1. Functional Requirements**

The Functional requirements for the system are divided into two main phases of Training and Classification (Machine Creation). For Further details, refer to the use cases.

1. This system should process the output given by the user only if it is an image file (jpg. Png. etc.)
2. System shall show the error message to the user when the input given is not on the required format.
3. System should detect characters present on the image.
4. System should retrieve character present in the image and display them to the user.

## **3.2. Non Functional Requirements**

# **3.2.1. Usability**

Our System is very simple and easy to use. Estimate time required to learn estimate time required to learn train data and then SVM can read the image 4ro 5 minutes.

# ***3.2.1.* Reliability**

**This software will work reliability for low resolution images and not for graphical images.** Our system will be 70% to 74% approx. reliable to reach the target point.

# ***3.2.2.* Performance**

Our system depends on system processor if the system speed is high the performance will be high.

# ***3.2.3.* Supportability**

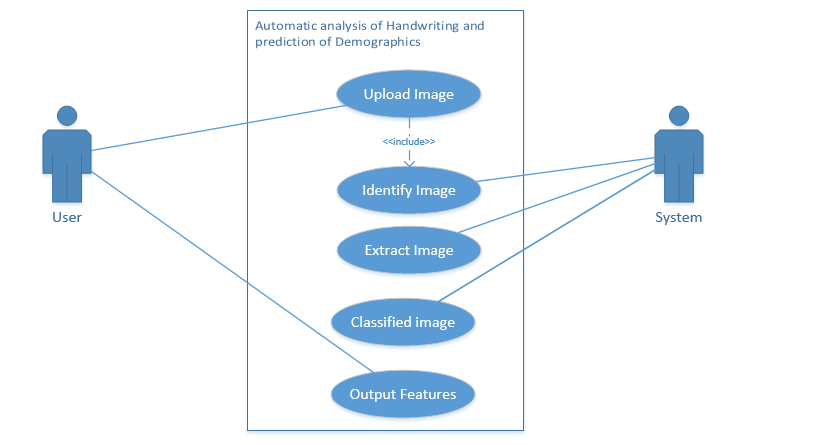
Our system can be support the SVM support vector machine to maintain the data in machine creation.

# **3.2.4. Design Constraints**

The developed system relies on binarizing the handwriting images and extracting a set of textural features namely histogram of oriented gradients and Haralick features. These features are used to train an SVM to classify a given writing on the basis of gender and handedness.

## **3.3. Use Case Model**

# **3.3.1. Use Case Diagram**

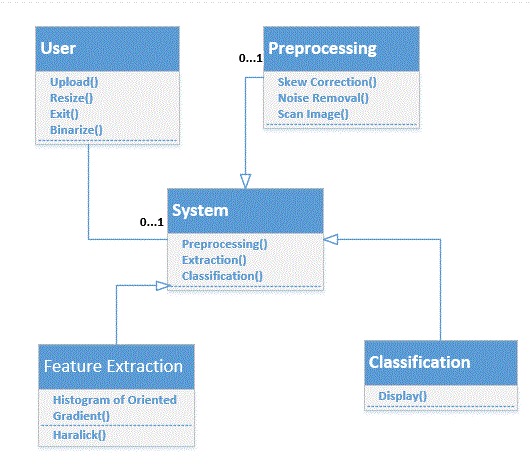


# 

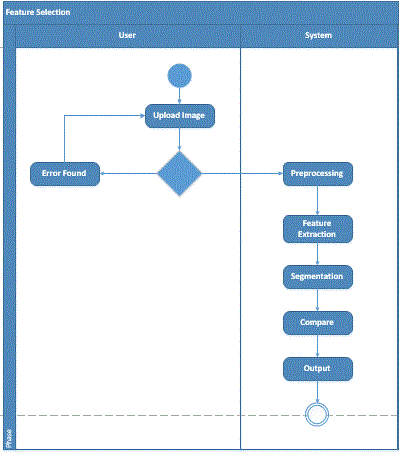
# **Chapter 4: Design and Architecture**

# **UML Behavioral Diagrams**

* + 1. Class diagram

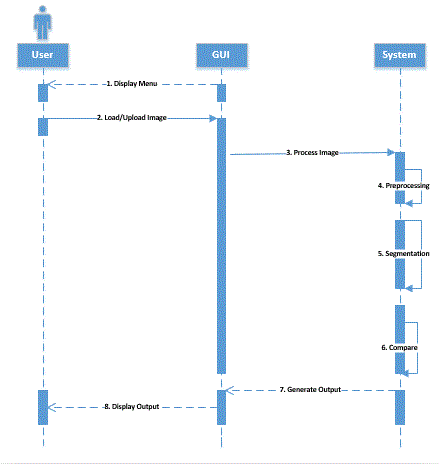


# **4.2.2 Activity Diagram**



## **4.3. UML Interaction Diagram:**

# **4.3.1 Sequence Diagram**



# Chapter 5: Implementation

In this chapter, we explain all the implementation of our project. Why we are using the visual studio 2019.

# Language C#

C# language is use to make a project for handwriting identification.

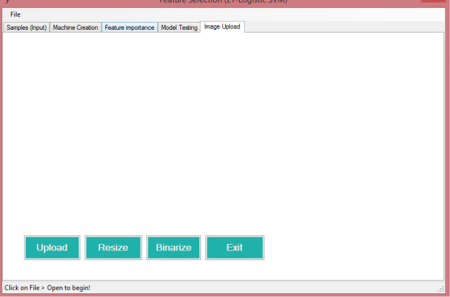
# Libraries

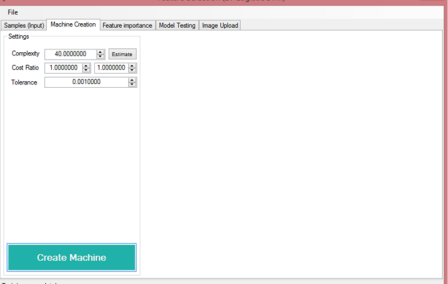
1. using System;
2. using System. Data;
3. using System. Drawing;
4. using System.IO;
5. using System.Windows.Forms;
6. using Accord;
7. using Accord.Controls;
8. using Accord.MachineLearning.VectorMachines;
9. using Accord.MachineLearning.VectorMachines.Learning;
10. using Accord.Math;
11. using Accord.Statistics.Analysis;
12. using AForge.Imaging.Filters;
13. using Accord.Imaging;
14. using System.Collections.Generic;

# Tool

**Visual studio 2019**

* Visual studio 2019 is smart code editor for C#
* Readability of the code is good due to the different color of the keyword, classes and function etc.
* Free available on internet
* Compatible with all operating system
* Helps to write code in C# quickly and efficiently





**Chapter 6**

**Testing and evaluation**

**6.1 Verification:**

The set of the test cases are used to test the functionality of each module of that module work properly then that test cases marked as pass or else fail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test id | Test case | Input description | Expected output | Status test |
| 1 | Uploading image | When user click on the button open filed box will be opened to select the image file | Image file should be selected and uploaded | pass |
| 2 | To preprocess images | Image will be taken for preprocessing | RGB to B/w image  (binarization) | pass |
| 3 | Feature extraction | A fray scale image | Character features should be executable | pass |
| 4 | Output file | Normalized character to the neural network | File containing only the test | Pass |

**6.2 Validation:**

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

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | Functions | Required output | Actual output |
| 1 | Uploaded the images with valid format | Image should be uploaded if supported | Valid images is uploaded successfully |
| 2 | Invalid image format | Error message should be displayed | Error message is displayed if the image format is not supported |
| 3 | Preprocessing of the upload image | Image should be preprocessed in ordered to convert to gray scale | Image is preprocessed |
| 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 |

**6.3 Unit testing**

**Test Cases**

**Test Case 1**

**Upload Image**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | | TC\_FUNCT\_01 | |
| **Description** | | Tests the uploading of images | |
| **Applicable for** | | Windows | |
| **Requirements** | | REQ\_FUNCT\_01 | |
| **Initial conditions** | | The images should be stored in the database | |
| **step** | **Full/reg** | **Task and expected result** |  |
| 1 |  | Switch on pc |  |
| 2 |  | Open the desktop application of developed system | pass |
| 3 |  | Click on the upload button |  |
| 4 | R | Given the directory of first image | pass |
| 5 | R | System automatically rest of the images | pass |

**Test Case 1: Upload Image**

**Test Case 2**

**Binarize Image**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | | TC\_FUNCT\_02 | |
| **Description** | | Binarizing the uploaded images | |
| **Applicable for** | | Windows | |
| **Requirements** | | REQ\_FUNCT\_02 | |
| **Initial conditions** | | The images should be uploaded in the machine | |
| **step** | **Full/reg** | **Task and expected result** |  |
| 1 |  | Click on the UPLOAD button |  |
| 2 |  | let the training dataset get uploaded | pass |
| 3 |  | click on the BINARIZE button |  |
| 4 | R | the images are binarized | pass |

**Test Case 2: Binarize Image**

**Test Case 3**

**Feature Extraction**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | | TC\_FUNCT\_03 | |
| **Description** | | Extract features | |
| **Applicable for** | | Windows | |
| **Requirements** | | REQ\_FUNCT\_03 | |
| **Initial conditions** | | The images should be uploaded in the machine | |
| **step** | **Full/reg** | **Task and expected result** |  |
| 1 |  | Click on the UPLOAD button |  |
| 2 |  | Let the training dataset get uploaded | pass |
| 3 |  | 3 Click on the BINARIZE button |  |
| 4 | R | The images are binarized | pass |
| 5 | R | The binarized images then extract the HOG and HARALICK features | pass |

**Test Case 3: Feature Extraction**

**Test Case 4**

**Machine Creation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | | TC\_FUNCT\_04 | |
| **Description** | | Feeding the extracted features into the machine for training purpose | |
| **Applicable for** | | Windows | |
| **Requirements** | | REQ\_FUNCT\_04 | |
| **Initial conditions** | | The extracted features must be uploaded on the database | |
| **step** | **Full/reg** | **Task and expected result** |  |
| 1 |  | Open the FILE button |  |
| 2 |  | It will upload the extracted features of training images | pass |
| 3 |  | Click on the machine creation button |  |
| 4 | R | Trains the machine on the basis of extracted features | pass |

**Test Case 4: Machine Creation**

**Test Case 5**

**Model Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | | TC\_FUNCT\_05 | |
| **Description** | | Feeding the extracted features into the machine for training purpose | |
| **Applicable for** | | Windows | |
| **Requirements** | | REQ\_FUNCT\_04 | |
| **Initial conditions** | | The machine must be trained on the training dataset | |
| **step** | **Full/Reg** | **Task and expected result** |  |
| 1 |  | Open the FILE button |  |
| 2 |  | It will upload the extracted features of test images | pass |
| 3 |  | Click on the RUN button in model testing tab |  |
| 4 | R | Shows the results in data grid views | pass |

**6.4 Integration Testing**

Integration testing focuses on finding defects which mainly arises due to combining various components for testing. The system as a whole is tested here. The system is said to be operating correctly if it passes these tests. After the different modules have been individually tested, we have to integrate them and tackle the issues during the integration

**6.5 System Testing**

System testing enhances usability and reliability of the application. It examines whether each module of the system is working properly or not as a group of interrelated modules. The technique of testing behavior of complete and fully-integrated software product whether it sits on Software Requirement Specification (SRS) document or not meeting requirements due to errors or bugs. Key objective of system testing is to test behavior of the application i.e. how would the application respond when user wants to perform an action. For example, add, update, view, delete, upload, download and search. System testing is usually performed by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing. System testing recognizes each action performed against specified. The main aim of system testing is to appraise functional, nonfunctional and End-user needs. As system testing is performed by end user and users are not allowed to access source code. So, this is black-box testing where the system is assessed externally without having the knowledge of system's architecture and logic.

**Chapter 7: Conclusion and Future Work**

In this chapter we are concluding our FYP work by analyzing what we were aim to do what we have done till now and what are improvements that still needs in this project or what kind of future work can anyone perform.

**7.1 Conclusion**

A system for automatic analysis of handwriting to predict gender and handedness of writers is developed in this study. Although a number of studies have been carried out on this problem by researchers in psychological studies, computerized analysis of handwriting for this task is a relatively recent area of research. The main hypothesis of this study was to investigate the features that could be exploited for prediction of writer demographics from handwriting samples. The developed system relies on binarizing the handwriting images and extracting a set of textural features namely histogram of oriented gradients and Harlick features. These features are used to train an SVM to classify a given writing on the basis of gender and handedness. The system was evaluated on the QUWI database and the realized results validated the ideas proposed in this study. The developed system can be effectively employed in forensic document analysis systems where the search space can be reduced by focusing on a set of writers with certain demographic attributes.

**7.2 Future work:**

The present study focused on writer demographic classification from offline handwriting images. There are certain enhancements and extensions that can be made to the present system. For instance, in addition to offline images, it would also be interesting to work on online handwriting where additional features in terms of pressure, speed and pen-up/pen-down times etc. are available. Moreover, in addition to gender and handedness, the prediction of other demographic attributes like age, nationality and educational background etc. can also be explored. Likewise, in addition to histogram of oriented gradients and Harlick features, other descriptors can also be employed. Features like Local Binary Patterns (LBP), statistics computed from Gabor filters, fractal features and local features (slant, slope, spacing etc.) can also be studied. The system can also be evaluated on larger databases to match the real world problem scenarios.