UMT PRESS
Journal of Mathematical Sciences and Informatics
Journal Homepage: https://journal.umt.edu.my/index.php/jmsi
eISSN: 2948-3697
DOI: http://doi.org/10.46754/jmsi.2024.10.002

STATIC SIGN LANGUAGE TRANSLATOR USING HAND GESTURE AND SPEECH RECOGNITION

EKO BUDI SETIAWAN*, AGUS DARMAWAN AND BUDI HERDIANA

Faculty of Engineering and Computer Science, Universitas Komputer Indonesia, Bandung, 40132 West Java, Indonesia.

**Corresponding author: eko@email.unikom.ac.id*

ARTICLE INFO

Article History:
Received 5 June 2024
Accepted 9 September 2024
Published 20 October 2024

Keywords:
Sign language;
hand gesture;
image recognition;
speech recognition;
MediaPipe.

ABSTRACT

Communication between ordinary and deaf people often has issues because ordinary people lack knowledge of sign language. This research aims to help ordinary people communicate with hearing impaired (deaf) people using sign language. This research aims to produce an Android-based mobile application that can translate static sign language using hand movements into text and also convert the spoken voice into sign language using speech recognition. The framework in this research for hand gesture detection uses the MediaPipe software program. This framework allows the creation of applications that translate hand movements into text that help ordinary people understand sign language. Speech recognition in this research uses the Android speech library. This research succeeded in detecting static letters of the alphabet from A to Z and numbers 0 to 9. Tests of 540 hand gestures carried out in the morning, afternoon, evening, and night had an average detection time of 4.37 seconds. The fastest object detection times were in the morning at a distance of 30 cm with an average detection time of 2.5 seconds. Based on acceptance testing, 83.13% of the features in this static sign language translator have met the users' needs when communicating with the deaf using sign language.

2020 Mathematics Subject Classification:

Introduction

Human beings interact or communicate in two ways. The first is via verbal communication with words and the second method of communication is without speech. The second form of communication also called non-verbal communication involving many modalities, including body, touch [1], eye movements, monitoring, and facial expressions in the interaction process to provide information [2]. This form of communication is usually used by people with hearing impairments (deaf). Communicating with deaf people is often problematic due to the general public's lack of knowledge about sign language [3].

SIBI and BISINDO are two sign languages widely used in Indonesia [4]. *Sistem Bahasa Isyarat* (SIBI) is an official form of sign language authorised by the Indonesian Education and Culture Ministry [5]. Meanwhile, *Bahasa Isyarat Indonesia* (BISINDO) is a widely-used means of communication used by both deaf and ordinary people to "talk" to each other in the course when carrying out their daily activities [6].

A sign language translator is needed to help ordinary people communicate with the hearing impaired [7]. However, the reality is that a translator is not always required for every communication with the deaf [8]. A translator that can be used at any time is needed to translate sign language into a language that listening friends can easily understand. A good sign language translator should therefore be able to identify a specific set of gestures using gesture recognition and convert it into text or speech [9].

Gesture recognition is a topic in computer science and language technology that makes computers understand human movement. These movements generally are made by the hands or face. Gesture recognition allows humans and computers to interact naturally without the need for a specific input device. This manner of communication can be achieved using computer vision and image processing techniques that will convert hand gestures into their text or voice equivalents [10].

Gestures are considered a natural means of communication between humans, especially between those suffering from hearing loss. Gestures can be either hand or body movements that convey messages or contain meaningful commands. Among the examples of implementing this gesture is sign language recognition [11], human-computer interaction [12], virtual reality [13], and lie detection [14].

This research aims to help ordinary people communicate via sign language with the hearing impaired (deaf). This research aims to produce a sign language translator for Android mobile phones using the MediaPipe framework. MediaPipe is a framework for building pipelines and will include incoming data arbitrarily and is designed for those who want to incorporate artificial intelligence elements into their applications [15]. MediaPipe allows the development of cross-platform applications that can be run on various devices [16] and that can be used to create a sign language translation application [17].

Materials and Methods

Hand gesture recognition software is a form of technology that can read hand gestures and convert them into their text or voice equivalents [18]. Gesture recognition is a topic in computer science and language technology that aims to make computers understand human movements that generally come from the hands or face [19]. Technology that can analyse gestures, among others [20]:

Vision-based

Vision-based computer input requires a camera to take images to be processed. Vision-based computer input is challenging to execute. Challenges include factors like lighting and the colour of the detected object especially where it is the same colour as its surroundings, and challenges from the background images that conflict with the image to be detected. Vision-based computer input requires speedy, accurate, and efficient detection systems.

Glove-based

The glove-based computer input methods require sensors to capture hand coordinates and their displacement. The benefit of using this method is that it makes it easier to receive information such as palm coordinates, fingers, and orientation. The disadvantages of this method is that it requires a direct connection between the user and the system and the cost of making the tool quite expensive.

Colour Marker

Colour markers need gloves with several colours to direct the tracking process. This method also provides the ability to shape hands by extracting geometric features. The application of hand gesture recognition is for sign language translators, robotics, virtual reality (VR) games, and human-computer interaction [21].

MediaPipe is a framework written using C++ and can build applications that can cross between platforms like Windows, Linux, Macintosh, and Android and can even run on websites. MediaPipe can even handle data image processing using multithreading and GPU acceleration [22].

MediaPipe has a calculator that will run when the program starts and will stop when the program is finished. This calculator is critical because MediaPipe uses multithreading. The calculator in MediaPipe has four important processes, which is GetContract (), Open (), Process (), and Close (). Google has been using MediaPipe since 2012 for its products and made it an open-source program at the Conference on Computer Vision and Pattern Recognition in June 2019. Google uses MediaPipe on YouTube to define several characteristics in videos. These characteristics include determining video thumbnails, identifying video copyright, video objects that are not permitted like porn, violence, and advertisements.

The Google team collected all the datasets by hand and then labelled each landmark manually. Landmarks in MediaPipe can be observed in Figure 1.

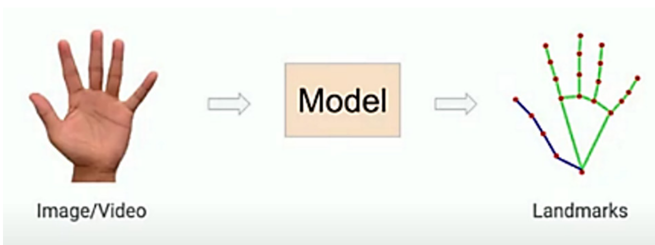


Figure 1: Landmarks in MediaPipe

Apart from detecting hands, MediaPipe also provides other machine learning solutions that are ready to use. Almost all code collection can be implemented to create Android-based programs, including:

1. Face detection, which is used to detect faces with six landmarks (signs) and allow multiple faces.
2. Face mesh detection, which is used to detect facial geometry in real-time and generate 468 facial landmarks (marks) in 3D.
3. Iris is used to track landmarks, including the iris, pupil, and eye contours.
4. Hands are used to trace hands and form 21 3D hand landmarks.
5. Pose, used to trace body poses and form 33 landmarks for the entire body and 25 for the upper body.
6. Other machine learning solutions include instant motion tracking, objection, hair segmentation, box tracking, object detection, and knife.

Application Programming Interface (API) is a system that integrates information from different applications simultaneously and runs behind the scenes [23]. API is a virtual interface between two software functions that work with each other. API describes how a programmer uses a feature of the computer. Initially, API technology provides system modifiability and makes materials databases accessible and interoperable [24] to drive the exchange of resources data between several applications [25].

Voice is the most basic general and efficient communication method for conveying ideas between people. Currently, speech recognition technology is widely available to help humans work. This technology allows computers or machines to understand human speech. The use of voice is preferred because it is faster than using a keyboard when communicating with a computer [26].

For computers to recognise and understand human voices, speech-to-text technology is needed. This technology allows computers to recognise sounds and translate them into text for later processing according to user needs. Among the methods that can be used to make computers understand human voices [27] include Support Vector Machines, (SVM) [28], Artificial Neural Networks [29], Naive Bayes Classifications [30], or all three methods SVM, Artificial Neural Network, and Naive Bayes Classifications [31] and K-Nearest Neighbour Classifier [32]. Many studies have used text to speech technology including ERICA [33], smart home [34], people with hearing problems (deaf) [35], people with vision problems (blind) [36, 37], and people who are both deaf and blind [38].

Flowchart Hand Gesture Recognition

A flowchart explains hand detection. The flowchart also checks palm gestures and gestures involving the back of the hand to detect character. This hand gesture recognition flowchart can be seen in Figure 2.

Below is a more detailed explanation of Figure 2.

1. The device opens the camera.
2. The device takes an image of the hand.
3. The palm detection system uses a Palm Detection model.
4. If a palm is found, the system will crop the image (box size) to the part of the palm that was successfully detected.
5. The cropped hand image is sent to the Landmark Detection model, which can detect 21 points on a hand.
6. The hand image is matched against the 21-point Palm Detection model.
7. At the same time, the system will match the 21-point with letters and numbers in the SIBI database.
8. The system will also check whether the detected image is the palm or the back of the hand.
9. Character detection for the palm and the back of hand can be see in the subchapters: Flowchart Checks Palm Gestures and Flowchart Checks the Back of the Hand.
10. After successfully converting 21 dots to letters or numbers, the system will combine the entire hand, 21 hand points, and gestures onto the screen.

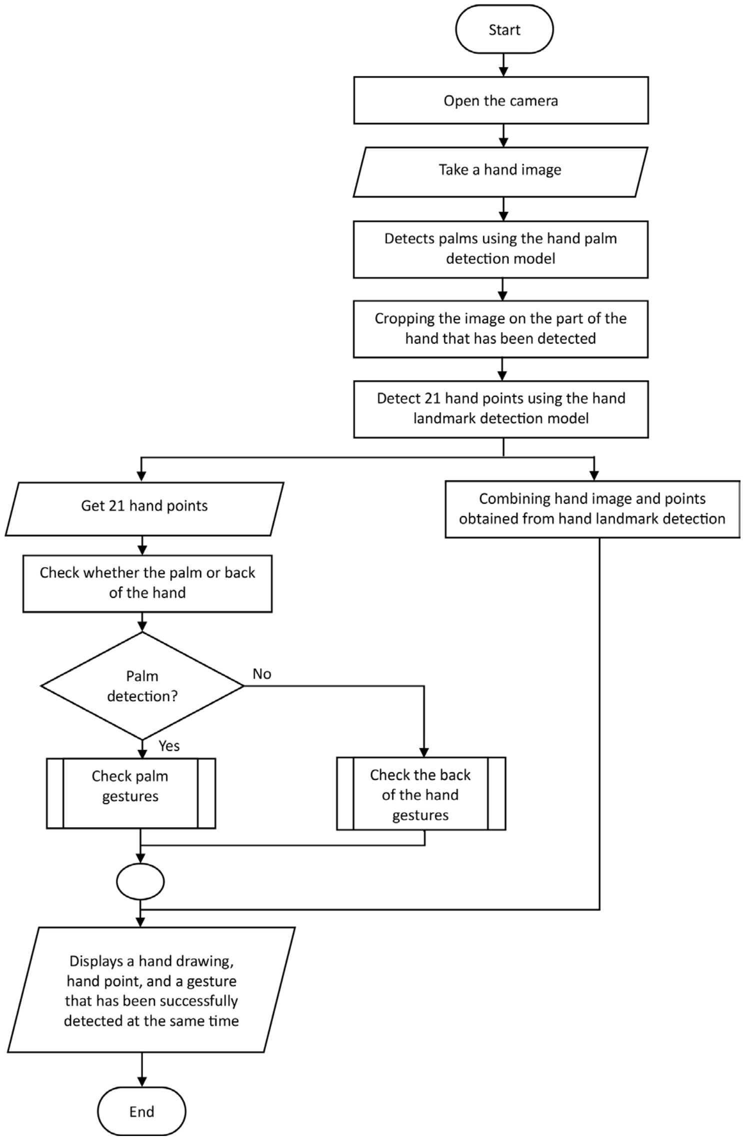


Figure 2: Flowchart of hand gesture recognition

To get the coordinates of 21 hand points, MediaPipe uses a second-hand landmark detection model. The output of this model is 21 hand points (x and y), markers of the possible presence of hands on the detected image, and right and left-hand classification. MediaPipe uses original images to provide markers/landmarks.

After landmark detection has succeeded in getting 21 hand points, mapping the points obtained on the hand image, these points consist of three points on the thumb, four points for each index finger, four points for the points for the ring finger, four points for the minor finger, and two points at the base of the palm.

Flowchart Checks Palm Gestures

This flowchart explains the flow of character detection for the palm. The results obtained from this flowchart cover the numbers 0 to 9 and some alphabets. Flowchart checks for palm gestures can be seen in Figure 3.

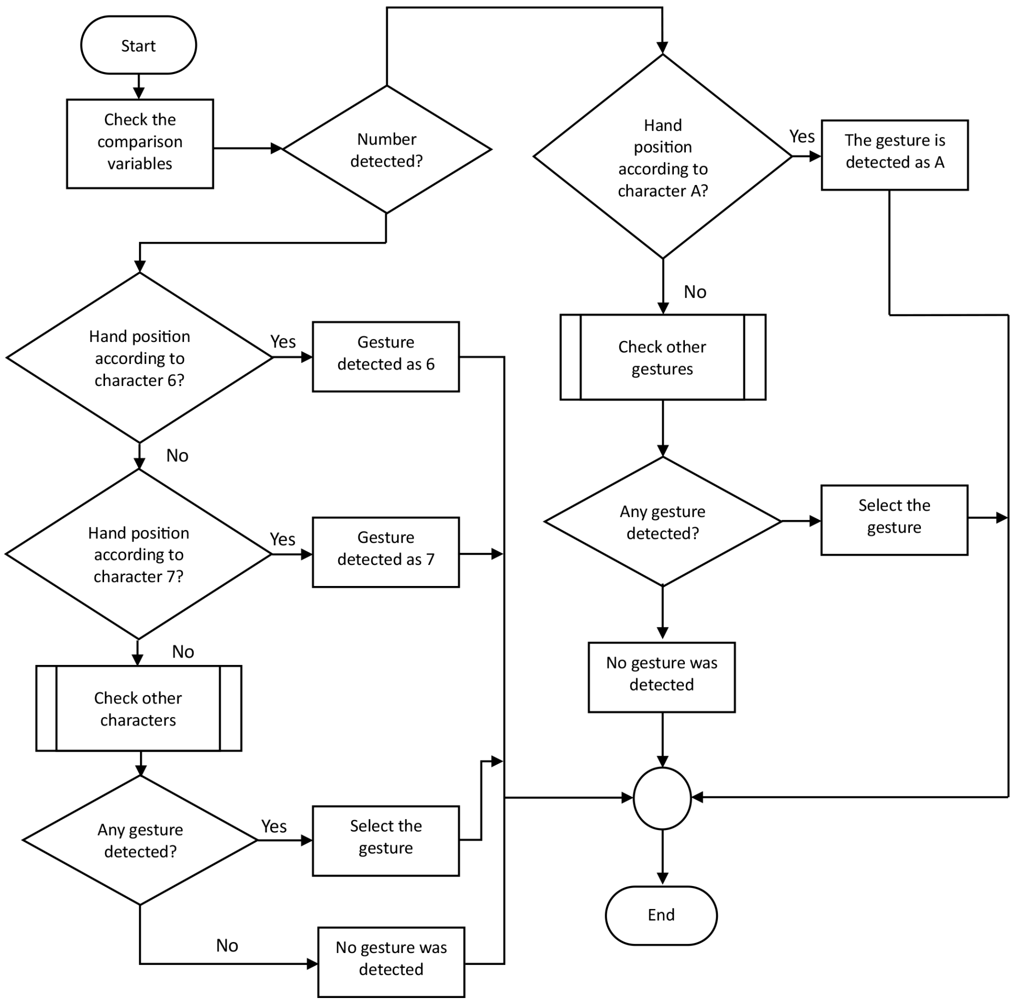


Figure 3: Flowchart checks palm gestures

MediaPipe's palm detection process uses a palm detection model [11]. When it first detects hand location, MediaPipe uses a single-shot detector model optimised for mobile devices in real-time. The single-shot detector helps score and box the detected palm image [19].

To overcome detection difficulties, MediaPipe uses several strategies, which is:

1. Using a palm detection model. The section of the palm detected is compared to hand detections as it can reduce detected areas. Besides the palms having fewer objects, the non-maximum suppression algorithm runs well even when there are obstacles such as shaking hands [11].
2. Using an extractor-encoder-decoder feature similar to that of a Feature Pyramid Network (FPN). An FPN is a system used to detect objects with different object sizes/scales [20].
3. Minimising focal point loss [21] when training data to optimise the image when scaling.

Flowchart Checks the Back of the Hand

This flowchart explains the flow of character detection using the back of the hand. The results obtained from this flowchart are several alphabets. Flowchart checks for the back of the hand can be observed in Figure 4.

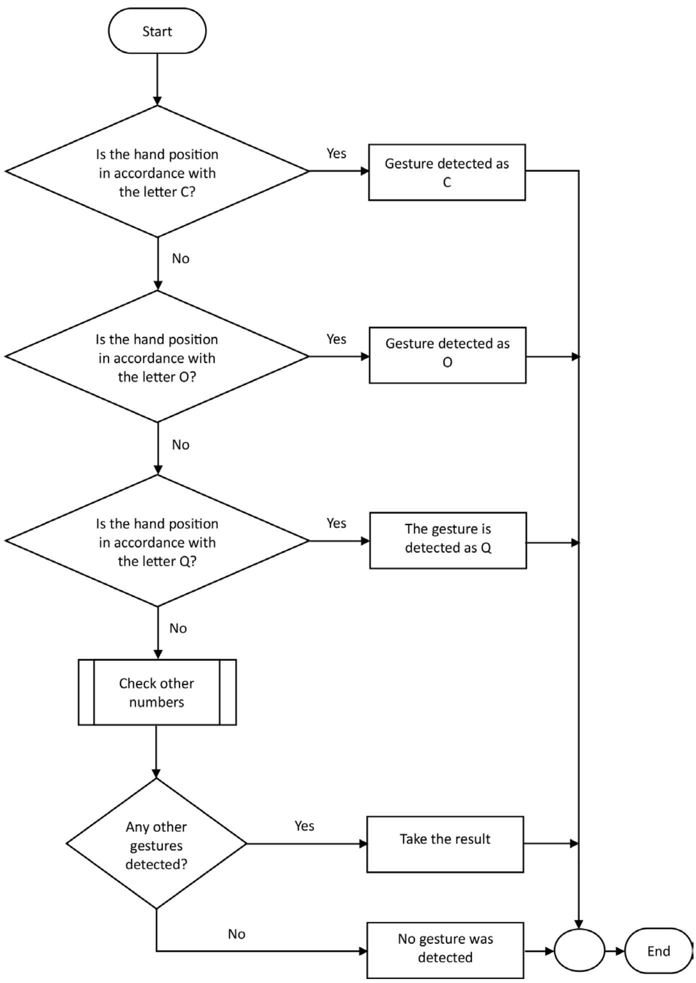


Figure 4: Flowchart checks the gesture of the back hand

Hand gesture recognition technology is used to translate hand gestures to text. Hand gesture recognition can be added to Android applications, one of which uses a framework from Google called MediaPipe. There are several stages to go through so that hand movement can be translated into text, including:

1. Take a picture using the camera.
2. Initialisation process. It is the process of reading the "palm_detection_labelmap.txt" file, loading the palm detection model stored in the "palm_detection_without_custom_op.tflite" file, and loading the landmark detection model stored in the "hand_landmark.tflite" file.
3. Detect palms using the palm detection model.
4. If the palm is found, it will cut the image around the palm to reduce the load on the system when processing the image.
5. Detecting 21 hand landmarks using landmark detection models. Hand landmarks can be observed in the discussion on hand modelling.
6. Checks whether the selected gesture type is letters or numbers. This process is used as the signal in the SIBI database for some letters and numbers is the same. For example, the number 4 and letter B, the number 1 and letter D, the number 2 and letter U, and the number 9 and letter F.
7. Checks whether gesture detected is the palm or the back of the hand. This process is used because the palm and the back of the hand produce different results.
8. Mapping the sign characters based on the points obtained from the landmark detection model.
9. Combine original images and hand landmarks to display on the screen. Then, displays the detected characters into the available text view.

Voice Detection Method

In Android Studio, there is a library that can translate voice into text, which is Android Speech. This library has been available since Android 1.5 with API level 3. There is a constant call RecogniserIntent.ACTION_RECOGNISE_SPEECH and EXTRA_LANGUAGE_MODEL. This constant will display a voice recording popup and send the detected sound to the speech recogniser. The received sound results can be processed in the OnActivityResult () activity in, getting sound detection results.

Results and Discussion

Figure 5 (a) gesture to text implementation can observe the type of gesture and delay provided. Gesture type is used to select the type of characters to be detected. The available character types are numbers and letters. At the same time, the delay is used to provide a delay in detection between one character and the next character. Examples of number detection can be seen in Figure 5 (b) and Figure 5 (c). Furthermore, examples of letter detection are shown in Figure 5 (d) and Figure 5 (e).

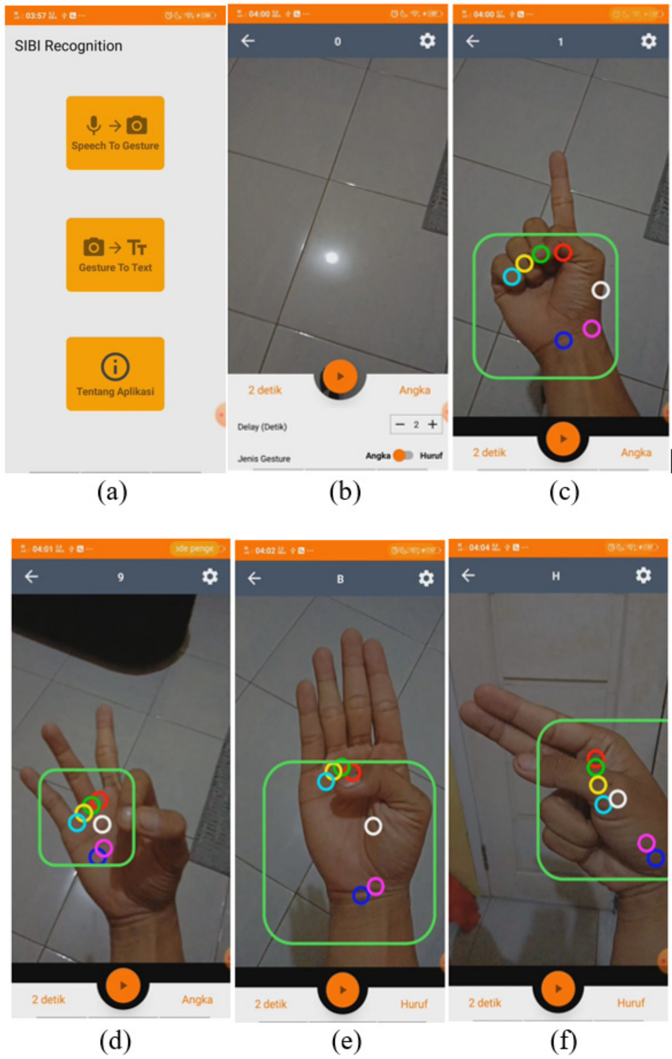


Figure 5: Gesture to text implementation

In Figure 5 (a), speech to gesture can be observed in several available fields and buttons. The text view “Enter a word or sentence” refers to entering the word or sentence you want to display the signal. This text view can enter manually, pasted from clipboard, or speech-to-text (by pressing the mic icon). Below the text view, there are two buttons for paste and delete. The paste button inserts text from the clipboard into the text view. The delete button is used to delete all text in the text view. There are two buttons with (–) and (+) icons. These two buttons are used to adjust the delay when displaying image signs. At the same time, the mic icon is used for text input from voice (speech-to-text) and will enter the text view. The last button is the run gesture button to display each character’s word or sentence’s gesture in the text view.

The characters whose signs that can be displayed can be seen in Figure 6 (b). Examples of displayed numbers can be seen in Figure 6 (c) and Figure 6 (d). Meanwhile, letter signs can be seen in Figure 6 (e) and Figure 6 (f).

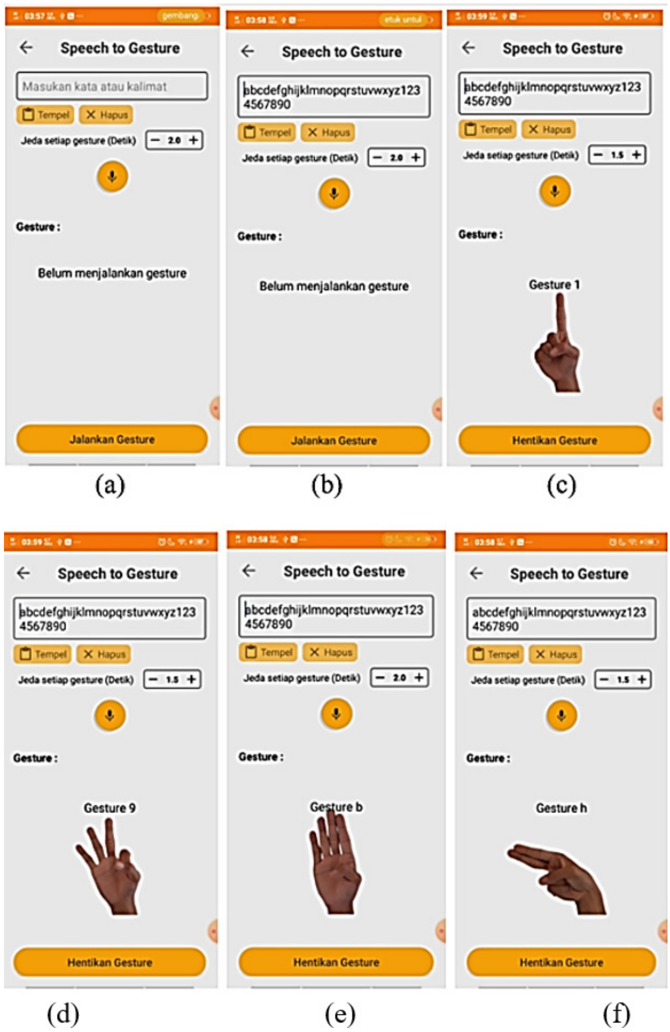


Figure 6: Speech to gesture implementation

Detection Performance Result

Application testing was carried out using a device with the specifications of the Vivo 1820 hp, Funtouch OS, Android 8.1.0 version, 2.0 GHz Octa-core processor, and 2 GB ram. The summary of detection by character is shown in Table 1.

Table 1: Test result of character SIBI detection

Characters	Number of Tests	Number of Detected	Number of Not Detected	Percentage of Success Detection	Percentage of Accuracy of Detection Results	Average Time Required to Detect (seconds)
A	15	15	0	100%	100%	6.87
B	15	15	0	100%	100%	2.07
C	15	15	0	100%	100%	4.40
D	15	15	0	100%	100%	2.47
E	15	15	0	100%	100%	4.67
F	15	15	0	100%	100%	2.53
G	15	15	0	100%	100%	4.60
H	15	15	0	100%	100%	3.60
I	15	15	0	100%	100%	3.47
J	-	-	-	-	-	-
K	15	15	0	100%	100%	3.07
L	15	15	0	100%	100%	2.07
M	15	15	0	100%	100%	10.47
N	-	-	-	-	-	-
O	15	15	0	100%	100%	2.27
P	15	15	0	100%	100%	6.93
Q	15	15	0	100%	100%	11.0
R	15	15	0	100%	100%	12.67
S	15	15	0	100%	100%	2.60
T	15	15	0	100%	100%	10.07
U	15	15	0	100%	100%	2.53
V	15	15	0	100%	100%	6.60
W	15	15	0	100%	100%	2.33
X	15	15	0	100%	100%	5.73
Y	15	15	0	100%	100%	2.13
Z	-	-	-	-	-	-
0	15	15	0	100%	100%	3.67
1	15	15	0	100%	100%	2.73
2	15	15	0	100%	100%	1.87
3	15	15	0	100%	100%	2.00
4	15	15	0	100%	100%	1.87
5	15	15	0	100%	100%	1.93
6	15	15	0	100%	100%	3.60
7	15	15	0	100%	100%	5.67
8	15	15	0	100%	100%	5.40
9	15	15	0	100%	100%	13.47

In the test of detection SIBI Character result, the application can translate 91.6% of the A to Z and 0 to 9 characters. Several characters were not successfully detected. These characters are J, N, and Z.

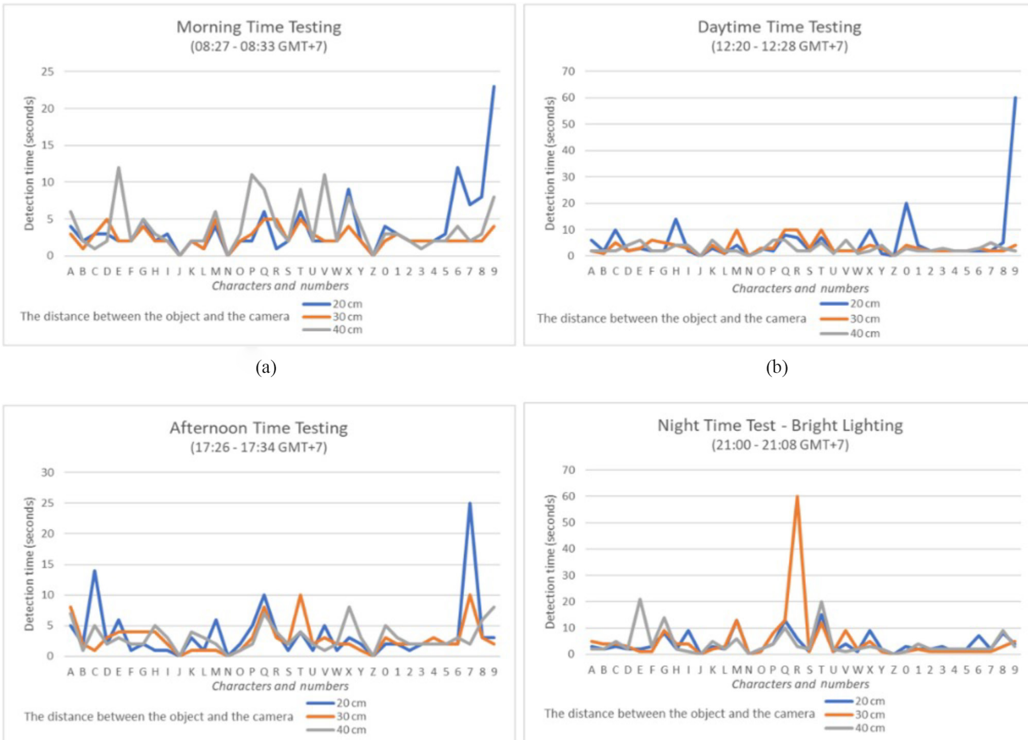
1. Letter J cannot be detected because the model of the MediaPipe does not provide a hands down pose. This causes the character J to be detected as a character I. Also, the character J is included in a dynamic gesture.
2. The letter N cannot be detected because the coordinates of the hand landmark S, M, and N are identical. This results in the character N being totally undetectable.
3. While the letter Z cannot be detected because it is a dynamic gesture. Further research is needed to detect dynamic hand movements.

The results based on the shooting time can be observed in Table 2.

Table 2: Detection based on distance and shooting time

Shooting Time	**Testing Time**	**Detection Distance**	**Lighting Condition**
Morning	08:31 – 08:33	40 cm	-
Morning	08:29 – 08:31	30 cm	-
Morning	08:27 – 08:28	20 cm	-
Day	12:26 – 12:28	40 cm	-
Day	12:24 – 12:25	30 cm	-
Day	12:20 – 12:23	20 cm	-
Afternoon	17:31 – 17:34	40 cm	-
Afternoon	17:29 – 17:30	30 cm	-
Afternoon	17:26 – 17:28	20 cm	-
Night	21:02 – 21:05	40 cm	Dim
Night	21:00 – 21:02	40 cm	Light
Night	20:55 – 20:59	30 cm	Dim
Night	20:53 – 20:55	30 cm	Light
Night	20:45 – 20:49	20 cm	Dim
Night	21:06 – 21:08	20 cm	Light

The graph in Figure 7 (a) shows the average time required to detect a sign; language character with a smartphone camera when tested in the morning was 3.75 seconds for a distance of 20 cm, 2.5 seconds for a distance of 30 cm, and 3.88 seconds for a distance of 40 cm. The sign language character for number 9 had the longest detection time, at 23 seconds.



(a) (b)

(c) (d)

Figure 7: Testing result diagram based on time of day and light conditions

The graph in Figure 7 (b) shows the average time required to detect a gesture character with a smartphone camera when tested in the daytime was 5.5 seconds for a distance of 20 cm, 3.44 seconds for a distance of 30 cm, and 2.83 seconds for a distance of 40 cm. The character for number 9 had the longest detection time, at 60 seconds.

The graph in Figure 7 (c) shows the average time required to detect a gesture character with a smartphone camera when the tested in the afternoon was 3.5 seconds from a distance of 20 cm, 2.86 seconds for a distance of 30 cm, and 3.05 seconds for a distance of 40 cm. The sign language character for number 7 had the longest detection time, at 25 seconds.

The graph in Figure 7 (d) shows the average time required to detect a sign language character with a smartphone camera when tested at night with bright lighting, which was 4 seconds at a distance of 20 cm, 5.08 seconds for a distance of 30 cm, and 4.02 seconds for a distance of 40 cm. The character for R had the longest detection time, at 60 seconds.

The all-graph diagram in Figure 7 can conclude that detection time is faster when done in the morning with a distance of 30 cm. The average time taken is 2.5 seconds. The longest detection time at night in the dark at a distance of 20 cm. The average time needed was 7.86 seconds. The most challenging character to detect was the character for R when tested at night with dark lighting and a distance of 20 cm. The time it took was 72 seconds.

Conclusions

This research has resulted in an Android-based mobile application that can assist ordinary able-bodied people communicate with deaf people. Based on the acceptance testing results for 30 users in Indonesia, 83.13% of the applications' features have met the users' needs. The speech to gesture feature in the sign language application can display all the user's alphabet and number hand gestures. The application can translate 91.6% of the alphabets and numbers. The average time for the 540 gesture tests performed in the morning, afternoon, evening, and night was 4.37 seconds. The fastest time for object detection was in the morning at a distance of 30 cm, with an average time of 2.5 seconds.

Acknowledgements

The authors wish to acknowledge Universitas Komputer Indonesia for funding this research.

Conflict of Interest Statement

The authors declare that they have no conflict of interest.

References

[1] Hall, J. A., Horgan, T. G., & Murphy, N. A. (2019). Nonverbal communication. *Annual Review of Psychology*, *70*, 271-294. https://doi.org/10.1146/annurev-psych-010418-103145

[2] Burgoon, L. K. G., Judee, K., & Valerie, M. (2021). *Nonverbal communication*. Routledge.

[3] Parera, C. H., Harlili, & Satya, D. P. (2019). Design of translator android application for sign language and Indonesian using user-centered design approach. In *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Yogyakarta, Indonesia, 2019,* (pp. 1-6). https://doi.org/10.1109/ICAICTA.2019.8904098

[4] Aditama, P. W., Putra, P. S. U., Yusa, I. M. M. & Putra, I. N. T. A. (2021). Designing augmented reality sibi sign language as a learning media. *Journal of Physics: Conference Series*, *1810*(1), 012038. https://doi.org/10.1088/1742-6596/1810/1/012038

[5] Rakun, E., & Darmana, I. S. A. (2020). Generating of SIBI animated gestures from Indonesian text. In *ICEMT 2020: 2020 The 4th International Conference on Education and Multimedia Technology*, pp. 256-264. https://doi.org/10.1145/3416797.3416822

[6] Indra, D., Madenda, S., & Wibowo, E. P. (2017). Recognition of Bisindo alphabets based on chain code contour and similarity of Euclidean distance. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(5), 1644-1652. https://doi.org/10.18517/ijaseit.7.5.2746

[7] Aljabar, A., & Suharjito (2020). BISINDO (Bahasa isyarat indonesia) sign language recognition using CNN and LSTM. *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, *5*(5), 282-287. https://doi.org/10.25046/AJ050535

[8] Yusnita, L., Rosalina, R., Roestam, R., & Wahyu, R. B. (2017). Implementation of real-time static hand gesture recognition using artificial neural network. *CommIT (Communication and Information Technology) Journal*, *11*(2), 85-91. https://doi.org/10.21512/commit.v11i2.2282

[9] Suchithra, M., Snehaa, S., Nambiar, G., Arya, C., & Diviya Bharrathi, L. B. (2015). Indian sign language recognition system for asymmetrical hand gestures. *International Journal of Applied Engineering Research (IJAER)*, *10*(55), 2628-2632.

[10] Raheja, J., Singhal, A., Sadab, N., & Chaudhary, A. (2015). Android based portable hand sign recognition system. In *Gate to Computer Science and Research* (pp. 1-18). https://doi.org/10.15579/gcsr.vol3.ch1

[11] Handhika, T., Zen, R. I. M., Murni, N., Lestari, D. P., & Sari, I. (2018). Gesture recognition for Indonesian Sign Language (BISINDO). *Journal of Physics: Conference Series*, *1028*(1), 012173. https://doi.org/10.1088/1742-6596/1028/1/012173

[12] Tsai, T., Huang, C., & Zhang, K. (2019). Design of hand gesture recognition system for human-computer interaction. *Multimedia Tools and Applications*, *79*(9-10), 5989-6007. https://doi.org/10.1007/s11042-019-08274-w

[13] Thao, N. D., Anh, N. V., Ha, L. T., & Duyen, N. T. (2021). Robustify hand tracking by fusing generative and discriminative methods. *VNU Journal of Science Computer Science and Communication Engineering*, *37*(1), 16-25. https://doi.org/10.25073/2588-1086/vnucsce.261

[14] Li, S., & Deng, W. (2020). Deep Facial Expression Recognition: A survey. *IEEE Transactions on Affective Computing*, *13*(3), 1195-1215. https://doi.org/10.1109/taffc.2020.2981446

[15] Halder, A., & Tayade, A. (2021). Realtime vernacular sign language recognition using MediaPipe and machine learning. *International Journal of Research Publication and Reviews*, *2*(5), 9-17. https://ijrpr.com/uploads/V2ISSUE5/IJRPR462.pdf

[16] Bazarevsky, V., & Zhang, F. (2019). On-Device, RealTime hand tracking with MediaPipe. *Google AI Blog*. https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/

[17] Subhalaxmi Chakraborty, N., Bandyopadhyay, P., Chakraverty, S. B., & Zinnia Sarkar, S. G. (2021). Indian Sign Language Classification (ISL) using Machine Learning. *American Journal of Electronics and Communication*, *1*(3), 17-21. https://doi.org/10.15864/ajec.1304

[18] Kumar, A., Raushan, R., Aditya, S., Jaiswal, V. K., & Divyashree, M. (2017). An innovative communication system for deaf, dumb and blind people. *International Journal Research Applied Science and Engineering Technology*, *5*(6), 1933-1939. https://www.ijraset.com/fileserve.php?FID=8560

[19] Abdallah, K. (2014). Gesture-based human computer interactions: Signs language recognition. *Open Science Repository*. https://doi.org/10.7392/openaccess.23050480

[20] Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, *6*(8), 73. https://doi.org/10.3390/jimaging6080073

[21] Cheok, M. J., Omar, Z., & Jaward, M. H. (2017). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, *10*(1), 131-153. https://doi.org/10.1007/s13042-017-0705-5

[22] Anilkumar, A., KT, A., Sajan, S., & KA, S. (2021). Pose estimated yoga monitoring system. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3882498

[23] Mi, W., Zhou, N., & Jin, Y. (2017). A real-time Cyber-Physical system for indoor environment monitoring. *Computer Science and Engineering*. https://doi.org/10.12783/dtcse/icmsie2016/6356

[24] Andersen, C. W., Armiento, R., Blokhin, E., Conduit, G. J., Dwaraknath, S., Evans, M. L., Fekete, Á., Gopakumar, A., Gražulis, S., Merkys, A., Mohamed, F., Oses, C., Pizzi, G., Rignanese, G., Scheidgen, M., Talirz, L., Toher, C., Winston, D., Aversa, R., . . . Yang, X. (2021). OPTIMADE, an API for exchanging materials data. *Scientific Data*, *8*(1), 217. https://doi.org/10.1038/s41597-021-00974-z

[25] Norton, B. (2021). APIs: A common interface for the Global Biodiversity Informatics community. *Biodiversity Information Science and Standards*, *5*, e75267. https://doi.org/10.3897/biss.5.75267

[26] Black, D., Rapos, E. J., & Stephan, M. (2019). Voice-driven modeling: Software modeling using automated speech recognition. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, *Munich*, *Germany*, *2019*, (pp. 252-258). https://doi.org/10.1109/models-c.2019.00040B

[27] Jolad, B., & Khanai, R. (2019). An art of speech recognition: A review. In *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, *Coimbatore*, *India*, *2019*, (pp. 31-35). https://doi.org/10.1109/icspc46172.2019.8976733

[28] Africa, A. D. M., Tabalan, A. R. V., & Tan, M. A. A. (2020). Speech emotion recognition using support vector machines. *International Journal of Emerging Trends in Engineering Research*, *8*(4), 1212-1216. https://doi.org/10.30534/ijeter/2020/43842020

[29] Kamble, B. C. (2016). Speech recognition using artificial neural network – A review. *International Journal of Computing*, *Communication and Instrumentation Engineering, 3*(1), 1-4. https://doi.org/10.15242/ijccie.u0116002

[30] Bhakre, S. K., & Bang, A. (2016). Emotion recognition on the basis of audio signal using Naive Bayes classifier. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, *Jaipur*, *India*, *2016* (pp. 2363-2367). https://doi.org/10.1109/ICACCI.2016.7732408

[31] Kose, M. R., Ahirwal, M. K., & Kumar, A. (2021). A new approach for emotions recognition through EOG and EMG signals. *Signal Image and Video Processing*, *15*(8), 1863-1871. https://doi.org/10.1007/s11760-021-01942-1

[32] Hidayat, R., Bejo, A., Sumaryono, S., & Winursito, A. (2018). Denoising speech for MFCC feature extraction using wavelet transformation in speech recognition system. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, *Bali*, *Indonesia*, *2018* (pp. 280-284). https://doi.org/10.1109/iciteed.2018.8534807

[33] Inoue, K., Milhorat, P., Lala, D., Zhao, T., & Kawahara, T. (2016). Talking with ERICA, an autonomous android. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 212-215. https://doi.org/10.18653/v1/w16-3625

[34] Agustin, E. I., Yunardi, R. T., & Firdaus, A. A. (2019). Voice recognition system for controlling electrical appliances in smart hospital room. *Telecommunication Computing Electronics and Control*, *17*(2), 965. https://doi.org/10.12928/TELKOMNIKA.V17I2.11781

[35] Meliones, A., & Duta, C. (2019). SeeSpeech: An android application for the hearing impaired. In *PETRA '19: Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. https://doi.org/10.1145/3316782.3324013

[36] Sagale, U., Bhutkar, G., Karad, M., & Jathar, N. (2018). An eye-free android application for visually impaired users. In Ray, G., Iqbal, R., Ganguli, A., Khanzode, V. (Eds.), Ergonomics in caring for people. Singapore: Springer. https://doi.org/10.1007/978-981-10-4980-4_36

[37] Barata, M., Salman, A. G., Faahakhododo, I., & Kanigoro, B. (2018). Android based voice assistant for blind people. *Library Hi Tech News*, *35*(6), 9-11. https://doi.org/10.1108/lhtn-11-2017-0083

[38] Landicho, J. A. (2016). Voisee Communicator: An android mobile application for hearing impaired and blind communications. *International Journal of Interactive Mobile Technologies* (iJIM), *10*(4), 26. https://doi.org/10.3991/ijim.v10i4.5859