**Chapter 3**

**METHODOLOGY**

**3.1 Introduction**

The main objective of this study is to create a system that translates an image of a Gregg shorthand stroke into its corresponding English word. This shall help in translating documents in Gregg shorthand easier for stenography newbies or even to non-shorthand writers. To accomplish this, the researchers shall use Optical Character Recognition (OCR) with k-Nearest Neighbors (k-NN) algorithm.

An overview of the research process shall be discussed in this chapter. Information regarding the \*insert parts here\* shall be provided. This chapter shall serve its purpose to explain the design and implementation process as well as the methods to be used to create the proposed system.

* 1. **Experimental Design**

Experimental method shall be applied for this study. Concepts that shall be utilized include computer vision, image processing, and machine learning. The researchers shall take into account factors that are vital to the integrity of the proposed system. Factors include diversity and amount of the training data, accuracy of the Gregg shorthand translation, and user satisfaction. In times when a Gregg shorthand document needs to be translated but a stenographer is not present, the proposed system shall alleviate such problem. For learners of the writing system, it can be used as a tool to aid learning.

**3.2.1 Research Procedure**

Data gathering

Develop the system

Training the system

Testing the system

Collect user feedback

Deploy to participating users

**Figure N Research Procedure Block Diagram**

To guide the researchers in conducting the study, Figure N shall be utilized. The study starts with the data gathering which includes images of training and testing Gregg shorthand strokes. An open source dataset which is comprised of 15280 cropped images from a Gregg shorthand dictionary shall be used for training the KNN classifier. As for the testing data, handwritten Gregg shorthand strokes shall be gathered from different 2nd year BS Office Administration students of Bohol Island State University – Main Campus.

Once the dataset has been prepared, the system development phase shall begin. The researchers shall develop the OCR backend of the system which utilizes Python and the OpenCV library. These two will also be used to create the KNN classifier. The design shall be discussed in more detail later on in this paper.

Following are the training and testing of the system. This process shall be repeated until the system’s accuracy reaches the maximum possible. The user interface shall then be developed and tested. Once ready, the system shall be distributed to participating BS Office Administration students of Bohol Island State University – Main Campus to collect user feedback.

**3.2.1 Software Development Life Cycle**

Requirements Gathering

System Design and

Software Design

Implementation

System Testing

Operation and

Maintenance

**Figure N Iterative Waterfall Model**

For the life cycle of the software development, the iterative waterfall model shall be used as seen in Figure N. Similarly to the classical waterfall model, this model allows the researchers to approach the development of the system in an organized manner which also has the flexibility of going back to certain stages after gathering feedback which results to parts of the system that needs to be redone.

Requirements gathering includes gathering of training and testing data. The training data is composed of 15280 English words from an open-source Gregg Shorthand dictionary. Meanwhile, the test data shall be collected from BS Office Administration student volunteers whom shall write in Gregg Shorthand on a clean bond paper. Designing the software is the next step wherein the OCR algorithm along with a KNN model shall be used to pre-process the data images which shall be used for training. Once the design is polished, the implementation shall commence. Following is the testing for the accuracy and operation and maintenance.

**3.2.2 System Design**

[insert here]

Image acquisition via user input

Image preprocessing

Features extraction

Stroke recognition with KNN classifier

Output result to user interface

**Figure N System Flow Diagram**

As seen on Figure N, the system first acquires an image from the user which will then undergoes preprocessing which includes converting to grayscale, blurring, thresholding, and converting to a binary image. After the features are extracted, these will be fed to the KNN classifier for the stroke to be recognized and translated to its corresponding English word. The result shall then be displayed to the user interface.

**Image Preprocessing**

[insert here]

**K-Nearest Neighbours Classifier**

[insert here]

**3.3 Statistical Treatment**

A confusion matrix shall be utilized to calculate the accuracy of the system. The said matrix is a two-by-two table which contains false positives, false negatives, true positives, and true negatives for a test. In this study, the following definitions are made of the classes:

* “Gregg shorthand stroke” is the positive class
* “Non-Gregg shorthand stroke” is the negative class

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Predicted** | |
| **Negative** | **Positive** |
| **Actual** | **Negative** | True Negative (TN) | False Positive (FP) |
| **Positive** | False Negative  (FN) | True Positive (TP) |

**Table N Confusion Matrix for system accuracy evaluation**

As seen on Table N, an actual non-Gregg shorthand stroke predicted as is, yields a true negative result. False negative is the result for an actual Gregg shorthand stroke predicted as the contrary. False positive is the result of an actual non-Gregg shorthand stroke predicted as the opposite. A Gregg shorthand stroke actual value predicted as is, results to a true positive.

The accuracy of the system is defined as:

Where:

= accuracy of the system’s translation in percentage

= True Positive

= True Negative

= False Positive

= False Negative