Sign Language Translator Application

Nur Fatin Nadzirah Binti Romizi
Universiti Kuala Lumpur Malaysian
Institute of Information Technology
Kuala Lumpur, Malaysia
fatin.romizi@s.unikl.edu.my

Shahidatul Arfah Baharudin
Universiti Kuala Lumpur Malaysian
Institute of Information Technology
Kuala Lumpur, Malaysia
shahidatularfah@unikl.edu.my

Abstract — **Communication plays an important role for human beings and it is treated as a compulsory life skill. Normal people could communicate easily without any problems, however there is a hindrance for normal people to communicate with disabled people such as deaf and mute. This happened because disabled people communicate using sign language but only minority of people learn and understand the sign language. Human translator also might not be there every time to help and it can be quite difficult to understand the sign language. Hence, the communication is not going well. Thus, this project aims to develop a sign language translator application to help the communication between normal people and disabled people going efficiently. This project will be using some features in the OpenCV such as image processing, features extraction and features matching. This project also will use Android Studio for generating the android application. For beginning, this project will develop an application that only translate the 26 alphabets. All the analysis, equipment and methods used, application testing and results has been documented in this article.**

Keywords — Sign language, OpenCV, Android Studio

I. Introduction

A. Background

This project come up with the idea of sign language translator application consists of designing and developing an intelligent system using image processing, and machine learning concepts to take visual input of sign language's hand gestures and generate easily recognizable form of text output. Hence, this system can act as a translator between the sign language and the spoken language and would also make the communication between both disabled people (deaf and mute) and normal people efficient.

The objectives of this project are is to develop a sign language translator Android application that able to translate between sign language and spoken language and to test this application whether it can be used to translate the sign language.

There is a gap or barrier between people who have speaking or hearing impairment and normal people to communicate. Human translator might not be there every time to help and it can be difficult to understand sign language. Hence, the communication is not going well. This application project is to overcome this problem by acting as a translator or a medium for them to communicate with each other efficiently.

There are some limitations found while doing this project. The limitations are:

- The past articles for references are mainly about system rather than application.
- The application only captures static figure, so it takes time to translate the sign language.
- The application only configured with one type of sign language.
- The application can only translate alphabets.

II. Literature Review

A. Overview

This chapter contains discussion regarding some topics that related to this project. Sign Language, OpenCV and Android Studio also will be explained briefly. In this chapter, the literature review that consists of the works from previous researchers that has relation towards this area of project also will be summarized. This chapter is important to show the originality and relevance of this project. It describes how this project is related to prior research in statistics and justifies the proposed methodology of this project.

B. Sign Language

Sign languages (also known as signed languages) are languages that use the visual-manual modality to convey meaning. There are over 135 different sign languages around the world, and different countries uses different sign languages. They also have their own grammars. This project will use *Bahasa Isyarat Malaysia* (BIM or Malaysian Sign Language). BIM has its own grammar that focuses more on what the deaf person can see. American Sign Language (ASL) has a strong influence on BIM, but both are considered different languages. But this project will focus only on translating 26 alphabets.

C. *OpenCV*

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. The library has more than 2500 optimized algorithms that can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, and track moving objects and more. OpenCV has C++, Python, Java, and MATLAB interfaces and supports various operating system including Windows, Linux, Android, and Mac OS.

D. *Android Studio*

Android Studio is the official Integrated Development Environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for android development. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building android apps, such as a fast and feature-rich emulator, extensive testing tools and frameworks, C++ and NDK support and more. It can be written in three programming languages which are Java, Kotlin and C++.

E. *Comparison on previous related projects*

American Sign Language Alphabet Translator Android Application: Hand Shapes into Text [1] - The aim of this project is to develop a mobile application capable of recognizing sign language alphabet poses in an android phone. The software and hardware required to develop this application are OpenCV, Android Studio, Support Vector Machine (SVM), an android smartphone and this project uses Java and C++ sign languages. The methods used by the authors are case diagram, system architecture and thresholding for hand segmentation, feature extraction and the algorithm to build the training model and make prediction. The developed application managed to recognize 18 out of 26 letters in the alphabets with percentage of 69%. The higher the percentage of seconds the translator to predict the right letter, the slower the translator cam recognize the hand shapes.

Sign Language Recognition System Simulated for Video Captured with Smart Phone Front Camera [2] - The aim of this project is to bring sign language closer to real time implementation on mobile platforms. The software and hardware requirements stated by these authors are OpenCV, Asus Zen phone II and Samsung Galaxy S4 equipped with 5M pixel front camera. The methods used by them are video processing, segmentation, feature extraction and classification. The classification of the words was tested with Euclidian and Mahalanobis distance functions. The outputs present are the camera can capture 18 words "Hai, Good Morning, I am P R I D H U, Have A Nice Day, Bye, Thank You".

Android Application for Sign Language Recognition [3] – The aim of this project is to develop an android mobile application that uses gesture recognition to understand American Sign Language. OpenCV and MATLAB are the software requirements that were listed by the authors. They also used programming language Java and C++. The methods used by the authors are image processing, color-based segmentation, background subtraction, morphological operations, contours, and extract contours information, removing false positives in convex hull and recognizing features. Highest recognition rate was for gestures "B", "D", "I", "K", "L", "R", "V", "W", and "Y" while the lowest recognition rate was observed for gestures "G", "H", "N", "P", "Q", "S", "T", and "U". The application is not programmed to create for non-static gestures "J" and "Z".

III. METHODOLOGY

A. *Overview*

This chapter contains discussion about the methodology used in this project. It provides the research method, flowchart, and the timeline for certain tasks to be done in this project. Project methodology is important because it will be the guide to develop and control the development of this application project.

B. *Reseacrh Method*

The research method chosen for this project is Rapid Application Development (RAD). It describes a method of software development that emphasizes rapid prototyping and iterative delivery. The reason for choosing this method is because it is more adaptable and flexible to changes as there may be a lot of changes and improvement made after the testing phase to produce a working final application.

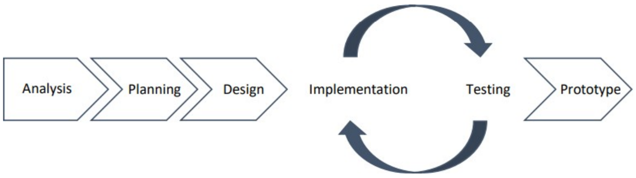

Figure 1: Rapid Application Development (RAD) phases

There are six phases in Rapid Application Development (RAD) model which are analysis, planning, design, implementation, testing and prototype. The first phase of this model is analysis which plays crucial role that will ensure the progress of theproject. It includes the activities of determining the initial requirements for this project. The outcome for this phase is to comprehend the gathered requirements and develop the application that direct to these requirements. The second phase is planning. After analyzing all the requirements, next is to plan the time management, cost, risk, and issues of the project. This phase also can help to guide the project through its execution phase. Next, is the design phase.

In this phase, the flowchart of the project and the interface of the application is designed. All the software and hardware requirements gathered during previous phase are used to sketch and construct the sign language translator application. Then, in the implementation phase, the requirements, coding, and designs are composed to develop the application. After that, the designed and developed application will need to go through some constant testing to check if there are any errors and flaws. Any improvements or corrections should be done in this phase. Lastly, after the application has been tested and corrected, the final prototype is produced. In this prototype phase, the application produced is fully functional and working and there are no changes needed. After the project is completed and concluded, a report of the whole process is documented. The other documentation that required is the information on how to maintain the application and any possible future enhancements.

C. *Flowchart*

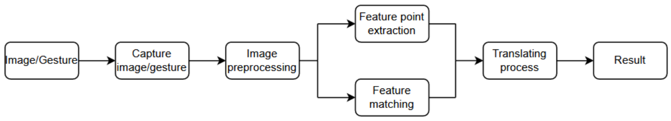

Figure 2: Translating Process Flowchart

Firstly, the application will capture image or video of gestures of the sign language using the mobile camera. Then the application will perform image preprocessing. During this step, the application will do the color-based image segmentation and background subtraction. Image segmentation is the process of dividing a digital image into different segments. Color-based image segmentation uses the color feature of every pixel to differentiate meaningful pixels in the image. Background subtraction is another way of image preprocessing that extracts pixels of moving objects from static background. After that is the feature point extraction and feature matching. This step is to recognize the hand gestures. This application uses a concept of convex hull for this recognition purpose. Convex hull is produced using OpenCV functions to determine the fingertips and basic handshape which then will be used to understand the gestures. It uses the lengths of the lines connecting the centroid and vertices of the convex hull to recognize the gestures and differentiate it from other gestures. Finally, the application will do the translating process and produce the result.

IV. Result & Discussion

A. *Overview*

This chapter contains the explanation of the result and analysis of this application testing. The main purpose is to identify whether this application is functioning properly like it should, and all the analysis recorded in a bar chart form will be explained more in this chapter. Other than that, the result on the user testing and a further discussion about the comparison of the result analysis are also included in this chapter.

B. *Result Analysis*

This section shows the results of the alphabet detection rate analysis in a table form. The analysis table includes what percentage that this application detects the alphabets and what percentage that this application has mistaken the alphabets with other alphabets that have similar gesture. The total test that has been carried out was 25 times.

ALPHABET	Prediction detection rate %	Not Detect Rate %
A	80%	20%
B	88%	12%
C	100%	0%
D	72%	28%

Table 1: A, B, C, D Detection Rate

This is the chart analysis for the letters “A”, “B”, “C”, and “D”. For the letter “A”, its detection rate is 80%. The other 20% is because this application has mistaken this letter with the letters “E”, “M”, “N” and “S”. The percentage of detection rate for the letter “B” is 88% and another 12%, this application sometimes mistook this letter with the letters “U” and “W”. Letter “C” has the perfect detection rate which is 100% and the letter “D” has 72% of detection rate. Another 28% is when this application has mistaken this letter with the letter’s “K”, “R” and “Z”.

ALPHABET	Prediction detection rate %	Not Detect Rate %
E	100%	0%
F	100%	0%
G	64%	36%
H	88%	12%

Table 2: E, F, G, H Detection Rate

Next is the chart analysis for the letters “E”, “F”, “G” and “H”. The letters “E” and “F” has 100% detection rate. While, the letter “G” has detection rate of 64% and the other 36% is because the application has mistaken this letter with the letters “H”, “P” and “Q”. The percentage of the detection rate for the letter “H” is 88%. Another 12% is when this application mistook this letter with the letter “G”.

ALPHABET	Prediction detection rate %	Not Detect Rate %
I	100%	0%
J	60%	40%
K	84%	16%
L	100%	0%

Table 3: I, J, K, L Detection Rate

The third chart analysis is for the letters “I”, “J”, “K” and “L”. The letters “I” and “L” has 100% of the detection rate. The detection rate for the letter “J” is 60% and another 40%, this application has mistaken this letter with the letters “A”, “M” and “T”. Meanwhile, the percentage for detection rate of the letter “K” is 84% with another 16% is because the application mistook this letter with the letters “L”, “V” and “W”.

ALPHABET	Prediction detection rate %	Not Detect Rate %
M	100%	0%
N	68%	32%
O	100%	0%
P	88%	12%

Table 4: M, N, O, P Detection Rate

Next is the fourth chart analysis for the letters “M”, “N”, “O” and “P”. The percentage of the detection rate for the letters “M” and “O” are 100% while the percentage for the letter “N” is 68% with another 32% is when the application has mistaken this letter with the letters “A”, “E”, “M” and “S”. For letter “P”, its percentage of detection rate is 88% with another 12% is because the application mistook this letter with the letter “M”.

ALPHABET	Prediction detection rate %	Not Detect Rate %
Q	76%	24%
R	100%	0%
S	56%	44%
T	100%	0%

Table 5: Q, R, S, T Detection Rate

This is the chart analysis for the letters “Q”, “R”, “S”, and “T”. The percentage of the detection rate for the letters “R” and “T” is 100%. The percentage of the detection rate for the letter “Q” is 76% with another 24% is because this application has mistaken this letter with the letters “G”, “H”, and “P” while the percentage of the detection rate for the letter “S” is 56% with another 44% is because this application mistook this letter with the letters “A”, “E” and “T”.

ALPHABET	Prediction detection rate %	Not Detect Rate %
U	92%	8%
V	100%	0%
W	100%	0%
X	100%	0%
Y	100%	0%
Z	92%	8%

Table 6: U, V, W, X, Y, Z Detection Rate

The last chart analysis is for the letters “U”, “V”, “W”, “X”, “Y”, and “Z”. The percentage of the detection rate for the letters “V”, “W”, “X” and “Y” are 100%. The percentage for the letter “U” is 92% with another 8% is because this application has mistaken this letter with the letters “D” and “Z” while the percentage of the detection rate for the letter “Z” is 92% with another 8% is when this application mistook this letter with the letters “D” and “U”.

C. *User Testing*

This application has been tested by the selected 25 respondents. These respondents consisted of people who might deal with disabled people (deaf and mute) such as receptionist, regular netizens, and people who daily works with disabled people such as at the OKU welfare center. A user testing form was distributed to survey the application feedback regarding its features and the level of its usability towards them. The result from the survey is shown using a pie chart.

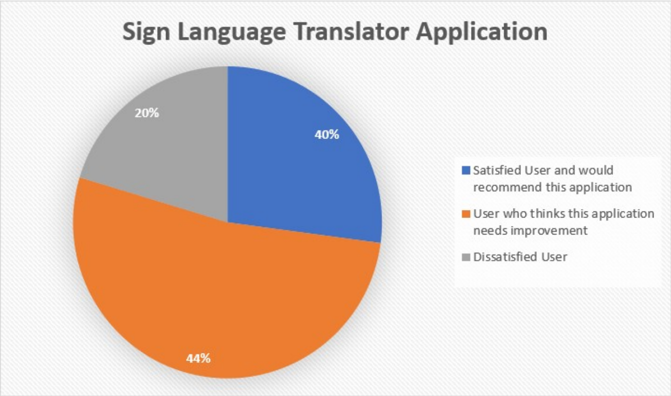

Figure 3: User Testing Result

The results of the survey are divided into three parts which are satisfied user and would recommend this application to others, user who thinks this application needs improvement and dissatisfied user. The percentage for the satisfied users and would recommend this application to others is 40%. This means that there were 10 out of the 25 respondents would recommend this application. Meanwhile, the biggest percentage with 44% which also means that 11 out of the 25 respondents think that this application needs to improve. Lastly, the dissatisfied users hold the least percentage which is 20%. This means that only 5 out of the 25 respondents did not like the application and would not recommend this application to others.

D. *Summary*

Based on the result, it can be concluded that this application can function properly in translating all 26 alphabets. Half of the total alphabets has the perfect detection rate which is 100%. The alphabets are “C”, “E”, “F”, “I”, “L”, “M”, “O”, “R”, “T”, “V”, “W”, “X”, and “Y”. Meanwhile, the alphabet “S” has the lowest rate of detection which is 56%. The overall success rate for this Sign Language Translator Application is 50% as it can translate half of the total alphabets. By looking at the result analysis, this application is frequently confused the alphabets that have similar gesture. This application needs to be train and focus more on detecting the key points of the hand of the alphabets that have the similar gesture. Next is the result of the user testing. Majority of the 25 respondents agreed that this application needs some improvement. From the feedbacks of the user testing form, the majority respondents were disagreed that this application translating process is going smoothly and efficiently although it has achieved its purpose which is to translate the sign language. In comparison between the application result analysis and the user testing form results, this application can function properly and able to translate all the alphabets even though it is not perfect. As the conclusion, it can be said that this application is able to work and run. It has attained its goals as mentioned in the requirements but somehow it still needs to be improved in the future.

V. Conclusion

A. *Overview*

This chapter discusses about the conclusion of the development of this sign language translator application and its future recommendation. The conclusion is made based on the requirements that have been stated earlier. Any improvements that can be made to make this application better and more convenience are also provided in this section.

B. *Conclusion*

The purpose of this sign language translator application is to translate between the spoken language and sign language. It uses the algorithms in OpenCV and is generated using Android Studio. This application captures a visual data of the sign language and it then processes and translates it into a recognizable form of text output. Developing this application is beneficial for people who just started to learn the sign language. With some addition of extras features, this application should be able to help people who needs to communicate with disabled people. As for the testing result of this application, it was able to translate most of the alphabets. But there was also error such as some alphabets was mistaken with another alphabets. For example, this application mistook the alphabet “S” with “M” as they both have almost similar gesture. There were also alphabets that cannot be translated by this application such as “J” and “Z” because these alphabets are a moving gesture. In conclusion, even though this basic sign language translator application fails in translating all the 26 alphabets, but it is still useful as it can translate majority of the alphabets. With further modifications and improvements, this application should be able to provide convenience in communication between normal people and disabled people.

C. Future Recommendations

There are some recommendations to enhance this application’s features so it can function efficiently and provide the convenience for normal people to communicate with disabled people. Firstly, this application can be programmed to translate a few more basic gestures or words of the sign language as for now, this application was only programmed to translate the 26 alphabets. Next, this application should be upgrade so it can capture the motion of gestures of the sign language. For now, this application can only translate static figures, so it takes some time to translate the alphabets. Lastly, the availability of this application could be widened and not limited to Android only but to other platforms too so more people could benefit from this application.

REFERENCE
S

[1] Bautista, J. M. P., Custodio, D. L., Lagundino, L. R., & Manaig, K. A. (2015). PSYCHOLOGY AND SOCIAL SCIENCE RESEARCH Open Access Presented in 2 nd International Conference in Education , Psychology , and Social Science (ICEPSS) International Research Enthusiast Society Inc . (IRES Inc .) Barriers , Strengths And AcademicPerform.

[2] Rao, G. A., & Kishore, P. V. V. (2016). Sign language recognition system simulated for video captured with smart phone front camera. *International Journal of Electrical and Computer Engineering*, *6*(5), 2176–2187. https://doi.org/10.11591/ijece.v6i5.11384

[3] Tiwari, V. M. (2017). University of Manchester Final year project Android Application for Sign Language Recognition.

[4] Jin, C. M., Omar, Z., & Jaward, M. H. (2016). A mobile application of American sign language translation via image processing algorithms. *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, 104– 109. https://doi.org/10.1109/TENCONSpring.2016.7519386

[5] Memon, Z. A., Ahmed, M. U., Hussain, S. T., Baig, Z. A., & Aziz, U. (2017). Real Time Translator for Sign Languages. *Proceedings - 2017 International Conference on Frontiers of Information Technology, FIT 2017*, *2017-January* (December), 144–148.https://doi.org/10.1109/FIT.2017.00033

[6] Triyono, L., Pratisto, E. H., Bawono, S. A. T., Purnomo, F. A., Yudhanto, Y., & Raharjo, B. (2018). Sign Language Translator Application Using OpenCV. *IOP Conference Series: Materials Science and Engineering*, *333*(1). https://doi.org/10.1088/1757-899X/333/1/012109

[7] V. Purushotham Vijay Naidu, M.R.Sai Hitesh & T.Dhikhi (2017). SOFTWARE ASSISTANCE TO DEAF AND DUMB USING HANDSHAPE ALGORITHM

[8] [8] Park, E., & Frogoso, R. (2016). *Senior Design I Project Specification: American Sign Language Translator using Gesture Recognition Fall 2016 By:*