Paul Gibbison

Candidate Number: 3226

Centre Number: 25278

NEA: British Sign Language Alphabet Detection System

Contents

[1. Analysis of the Problem 4](#_Toc178322885)

[1.1 Researching Existing Solutions 4](#_Toc178322886)

[1.1.a *British Sign* 4](#_Toc178322887)

[1.1.b *Lingvano* 5](#_Toc178322888)

[1.1.c *Online Webcam Test* 6](#_Toc178322889)

[1.2 Stakeholders 6](#_Toc178322890)

[1.2.a *Deaf People* 6](#_Toc178322891)

[1.2.b *Friends and family of deaf people* 6](#_Toc178322892)

[1.2.c *People whose job requires the use of BSL.* 7](#_Toc178322893)

[1.3 Proposed Solution 7](#_Toc178322894)

[1.3.a *Functionality* 7](#_Toc178322895)

[1.3.b *User Experience* 7](#_Toc178322896)

[2. Design of the Solution 8](#_Toc178322897)

[2.1 Decomposition of the Problem 8](#_Toc178322898)

[2.2 Algorithm Justification 8](#_Toc178322899)

[2.2.1 Searching Algorithm 8](#_Toc178322900)

[2.2.2 AI Algorithms 9](#_Toc178322901)

[2.3 Entity Relationship Diagram (ERD) 9](#_Toc178322902)

[2.4 Design of web pages 9](#_Toc178322903)

[(types of data: Boundary, Erroneous, Normal) 9](#_Toc178322904)

[2.5.1 Boundary Data Testing 10](#_Toc178322905)

[2.5.2 Erroneous Data Testing 10](#_Toc178322906)

[2.5.3 Normal Data Testing 10](#_Toc178322907)

[3 Developing the Solution 11](#_Toc178322908)

[3.2 Development Process 11](#_Toc178322909)

[3.2.1 Learning the Necessary Modules 11](#_Toc178322910)

Table of Figures

[Figure 1 - Sign of the Day 4](#_Toc177461039)

[Figure 2 - Flexibility 4](#_Toc177461040)

[Figure 3 - Gift Enrolment 5](#_Toc177461041)

[Figure 4 - Daily Streaks 5](#_Toc177461042)

[Figure 5 - Daily Learning Goals 5](#_Toc177461043)

[Figure 6 - Slow Mode 6](#_Toc177461044)

[Figure 7 - Troubleshooting page 6](#_Toc177461045)

[Figure 8 - Decomposition of the project 8](#_Toc177461046)

[Figure 9 - Entity Relationship Diagram 9](#_Toc177461047)

[Figure 10 - Learning OpenCV 10](#_Toc177461048)

[Figure 11 - Learning Flask with Python 11](#_Toc177461049)

[Figure 12 - Learning Flask with Python and HTML (Python Code) 11](#_Toc177461050)

[Figure 13 - Learning Flask with Python and HTML (HTML Code) 12](#_Toc177461051)

# 1. Analysis of the Problem

## Researching Existing Solutions

### 1.1.a *British Sign*

A person with his hands crossed

Description automatically generated

Figure 1 - Sign of the Day

British Sign has a ‘Sign of the Day’ feature on the front page of their website. I like this feature because it could be useful for people who do not have much time on their hands as it enables them to learn one small sign when they have time each day. Although the issue with this is that because it is a different sign every day, it could become too easy for people to forget the signs they have previously learned.

A close up of a sign

Description automatically generated

Figure 2 - Flexibility

The website also advertises the ability to learn on many different devices. Although this would be a desirable addition to my project. It would not be feasible to be completed completely within the allocated time frame.

A blue and white card with text

Description automatically generated

Figure 3 - Gift Enrolment

Due to the website’s courses being paid, the addition of a ’Gift Enrolment’ were to also be a desirable addition although one of the main elements of my project is that it will not cost the user to enrol to utilize all available features. Therefore, this feature would not be a possible addition within the limitations set by the parameters.

### 1.1.b *Lingvano*

A black text on a white background

Description automatically generated

Figure 4 - Daily Streaks

Lingvano introduces the concept of consistently working towards the user’s learning target. The concept of ‘daily streaks’ would be a desirable feature of my project as the encouragement of a consistent effort would allow for a better understanding of British Sign Language

A screenshot of a cell phone

Description automatically generated

Figure 5 - Daily Learning Goals

The website also introduces different ‘learning goals’, this inclusive feature takes into consideration people’s limited free time. This when linked with the aforementioned ‘daily streaks’ would be a desirable inclusion to my project as this would enable people to go at their own pace with their learning.



Figure 6 - Slow Mode

As the website features short videos to describe the different signs, a ‘slow’ feature is included to slow down these videos into a more manageable pace. This would be a desirable feature for my project although the alphabet of the British Sign Language does not include much movement (aside from the letters J and H) and therefore does not require a slow feature.

### A screenshot of a computer test Description automatically generated1.1.c *Online Webcam Test*

Figure 7 - Troubleshooting page

Online mic test includes webcam testing software. I like the inclusion of a list as it allows people to troubleshoot any issues, they might have with setting up their camera.

## 1.2 Stakeholders

My stakeholders would include the following:

* Deaf People of whom do not know BSL (Primary Stakeholder)
* Friends and family of deaf people
* People whose job requires the use of BSL.

### 1.2.a *Deaf People*

There are many people that do not know how to sign using BSL, some of whom are deaf themselves. Said people would become my primary stakeholders as it would be highly likely for them to utilise the knowledge gained from my program in their day-to-day lives.

### 1.2.b *Friends and family of deaf people*

Due to the main mode of contact from people with a lack of hearing being through sign language. Another one of my stakeholders would be the friends and family of which would have to be able to use BSL (as gained from my project) fluently to maintain a conversation with the deaf people.

### 1.2.c *People whose job requires the use of BSL.*

There exist many people up and down the country of whom work within a job that may require the use of BSL. These people would be able to use my website to begin their journey of learning BSL to aid in their job.

## 1.3 Proposed Solution

### 1.3.a *Functionality*

For the success of my project, it would have to be able to be able to take multiple screenshots of the user’s camera, these screenshots would then have to be analysed against a pre-made database to identify if the user has done the correct symbol. My database will also have to be organised so that an outside developer would be able to intuitively analyse it. The back-end code will also have to be clear, precise, and efficient and utilise comments effectively.

### 1.3.b *User Experience*

The user would have to be able to navigate the website easily and be able to get feedback about each of their attempts at getting the correct letter. There would also need to be a temporary section designated to aiding someone in getting their webcam setup.

# 2. Design of the Solution

## 2.1 Decomposition of the Problem

Figure 8 - Decomposition of the project

## 2.2 Algorithm Justification

### 2.2.1 Searching Algorithm

One of the functions of the AI is having to search through many pictures of different letters to find one which is the most like what the user is doing. A binary search will be the most efficient algorithm to accomplish this. This is due to the ‘divide-and-conquer’ nature of the algorithm. This means that the algorithm will be more efficient with the numerous pictures it will have to go through. If this program were to be developed further, there will be more pictures of varying variables, this would mean that there would be more pictures that the algorithm would have to search through and compare, making the algorithm even more efficient.

### 2.2.2 AI Methodologies

#### 2.2.2.a Supervised Learning

<https://www.ibm.com/topics/supervised-learning>

Supervised learning involves using labelled data sets so that the AI can classify data. The accuracy for this algorithm is measured through the ‘loss function’. The ‘loss function’ is a graph that maps the accuracy of the AI, the exact implementation of the function can vary on a case-by-case basis. Generally, as more variables are added, the quality off the AI and subsequently the function increases. While this method is typically used for predicting outcomes such as earthquake simulations, this method would be better suited in my project towards classifying the data into categories of different letters based on what the AI’s guess is. Supervised learning excels at following patterns, would be highly beneficial within the context of the detection system.

#### 2.2.2.b Unsupervised Learning

<https://www.ibm.com/topics/unsupervised-learning>

Unlabelled data sets

Train to classify data.

Can discover hidden patters or data groupings without human interaction.

Typically used for anomaly detection

#### 2.2.2.c Reinforcement Learning

<https://www.ibm.com/topics/reinforcement-learning>

AI learns by interacting with an environment.

Prioritises autonomous decision making and sequential decision making.

Typically used in control problems (where getting an incorrect answer could be critical)

## 2.3 Entity Relationship Diagram (ERD)

A screenshot of a computer

Description automatically generated

Figure 9 - Entity Relationship Diagram

## 2.4 Design of web pages

A screenshot of a computer

Description automatically generated

2.5 Testing

(what types of testing I am doing: e.g. boundary, erroneous, normal)

### 2.5.1 Boundary Data Testing

#### 2.5.1.a What is being tested?

#### 2.5.1.b Why is it being tested?

#### 2.5.1.c What is the test data?

#### 2.5.1.d What outcome is being tested?

### 2.5.2 Erroneous Data Testing

#### 2.5.2.a What is being tested?

#### 2.5.2.b Why is it being tested?

#### 2.5.2.c What is the test data?

#### 2.5.2.d What outcome is being tested?

### 2.5.3 Normal Data Testing

#### 2.5.3.a What is being tested?

#### 2.5.3.b Why is it being tested?

#### 2.5.3.c What is the test data?

#### 2.5.3.d What outcome is being tested?

## 3 Developing the Solution

## 3.2 Development Process

### 3.2.1 Learning the Necessary Modules

#### 3.2.1.a Utilising the user’s webcam

Utilising the external module OpenCV (cv2) enables me to be able to capture the user’s webcam so that the AI can tell if what the user is signing is correct. OpenCV enables the use of facial detection. This is not a necessary feature of my project as the British Sign Language alphabet does not require the use of facial expressions and only requires the use of either one or two hands. A screenshot of a computer program

Description automatically generated

Figure 10 - Learning OpenCV

This code enables me to capture the user’s webcam, (as set by their computer’s ‘default webcam’ setting) in a separate ‘pop-up’ window that simply displays the user’s camera. Lines 4 and 5 enable me to set the scale of the camera. This feature will become useful as when utilising this within my website, I don’t want the camera to take up all the space; as other information will also be necessary. Line 12 gives me the ability to close the webcam upon a specific keystroke. This is useful for when the webcam is in a window, however this will not be as mandatory for my final product. This could be a useful ‘quality of life’ feature as some people will want to practice their ability to use sign language without being able to see themselves to simulate how it will be within a more accurate scenario.

#### 3.2.1.b Creating a webpage with Python

Flask is an external module for python that enables back-end development. This is a necessary module for my final product as without it I am unable to display the user’s webcam, and the AI will not be connected to the website. Flask also has the benefit of being very modular and customisable, this is necessary for making a website that has an appealing and intuitive design. Flask’s high customisability means that I will require the use of many other modules as Flask does not contain many advanced functions within it. While Flask is typically used within a server to host a website, my project will not end up having a public page, so therefore a server is not necessary. This could however be used to advance my program in the future.

A computer screen shot of a program code

Description automatically generated

Figure 11 - Learning Flask with Python

This code produces a basic website with multiple pages. The lines 6 - 8 define the home page of the website at *http://<website>/home.* The return statement on line 8 is basic HTML Code to display the text ‘Welcome to AI BSL Learning’ in the ‘header 1’ size. Both lines 10 – 12 and 14 – 16 follow a similar structure with lines 14 – 16 testing the ‘redirect’ function within flask. These lines redirect the user back to the home page if they are not an admin; though it is not possible for the user to be an admin at this stage, so the main aspect of these lines is to demonstrate a redirect on condition. Lines 10 -1,2 on the other hand, are used as a preventative measure to tell the user that they have gone to a page that has either not yet been programmed or has been typed incorrectly.

#### 3.2.1.c Creating a webpage with Python and HTML

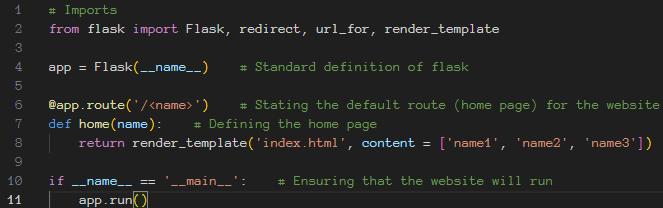


Figure 12 - Learning Flask with Python and HTML (Python Code)

Unlike the code from Figure 10, this code utilises the template function of Flask to allow for a more seamless use of HTML. Using the template function also allows for the code to be more compact and easier to read. This is mainly due to the separate HTML files (figure 13) that make up the templates (designs) of the pages. Line 8 shows how the list ‘content’ can be assigned in python and the converted into being utilised within HTML (line 8 in figure 13).

A screen shot of a computer screen

Description automatically generated

Figure 13 - Learning Flask with Python and HTML (HTML Code)

Figure 13 is the demonstration of a basic HTML template file for a home page that shows every item in the list ‘content’. This is done through a for loop that changes in length based on the length of the list. The use of the {% %} allows for code like pseudocode, and therefore allows for loops such as the for loop displayed above.