information

MDPI

Article

Interpretation of Bahasa Isyarat Malaysia (BIM) Using SSD-MobileNet-V2 FPNLite and COCO mAP

Iffah Zulaikha Saiful Bahri [1], Sharifah Saon [1,*], Abd Kadir Mahamad [1,*], Khalid Isa [1], Umi Fadlilah [2], Mohd Anuaruddin Bin Ahmadon [3] and Shingo Yamaguchi [3]

[1] Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia; iffahsaiful98@gmail.com (I.Z.S.B.); halid@uthm.edu.my (K.I.)

[2] Teknik Elektro, Fakultas Teknik, Kampus 2, Universitas Muhammadiyah Surakarta (UMS), Jl. Ahmad Yani, Tromol Pos 1, Surakarta 57169, Jawa Tengah, Indonesia; umi.fadlilah@ums.ac.id

[3] Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Tokiwadai 2-16-1, Ube 755-8611, Japan; anuar@yamaguchi-u.ac.jp (M.A.B.A.); shingo@yamaguchi-u.ac.jp (S.Y.)

* Correspondence: sharifa@uthm.edu.my (S.S.); kadir@uthm.edu.my (A.K.M.)

Abstract: This research proposes a study on two-way communication between deaf/mute and normal people using an Android application. Despite advancements in technology, there is still a lack of mobile applications that facilitate two-way communication between deaf/mute and normal people, especially by using Bahasa Isyarat Malaysia (BIM). This project consists of three parts: First, we use BIM letters, which enables the recognition of BIM letters and BIM combined letters to form a word. In this part, a MobileNet pre-trained model is implemented to train the model with a total of 87,000 images for 29 classes, with a 10% test size and a 90% training size. The second part is BIM word hand gestures, which consists of five classes that are trained with the SSD-MobileNet-V2 FPNLite 320 × 320 pre-trained model with a speed of 22 s/frame rate and COCO mAP of 22.2, with a total of 500 images for all five classes and first-time training set to 2000 steps, while the second- and third-time training are set to 2500 steps. The third part is Android application development using Android Studio, which contains the features of the BIM letters and BIM word hand gestures, with the trained models converted into TensorFlow Lite. This feature also includes the conversion of speech to text, whereby this feature allows converting speech to text through the Android application. Thus, BIM letters obtain 99.75% accuracy after training the models, while BIM word hand gestures obtain 61.60% accuracy. The suggested system is validated as a result of these simulations and tests.

Keywords: Bahasa Isyarat Malaysia (BIM); SSD-MobileNet-V2 FPNLite; COCO mAP; TensorFlow Lite; Android application

check for updates

Citation: Saiful Bahri, I.Z.; Saon, S.; Mahamad, A.K.; Isa, K.; Fadlilah, U.; Ahmadon, M.A.B.; Yamaguchi, S. Interpretation of Bahasa Isyarat Malaysia (BIM) Using SSD-MobileNet-V2 FPNLite and COCO mAP. *Information* **2023**, *14*, 319. https://doi.org/10.3390/info14060319

Academic Editor: Danilo Avola

Received: 12 March 2023
Revised: 2 May 2023
Accepted: 12 May 2023
Published: 31 May 2023

1. Introduction

Every normal human being has been granted the most precious gift that cannot be replaced: the ability to express themselves by responding to the events occurring in their surroundings, where they observe, listen, and then react to circumstances through speech [1]. Unfortunately, there are those unfortunate ones who lack this precious gift. This creates a difference between normal human beings and disadvantaged ones, creating a massive gap between them [1–5]. Because communication is a necessary element of regular people's lives, deaf/mute individuals must communicate as normally as possible with others.

Communication is a tedious task for people who have hearing and speech impairments. Hand gestures, which involve the movement of hands, are used as sign language for natural communication between ordinary people and deaf people, which is just like speech for vocal people [4,6–9]. Nonetheless, sign languages differ by country and are used for a variety of purposes, including American Sign Language (ASL), British Sign Language (BSL), Japanese Sign Language [10,11], and Turkish Sign Language (TSL) [12]. This project focuses on Bahasa Isyarat Malaysia (BIM), also known as Malaysian Sign Language (MSL). BIM began its journey by forming a deaf school in Penang called the Federation School for the Deaf (FSD) in 1954. Studies have revealed that indigenous sign words arose through gestural communication amongst deaf students at the FSD outside their classroom. With the aim of educating deaf students, in 1964, American Sign Language (ASL) was made available in Johor, while Kod Tangan Bahasa Malaysia (KTBM) started to settle in Penang in 1978 when Total Communication was introduced into education for deaf students [13]. BIM has been the main form of communication amongst the deaf population in Malaysia for many years since it was first developed [14–16].

Communication is a vital aspect of everyday life; deaf/mute individuals must communicate as normally as possible with others [9]. The inability to speak is considered a problem amongst people [17] because they cannot clearly understand the words of normal people and, hence, cannot answer them [17]. This inability to express oneself verbally generates a significant disadvantage and, thus, a communication gap between the deaf/mute society and normal people [1,2,5,14]. The deaf/mute population or sign language speakers experience social aggregation challenges [4], and they constantly feel helpless because no one understands them and vice versa. This major humanitarian issue requires a specialised solution. Deaf/mute individuals face difficulties connecting with the community [3,18], particularly those who were denied the blessing of hearing prior to the development of spoken language and learning to read and write [3].

Usually, the ancient technique utilised for the deaf/mute to communicate with normal people is a human translator that can aid them with the discussion. However, it might be challenging due to the lack of a human translator [7], they might not always be accessible [19] for the deaf/mute, and paying for them can be expensive. It also makes such persons dependent on interpreters [2]. This procedure may also be relatively slow. It makes talking seem unnatural and boring between deaf/mute and normal people, which indirectly causes a lack of engagement in social activities [2]. Correspondingly, as previously stated, the deaf/mute use sign language to communicate with others and those who understand sign language. This causes a challenge if the deaf/mute are required to communicate with normal people, as they must be proficient in sign language, which only a minority of people learn and understand [19].

In addition, the issues of sign language are due to the substantial variation in gesture form and meaning amongst many cultures, situations, and people; gesture detection is a challenging undertaking. It is difficult to create precise and trustworthy models for gesture recognition because of this heterogeneity. Some of the most important elements that influence gesture recognition are (i) gestures can vary in terms of their speed, amplitude, duration, and spatial placement, which can make it challenging to consistently identify them [20], (ii) gestures can indicate a variety of things depending on the situation, the culture, and the individual's perception, (iii) different modes of interfering: speech, facial expressions, and other nonverbal clues can all be used in conjunction with gestures to affect how they are perceived [21], (iv) individual variations: different gesturing techniques can influence how accurately recognition models work [20], (v) the distinctions between spoken languages and sign languages present extra difficulties for sign language recognition [22], (vi) the unique grammar, syntax, and vocabulary of sign languages can make it difficult to effectively translate them into written or spoken language, and (vii) the difficulty of recognising sign languages can also be complicated by regional and cultural variances.

Undoubtedly, the advancement in technology, such as smartphones that can be used to make calls or send messages, has significantly improved people's quality of life. This includes the numerous assistive technologies available to the deaf/mute, such as speech-to-text and speech-to-visual technologies and sign language, which are portable and simple. Several applications are accessible to normal people; however, each has a set restriction today [16,23]. Additionally, there is a shortage of excellent smartphone translation programs that encourage sign language translation [14] between deaf/mute and normal people. Therefore, despite the tremendous benefits of cutting-edge technologies, deaf/mute and normal people cannot benefit from them. Unknowingly, Malaysians are unfamiliar with BIM, and present platforms for translating sign language are inefficient, highlighting the limited capability of the market's existing mobile translating application [16].

As previously said, the smartphone is a dependable technology for the deaf/mute to connect with normal people. Thus, this project intends to develop and build a two-way communication system for a Bahasa Isyarat Malaysia (BIM) application between deaf/mute and normal users, allowing both groups to engage freely. The deaf/mute community will benefit from the sign-language-to-text module, while the normal community will benefit from the speech-to-text module. This application will make it simpler for deaf/mute individuals to converse with the normal and vice versa. This can help to decrease the amount of time spent communicating. It will also be advantageous if the deaf/mute attend a meeting or a ceremony, where they can easily interpret the speech using the Android application without the assistance of a translator.

Today, there are many applications available for deaf/mute individuals to communicate with non-deaf/mute individuals. Despite all the benefits of state-of-the-art technology, each application has certain limitations [23,24], and there are fewer BIM mobile translation applications on the market [7]. This is because BIM is little known amongst Malaysians, and existing sign language translation platforms are inefficient, not to mention the incomplete functionality of existing mobile translation applications on the market [16]. Therefore, people with hearing and speech disabilities cannot fully benefit from them [24]. The main challenge in this study is the availability of a recognisable character database. Existing databases, especially BIM databases, are often provided without adequate standards for image resolution, structure, and compression that are good enough [25]. Hence, this project aims to reduce the communication gap between deaf and normal people by easily communicating using an Android application. This project can also eliminate the need to hire a human translator, thus, becoming substantially more cost-effective and simultaneously developing briefer and more interesting conversations. Finally, this app can also increase the utilisation of Bahasa Isyarat Malaysia, which boosts the acknowledgment of this language in Malaysian society.

2. Related Work

Bahasa Isyarat Malaysia (BIM), also known as Malaysian Sign Language (MSL), was initially developed in 1998, shortly after the Malaysian Federation of the Deaf was founded. This paper aims to create a mobile application that will bridge the communication gap between hearing people and the deaf–mute community by assisting the community in learning BIM.

In [14], a survey was conducted for possible consumers as its methodology. The target populations were University of Tenaga Nasional (UNITEN) students and Bahasa Isyarat Malaysia Facebook Group (BIMMFD) members. Multiple-choice, open-ended, and dichotomous items were included in the surveys. This research demonstrates that the software is thought to be helpful for society and suggests creating a more user-friendly and accessible way to study and communicate using this app utilising BIM.

The current state of the art with modern and more efficient gesture recognition methods has been discussed in several papers. In [26], the author introduced two deep-neural-network-based models: one for audio–visual speech recognition (AVSR) using the Lip Reading in the Wild Dataset (LRW) and one for gesture recognition using the Ankara University Turkish Sign Language Dataset (AUTSL). This paper uses both visual and acoustic features and fusion approaches, achieving 98.56% accuracy and demonstrating the possibility of recognizing speech and gestures using mobile devices. The authors of [27] worked on training models on datasets from different sign languages (Word-Level American Sign Language (WLASL), AUTSL, and Russian (RSL)) to improve sign recognition quality and demonstrate the possibility of real-time sign language recognition without using GPUs, with the potential to benefit speech- or hearing-impaired individuals, using VideoSWIN transformer and MViT. However, this paper focuses on the development of BIM letter and word recognition using SSD-MobileNet-V2 FPNLite and COCO mAP.

2.1. SSD-MobileNet-V2 FPNLite

SSD-MobileNet-V2 can recognise multiple items in a single image or frame. This model detects each image's position, producing the object's name and bounding boxes. Ninety different objects can be classified using the pre-trained SSD-Mobile model.

Due to the elimination of bounding box proposals, Single-Shot Multibox detector (SSD) models run faster than R-CNN models. The processing speed of detection and the model size were the deciding factors in the choice of the SSD-MobileNet-V2 model. As demonstrated in Table 1, the model requires input photos of 320 × 320 and detects objects and their locations in those images in 19 milliseconds, whereas other models require more time. For example, SSD-MobileNet-V1-COCO, the second-fastest model, takes 0.3 milliseconds to categorise objects in a picture compared to SSD-MobileNet-V2-COCO, the third-fastest model, and so on. Compared to the second-fastest model SSD-MobileNet-V1-COCO, SSD-MobileNet-V2 320 × 320 is the most recent MobileNet model for Single-Shot Multibox detection. It is optimised for speed at a very low cost, with a mean average precision (mAP) of only 0.8 [28].

Table 1. Model comparison [28].

Model Name	Speed (ms)	COCO mAP	TensorFlow Version
SSD-MobileNet-V2 320 × 320	19	20.2	2
SSD-MobileNet-V1-COCO	30	21	1
SSD-MobileNet-V2-COCO	31	22	1
Faster R-CNN ResNet50 V1 640 × 640	53	29.3	2
Faster RCNN Inception V2 COCO	58	28	1

2.2. TensorFlow Lite Object Detection

An open-source deep learning framework called TensorFlow Lite was created for devices with limited resources, such as mobile devices and Raspberry Pi modules. TensorFlow Lite enables the use of TensorFlow models on mobile, embedded, and Internet of Things (IoT) devices. It allows for low latency and compact binary size on-device machine learning inference. As a result, latency is increased and power consumption is decreased [28].

For edge-based machine learning, TensorFlow Lite was explicitly created. It enables us to use various resource-constrained edge devices, such as smartphones, micro-controllers, and other circuits, to perform multiple lightweight algorithms [29].

An open-source machine learning tool called TensorFlow Object Detection API is utilised in many different applications and has recently grown in popularity. When installing the TensorFlow Object Detection API, an implicit assumption is that it can be provided with noise-free or benign datasets. This open-source software is now being used in many object detection applications. However, in the real world, the datasets could contain inaccurate information due to noise, naturally occurring adversarial objects, adversarial tactics, and other flaws. Therefore, for the API to handle datasets from the real world, it needs to undergo thorough testing to increase its robustness and capabilities [30].

Another paper also defines TensorFlow Object Detection as a class of semantic things (such as people, buildings, or cars) that can be detected in digital photos and videos using object detection, a computer technology linked to computer vision, and image processing. The study areas for target detection include pedestrian and face detection.

Many computer vision applications require object detection, such as image retrieval and video surveillance. Applying this method to an edge device could let you perform a task, such as an autopilot [29].

2.3. MobileNets Architecture and Working Principle

Efficiency in deep learning is the key to designing or creating a helpful tool that is feasible to use with as little computation as possible. There are other ways or methods to solve efficiency issues in deep learning programming, and MobileNet is one of the approaches for said problem. MobileNets reduce the computation by factorising the convolutions. The architecture of MobileNets is primarily from depth-wise separable filters. MobileNets factorise a standard convolution into a depth-wise convolution and a 1 × 1 convolution (pointwise convolution) [31]. A standard convolution filters and combines inputs into a new set of outputs in one step. In contrast, depth-wise separable convolution splits the information into the filtering layer and the combining layer, decreasing the computation power and model size drastically.

2.4. Android Speech-to-Text API

Google Voice Recognition, or GVR, is a tool with an open API that converts the speech from the user to text to be read. GVR usually requires an internet connection from the user to the GVR server. GVR uses neural network algorithms to convert raw audio speech to text and works for several languages [32]. This tool has two-thread communication. The first thread is to receive the user's audio speech and send it to Google Cloud to be converted into text and stored as strings. After that, the other communication thread reads the strings, sends them to the server, and resides in the user workstation.

Google Cloud Speech-to-Text or Cloud Speech API is another tool for the speech-to-text feature. It has far more features than standard Google Speech API. For example, it has 30+ voices available in multiple languages and variants. However, this is not just a tool; it is a product made by Google, and the user needs to subscribe and send some fees to use this tool. Table 2 lists the advantages and disadvantages of these tools.

Table 2. Advantages and disadvantages of Google Cloud API and Android Speech-to-Text API.

	Advantages	**Disadvantages**
Google Cloud API	It supports 80 different languages.	Not free.
	Can recognise audio uploaded in the request. Returns text results in real time. Accurate in noisy environments. Works with apps across any device and platform.	Requires higher-performance hardware.
Android Speech-to-Text API	Free to use. Easy to use. It does not require high-performance hardware. Easy to develop.	Need to pass local language to convert speech to-text. Not all devices support offline speech input. It cannot pan an audio file to be recognised. It only works with Android phones.

3. Materials and Methods

This project includes three main categories: BIM letters, BIM words, hand gestures, and Android application development. These three main categories are divided into the database acquisition phase, the system's design phase, and the system's testing phase. The BIM sign language implemented utilises the static hand gesture, which only involves capturing a single image at the classifier's input.

3.1. BIM Letters

The first category, BIM letters, had three phases: the database acquisition phase, system's design phase, and system's testing phase. Phase 1: In the database acquisition phase, datasets were obtained from deaf/mute teacher datasets, Kaggle, and self-generated datasets. BIM datasets in Kaggle are limited; thus, ASL letters were implemented with a replacement of self-generated letters G and T. Phase 2: for the system's design phase, TensorFlow/Keras was implemented into the system as deep learning neural network to train the dataset. Lastly, Phase 3: the system's testing phase was tested to ensure the functionality was well executed by generating a confusion matrix table.

Collected data were processed for classification using the CNN model, in this case, MobileNet, and the datasets were trained by implementing 10% of the datasets for testing and 90% of the datasets for training. Once the result was obtained, the model was converted to TensorFlow Lite to be imported into Android Studio for application making. A flow process is shown in Figure 1.

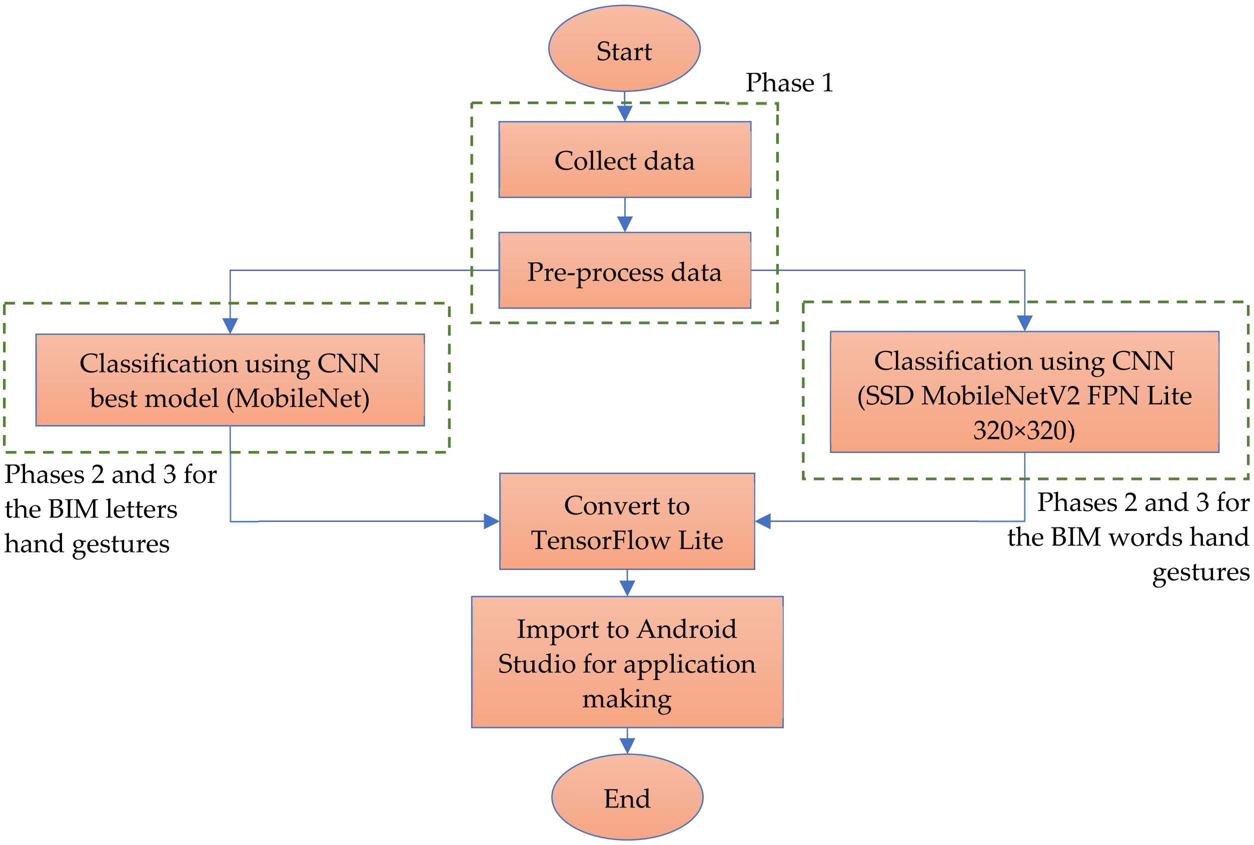
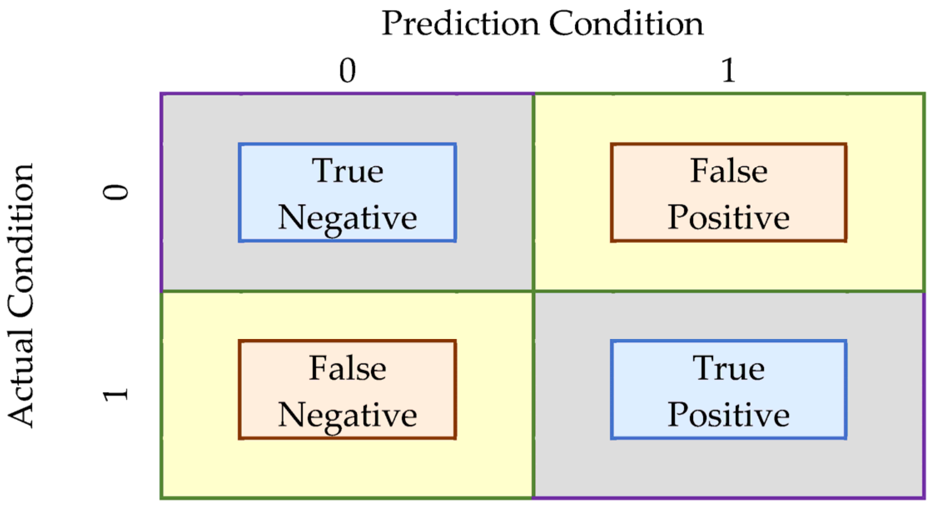

Start
Phase 1
Collect data
Pre-process data
Classification using CNN best model (MobileNet)
Phases 2 and 3 for the BIM letters hand gestures
Classification using CNN (SSD MobileNetV2 FPN Lite 320×320)
Phases 2 and 3 for the BIM words hand gestures
Convert to TensorFlow Lite
Import to Android Studio for application making
End

Figure 1. Flowchart of BIM letters and word hand gestures.

There are 29 letters in the datasets, including delete, nothing, and space, which is beneficial for real-time applications. The data collected from a total number of 3000 images in each class, comprising 87,000 images, were then resized to 200 px $\times$ 200 px before being provided as input because smaller images can allow training to be faster.

The system was tested to ensure its operation was executed effectively using a confusion matrix, as seen in Figure 2.

The confusion matrix consists of True Negative, True Positive, False Positive, and False Negative, and zero means false, while one means true. Therefore, there are two classes: (class 0) and (class 1). Thus, anything that the confusion matrix stated as zero or (class 0) is where the prediction is incorrect, such as True Negative, False Positive, and False Negative, whereas (class 1) is the number of samples that the model correctly classified as true, that is, True Positive.



		Prediction Condition	
		0	1
Actual Condition	0	True Negative	False Positive
	1	False Negative	True Positive

Figure 2. Confusion matrix.

3.2. *BIM Word Hand Gestures*

The dataset includes five classes, three of which are from family (*keluarga*) and contain the words brother (*abang*), father (*bapa*), and mother (*emak*); one from feelings (*perasaan*), which is love (*sayang*); and one from pronouns (*ganti nama*), which is I (*saya*). Data were gathered and processed to be classified using the CNN model. A pre-trained model from TensorFlow 2 Model Zoo was used to ensure that it achieved the best accuracy. This process includes changing the ratios, which are 25% for testing and 75% for training. The model was converted to TensorFlow Lite to construct apps and put into Android Studio. Database acquisition, system design, and system testing were the three steps that make up this category. A flow process of BIM words and hand gestures is shown in Figure 1.

The datasets were self-generated, in which 100 images for each class were used, and 500 images in total were collected, with a size of 512 px × 290 px. The images captured were based on different positions and light intensities, including the distance from the camera and the brightness. The pictures were also mirrored to acquire a variety of images. To differentiate the images, labelImg was downloaded and used. This software generates an XML file for each image labelled so it can be detected using TensorFlow Object Detection API. Figure 3 shows an example of selecting and labelling hand gestures for brother (*abang*) using labelImg software.

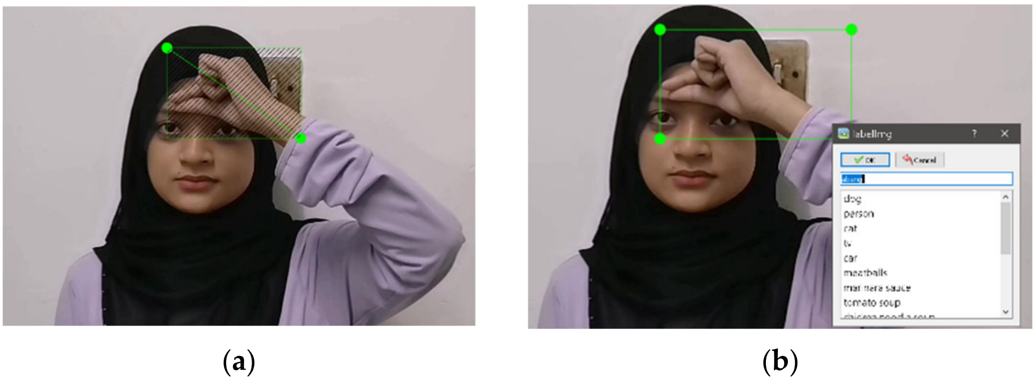


Figure 3. (**a**) Selecting and (**b**) labelling hand gestures of the class of *abang* on labelImg.

While the pre-processed datasets were classified using TensorFlow 2 Detection Model Zoo, the SSD-MobileNet-V2 FPNLite 320 × 320 model with a speed of 22 s/frame rate and COCO mAP of 22.2 was used to determine the model's accuracy before being transformed into TensorFlow Lite and exported to Android Studio. Then, a collection of

500 (512 px × 290 px) images was used, of which 25%, or 125 images, was utilised for testing and 75%, or 375 images, was used for training.

Once the training process was completed, the hand gesture was detected in real time using TensorFlow Object Detection API and TensorFlow 2 Detection Model Zoo, the SSD-MobileNet-V2 FPNLite 320 × 320 model with a speed of 22 s/frame rate and COCO mAP of 22.2 was used as the pre-trained model. This is because the pre-trained model was trained with a large dataset, and this saves much more time rather than creating a model. COCO is an extensive dataset for object identification, segmentation, and captioning. Therefore, since a larger COCO mAP is advised, other models may also be employed to recognise the objects correctly. TensorFlow Records (TFRecords) can be used, and these TFRecords are a binary file format for storing data. Using this helps speed up training for custom object detection, in this case, hand gestures. The model was trained three times, in which the number of steps was changed to 2000 steps and 2500 steps to evaluate the model's accuracy.

3.3. Android Application

This application has features of converting speech to text, converting BIM letter hand gestures that can form words, and converting BIM word hand gestures into text. The datasets for the BIM letter and word hand gestures were obtained by trained models that were converted into TensorFlow Lite. Android Studio was used to make the Android application. Users need to sign up and log in to the application to gain access to the feature. To ensure the system functions properly, the system was tested towards the objectives of this project. This, in turn, ensures the developer can improve the developed application. The flow process of the Android application of this BIM recognition is presented in Figure 4.

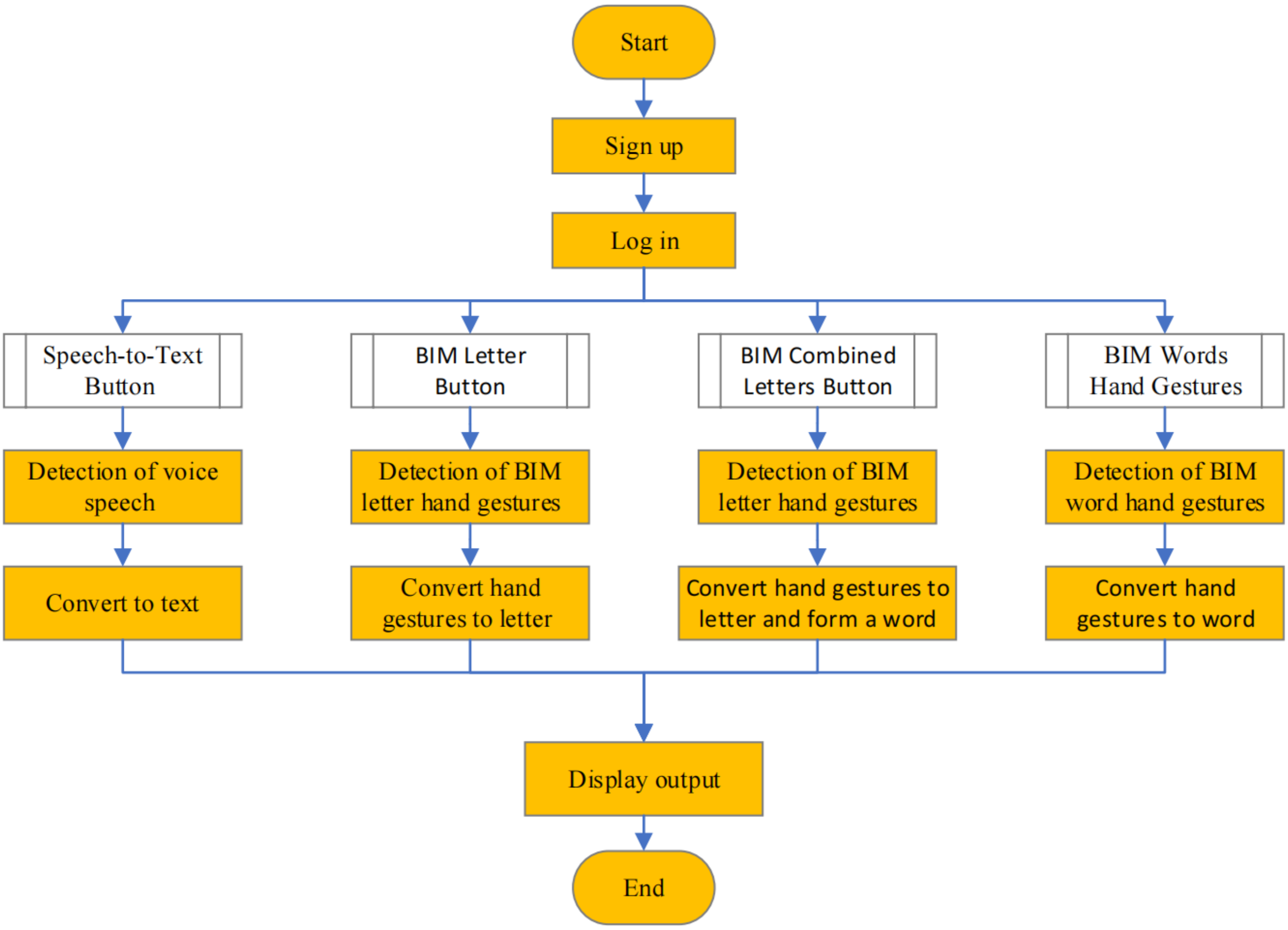

Start
Sign up
Log in
Speech-to-Text Button
BIM Letter Button
BIM Combined Letters Button
BIM Words Hand Gestures
Detection of voice speech
Detection of BIM letter hand gestures
Detection of BIM letter hand gestures
Detection of BIM word hand gestures
Convert to text
Convert hand gestures to letter
Convert hand gestures to letter and form a word
Convert hand gestures to word
Display output
End

Figure 4. Android application flowchart.

For this phase of developing the Android application, two files from the BIM letter and BIM word hand gestures were included. To acquire the mentioned files, a trained model of BIM letter and BIM word hand gestures was converted into TensorFlow Lite files and used for application making.

The BIM letters, BIM word hand gestures, and Android speech to text were developed using Android Studio. To enable real-time hand gesture detection in the application, the trained model of BIM letters and BIM word hand gestures are translated to TensorFlow Lite and imported into Android Studio. By importing the SpeechRecognizer class, which gives access to the speech recognition service, the application of speech-to-text capability can also be accomplished in Android Studio. The speech recogniser can be accessed through this service. This API's implementation involves sending audio to distant servers for speech recognition, such as converting microphone input to text.

In this project, the trained models are created with TensorFlow and converted into TensorFlow Lite format. Then, the converted format is then used to develop an Android app that analyses a live video stream and identifies things using a machine learning model; in this case, it analyses the BIM letters and BIM word hand gestures.

This machine learning model detects objects, which are BIM hand gestures, and it evaluates visual data in a prescribed manner to categorise components in the image as belonging to one of a set of recognised classes it was taught to identify. Milliseconds are frequently used to assess how long a model takes to recognise a known item (also known as object prediction or inference). In reality, the amount of data being processed, the size of the machine learning model, and the hardware hosting the model all affect how quickly inferences are made.

For the user's Android application, there are a few stages and features that need to be fulfilled by the user, such as:

1. The user needs to turn on the internet connection.
2. The user needs to download and install the app on their smartphone.
3. The user needs to register to the app if they are a first-time user (input name, email address, and password).
4. The user needs to log in as a user with their successfully registered account (input name and password).
5. The user must allow the app to use the camera and record audio.

4. Results

The implementation of the Android application that allows two-way communication between deaf/mute and normal people, which integrates Bahasa Isyarat Malaysia (BIM), consists of four main buttons that enable users to choose whether they want to use speech-to-text conversion, BIM letters to text conversion, BIM letters to create words conversion, and BIM word hand gestures to text conversion.

Three main categories help the application to be fully functional: BIM letters, BIM word hand gestures, and the development of the Android application itself. For BIM letters, the trained model achieved the highest accuracy of 99.78% by utilising the MobileNet pre-trained model with a 10% test size and a 90% training size. The result was evaluated by using a normalised confusion matrix. As for BIM word hand gestures, by implementing the TensorFlow 2 Detection Model Zoo, which uses SSD-MobileNet-V2 FPNLite 320 × 320, the average precision was 61.60% after being trained three times with 2000 steps and 2500 steps. Lastly, for the development of the Android application, '2 CUBE' is the name of the application, which means '2 ***C****ara* ***U****ntuk* ***BE****rkomunikasi dalam Bahasa Isyarat Malaysia*'. Furthermore, a feature of this application includes speech-to-text conversion, and the trained models of letters and BIM word hand gestures were converted to TensorFlow Lite, which can be implemented for real-time hand gesture detection.

4.1. BIM Letters

Using MobileNet pre-trained models, 29 BIM letters were trained and evaluated. Figure 5 displays a normalised confusion matrix for the trained model with the 10% test size and 90% training size. The diagonal elements represent the total correct values predicted for the classes based on the normalised confused matrix. The result demonstrates that the model accurately predicted all classes with a value of about 99 per cent.

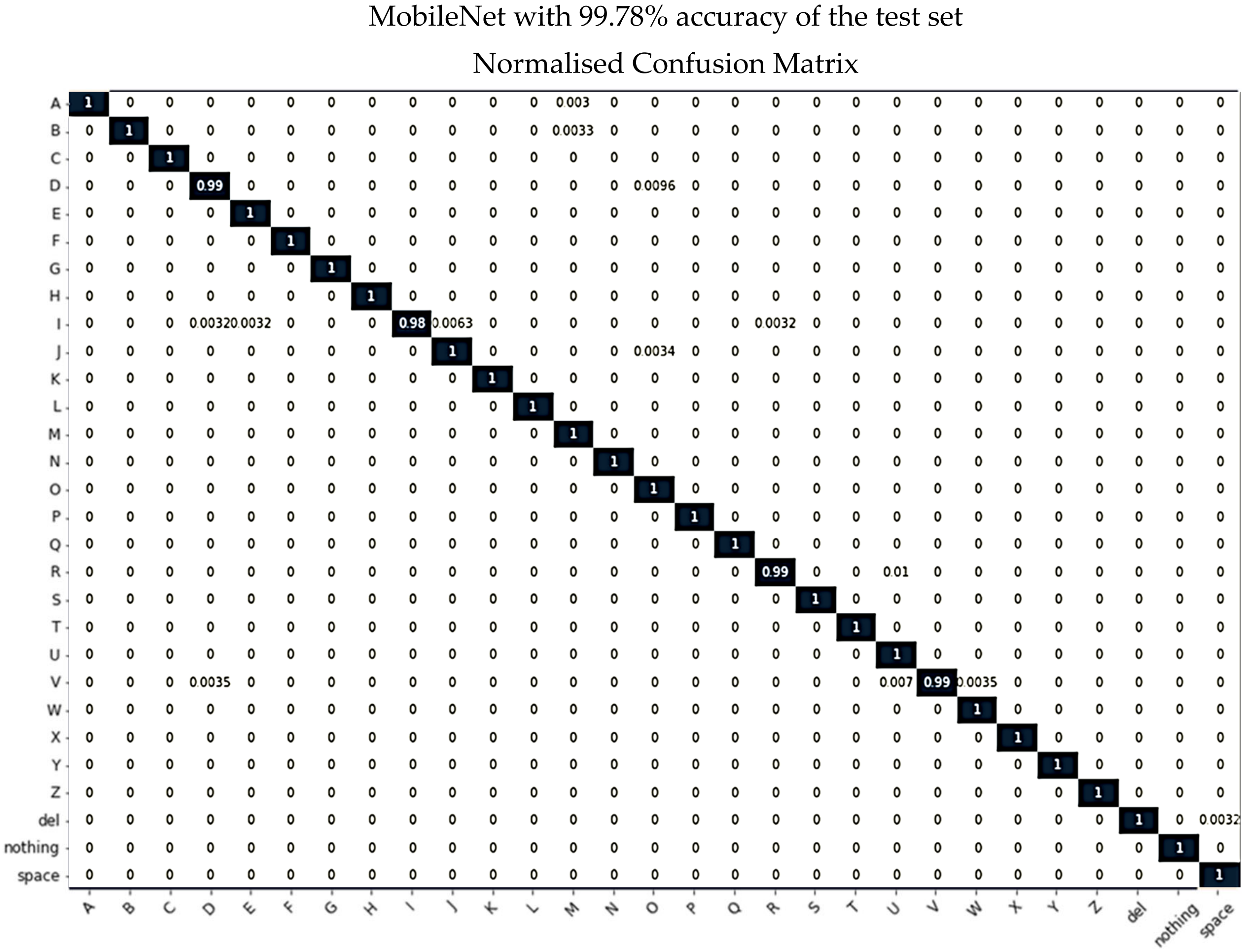


Figure 5. Normalised confusion matrix for the model with 10% test size and 90% training size.

4.2. BIM Word Hand Gestures

The training results of BIM words using hand gestures conducted using TensorBoard, as explained in Section 3.2, presents the loss, learning rate, and steps per second. The first-time training was set to 2000 steps, while the second- and third-time training were set to 2500 steps, and Figure 6 shows the training results of classification loss via TensorBoard, while Table 3 shows the training results of loss, learning rate, and steps per second.

For the evaluation result, this model obtained 0.616, which is 61.60% average precision (AP), with intersection over union (IoU) between 0.50 and 0.95 in all datasets with a maximum detection of 100. The precision is not that high since the datasets collected are in a small volume because the laptop capacity used for this project was low, and it required a lot of time running on CPU instead of GPU. Other than that, the classes for the hand gestures of father (*bapa*), mother (*emak*), and I (*saya*) are almost the same; hence, these were detected as the same classes. For the average recall (AR), the model obtained a value of 0.670, or 67%, with IoU between 0.50 and 0.95 in all datasets with a maximum detection of one. The evaluation results can be seen in Figure 7.

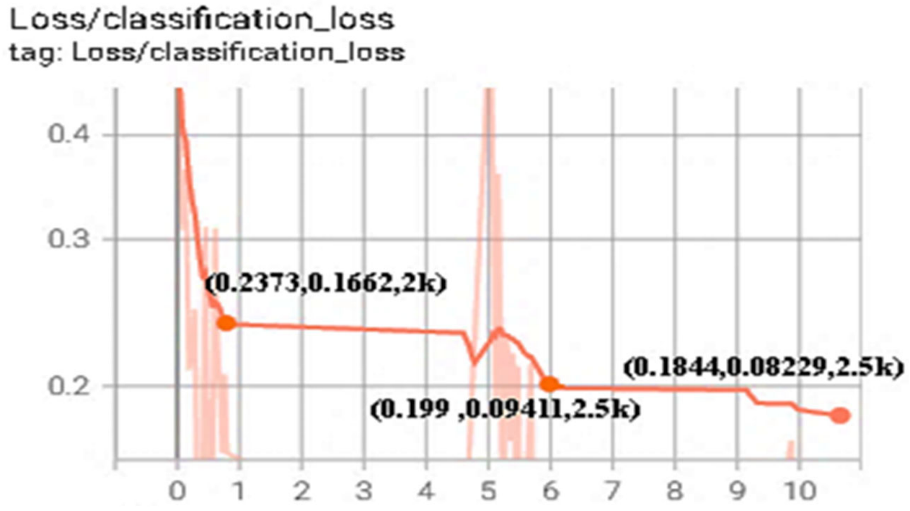
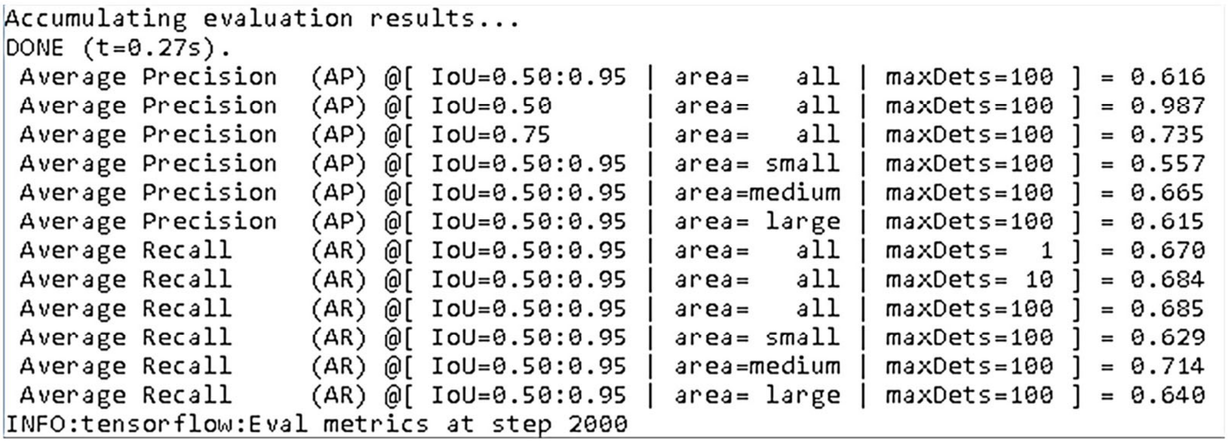


Figure 6. Results of loss, learning rate, and steps per second for the trained model of classification loss for three times training.

Table 3. Loss, learning rate, and steps per second for the trained model with three times training results.

	1st Training with 2000 Steps		**2nd Training with 2500 Steps**		**3rd Training with 2500 Steps**	
	Smoothed	**Loss Value**	**Smoothed**	**Loss Value**	**Smoothed**	**Loss Value**
Classification loss	0.23730	0.16620	0.19900	0.09411	0.18440	0.08229
Localisation loss	0.16730	0.09643	0.15180	0.04301	0.13880	0.05348
Regularisation loss	0.15240	0.14820	0.15130	0.14510	0.15050	0.14510
Total loss	0.55700	0.41100	0.50210	0.28220	0.47360	0.28090
Learning rate	0.06807	0.07992	0.06560	0.06933	0.07206	0.07982
Steps per second	0.65980	0.74190	0.65720	0.63350	0.62930	0.06296



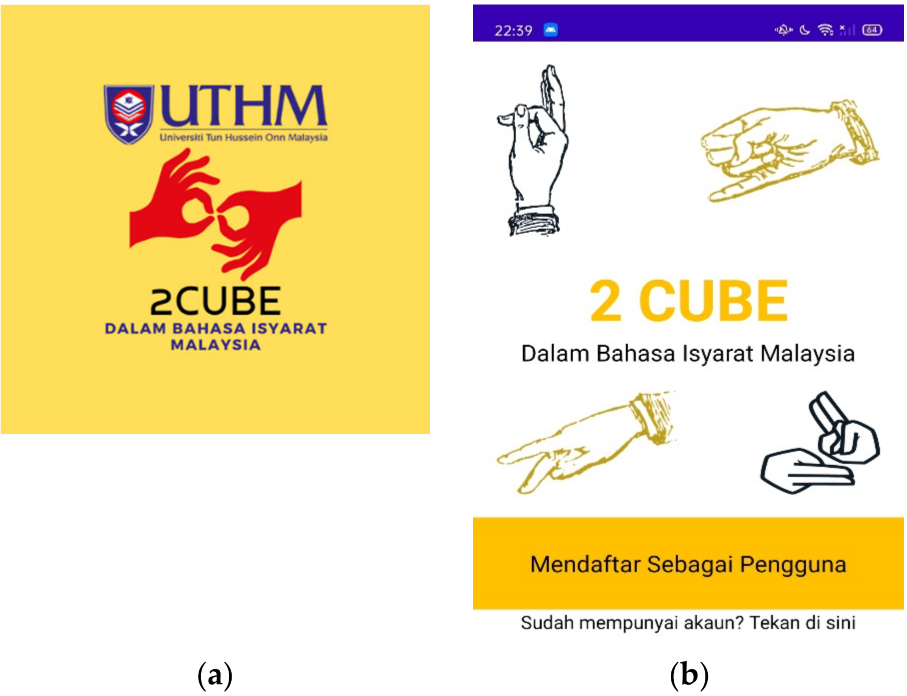
```
Accumulating evaluation results...
DONE (t=0.27s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.616
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.987
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.735
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.557
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.665
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.615
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.670
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.684
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.685
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.629
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.714
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.640
INFO:tensorflow:Eval metrics at step 2000
```

Figure 7. Evaluation result of the model after being trained three times.

The model accuracy was determined by using images to estimate the percentage of the accuracy of each class, with brother (*abang*) at 86%, father (*bapa*) at 88%, mother (*emak*) at 92%, I (*saya*) at 97%, and love (*sayang*) at 98%, while the results of using a live camera from a webcam to detect the hand gestures in real time show that the accuracy of *saya* is 83%, *sayang* is 94%, and *emak* is 93%.

4.3. Development of Android Application

Figure 8a shows the launcher icon for the application, a graphic representing the mobile application. This icon appears on the user's home screen whenever the user downloads this application. The main page for this application is shown in Figure 8b, where users need to register before they can use the application. If the user already has a registered account, they can log in with their successfully registered account.


UTHM
Universiti Tun Hussein Onn Malaysia
2CUBE
DALAM BAHASA ISYARAT MALAYSIA
22:39
Mendaftar Sebagai Pengguna
(a)
(b)

2 CUBE

Dalam Bahasa Isyarat Malaysia

Sudah mempunyai akaun? Tekan di sini

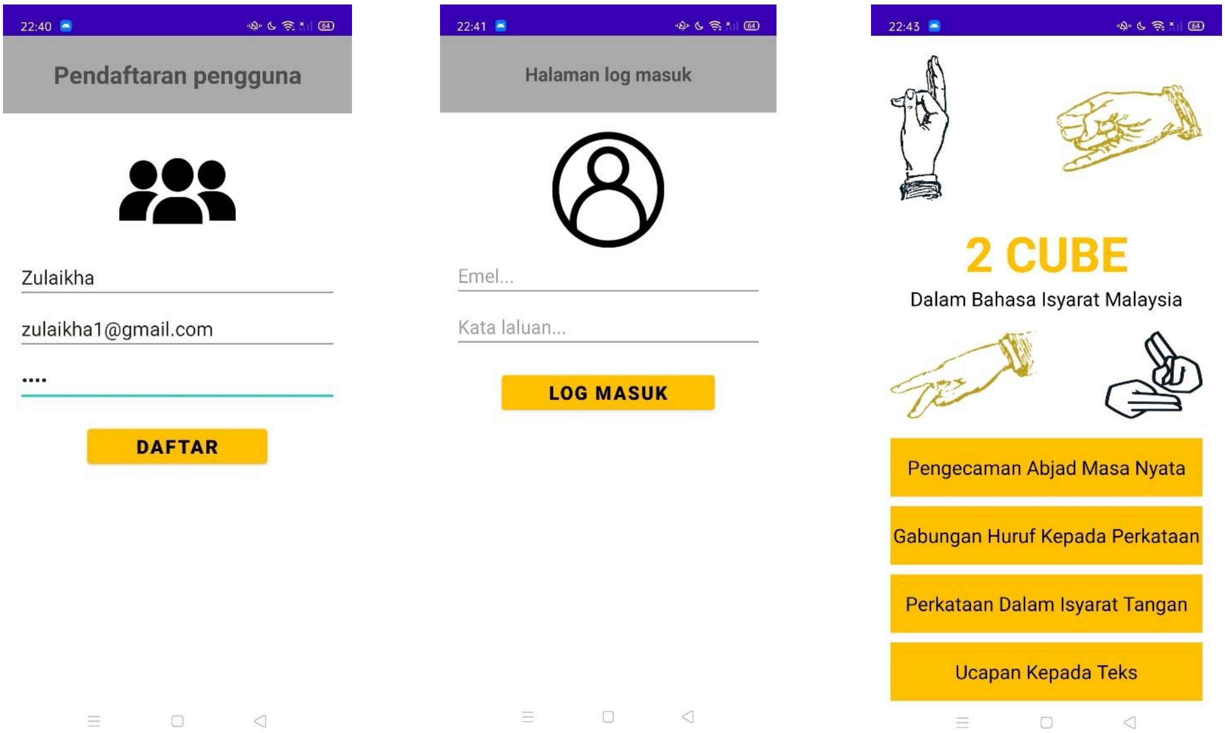
Figure 8. (**a**) Launcher icon; (**b**) main page.

Figure 9a shows the user's registration page for the application. Users need to input their name, email address, and password before clicking on the register button, and Figure 9b shows the user's login page. The user must enter their successfully registered email and password to log in to the application by clicking on the login button.

Figure 9c shows the home page of the application after the user successfully logs into the application, where there are four clickable buttons with different functions for them to choose from, which are BIM letter hand gestures, BIM letter hand gestures to create a word conversion, BIM word hand gesture to text conversion, and, lastly, speech-to-text conversion.

Figure 10 shows the page after clicking on the BIM letters recognition button, whereas Figure 10a shows that users need to click on the start camera (*mulakan kamera*) recognition before using this feature. Figure 10b shows that the user needs to allow the app to take pictures and record videos if they are a first-time user before proceeding.

Figure 11 shows the BIM letters page when the camera recognition has been allowed, whereas Figure 11a shows the camera detected the letter 'D' when the BIM hand gesture is shown. In contrast, Figure 11b shows the camera detects the letter 'I' when the BIM hand gesture is directed to the camera. As for Figure 11c, when the camera does not recognise the hand gesture shown, the app displays *Tidak dapat dikesan*, which means it cannot be detected.


22:40
Pendaftaran pengguna
DAFTAR
22:41
Halaman log masuk
LOG MASUK
22:43
2 CUBE
Dalam Bahasa Isyarat Malaysia
Pengecaman Abjad Masa Nyata
Gabungan Huruf Kepada Perkataan
Perkataan Dalam Isyarat Tangan
Ucapan Kepada Teks

Zulaikha

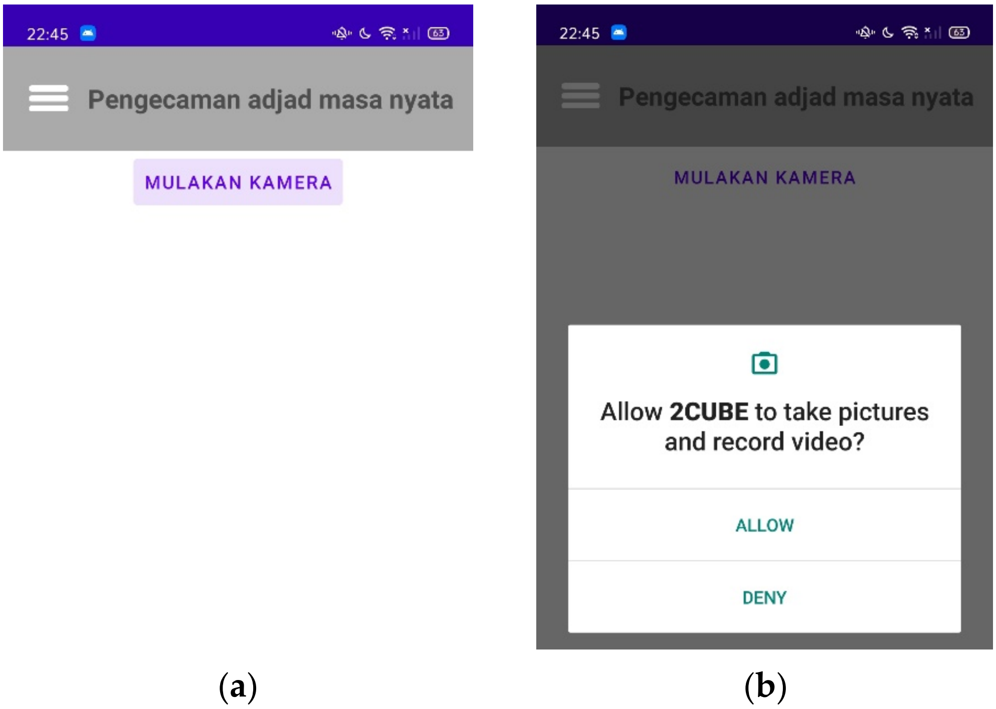
zulaikha1@gmail.com

••••

Emel...

Kata laluan...

Figure 9. (**a**) Registration (*Daftar*) page; (**b**) login (*Log Masuk*) page; (**c**) home page.



Allow **2CUBE** to take pictures and record video?

(**a**) (**b**)

Figure 10. BIM letters recognition page for (**a**) starting camera recognition; (**b**) allowing the app to use the camera.

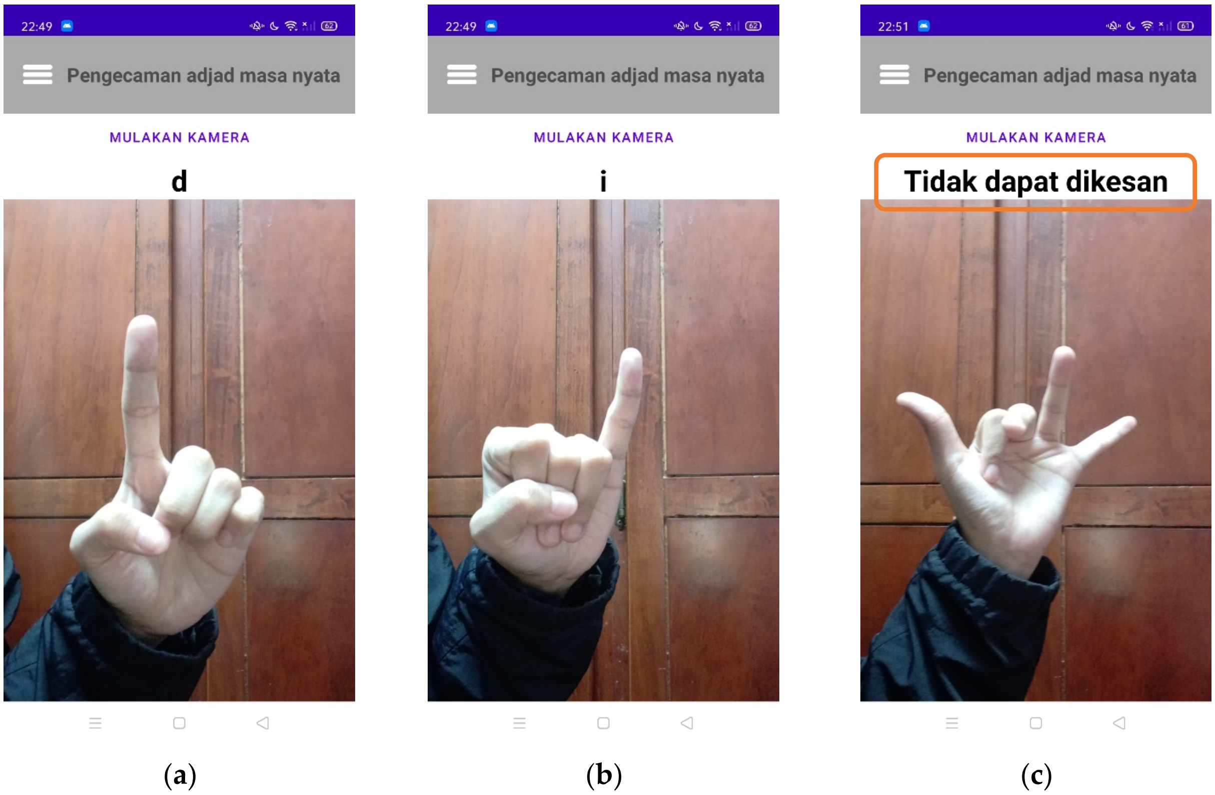


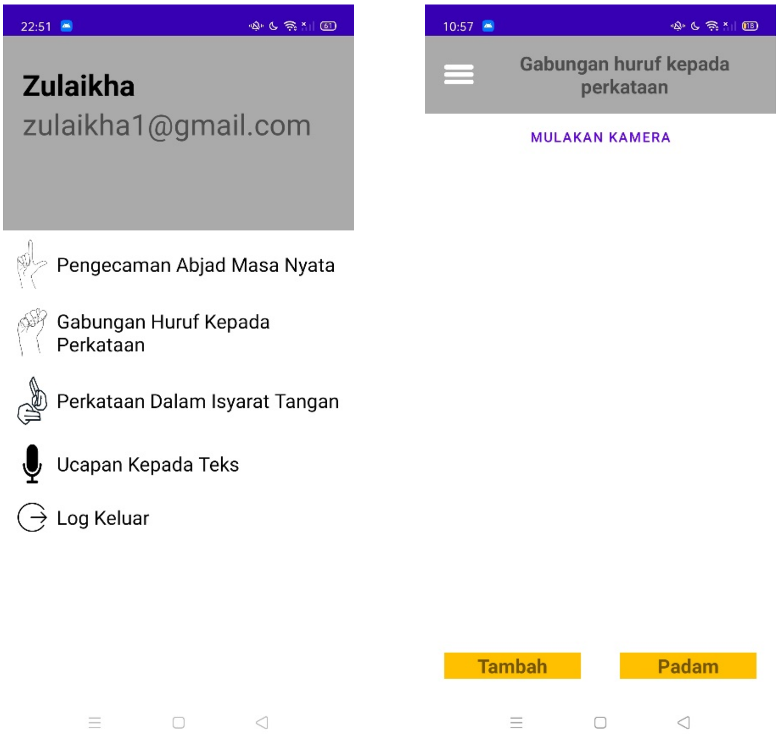
Figure 11. BIM letter page for (**a**) recognition of BIM hand gesture letter 'd'; (**b**) recognition of BIM hand gesture letter 'i'; (**c**) the BIM hand gesture shown is not recognised.

Figure 12a shows the sidebar menu on which the user can click, and they can see their name and the registered email address they use. In addition, the sidebar menu includes four buttons with different features that they can click on, and they can also sign out from the application when they do not want to use it anymore.

Figure 12b shows the BIM combined letter page where the user needs to click on the start camera recognition button, and this page also has an additional and clear button for the user to use when they want to combine the hand gesture they show or erase the letter they want.

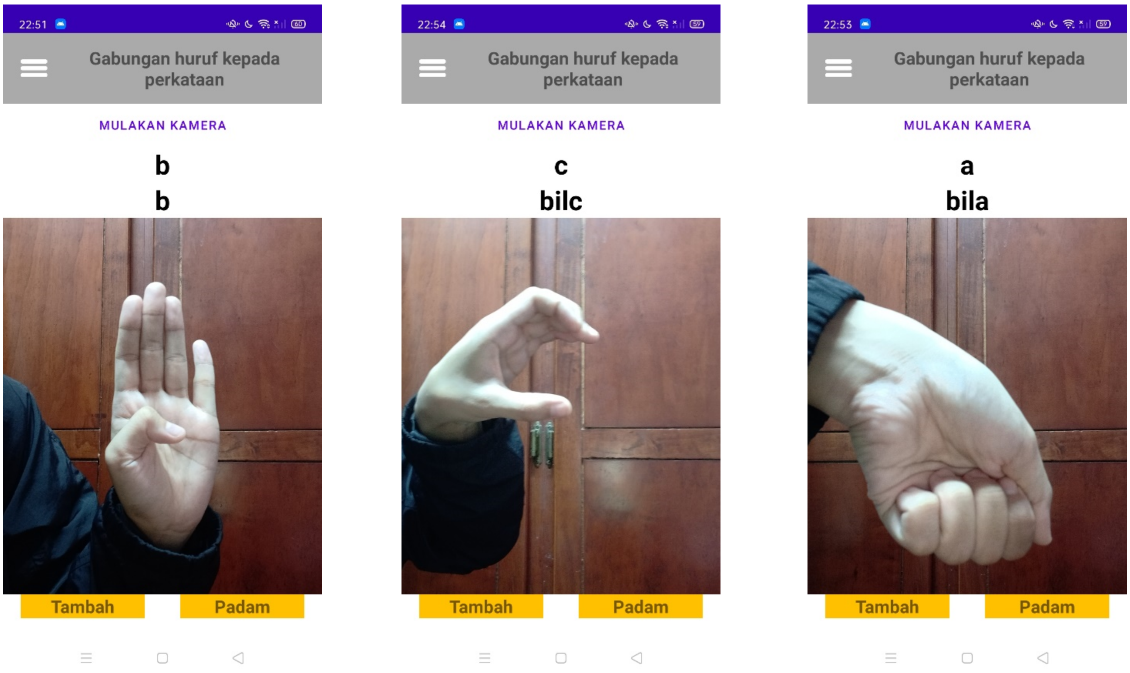
Figure 13 shows the BIM combined letter page, whereas Figure 13a shows the hand gesture of the letter 'B', and this is added to the app by clicking on the add (*Tambah*) button. Figure 13b shows that the hand gesture of the letter 'C' is shown and is added to the app, resulting in the word '*bilc*', which is an incorrect word; therefore, the user must click on the delete (*Padam*) button to delete the letter 'C'. Lastly, Figure 13c shows the hand gesture of the letter 'A' after deleting the letter 'C'; hence, the resulting word is '*bila*'.

Figure 14 shows the BIM word hand gesture page, where the user can use this feature to detect the BIM word hand gesture. Users need to click on the start camera recognition button to start detecting the hand gesture they show. Figure 14a shows the BIM hand gesture being translated to brother (*abang*) in a text, while Figure 14b shows mother (*emak*) being translated by the app when the user displays the mother (*emak*) hand gesture. Lastly, Figure 14c shows the letter I (*saya*) being translated from the hand gesture shown by the user.


22:51
Zulaikha
zulaikha1@gmail.com
Pengecaman Abjad Masa Nyata
Gabungan Huruf Kepada Perkataan
Perkataan Dalam Isyarat Tangan
Ucapan Kepada Teks
Log Keluar
10:57
Gabungan huruf kepada perkataan
MULAKAN KAMERA
Tambah
Padam

(**a**) (**b**)

Figure 12. (**a**) Sidebar menu; (**b**) BIM combined letter starts camera recognition.



(**a**) (**b**) (**c**)

Figure 13. BIM combined letter page for (**a**) hand gesture of letter 'b'; (**b**) hand gesture of letter 'c'; (**c**) hand gesture of letter 'a' after deleting letter 'c'.

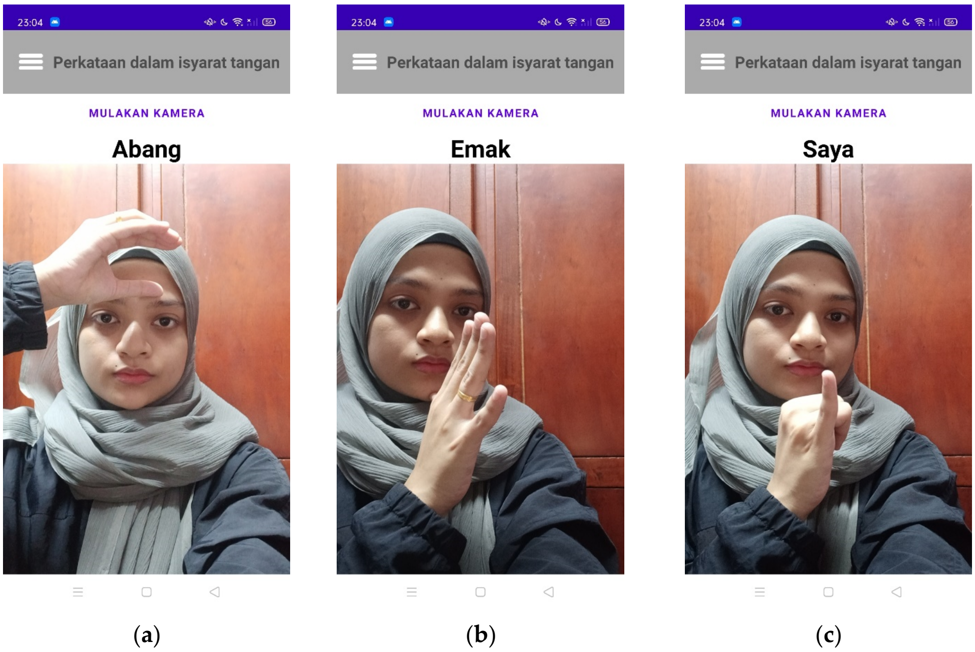
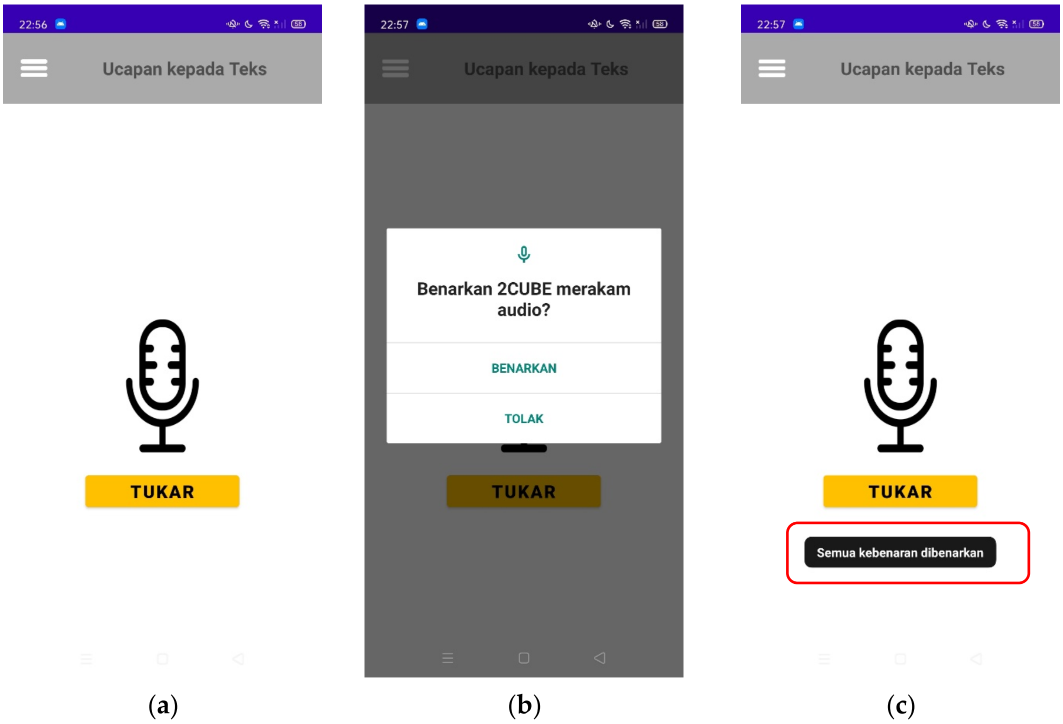


Figure 14. BIM word hand gesture page for (**a**) *abang* hand gesture; (**b**) *emak* hand gesture; (**c**) *saya* hand gesture.

Figure 15 shows the speech-to-text page, whereas Figure 15a shows the main page once the user clicks the speech-to-text button. After that, the microphone can be clicked, and for a first-time user of the app, the user needs to grant access to recording the audio, as shown in Figure 15b. Finally, Figure 15c shows the *Semua kebenaran dibenarkan* message, meaning the user has granted all access.


Benarkan 2CUBE merakam audio?

Semua kebenaran dibenarkan

Figure 15. Speech-to-text page of (**a**) the main page; (**b**) permission for the app to record the audio; (**c**) access granted.

Figure 16 shows the speech-to-text page, and in Figure 16a, it can be seen that the user needs to click on the microphone icon, the Google speech recogniser pops up, the user is able to talk and capture speech by using the microphone, and it detects the speech and converts it to text, as shown in Figure 16b. Users need to click the change (Tukar) button for the next speech-to-text process.

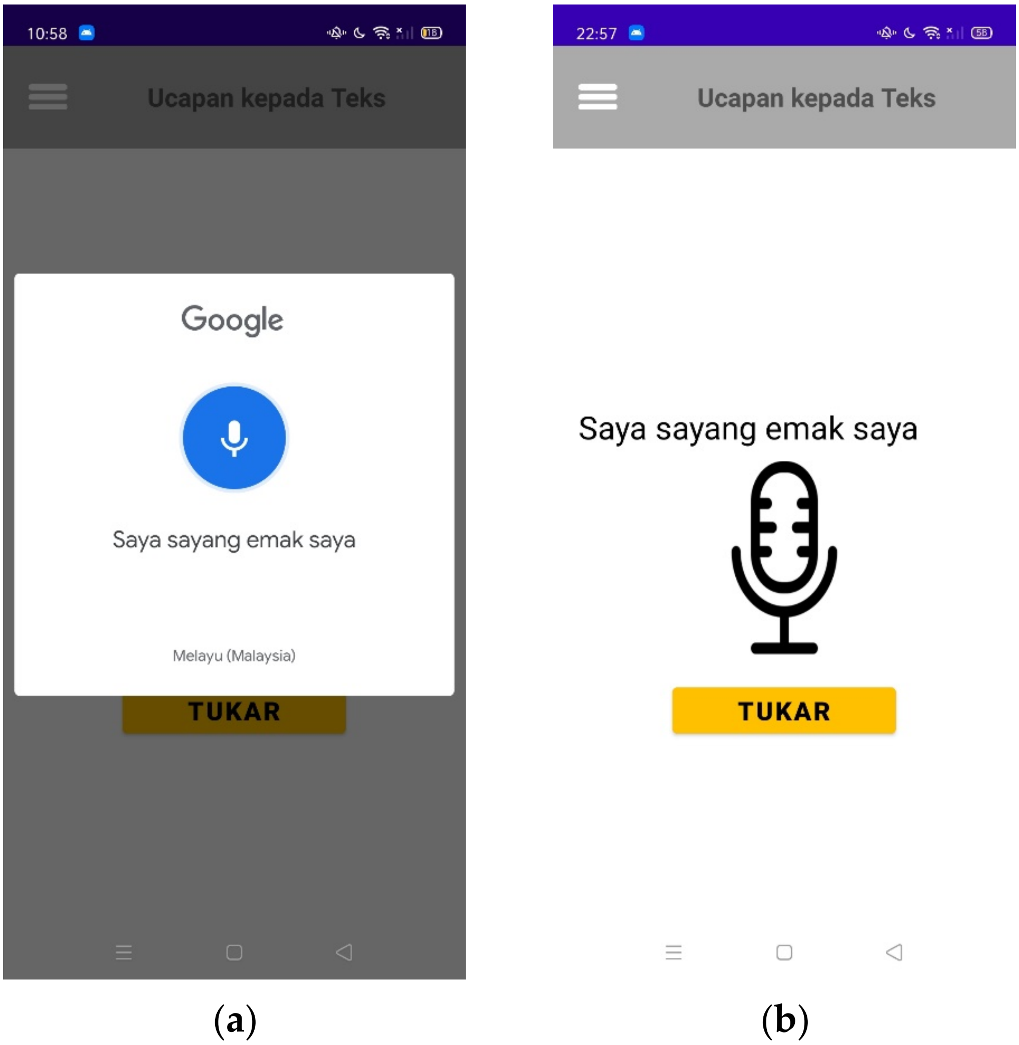


Figure 16. Speech-to-text page of (**a**) converting recorded speech to text; (**b**) the speech has been converted to text.

4.4. Analysis of Android Application

By selecting their preferred speech to text, BIM letter recognition, BIM letters to construct a word, and, finally, BIM word hand gesture buttons on the BIM Android application, deaf/mute and normal people can communicate with one another.

This test can be conducted by repeating a hand gesture of each BIM letter and captured by phone camera ten times, and the accuracy results are tabulated in Table 4. The letters 'B', 'D', 'I', 'M', and 'V' have the highest accuracy from ten trials at 100%, while the lowest is the letter 'E', with 50% accuracy. The other stated letters have an average accuracy above 50%.

Table 4. Analysis of BIM letters from A to Z by using the app.

Letter	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Accuracy (%)	60	100	70	100	50	90	70	90	100	80	70	90	100	70	80	80	70	60	90	70	60	100	90	70	90	60

A speech-to-text analysis was conducted, and the accuracy results are presented in Table 5. The test aims to determine whether or not the application accurately recognises the speech. For example, the words '*abang*' and '*sayang*' have an accuracy of 100%, '*bapa*' has an accuracy of 90%, and '*emak*' and '*saya*' have an accuracy of 80%.

Table 5. Analysis of speech to text with chosen words.

Word	***Abang***	***Bapa***	***Emak***	***Saya***	***Sayang***
Accuracy (%)	100	90	80	80	100

5. Conclusions

In summary, Bahasa Isyarat Malaysia (BIM), an Android application, was successfully developed. This project's goals were all completed. This success can be seen in the findings for the BIM letters, which, after training the models, achieved 99.75% accuracy. The app was built successfully for testing and analysis to determine the effectiveness of the whole system, where the test analysis reveals that, after ten trials, the average accuracy of the letters hand gesture was greater than 50%. The same may be said for speech to text, where an acceptable accuracy of more than 80% was attained. Briefly, this application can help deaf/mute and normal people communicate at ease with each other. This project can also eliminate the hassle of a human translator, making it significantly more cost-effective while developing a shorter and more fascinating interaction.

Additionally, there are a number of potential areas for future research that can be taken into account: (i) to increase the accuracy of speech recognition, audio–visual speech recognition with lip-reading will be introduced and (ii) to increase the performance of hand gesture recognition, attention models that enable the system to concentrate on the most instructive portion of a sign video sequence can be used.

Author Contributions: Conceptualisation, I.Z.S.B., S.S. and A.K.M.; methodology, I.Z.S.B., S.S. and A.K.M.; software, I.Z.S.B.; validation, I.Z.S.B., S.S. and A.K.M.; formal analysis, I.Z.S.B. and S.S.; investigation, I.Z.S.B.; resources, I.Z.S.B.; data curation, I.Z.S.B. and S.S.; writing—original draft preparation, I.Z.S.B. and S.S.; writing—review and editing, S.S., A.K.M., K.I., U.F., M.A.B.A. and S.Y.; visualisation, I.Z.S.B. and S.S.; supervision, S.S., A.K.M. and K.I.; project administration, S.S. and A.K.M.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Universiti Tun Hussein Onn Malaysia (UTHM) through TIER 1 (vot H917).

Data Availability Statement: The alphabet dataset used in this study is partially and openly available in Kaggle, except for G and T (https://www.kaggle.com/datasets/grassknoted/asl-alphabet accessed on 7 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahire, P.G.; Tilekar, K.B.; Jawake, T.A.; Warale, P.B. Two Way Communicator between Deaf and Dumb People and Normal People. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 641–644. [CrossRef]
2. Kanvinde, A.; Revadekar, A.; Tamse, M.; Kalbande, D.R.; Bakereywala, N. Bidirectional Sign Language Translation. In Proceedings of the 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 25–27 June 2021; pp. 1–5. [CrossRef]
3. Alobaidy, M.A.; Ebraheem, S.K. Application for Iraqi sign language translation. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 5226–5234. [CrossRef]
4. Dewasurendra, D.; Kumar, A.; Perera, I.; Jayasena, D.; Thelijjagoda, S. Emergency Communication Application for Speech and Hearing-Impaired Citizens. In Proceedings of the 2020 From Innovation to Impact (FITI), Colombo, Sri Lanka, 15 December 2020; pp. 1–6. [CrossRef]
5. Patil, D.B.; Nagoshe, G.D. A Real Time Visual-Audio Translator for Disabled People to Communicate Using Human-Computer Interface System. *Int. Res. J. Eng. Technol. (IRJET)* **2021**, *8*, 928–934.
6. Mazlina, A.M.; Masrulehsan, M.; Ruzaini, A.A. MCMSL Translator: Malaysian Text Translator for Manually Coded Malay Sign Language. In Proceedings of the IEEE Symposium on Computers & Informatics (ISCI 2014), Kota Kinabalu, Sabah, 28–29 September 2014.
7. Ke, S.C.; Mahamad, A.K.; Saon, S.; Fadlilah, U.; Handaga, B. Malaysian sign language translator for mobile application. In *Proceedings of the 11th International Conference on Robotics Vision, Signal Processing and Power Applications, Penang, Malaysia 5–6 April 2021*; Mahyuddin, N.M., Mat Noor, N.R., Mat Sakim, H.A., Eds.; Lecture Notes in Electrical Engineering; Springer: Singapore, 2022; Volume 829. [CrossRef]
8. Asri, M.A.; Ahmad, Z.; Mohtar, I.A.; Ibrahim, S. A Real Time Malaysian Sign Language Detection Algorithm Based on YOLOv3. *Int. J. Recent Technol. Eng. (IJRTE)* **2019**, *8*, 651–656. [CrossRef]
9. Karbasi, M.; Zabidi, A.; Yassin, I.M.; Waqas, A.; Bhatti, Z. Malaysian sign language dataset for automatic sign language recognition system. *J. Fundam. Appl. Sci.* **2017**, *9*, 459–474. [CrossRef]
10. Jayatilake, L.; Darshana, C.; Indrajith, G.; Madhuwantha, A.; Ellepola, N. Communication between Deaf-Dumb People and Normal People: Chat Assist. *Int. J. Sci. Res. Publ.* **2017**, *7*, 90–95.
11. Yugopuspito, P.; Murwantara, I.M.; Sean, J. Mobile Sign Language Recognition for Bahasa Indonesia using Convolutional Neural Network. In Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia, (MoMM2018), Yogyakarta, Indonesia, 19–21 November 2018; pp. 84–91. [CrossRef]
12. Sincan, O.M.; Keles, H.Y. AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods. *IEEE Access* **2020**, *8*, 181340–181355. [CrossRef]
13. Yee, C.V. Development of Malaysian Sign Language in Malaysia. *J. Spec. Needs Educ.* **2018**, *8*, 15–24.
14. Siong, T.J.; Nasir, N.R.; Salleh, F.H. A mobile learning application for Malaysian sign language education. *J. Phys. Conf. Ser.* **2021**, *1860*, 012004. [CrossRef]
15. Sahid, A.F.; Ismail, W.S.; Ghani, D.A. Malay Sign Language (MSL) for Beginner using android application. In Proceedings of the 2016 International Conference on Information and Communication Technology (ICICTM), Kuala Lumpur, Malaysia, 16–17 May 2016; pp. 189–193. [CrossRef]
16. Hafit, H.; Xiang, C.W.; Yusof, M.M.; Wahid, N.; Kassim, S. Malaysian sign language mobile learning application: A recommendation app to communicate with hearing-impaired communities. *Int. J. Electr. Comput. Eng. (IJECE)* **2019**, *9*, 5512–5518. [CrossRef]
17. Monika, K.J.; Nanditha, K.N.; Gadina, N.; Spoorthy, M.N.; Nirmala, C.R. Conversation Engine for Deaf and Dumb. *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)* **2021**, *9*, 2271–2275. [CrossRef]
18. Jacob, S.A.; Chong, E.Y.; Goh, S.L.; Palanisamy, U.D. Design suggestions for an mHealth app to facilitate communication between pharmacists and the Deaf: Perspective of the Deaf community (HEARD Project). *mHealth* **2021**, *7*, 29. [CrossRef] [PubMed]
19. Mishra, D.; Tyagi, M.; Verma, A.; Dubey, G. Sign Language Translator. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 246–253.
20. Özer, D.; Göksun, T. Gesture Use and Processing: A Review on Individual Differences in Cognitive Resources. *Front. Psychol.* **2020**, *11*, 573555. [CrossRef] [PubMed]
21. Ferré, G. Gesture/speech integration in the perception of prosodic emphasis. In Proceedings of the Speech Prosody, Poznań, Poland, 13–16 June 2018; pp. 35–39. [CrossRef]
22. Hsu, H.C.; Brône, G.; Feyaerts, K. When Gesture "Takes Over": Speech-Embedded Nonverbal Depictions in Multimodal Interaction. *Front. Psychol.* **2021**, *11*, 552533. [CrossRef] [PubMed]
23. Tambe, S.; Galphat, Y.; Rijhwani, N.; Goythale, A.; Patil, J. Analysing and Enhancing Communication Platforms available for a Deaf-Blind user. In Proceedings of the 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur Odisha, India, 16–17 December 2020; pp. 1–5. [CrossRef]
24. Seebun, G.R.; Nagowah, L. Let's Talk: An Assistive Mobile Technology for Hearing and Speech Impaired Persons. In Proceedings of the 2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM), Balaclava, Mauritius, 25–27 November 2020; pp. 210–215. [CrossRef]

25. Maarif, H.A.Q.; Akmeliawati, R.; Bilal, S. Malaysian Sign Language database for research. In Proceedings of the 2012 International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 3–5 July 2012; pp. 798–801. [CrossRef]
26. Ryumin, D.; Ivanko, D.; Ryumina, E. Audio-Visual Speech and Gesture Recognition by Sensors of Mobile Devices. *Sensors* **2023**, *23*, 2284. [CrossRef] [PubMed]
27. Novopoltsev, M.; Verkhovtsev, L.; Murtazin, R.; Milevich, D.; Zemtsova, I. Fine-tuning of sign language recognition models: A technical report. *arXiv* **2023**, arXiv:2302.07693. [CrossRef]
28. Konaite, M.; Owolawi, P.A.; Mapayi, T.; Malele, V.; Odeyemi, K.; Aiyetoro, G.; Ojo, J.S. Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite. In Proceedings of the International Conference on Artificial Intelligence and its Applications (icARTi '21), Bagatelle, Mauritius, 9–10 December 2021; pp. 1–6. [CrossRef]
29. Dai, J. Real-time and accurate object detection on edge device with TensorFlow Lite. *J. Phys. Conf. Ser.* **2020**, *1651*, 012114. [CrossRef]
30. Kannan, R.; Jian, C.J.; Guo, X. Adversarial Evasion Noise Attacks Against TensorFlow Object Detection API. In Proceedings of the 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 8–10 December 2020; pp. 1–4. [CrossRef]
31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
32. Yu, T.; Gande, S.; Yu, R. An Open-Source Based Speech Recognition Android Application for Helping Handicapped Students Writing Programs. In Proceedings of the International Conference on Wireless Networks (ICWN), Las Vegas, NV, USA, 27–30 July 2015; pp. 71–77.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.