

Table of Contents

[Chapter 1: Introduction to the Study 4](#_Toc124109747)

[1.1 Background to the Project 4](#_Toc124109748)

[1.2 Problem Context 6](#_Toc124109749)

[1.3 Rationale 6](#_Toc124109750)

[1.4 Potential Benefits 7](#_Toc124109751)

[1.4.1 Tangible Benefits 7](#_Toc124109752)

[1.4.2 Intangible Benefits 7](#_Toc124109753)

[1.5 Target Users 7](#_Toc124109754)

[1.6 Scope and Objectives 7](#_Toc124109755)

[1.6.1 Aims 7](#_Toc124109756)

[1.6.2 Objectives 7](#_Toc124109757)

[1.6.3 Deliverables – Functionality of the proposed system 8](#_Toc124109758)

[1.6.4 Nature of Challenges 8](#_Toc124109759)

[1.7 Overview of this Investigation Report 9](#_Toc124109760)

[1.8 Project Plan 9](#_Toc124109761)

[Chapter 2: Literature Review 10](#_Toc124109762)

[2.1 Introduction 10](#_Toc124109763)

[2.2 Domain Research 10](#_Toc124109764)

[2.3 Similar Systems 12](#_Toc124109765)

[2.3.1 Copyfish 12](#_Toc124109766)

[2.3.2 SimpleOCR 13](#_Toc124109767)

[2.4 Summary 13](#_Toc124109768)

[Chapter 3: Technical Research 14](#_Toc124109769)

[3.1 Programming Language Chosen 14](#_Toc124109770)

[3.2 IDE Chosen 14](#_Toc124109771)

[3.3 Libraries/Tools Chosen 14](#_Toc124109772)

[3.4 Operating System Chosen 15](#_Toc124109773)

[3.5 Summary 15](#_Toc124109774)

[Chapter 4: System Development Methodology 16](#_Toc124109775)

[Chapter 5: Research Methods 17](#_Toc124109776)

[5.1 Introduction 17](#_Toc124109777)

[5.2 Design 17](#_Toc124109778)

[Chapter 6: Requirements Validation 21](#_Toc124109779)

[6.1 Analysis of Data 21](#_Toc124109780)

[6.1.1 First Section Data 21](#_Toc124109781)

[6.1.2 Second Section Data 25](#_Toc124109782)

[6.2 Summary 30](#_Toc124109783)

[Chapter 7: System Architecture 31](#_Toc124109784)

[7.1 System Architecture Introduction 31](#_Toc124109785)

[7.2 System Architecture Diagrams 31](#_Toc124109786)

[7.2.1 OCR Translation Use Case Diagram 31](#_Toc124109787)

[7.2.2 OCR Translation Flowchart 35](#_Toc124109788)

[Chapter 8: Project Plan 36](#_Toc124109789)

[8.1 Release Plan 36](#_Toc124109790)

[8.1.1 OCR Translation 1.0 36](#_Toc124109791)

[8.1.2 OCR Translation 2.0 36](#_Toc124109792)

[8.2 Project Test Plan 36](#_Toc124109793)

[Chapter 9: Implementation 36](#_Toc124109794)

[9.1 36](#_Toc124109795)

[9.2 36](#_Toc124109796)

[Chapter 10 System Validation 36](#_Toc124109797)

[Chapter 11 Conclusion 37](#_Toc124109798)

[References 38](#_Toc124109799)

[Appendix 40](#_Toc124109800)

[Log Sheets 40](#_Toc124109801)

[PPF 42](#_Toc124109802)

[PSF 44](#_Toc124109803)

[Ethics Form 46](#_Toc124109804)

# Chapter 1: Introduction to the Study

## Background to the Project

Communication is what allows human beings to prosper in this world as a civilization when compared to other species within the animal kingdom. It is true that other animals communicate with each other as well though they use different means to communicate that are unique to each species. Ants for example, communicate with each other using pheromones. These pheromones can be used as signs or indicators to guide other ants towards their nests for example (Arnold, 2013). The main difference that separates us humans with other animals however is we have the intelligence and organs necessary to produce a more complex form of communication. This form of communication is what we know as language. Language allows us to transmit complex and abstract ideas that may or not even real to other humans (Jose, 2015). These humans may then transmit their own thought of an idea as well, resulting in a discussion. These discussions have happened countless times throughout the history of mankind and has given birth to various ideas that are both productive and malicious.

Language has multiple forms and method of transmission as well. There is of course the regular way of using language as a method of verbal communication. Where words and sentences are spoken with the mouth and heard with the ear. As time passes however, humans have learned how to transmit messages and ideas visually, by writing symbols on objects such as stone tablets and later on paper. The idea of writing was revolutionary, as it allows humans to preserve ideas physically without the need of a speaker. The earliest known form of writing was discovered in an area near the Persian Gulf which is where ancient Mesopotamia was located (Brown, 2021). There, writing was used in government administration and artistic literature. We can see this technology still being used today though in different forms such as this paper that was written digitally. However, it is ultimate the same, a form of communication transmitted visually in the form of letters or symbols.

As wonderful as language is, just like other ideas and objects that humans have invented throughout history, it is not flawless. The problem with language is: it is only viable as a method of communication if both parties understand the language that is being used. If two parties who speak different languages were to attempt to communicate with one another, there would be difficulties as none of them would understand each other.

Before, the only solution to this issue was to learn the opposing party’s language or the other way around. Today in the information era however, tools have been made to make communication between members of societies with different languages possible. This process is known as translation. With the internet, it is possible to translate text or speech into tens of other languages, allowing people to access writing or any piece of media freely with minimal language barrier.

This project seeks to reduce the barrier between languages even further. By developing a computer software that runs as a lightweight overlay and reads texts from a computer display using OCR which will then be translated using machine translations. It aims to remove the need to manually extract texts from images and then paste them into a translator on a separate browser window. Instead, everything will be done in one program seamlessly.

## 1.2 Problem Context

Language barrier has always been a problem for mankind due to how diverse we are as a species, both biologically and culturally. In the past, to bypass this barrier, for two parties of different languages to communicate with one another, a person who speaks both languages need to serve as a translator. Otherwise, communication will be near impossible. This is problematic as learning a new language is very time consuming and takes dedication. Even today with the amount of knowledge we have available at our fingertips, around 40% of the global population are monolingual and another 43% of the global population are bilingual (Hyperpolyglots: How Many Languages Can You Learn?, 2022). This means nearly half of the world’s population are only able to communicate with people of their own, now imagine how even lower the numbers would be prior to globalization where communities tend to be isolated from one another other than the occasional merchants and travelers.

Nowadays however, things are not as bad, at least when communication is carried out through text electronically. As most translating platforms are able to receive texts as input from the user and have it translated into over a hundred languages. Among these platforms Google Translate is the platform with the biggest number of users with over 500 million users using the platform on a daily basis (Turovsky, 2016).

This technology is great especially considering the value it facilitates. However, using mostly text as an input on a desktop is limiting as that means it will be tricky for the users to use if they want to translate a phrase on an image for example as it means they will have to type it in manually and accurately. This is even harder if the phrase that they want translated is in a language where the user is not familiar with its script, making it impossible for them to type it in the first place.

## 1.3 Rationale

By using OCR and taking in the user’s screen as an image input, this project aims to solve the problem of having to type in the desired input manually. Not only will it be easier and accurate, we aim to cut off time as well as reading and typing in foreign sentences takes time and that is if the user is capable of doing so in the first place. Implementing machine translation aims to improve the accuracy of the translation as well as machine translation API’s such as DeepL tend to outperform Google Translation in terms of accuracy (Colin, 2022).

## 1.4 Potential Benefits

### 1.4.1 Tangible Benefits

* Communication between parties over media that does not involve direct text will be easier as the proposed project will allow translation via texts in images.
* Translation results will be more accurate in most cases due to the higher quality of datasets used when training machine translation models.

### 1.4.2 Intangible Benefits

* The barrier that stops people from enjoying or accessing pieces of media that are only available in languages foreign to them will be taken down.

## 1.5 Target Users

The system is not targeted towards any specialized groups in particular therefore it is aimed towards the general public. This is because the language barrier is a universal problem and since the system is trying to focus on that problem, everyone who does not speak every language in the world will benefit from the system.

## 1.6 Scope and Objectives

### 1.6.1 Aims

This project aims to provide a tool that can read text from any piece of media on a computer such as a digital scan of a book or even video games and seamlessly provide machine translations of such text into a variety of languages. The tool should be lightweight and work as an overlay that always works therefore users should still be able to use it when they are using a program that needs to be translated in full screen mode without the need to use it in windowed mode.

### 1.6.2 Objectives

The objectives of this project are:

* To develop an algorithm with the capabilities of reading text from different kinds of alphabet therefore not limited to the standard 26 letter English letters.
* The algorithm should be able to read from an input image such as but not limited to a photograph of a street sign, a scan of a book or any physical form of literature, or the currently displayed programs on the running computer.
* To provide translations seamlessly at an efficiency where it allows the user to read from a sentence to another sentence without having to wait for the program to translate them first.
* To develop a software that is capable of achieving all of the above features efficiently.

### 1.6.3 Deliverables – Functionality of the proposed system

The project aims to deliver a system that is able to use OCR technology to detect characters on a computer screen, and then utilize machine translation to translate the read text into a desired language. The initial target languages will be Bahasa Indonesia and English as those two are the languages I am familiar with, therefore will be the easiest to work with and can provide accurate results.

### 1.6.4 Nature of Challenges

Developing a system that implements OCR as means of reading text in an image and then hooking it unto a machine translation model is not an easy task. The OCR itself will be the main hurdle in this project as a model that can recognize characters and cleans out clutter in its vision is critical. Otherwise, the quality of translation will be terrible, no matter how accurate the machine translation model would be as the initial input is already false and inaccurate. This means, a model needs to be trained that can achieve three main things:

* Detect characters in an image.
* Identify what characters they are as different fonts have different appearances.
* Clean out noise and be able to separate between characters and noise.

After these three main points can be achieved, only then can the machine translation model attempt to do its job as it will finally have a clean and accurate input for it to process. The machine translation aspect will not be as hard as for this project, we will be using the DeepL model which is a neural machine translation service with an API that can be used for projects such as these.

## 1.7 Overview of this Investigation Report

The investigation report covers on the problem of basics of the language barrier and even though it is not as big of a problem as it used to be in the past, it is still a prevalent problem in today’s digital era. The report expands on a proposed system that is aimed towards tackling this problem. The proposed system is a program with the capabilities to capture a computer display as an image and then detect and recognize texts on the image. This text will later be used as an input to a neural machine translation engine to translate the text into a different language. Elaborates on the technical aspect of the system as in what libraries it would use and why and later on justifies its decisions based on analysed responses of a survey sent out to the public.

## 1.8 Project Plan

The investigation starts with expanding on the problems of texts on image and how much of a hassle they are to translate. Previous OCR technologies have been researched and papers on them have been reviewed and a conclusion that they have improved significantly can be made. After that, a research on the available technologies to make this possible will be made. As figuring out what technologies to use and how they can work with one another is half the work. Once the technicalities have been decided, a software development methodology needs to be decided for implementation during the development process as having a suitable development methodology is very critical to having a balanced workload and achieving a satisfying end product. A survey will then be made to understand the general public’s opinions, views, and preferences when it comes to translation systems. After all, the system is aimed towards the general public’s use and therefore their opinions and voices need to be listened to and kept in mind as reference when developing the system. Finally, everything can be summarized and a conclusion can be made.

# Chapter 2: Literature Review

## 2.1 Introduction

To delve deeper and achieve a firmer grasp on the topic of this investigation, more research needs to be conducted. Various topics such as but not limited to: OCR and machine translations are investigated and summarized to achieve supporting data and evidence.

## 2.2 Domain Research

According to Eikvil (1993), the first true OCR machine was in the form of an equipment that converts sales reports into punch cards which can then be used by computers which was used in the year 1954. A decade later, OCR machines advanced as they can now recognize multiple fonts with the number of recognizable fonts scale with the machine’s capabilities in methodologies such as but not limited to pattern matching. To tackle this problem, 1966 an OCR standard font was made as a form of standardization to make it easier for OCR machines to be made and used. The American standard font is OCR-A while the European standard font is called OCR-B. Both fonts are similar in nature with OCR-A looking more robotic and OCR-B looking more natural. The technology keeps progressing with today OCR technology is available at our fingertips with open-source libraries and software’s. (Eikvil, 1993)

In 2013, Ravina Mithe, Supriya Indalkar, Nilam Divekar published a paper on an OCR system that runs on an Android phone using the open-source OCR engine by Google, Tesseract. The system they made works by having the system receive an input in the form of a picture taken using the camera of a phone. The system first scans the image with methods such as thresholding being used to assist the process. After this, pre-processing is done to reduce noise in the image and Tesseract will then be used to recognize characters in the image. These characters will then be combined to form sentences and the system can then output an audio using a TTS synthesizer to convert the scanned text into synthesized speech. (Mithe, R et al., 2013)

The authors concluded that the system practical and relatively cheap as it uses an Android phone which is free and open source. The input device is also the same phone’s camera which is more affordable than the expensive scanners and cameras used in early OCR. The system however relies heavily on the quality of the input image as they tend to be noisy and hinder the process of scanning. (Mithe, R et al., 2013)

According to Stahlberg (2020), in the field of machine translation, the paradigm that used to dominate the field which is statistical machine translation, has been outclassed by neural machine translation. Initially, neural networks were used to as components to *boost* the capabilities of statistical machine translations. However, by using as single large neural network, neural machine translation surpassed statistical machine translation. This is evident as neural machine translation as a paradigm has been deployed by big players in the tech industry such as Facebook and Google. (Stahlberg, 2020)

## 2.3 Similar Systems

### 2.3.1 Copyfish



Figure 2.3.1.1 Copyfish UI

Copyfish is a browser extension that can capture a page on its installed browser and utilizes OCR technology to read the text on it. It works well and is lightweight. However, Copyfish is limited by its form as it is a browser extension. Therefore, it relies on a browser instead of running natively on the user’s device. Another weakness that Copyfish has is it takes in manually taken static images. Therefore, if a user would like to translate a currently playing video on their computer, Copyfish needs to take in images manually every time something new that needs to be translated appears.

### 2.3.2 SimpleOCR

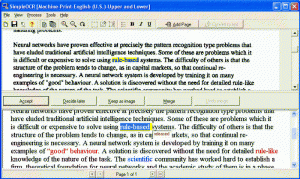


Figure 2.3.2.1 SimpleOCR UI

SimpleOCR is a freeware that uses OCR technology to read text from an input. SimpleOCR is typically and was intentionally made for usage in document scanning. Users can scan a document and have SimpleOCR read the document for them so they do not have to rewrite the whole document manually. Unlike Copyfish, it runs natively on the user’s device therefore it does not rely on a browser extension. However, it is relatively outdated as its features are pretty barebones and minimalistic.

## 2.4 Summary

In conclusion, OCR has come a long way over the past couple of decades. Neural machine translation is the go-to solution for machine translation and it is possible to implement such engines like the Tesseract OCR for example into portable software for usage. Systems should run natively to prevent reliance on other systems, and successful implementation of similar systems have been done successfully for decades.

# Chapter 3: Technical Research

## 3.1 Programming Language Chosen

When it comes to programming language, we have to choose based on the project that we are working on. In this case, it will be a mix between C++ and Python. This is because the libraries that are used such as Tesseract are written in C++ while the neural machine translation service DeepL can be imported as a Python library.

## 3.2 IDE Chosen

The chosen IDE for this project will be the Visual Studio Code as it is my preferred IDE and I have been using it for the past couple of years. The IDE itself is lightweight and is unique compared to other IDE’s as it is customizable. Visual Studio Code features the plugin system which allows users to create plugins for the IDE. These plugins can be something simple like changing the colours of texts, adding icons for specific file types, to powerful features such as text formatting and git implementation. Overall, due to its versatility and modularity Visual Studio Code will be a great choice for this project.

## 3.3 Libraries/Tools Chosen

The project has two main parts: The OCR and the neural machine translation. Each part will function differently and use different libraries.

The OCR part will be mainly powered by the open-source OCR engine, Tesseract. Tesseract was originally developed by Hewlett-Packard. However, it was later rereleased as open-source by Google in 2006 and they have been sponsoring its development since then (Vincent, 2006). When compared to other OCR engines such as EasyOCR, Tesseract outperforms it in terms of alphabetical recognition and CPU performance. EasyOCR however, outperforms it when it comes to numerical recognition and GPU performance (Liao, 2020). This means, Tesseract will be preferable for this project as the project focuses more on letters and sentences rather than numbers due to the translation nature of the project.

When it comes to the neural machine translation, the DeepL neural machine translation model will be used. This is because despite being less popular, DeepL is more accurate than other platforms such as Google Translate (Colin, 2022). According to DeepL themselves, they outperform their competitors by a landslide with a factor of 3:1 (Why DeepL?, n.d.). DeepL facilitates developers by proving API’s so developers can utilize their neural machine translation as well. These factors combined make DeepL a great choice for the project.

## 3.4 Operating System Chosen

When it comes to Operating System for both development and usage, there are two choices: Windows and Linux. In terms of development, ideally it should be done on a Linux virtual machine. As virtual machines or virtual environments in general allows developers to isolate their work environment from one another. Linux is better than Windows in this specific scenario as it is free, and lighter. Installing a Linux virtual machine takes less time and hassle than installing a Windows virtual machine. However, due to hardware limitations I will not be able to run a Linux virtual machine. Therefore, development will be done natively on a machine with a Windows operating system.

For usage, Windows will be better as there are simply more machines that use Windows than any other operating systems. According to StatCounter, over 75% of desktop machines out there are running Windows as their operating systems (Desktop Operating System Market Share Worldwide | StatCounter Global Stats, 2022). Therefore, the Windows operating system should be the main target with compatibility for other operating systems such as Linux can be set as an optional target.

## 3.5 Summary

To summarize it, the system will be built using the programming languages C++ and Python. Utilizing the Tesseract engine as for its OCR capabilities, and DeepL for its accurate machine translations. The system will be developed on Windows and be targeted towards Windows users with Linux compatibility still being considered.

# Chapter 4: System Development Methodology

Implementing the correct system development methodology to the development cycle of a system is very important. This is because by following a methodology, workflow can be organized and steady. However, there are different kinds of software development methodology out there and its crucial to pick the right one for the job.

First up is the agile methodology. Agile is a methodology where the development cycle is divided into phases and sprints. Where during each sprint a component is developed and implemented, and at the end of each sprint a meeting with stakeholders is held. These meetings are crucial as they are used as a way to gather feedback for the next sprint. The cycle repeats until the stakeholders are satisfactory with the system and all the requirements are met.

Next is the Kanban methodology. Kanban is a methodology where requirements are set on what is known as a Kanban board. Where they will all be placed in a single column on the left, and slowly moved to the right as each requirement progresses. What makes Kanban unique is typically, there can only be a limited number of requirements in active development at the same time. Therefore, it is not as flexible as other methodologies, however it is a great way to visualize a software’s development cycle (Stobierski, 2021).

The last methodology for consideration is the Waterfall methodology. The Waterfall methodology is a methodology where the development cycle is divided into different phases and each of the phases are done in order. This is great as it allows to synchronize a team with ease as everyone will be on the same page and working on the same phase. However, since everyone is on the same stage and the phases must be done in order, it is not flexible and therefore changing the requirements of a system mid-development will be difficult.

Overall, it is believed that Agile would be the best for the development of this system. The reason is Agile is relatively more flexible than the other two compared methodologies and allows the system to be developed piece by piece slowly. This is crucial as the development will be done during academic studies and not a lot of time can be put into it as there will be factors external of the project to consider.

# Chapter 5: Research Methods

## 5.1 Introduction

Since the project is designed with usage by a general non-specialized user, it is believed that quantitative research will be more beneficial for the project than qualitative research. This is because the detail and quality of the data is not as important as the size of the data and having a firmer grasp on a wider audience. The research aims to gather feedback via survey, to understand better on the public’s stance and opinions when it comes to what they look for in a translation system.

## 5.2 Design

A survey was made as a method to execute the quantitative research. The platform used to make the survey was Google Forms. This is because Google Forms is easy to use, fast, and free. The survey itself was separated into two different sections. The first section is to understand the demographic of the respondents, while the second section is to gather the respondents’ opinions and stance towards translation system related issues. Before both sections however, an introduction page containing a brief description of the survey as shown below is presented to the respondents so they can have an idea on what the survey is about.



As mentioned in the previous paragraph, the first section aims to understand the demographic of the survey’s respondents. To achieve this, the nature of the section’s questions are more general and less more about the system itself but more about the characteristics and background of the respondents such as age, gender, and their proficiency in languages such as how many languages they speak or how many writing systems do they know or use. The table below lists down the questions asked and elaborate on them.

|  |  |  |
| --- | --- | --- |
| Question | Question Type | Explanation |
| What is your gender? | Multiple Choice | General demographic question |
| What is your age group? | Multiple Choice | General demographic question |
| What is your nationality? | Short Answer | General demographic question |
| How many languages do you speak? | Multiple Choice | To gather data on how many languages can people speak on average |
| How interested are you in learning a new language? | Linear Scale | To gather data on how interested people are in learning new languages |
| How many writing systems or alphabet can you read?  (e.g., Latin alphabet, Arabic alphabet, Chinese, Thai, etc.) | Multiple Choice | To gather data on how many writing systems do people know on average when compared to languages |
| How many writing systems or alphabet can you write? (e.g., Latin alphabet, Arabic alphabet, Chinese, Thai, etc.) | Multiple Choice | To gather data on how many writing systems do people know on average when compared to languages |

The second and last section is filled with questions regarding the respondents’ opinions and experience towards translation systems and what they expect out of them or what they wish translation systems could be. These questions are geared towards finding out what features should be added when developing the system and what should be prioritized as there are so many variables and options to consider. The table below represents the questions within the second section of the survey.

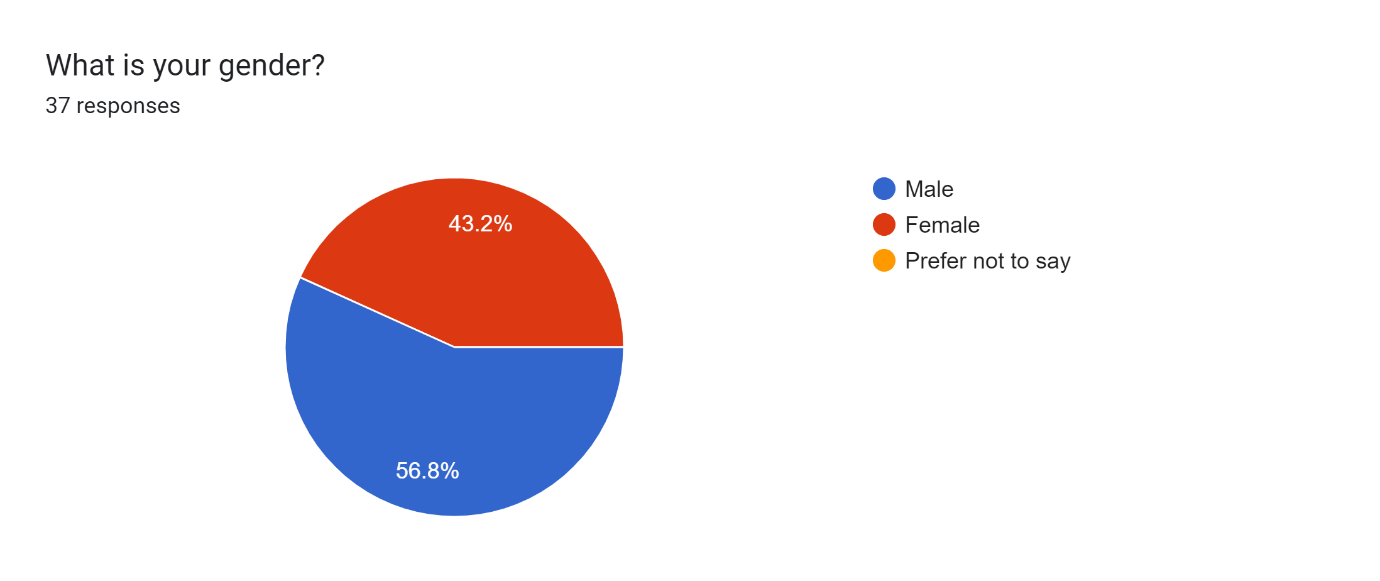
|  |  |  |
| --- | --- | --- |
| Question | Question Type | Explanation |
| Which of the platforms below do you use for translation the most? | Multiple Choice | To support the initial hypotheses that Google Translate is the most popular platform |
| How important is accuracy for you when it comes to translation? | Linear Scale | To understand the priority level of accuracy |
| It is irritating when websites or books I read are in a foreign language and I cannot translate them to a language I understand. | Linear Scale | To understand whether this survey’s problem statement of the need to translate images is important to the public or not |
| Would being able to translate words on images (e.g., street signs, shirts, book covers, etc.) such as the signs on the image below be beneficial to you? | Linear Scale | A physical example of the previous question to show respondents how it would be applied |
| How important is speech translation for you in translation software? (e.g., you can directly speak into your phone and have it translate your speech) | Linear Scale | To see the demand in speech translation and if it is a good addition to the system in the future or not |
| How interested would you be in a program that can translate foreign text on your screen such as foreign shows or websites? | Linear Scale | Some people do not understand that image translation is not limited to just street signs but it can be used in computer display as well. Data on that needs to be gathered. |
| I will consume more foreign media (e.g., books, shows, movies, etc.) if they can be translated automatically to a language I can understand. | Linear Scale | To gather data on how much language barrier is hindering people from accessing media of foreign language. |
| I will communicate with foreigners more online if both our messages can be translated automatically and accurately | Linear Scale | To gather data on how much language barrier is hindering people from communicating with foreigners of different language |
| It will be easier to acquire knowledge online if resources of all foreign languages can be translated into a language you understand automatically. | Linear Scale | To gather data on how much language barrier is hindering people from studying |

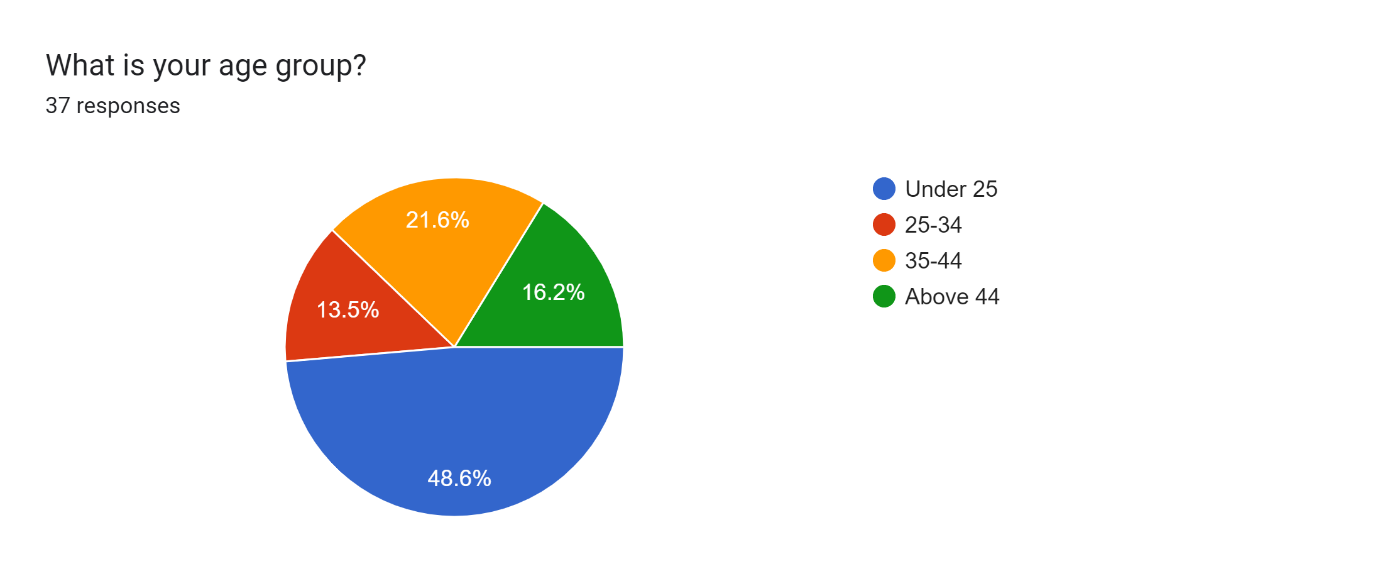
# Chapter 6: Requirements Validation

## 6.1 Analysis of Data

The survey received 37 responses and from those, a sizeable amount of data has been gathered and can therefore be analysed and summarized into information paramount to the development of the system.

### 6.1.1 First Section Data





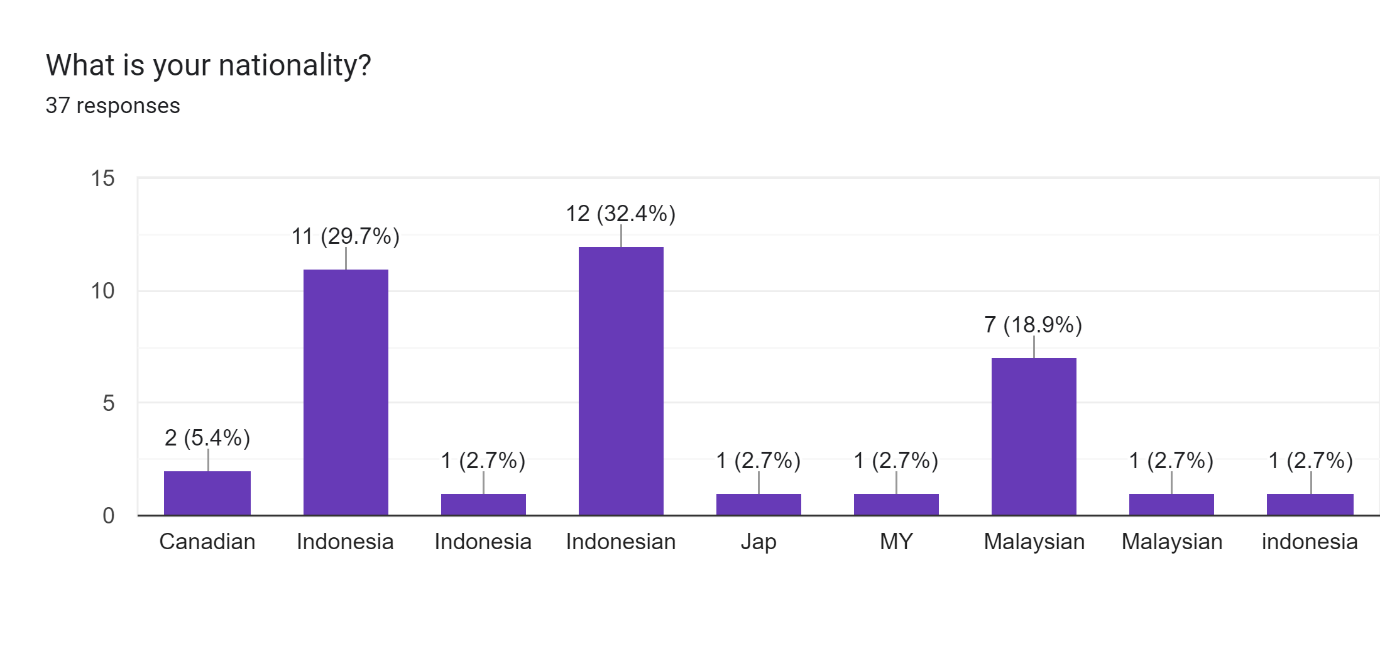


Figure 6.1.1 Demographic data

From the data above, we can conclude that a majority of the respondents are male. With close to half of the respondents being aged below 25, and the other half being split relatively evenly among the other three age groups. Over 90% of the respondents are Asian, with more than half of those being Indonesians. The rest are Malaysians with a single Japanese respondent.

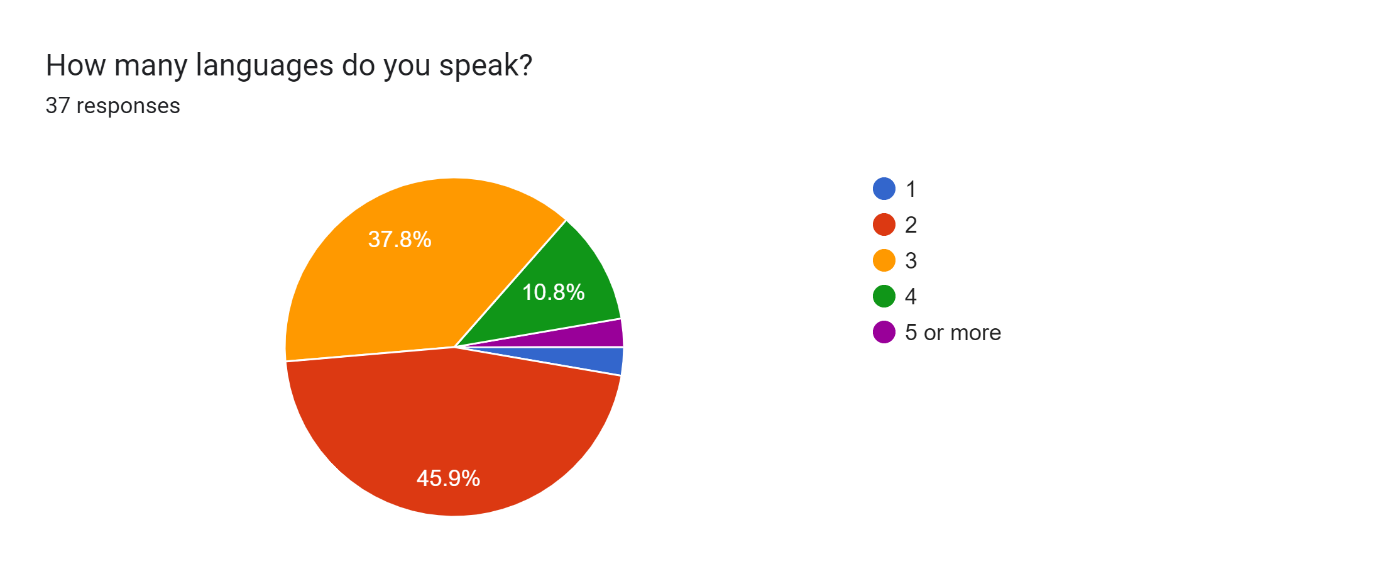


Figure 6.1.2 Language data

In this data, we can conclude that over 90% of the respondents speak at least two languages and around half of the respondents speak more than three languages and around 10% of the respondents are able to speak in four or more languages. The data is as expected as it is very common for Asians to be able to speak a second language – typically English – on top of their mother tongue. With some of them such as Indonesians and Malaysian Chinese being able to speak a third language. For example, a Javanese Indonesian would be able to speak Indonesian, English, and Javanese while a Chinese Malaysian would be able to speak Chinese, English, and Malay.

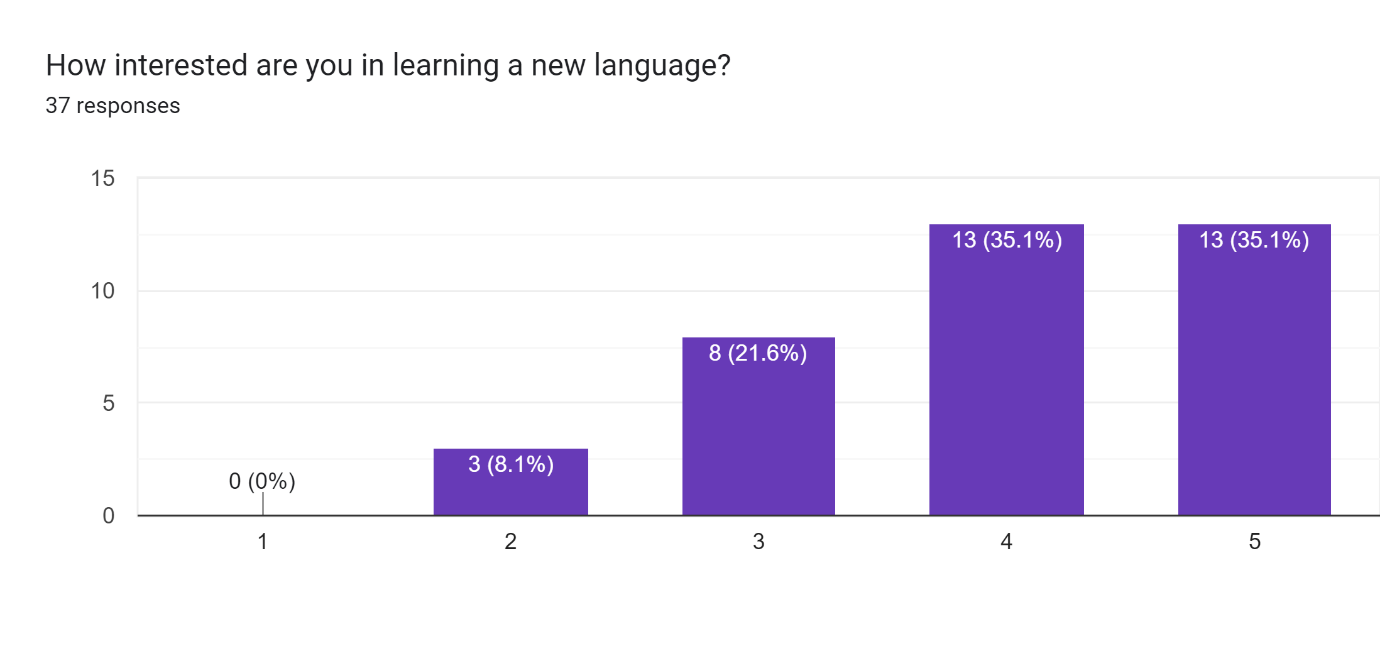
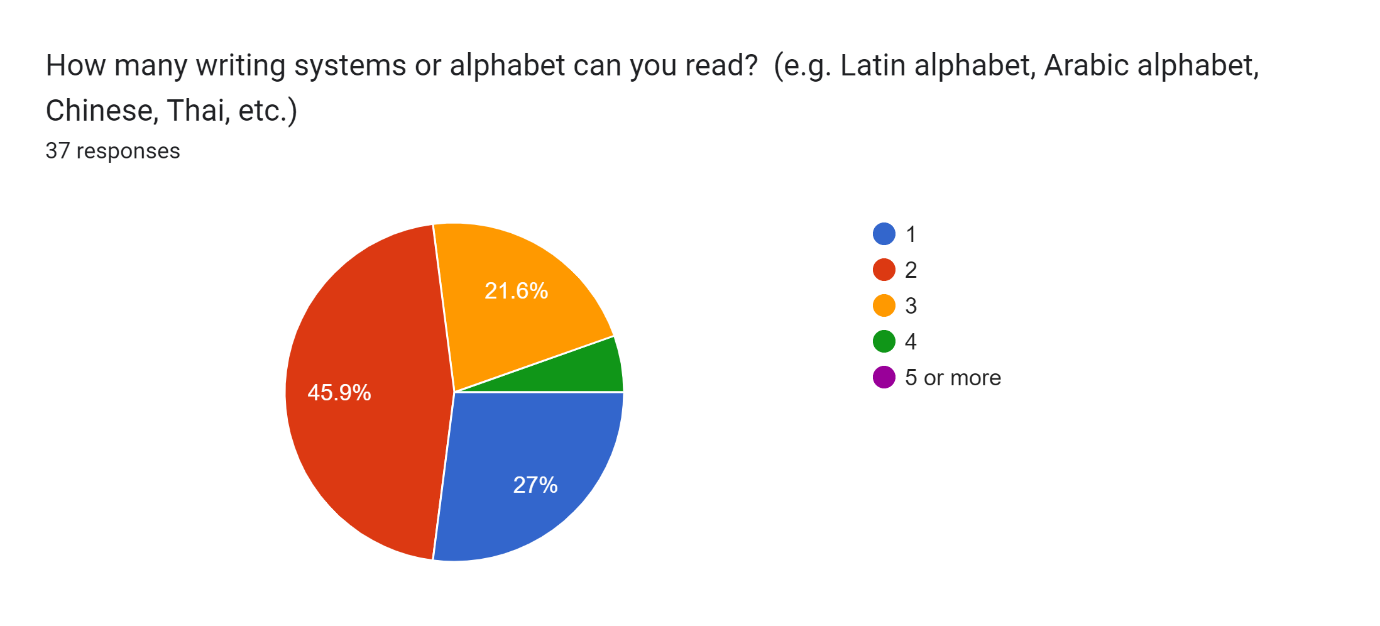


Figure 6.1.3 Data on interest in learning a new language

From this data, we can gather that the interest in learning a new language is actually quite high. With more than half of the respondents answering that they are somewhat interested in learning a new language and the rest are either neutral towards it or are simply uninterested with the idea.



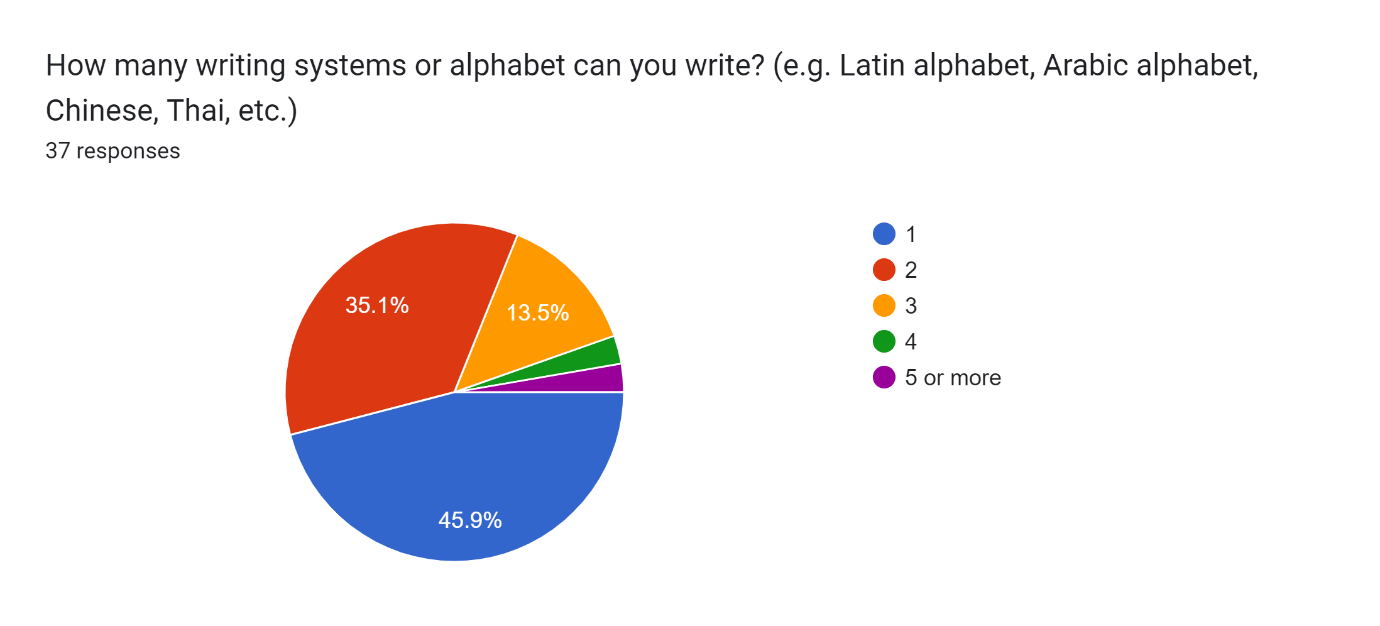


Figure 6.1.4 Writing systems known by respondents

From the two charts above, we can see that despite a wide majority of our respondents are able to speak multiple languages, not many of them can actually understand different scripts. This is because most languages share the same script and writing system which explains why the number is lower. It is also apparent that more people can read than write which shows a sign that the respondents may or may not be proficient in some languages. This is because when learning a new script, it is easier to read than it is to write. Therefore, this data is in line with the figure 6.1.3 that shows the respondents are interested in learning a new language.

This data can also be correlated with the data in figure 6.1.1 by comparing the nationalities of the respondents with the fact that a wide majority of the respondents are Indonesians and Malaysians. When it comes to Indonesians, 200 million Indonesians are Muslims (Investments, n.d.). Therefore, it can be explained that a majority of respondents can understand a different script as it is very common for Muslims to be able to read Arabic but not necessarily speak the language. When it comes to Malaysians, there are ethnicities within the country such as the Indians and the Chinese that have their own writing systems as well. Overall, the data is in line with one another.

### 6.1.2 Second Section Data

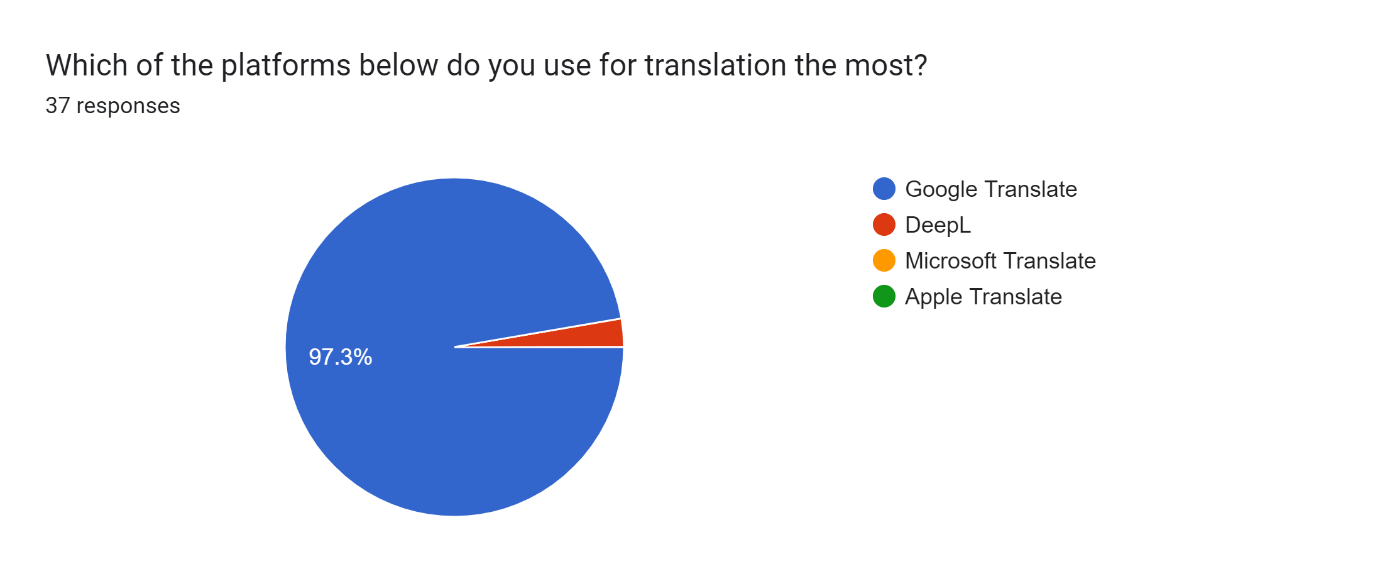


Figure 6.2.1 Platform popularity

The data shown in the chart above shows that the idea of Google Translate being the biggest player in the market with the most user is true when it comes to the respondents’ personal experience and preference with translation systems.

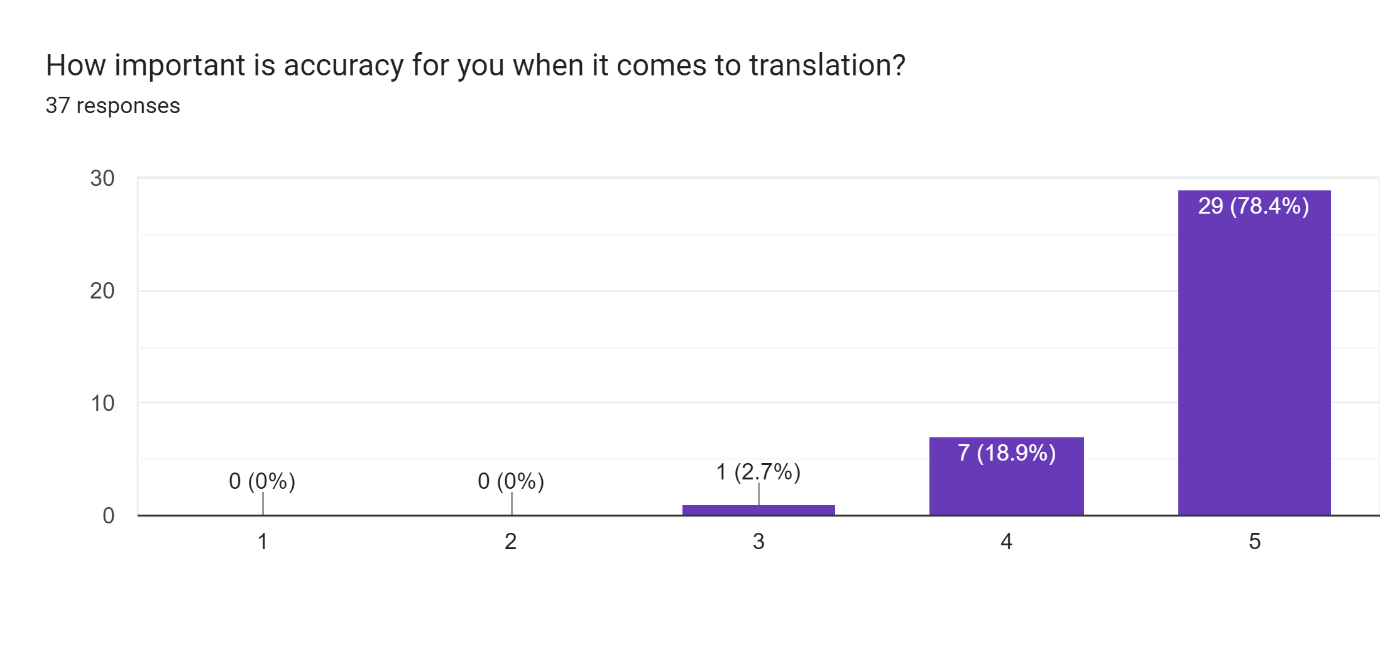


Figure 6.2.2 Importance of accuracy

The data above shows that over 97% of the respondents agreeing that accuracy is very important when it comes to translation. This means accuracy should be prioritized during the development of the system.

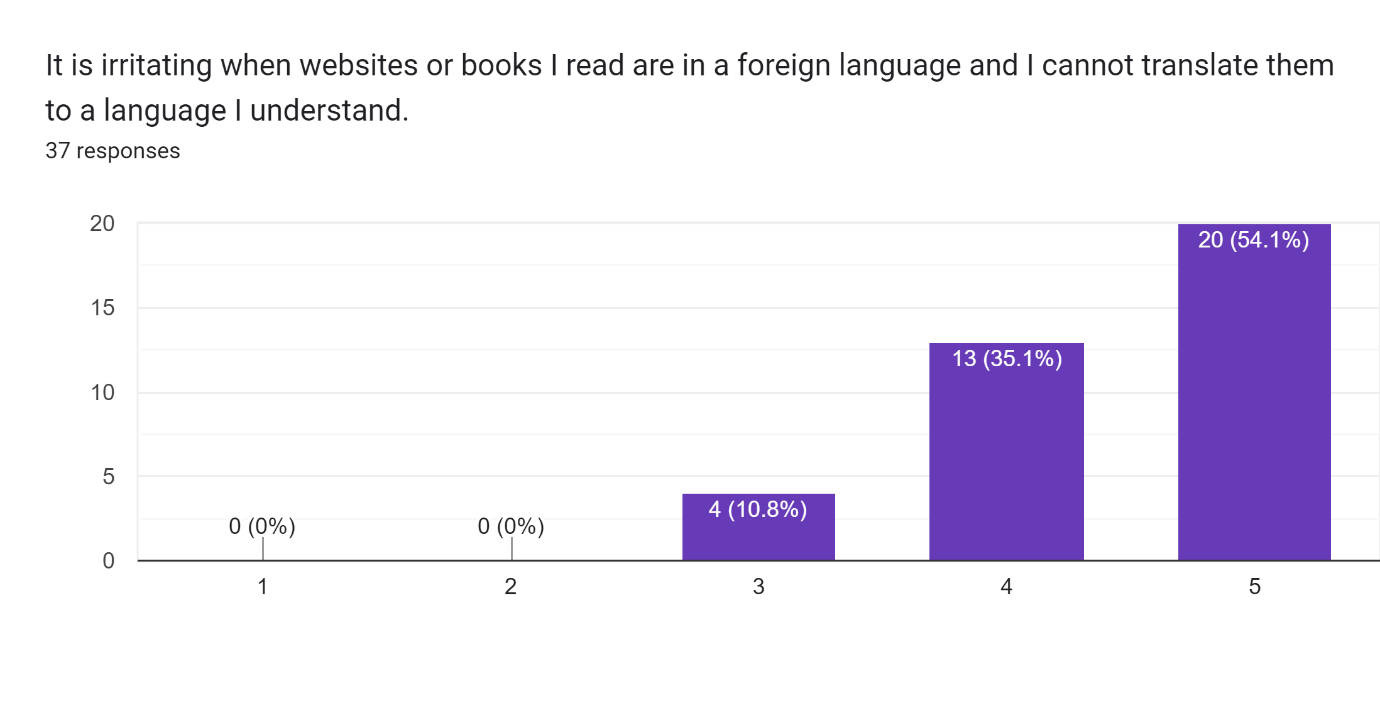


Figure 6.2.3 How problematic not being able to translate a media is

The data above shows that nearly 90% of the respondents agree that it is irritating when they cannot translate media into a language they understand. In most cases this is because websites tend to have texts as images and a lot of them are isolated from one another, making it difficult to copy and paste into a translator such as Google Translate.

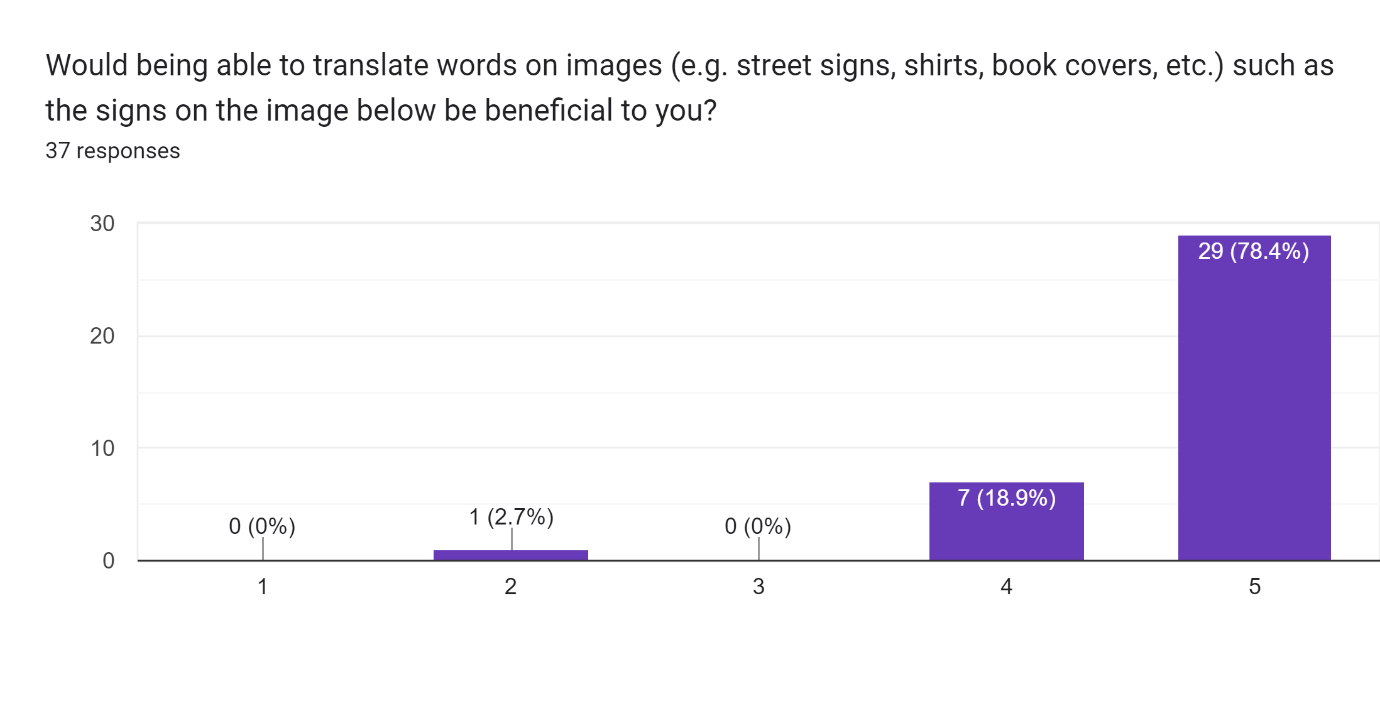


Figure 6.2.4 Importance of translating images of physical objects

In the data above, we can see that over 97% of the respondents find it beneficial when translation systems can translate images as an input. The question above uses an image of a street sign in foreign languages as a visual example.

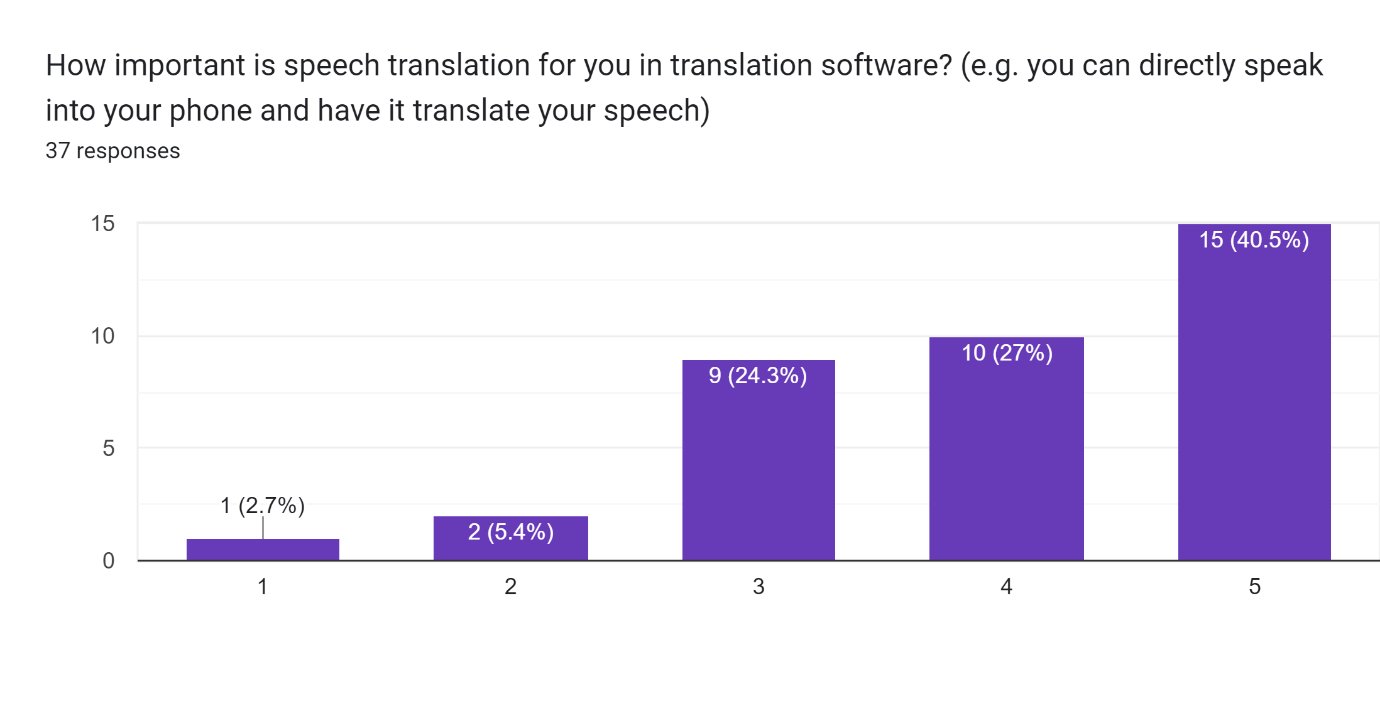


Figure 6.2.5 Importance of speech translation

The data above shows that speech translation does not have as high of a demand as the previous proposed features with this data being the first one to have a “Strongly Disagree” response. The general trend of the data still shows that over half of the respondents do think that it is important therefore speech translation might be considered as an added feature for the system.

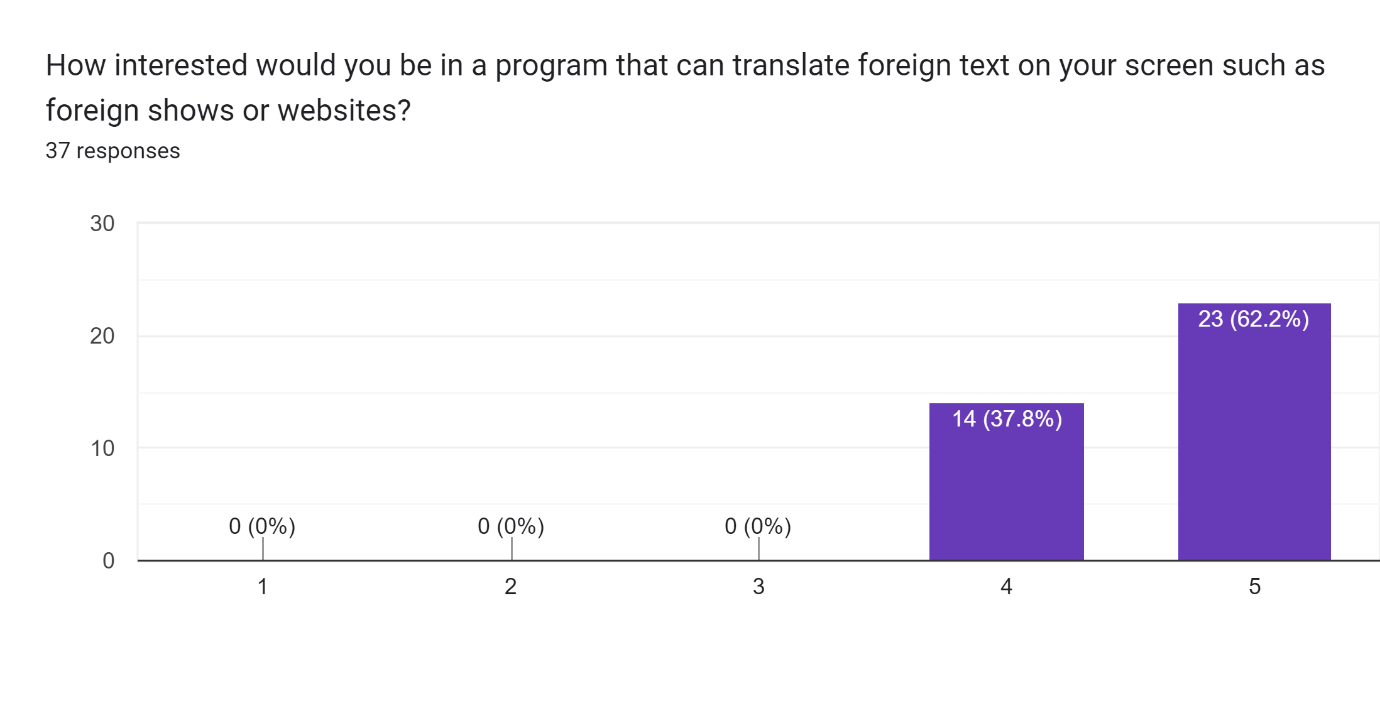


Figure 6.2.6 Importance of translating images of digital objects

The data above shows when the example is changed from street signs to digital media, the importance of image translation increases. This can be derived from the pandemic that has been hitting the world in the past two years and leading people to live an isolated life surrounded by technology. We can conclude from this graph that the system is indeed in demand.

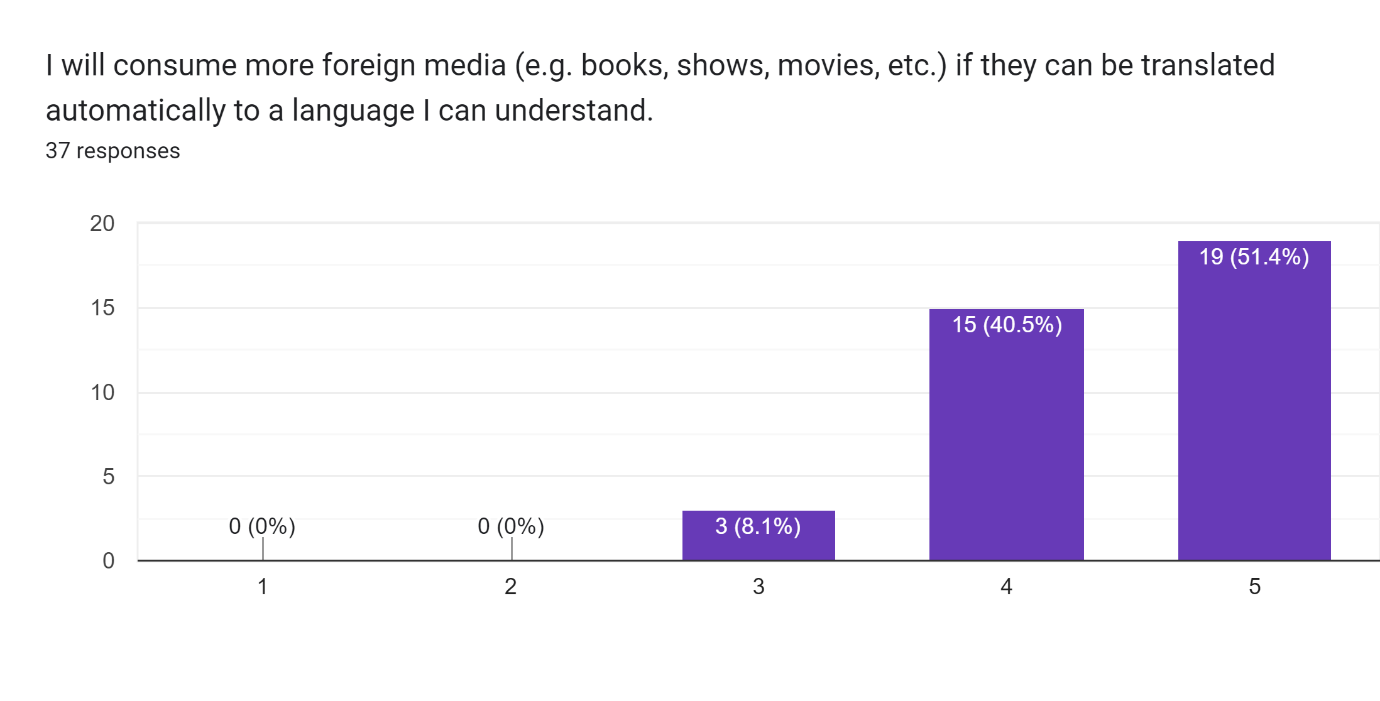
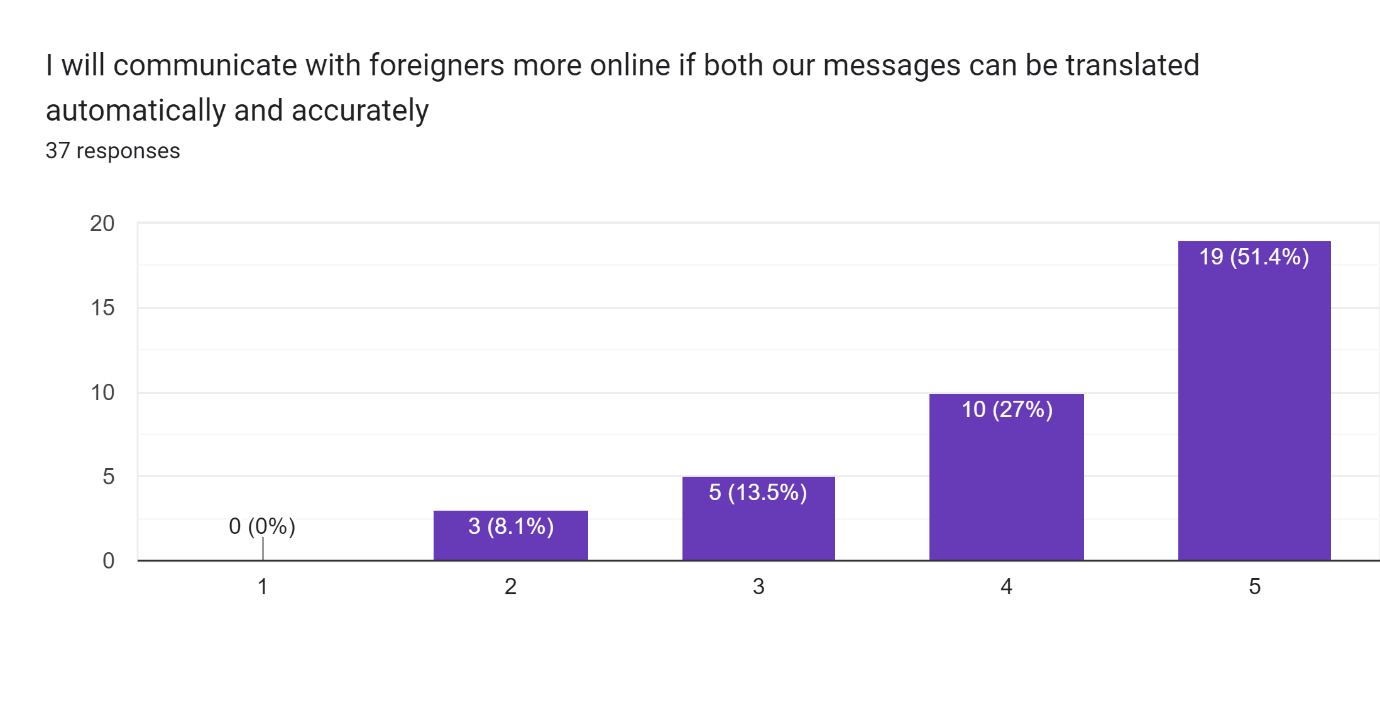
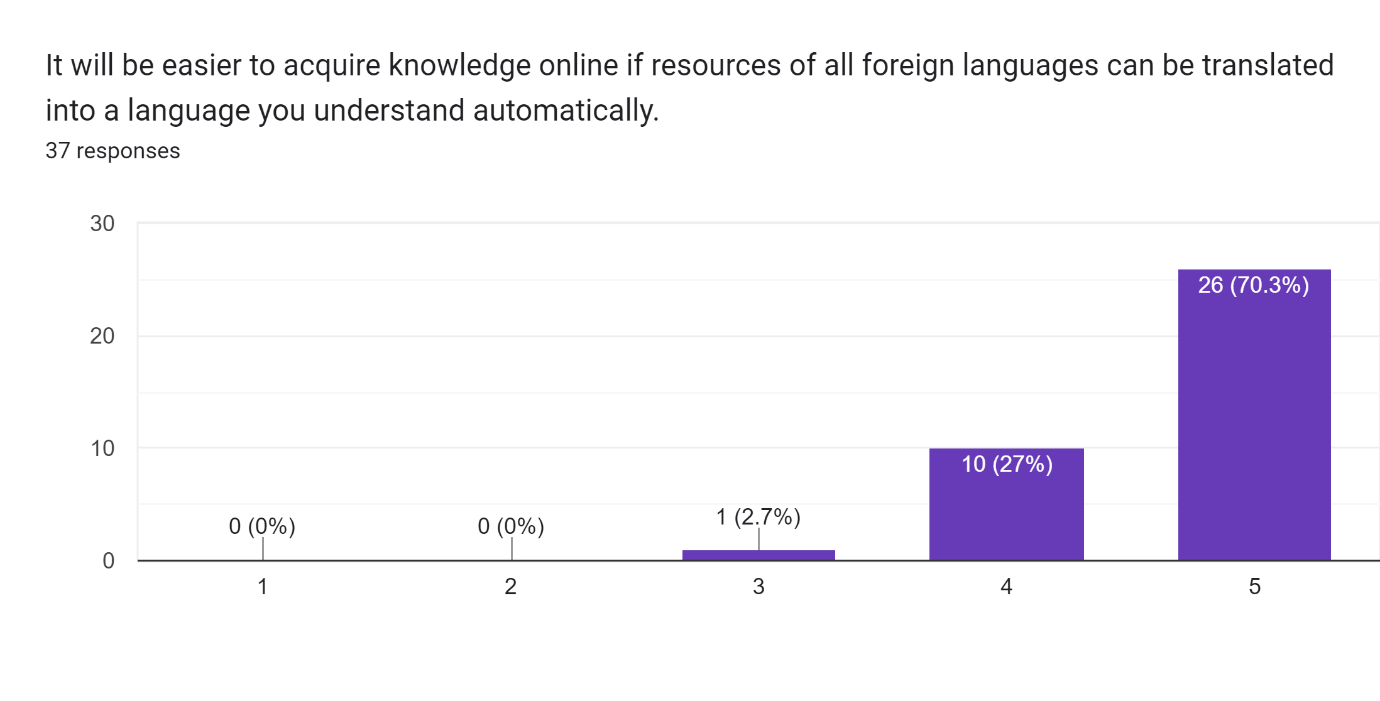
  

Figure 6.2.7 How much more people would do digitally if the language barrier were to be removed

The three graphs above shows that if the language barrier were to be removed, the respondents are willing to explore more outside of their current domain and touch new grounds.

## 6.2 Summary

Looking at the data that the survey has gathered, a conclusion can be achieved. Most of our respondents are multilingual with more than half of them being able to understand multiple scripts. The demand for the proposed system is there, and some features such as speech recognition is in demand as well however its demand is not as high as image translation. Accuracy is very important and people would do more if they are able to have foreign media translated into a language they can understand.

# Chapter 7: System Architecture

## 7.1 System Architecture Introduction

The OCR Translation software is a system that is capable of reading text from an image input and then translating them using machine translation. The purpose of this software is to implement and join the two technologies which are OCR and machine translation together, to form a single system capable of both running on the user’s device. These two technologies are very complex and advanced, to the point where developing either of them will be too much of a technical undertaking for a degree student’s final year project. To solve this problem, the system will be using pre-existing and already developed technologies. The OCR technology that this software uses is Tesseract OCR, while the machine translation is provided by DeepL. The purpose of this software is to create a platform for the two technologies to work together and serve a unified purpose which is OCR translation. The system uses a GUI for ease of use and allowing the program to utilize multiple functions of the system with button presses. The software will take in inputs in the form of images and process it accordingly to extract text from the images. This text will then be translated using DeepL’s machine translation API. The API is capable of translating a variety of languages, but for this software only Indonesian to English will be used as these two are the languages that I am proficient in.

## 7.2 System Architecture Diagrams

### 7.2.1 OCR Translation Use Case Diagram

This section shows the system architecture using a use case diagram.

Diagram

Description automatically generated

Figure 7.2.1.1 Use Case Diagram

|  |  |
| --- | --- |
| Use Case | Image |
| Description | The system will provide the user with the available options for Image translation |
| Actor | User |
| Prerequisite | User is on the main menu of the program |
| Workflow | The system will read the input within the GUI, and if the event matches the Image button, it will proceed to the image options menu. |

|  |  |
| --- | --- |
| Use Case | Video |
| Description | The system will provide the user with the available options for Video translation |
| Actor | User |
| Prerequisite | User is on the main menu of the program |
| Workflow | The system will read the input within the GUI, and if the event matches the Video button, it will proceed to the video options menu. |

|  |  |
| --- | --- |
| Use Case | PDF |
| Description | The system will provide the user with the available options for PDF translation |
| Actor | User |
| Prerequisite | User is on the main menu of the program |
| Workflow | The system will read the input within the GUI, and if the event matches the PDF button, it will proceed to the PDF options menu. |

|  |  |
| --- | --- |
| Use Case | Image Import |
| Description | The system will read for the requested file and run OCR. |
| Actor | User |
| Prerequisite | User is on the image menu of the program and has an image file within the same directory. |
| Workflow | The system will call a function that looks for the specified file within the same directory and runs the OCR function on the input image. |

|  |  |
| --- | --- |
| Use Case | Image Clipboard |
| Description | The system will read the image stored in the computer clipboard. |
| Actor | User |
| Prerequisite | User is on the image menu of the program and has an image stored in the clipboard. |
| Workflow | The system will call a function that pulls an image from the computer’s clipboard and runs the OCR function on the input image. |

|  |  |
| --- | --- |
| Use Case | Video Capture |
| Description | The system will capture a specified window and continuously screenshots the window. Sending the screenshots to the system for OCR. |
| Actor | User |
| Prerequisite | User is on the video menu, has a window open with a static window name. |
| Workflow | The system gets the window specification such as size and position on the screen. And then it creates a bitmap and returns it. The system then runs OCR on this returned image at certain user specified intervals. |

|  |  |
| --- | --- |
| Use Case | PDF OCR |
| Description | The system will capture pages of a PDF file and runs OCR on the pages for translation. |
| Actor | User |
| Prerequisite | User is on the PDF menu and has a PDF file ready. |
| Workflow | The system grabs a page from the PDF file and convert it into an image saved on the same directory as the system. and runs OCR on the image. |

|  |  |
| --- | --- |
| Use Case | PDF Reader |
| Description | The system will extract text directly from the input PDF file |
| Actor | User |
| Prerequisite | The user is on the PDF menu and has a PDF file ready. |
| Workflow | The system reads raw text from the PDF file and returns it. |

|  |  |
| --- | --- |
| Use Case | Translation |
| Description | The system translates the input text and returns the translation result |
| Actor | User |
| Prerequisite | User sent a text to this case using any of the three main functionalities. User also needs to have an active API key for DeepL translation to work. |
| Workflow | The system receives a text input and sends it for translation using the DeepL API. The received translation is then returned by the function. |

### 7.2.2 OCR Translation Flowchart

The flowchart below shows the flow of the program. It shows that at any point during the program running, the user is given options on what they want to do. And at any point after a process is finished, the results will be displayed, and the user is given an option whether they want to go back to the main menu or quit the program.

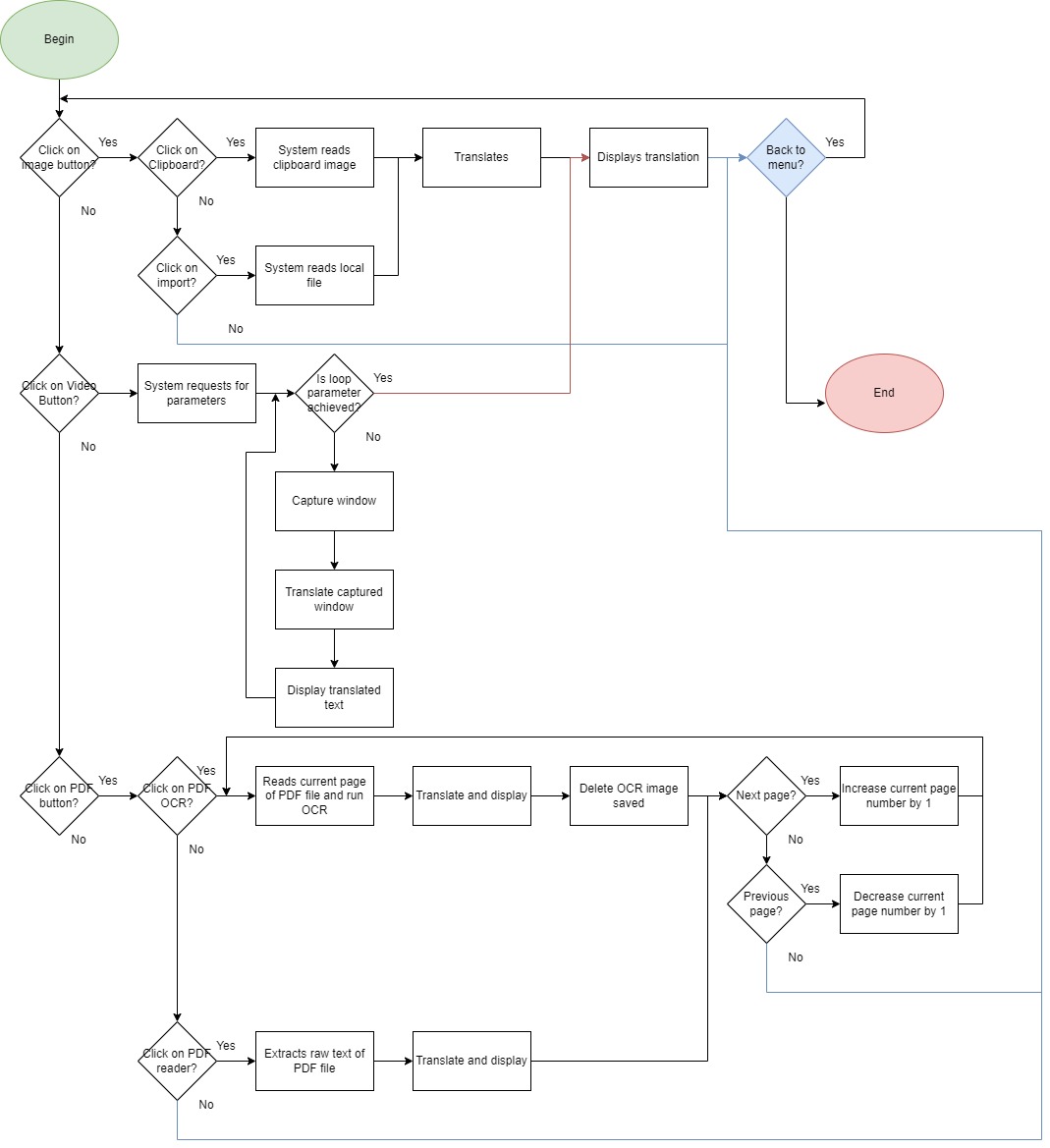


Figure 7.2.2.1 Flowchart of OCR Translation Software

# Chapter 8: Project Plan

## 8.1 Release Plan

### 8.1.1 OCR Translation 1.0

The first version of the OCR Translation software was released on December of 2022. This version includes the basic components running which includes translation from images, window capture video stream, and PDF files. Since this version was meant to be a prototype to proof that the system is capable of running the proposed function, it has no GUI and is fully CLI-based. The program will run through a terminal, and outputs all translations into the terminal. At this stage, each function of the program is their own python file. Which means the user will have to run the program according to the functionality that they need.

### 8.1.2 OCR Translation 2.0

The second version of the OCR Translation software was released at the start of 2023. This version adds a GUI to the software, and allows the user to interact with the software using buttons to navigate through the menu and execute actions. The output of the program will also be shown on the GUI instead of the terminal in the previous version. With the exception of the video stream feature which only outputs to the terminal due to technical limitations.

## 8.2 Project Test Plan

# Chapter 9: Implementation

## 9.1 UI

The UI of the system was made using the Python library PySimpleGUI. The GUI consists of buttons to execute actions, and text boxes to input or output text. Below is a sample of one of the pages in the system showing the UI of the system.

Graphical user interface, text, application

Description automatically generated

Figure 9.1.1 Image Clipboard Output

On some pages of system, the system will output the input image to show the user what the program is seeing. The image below shows Figure 9.1.1 on the right, and the image that was used as its input on the left.

Graphical user interface, text, application

Description automatically generated

Figure 9.1.2 Image Clipboard Output and Input image.

The way PySimpleGUI library work is the developer must design a layout, and then initialize windows using the designed layout. For this system, one main layout is used which contains columns using different layout within it. Pressing different buttons during the program will activate and deactivate the layouts within the system to show the corresponding menu based on the buttons pressed. The code snippet below shows a page layout at the top and the main layout using it at the bottom alongside the other page layouts.

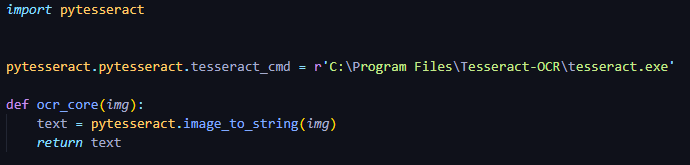
Text

Description automatically generated

Figure 9.1.3 Code snippet of layouts and its initialization at the bottom.

## 9.2 Libraries Implementation

The system uses the Tesseract OCR engine. To implement this OCR engine into the system, the pytesseract library was used. This library allows the implementation and usage of the Tesseract OCR engine in Python programs. All that needs to be done is for the path to the Tesseract OCR engine to be specified for the program to use, and a function to be called. In the function below, it takes in an image as an argument, and runs the Tesseract OCR engine on the image and returns a string of text for further processing.



In some cases however, the system itself needs to generate or find a way to extract image. Multiple libraries are used for different methods of obtaining images. The Pillow library was used to extract images from the computer’s clipboard for OCR. OpenCV was used to obtain imported images by the user as there was no processing needed, only a method to get it into the program.

For PDF files, there are mainly two libraries that were used: fitz and PyPDF2. Fitz was used to convert PDF pages into images for OCR processing. PyPDF2 was used to extract raw text from PDF files in case the user does not need OCR and just want to extract raw text from the PDF file.

For video capture the libraries numpy and wind32gui was used. The algorithm used here was cited from Ben Johnson (2020) where the program would find a window based on user input with the same name, gets the window’s specifications such as size and location on the screen, and return bitmaps as images to the OCR function.

Last but not least is the DeepL library which allows DeepL API implementation into Python programs. This library eases the use of the API as the API requires POST and GET requests which are tricky to implement. With the DeepL library however, the user only needs to input their API key into the authKey.py file and the system will be ready to use. API keys can be obtained on DeepL’s official website.

The table below shows all libraries used within the program and their versions. The development itself was done on Python version 3.9.12.

|  |  |
| --- | --- |
| Library | Version |
| DeepL | 1.11.0 |
| Fitz | 0.0.1.dev2 |
| numpy | 1.21.5 |
| Opencv | 4.6.0.66 |
| Pillow | 9.4.0 |
| PyMuPDF | 1.21.0 |
| PyPDF2 | 3.0.1 |
| PySimpleGUI | 4.60.4 |
| Pytesseract | 0.3.10 |
| Pywin32 | 302 |
| Win32gui | 221.6 |

# Chapter 10 System Validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **Expected Outcome** | **Actual Outcome** | **Pass/Fail** |
| 1 | Image Clipboard OCR | Reads image from clipboard and translates | As expected | Pass |
| 2 | Image Import OCR | Reads image from directory and translates | As expected | Pass |
| 3 | Video OCR | Reads window and translates at interval | As expected | Pass |
| 4 | PDF OCR | Converts PDF page to image and OCR for translation | Everything is functioning however the conversion process tends to output a small image, causing the OCR to not have a good input and interfering with translations | Fail |
| 5 | PDF Reader | Reads PDF Text and translate | As expected | Pass |

# Chapter 11 Conclusion

## 11. 1 Implementation Challenges

It was difficult to implement and develop a system as initially planned. Most of the difficulties were faced within the video capture feature as there were hardware limitations and simply not enough time to develop one. Constantly capturing a window and running OCR is very computationally demanding.

Implementing machine translation was also difficult as most engines only provide limited trials. To run this program, the user must register with DeepL to obtain their own DeepL API key which is required to run the machine translation API.

## 11.2 Future Plans

Ideally, the system should have its own OCR engine and machine translation model developed. However these two technologies are far too complexed to be developed by a single-man team. The technologies used in this system are cutting edge and are leaders in their own field. With the right team however, it will be possible for the OCR translation system to have its own in-house OCR engine and machine translation model.

## 11.3 Summary

In conclusion, the language barrier is something that is holding back mankind as a whole. Isolating communities and cultures from one another. Preventing communication between cultures with difference in language. The proposed system aims to tackle it, and open up new horizons. Technically, it can be achieved as the technology is available in this day and age. With tools like Tesseract OCR and DeepL implemented in the same system, an image translation system can be realized. The demand is there as shown by the survey done, and additional insights such as which properties to focus on and features to consider has also been received. With this, developing the system is very feasible.

# References

Arnold, C., 2013. *Shhh, the Ants Are Talking*. [online] Science.org. Available at: <https://www.science.org/content/article/shhh-ants-are-talking#:~:text=Scientists%20have%20known%20for%20decades,ants%20marching%20one%20by%20one.> [Accessed 23 June 2022].

Jose, J., 2015. *The Difference Between Animal and Human Communication*. [online] Owlcation. Available at: <https://owlcation.com/stem/The-difference-between-animal-and-human-communication#:~:text=The%20way%20that%20animals%20communicate%20are%20biological%2C%20or%20inborn.&text=Human%20language%20is%20symbolic%2C%20using,preserve%20ideas%20of%20the%20past.> [Accessed 23 June 2022].

Brown, S., 2021. *Where Did Writing Come From?*. [online] Getty.edu. Available at: <https://www.getty.edu/news/where-did-writing-come-from/> [Accessed 23 June 2022].

DayTranslationsblog. 2022. *Hyperpolyglots: How Many Languages Can You Learn?*. [online] Available at: <https://www.daytranslations.com/blog/how-many-languages-can-you-learn/> [Accessed 13 August 2022].

Turovsky, B., 2016. *Ten years of Google Translate*. [online] Google. Available at: <https://blog.google/products/translate/ten-years-of-google-translate/#:~:text=There%20are%20more%20than%20500%20million%20of%20you%20using%20Google%20Translate.> [Accessed 3 August 2022].

Colin, 2022. *DeepL vs Google Translate: Which Is Better? + How to Use Them (2022)*. [online] TranslatePress. Available at: <https://translatepress.com/deepl-vs-google-translate-comparison/> [Accessed 14 August 2022].

Eikvil, L. (1993). Optical character recognition. *citeseer. ist. psu. edu/142042. html*, *26*.

Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, *2*(1), 72-75.

Stahlberg, F. (2020). Neural machine translation: A review. *Journal of Artificial Intelligence Research*, *69*, 343-418.

Vincent, L., 2006. *Announcing Tesseract OCR*. [online] Googlecode.blogspot.com. Available at: <http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html> [Accessed 17 August 2022].

DeepL. n.d. *Why DeepL?*. [online] Available at: <https://www.deepl.com/en/whydeepl> [Accessed 17 August 2022].

StatCounter Global Stats. 2022. *Desktop Operating System Market Share Worldwide | Statcounter Global Stats*. [online] Available at: <https://gs.statcounter.com/os-market-share/desktop/worldwide#monthly-202204-202204-bar> [Accessed 17 August 2022].

Stobierski, T., 2021. *Agile vs. Scrum: What’s the Difference?*. [online] Northeastern University Graduate Programs. Available at: <https://www.northeastern.edu/graduate/blog/agile-vs-scrum/#:~:text=The%20key%20difference%20between%20Agile,used%20to%20facilitate%20a%20project.> [Accessed 17 August 2022].

Investments, I., n.d. *Introduction to Islam in Indonesia - Development of Indonesian Islam | Indonesia Investments*. [online] Indonesia-investments.com. Available at: <https://www.indonesia-investments.com/culture/religion/islam/item248#:~:text=Islam-,Islam%20in%20Indonesia,whom%20adhere%20to%20Sunni%20Islam.> [Accessed 18 August 2022].

Stobierski, T., 2021. *Agile vs. Scrum: What’s the Difference?*. [online] Northeastern University Graduate Programs. Available at: <https://www.northeastern.edu/graduate/blog/agile-vs-scrum/#:~:text=The%20key%20difference%20between%20Agile,used%20to%20facilitate%20a%20project.> [Accessed 17 August 2022].

Johnson, B., 2020. *OpenCV Tutorials*. [online] Available at: <https://github.com/learncodebygaming/opencv_tutorials/tree/master/004_window_capture> [Accessed 13 December 2022]

# Appendix

## Log Sheets



## PPF



## PSF



## Ethics Form

