



Translating SIBI (Sign System for Indonesian Gesture) Gesture-to-Text in Real-Time using a Mobile Device

Misael Jonathan & Erdefi Rakun*

Faculty of Computer Science, University of Indonesia,
Kampus Baru UI, Depok 16424, Indonesia

*E-mail: efi@cs.ui.ac.id

Abstract. The SIBI gesture translation framework by Rakun was built using a series of machine learning technologies: MobileNetV2 for feature extraction, Conditional Random Field for finding the epenthesis movement frame, and Long Short-Term Memory for word classification. This high computational translation system was previously implemented on a personal computer system, which lacks portability and accessibility. This study implemented the system on a smartphone using an on-device inference method: the translation process is embedded into the smartphone to provide lower latency and zero data usage. The system was then improved using a parallel multi-inference method, which reduced the average translation time by 25%. The final mobile SIBI gesture-to-text translation system achieved a word accuracy of 90.560%, a sentence accuracy of 64%, and an average translation time of 20 seconds.

Keywords: *Android; gesture-to-text translation; Indonesian sign language recognition; TensorFlow; mobile application; on-device inference.*

1 Introduction

Deafness is a condition where a person has a disturbance of their sense of hearing and suffers from impairment of oral communication. To overcome this hearing problem, alternative ways of communicating using other senses are needed. A common form of communication is the use of sign language, which can be captured visually. Sign language combines finger movements, hands/arms, body and facial expressions that describe representative words [1]. There are two types of Indonesian sign languages: SIBI (Sign System for Indonesian Language) and BISINDO (Indonesian Sign Language).

SIBI is an Indonesian language represented by gestures, which are arranged in a sentence according to Indonesian grammar. SIBI is a sign language system officially recognized by the Indonesian Ministry of Education and Culture in 1994 and is used in schools for students with special needs. BISINDO is a sign language that has developed naturally among the deaf community. BISINDO's grammar differs from Indonesian grammar. The grammatical differences include

Received May 23rd, 2022, 1st Revision September 14th, 2022, 2nd Revision October 31st, 2022, Accepted for publication November 17th, 2022.

Copyright © 2022 Published by IRCS-ITB, ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2022.16.3.5

all elements ranging from phonology, morphology, syntax, pragmatics and other elements. BISINDO is generally used for everyday communication.

Through the use of sign language, deaf people can hopefully be able to overcome the communication barriers they face. However, many people find it challenging to communicate with sign language. As a result, deaf people will still experience communication problems. An alternative method is needed to overcome this so that sign language can be used more easily. One such alternative method is to use an application that can translate sign language into text.

Gesture-to-text translator applications require ease of use in order to achieve their goal of being a communication bridge for the deaf. The development of mobile processor technology currently provides an opportunity to develop such a gesture-to-text translator application for mobile platforms. Another consideration is that the mobile phone has become a gadget owned by almost everyone. This research focused on developing a mobile gesture-to-text translator application for SIBI. After completion of this mobile SIBI application, we will start developing a similar translator for BISINDO.

As a comparison, we searched the App Store and Google Play for mobile apps related to sign language. The following were our findings:

1. Generally, the available mobile apps are applications that teach sign language. These types of applications translate text into gestures. In contrast, the application we developed translates gestures to text.
2. Many of the available applications are related to American Sign Language (ASL). SIBI is very different from ASL, both in terms of gestures and the way word signals are arranged in sentences.

In addition to the steps generally required to recognize gestures, such as feature extraction and determining a classification model, our application needed to add processes that are specifically required by SIBI. The uniqueness of SIBI gestures lies in the presence of inflectional word gestures. Inflectional words are root words to which prefixes, suffixes, particles, and prefix + suffix pairs (confixes) are added. The addition of these affixes serves to give additional meaning to the root word.

Among the various types of SIBI gestures, inflectional word gestures are the most common type of gestures. Inflectional word gestures do not have a unique gesture but are formed by connecting the component word gestures that make up the inflectional word. Figure 1 below shows some inflectional words that are formed from the root word 'main' (= to play), such as 'bermain' (= to play), 'memainkan' (= the act of playing), 'pemain' (= player), and 'permainan' (= game).

Previous research [2] has shown that we can improve the accuracy of the inflectional word recognition system by separating words into their components. This component separation technique is much smaller than generating a unique set of feature vectors for each inflectional word in the Indonesian language. The implementation of this separation method notably reduces the computational time required to interpret inflectional word gestures and improves the efficiency of the translation system. Our mobile gesture-to-text translation system also employs this technique to recognize inflectional word gestures using only three feature vector sets: prefix, root word, and suffix feature sets.











No.	Inflectional Word	Prefix Gesture		Root Word Gesture		Suffix Gesture
1.	bermain (= verb playing) consist of: prefix "be" + root word "main"		+			
2.	memainkan (= verb - the act of playing) consist of: prefix "me" + root word "main" + suffix "kan"		+		+	
3.	pemain (=noun -player) consists of prefix "pe" + root word "main"		+			
4.	permainan (=game) - noun prefix "pe" + root word "main" + suffix "an"		+		+	

Figure 1 Inflectional word gestures.

Another aspect that distinguishes our application is the epenthesis removal process. Epenthesis is a transitional movement that has no meaning. The epenthesis connects the gestures of the main components to form an overall unified gesture. In SIBI, epenthesis connects the infix, root word and suffix in an inflectional word and connects the words in a sentence. Figure 2 shows an example of a breakdown of the sentence 'Bagaimana cara mengirim uang melalui

bank? (= How can I send money through a bank?)' video frames. From Figure 2, we can see that the epenthesis in the SIBI sentence gesture can be located:

1. at the beginning of a sentence
2. between two consecutive words
3. between a prefix and the root word
4. between the root word and a suffix
5. between a suffix and the next word
6. at the end of a sentence

There are so many possibilities for epenthesis that it makes it difficult for the translation system to recognize them. Since the epenthesis itself is meaningless and difficult to identify, we developed the translation system by excluding all epenthesis found in the SIBI sentence gestures.

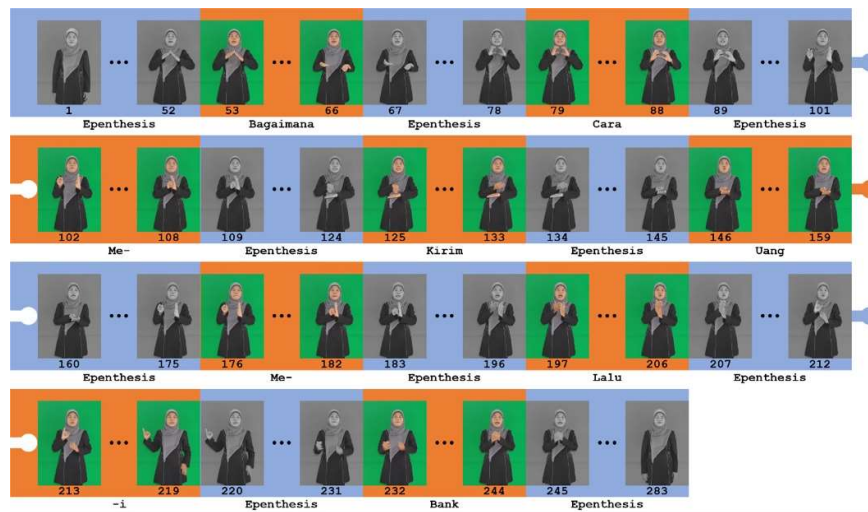


Figure 2 Pieces of SIBI sign movement 'Bagaimana cara mengirim uang melalui bank? (How can I send money through a bank?)'

These processes were built into a separate machine learning model. This research focused on implementing the patented system using the models shown in Figure 3 into a mobile device application, measuring the capability of mobile devices to handle high computational systems, and the improvement approaches which can also be applied in general on-device machine learning problems.

The next section will discuss the architecture of our gesture-to-text translation system, and how we built each process on top of that architecture. The rest of this paper is organized as follows. Section 3 discusses the implementation of the design presented in Section 2 as a mobile application. In Section 4, we analyze the performance of the mobile application. Section 5 discusses the improvements we made to the accuracy and execution time of the mobile application. Finally, Section 6 contains our conclusions and future work that can further improve the performance of the application.

2 Design of SIBI Gesture-To-Text Translation System

The process of recognizing a SIBI gesture and translating it into Indonesian text can be seen in Figure 3, with the following details:

1. SIBI sentence gestures are captured by the smartphone's camera.
2. This gesture video is then broken down into a sequence of frames from the sentence gesture.
3. Feature extraction is performed for each frame of the sentence gesture.
4. All frames containing epenthesis movements are recognized using TCRF and discarded. The result of this process is a sequence of frames for all the words in the sentence. Each word has a varying length of frames depending on how fast or slow the gesture is done.
5. We use the BiLSTM model as the classifier, which requires an input sequence with the same number of frames. This stage performs frame equalization for all the words in the input sentence, to equalize the number of frames for each word sequence into the average number of frames of our word dataset. In this research, the average length of word in the dataset is thirteen frames. The frame equalization process is shown in Figure 4.
6. BiLSTM recognizes each word according to the order of its appearance in a sentence.
7. The last process displays the BiLSTM prediction results on the smartphone screen arranged according to Indonesian grammar.

Since 2018, we have been working on developing this app using four datasets that we have been gradually building. The four datasets are: root words, inflectional words, finger-spelling alphabets, and numbers, and SIBI sentences. The research began with finding a feature extraction technique and a classifier model that are suitable for SIBI.

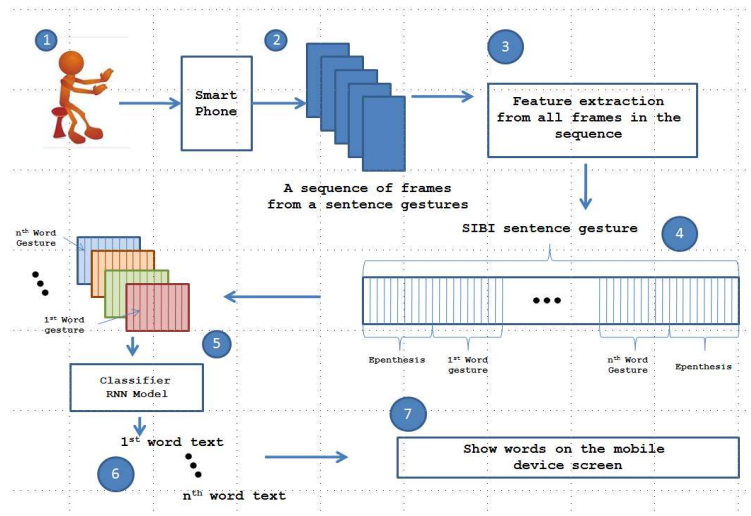


Figure 3 Gesture-to-text architecture [3].

Algorithm 1 Equalize number of frames in array of frames

Input: array of frames (2 dimensional array)

Output: equalized array of frames (2 dimensional array)

```

1: procedure EQUALIZEFRAMELENGTH(A)
2:   threshold ← 13
3:   copyArray ← FloatArraySizeThreshold      ▷ Result, empty array with size of threshold
4:   if inputArray.length ≤ threshold then    ▷ Expand frame length if ≤ threshold
5:     copyArray ← inputArray                ▷ Copy frame data to copyArray
6:     for i ← inputArray.length to threshold do
7:       copyArray[i] ← copyArray[inputArray.length - 1]  ▷ Copy the last frame
8:     end for
9:   else                                     ▷ Cut the frame length if > threshold
10:    copyArray ← inputArray.subArray(0, threshold)
11:   end if
12:   Return copyArray
13: end procedure

```

Figure 4 Frame equalizer pseudo code.

Table 1 summarizes our research to obtain feature extraction techniques, classifier models, and ways to recognize and discard epenthesis movement frames. The best research results were used in the mobile application.

The best research results used by mobile apps are:

1. MobileNetV2 for feature extraction
2. Two-layer Bi-directional LSTM as the classifier
3. Threshold CRF to find the epenthesis movement frame.

Table 1 Summary of previous research.

Title	Method	Dataset	Best Result
	<i>Feature Extraction</i>		
Feature Extraction from Smartphone Images by Using Elliptical Fourier Descriptor, Centroid and Area for Recognizing Indonesian Sign Language SIBI (Sistem Isyarat Bahasa Indonesia) [4]	<ul style="list-style-type: none"> • YCrCb Skin color segmentation [5] • Viola-Jones algorithm to find face [6] • Elliptical Fourier Descriptor (EFD) to find the coordinate of the hand's contour [7] • LSTM for classification [8] 	Inflectional words	<ul style="list-style-type: none"> • Root word:99.59 % accuracy • Prefix: 71.75% accuracy • Suffix: 86.88% accuracy
Human Skeleton Feature Extraction from 2-Dimensional Video of Indonesian Language Sign System (SIBI [Sistem Isyarat Bahasa Indonesia]) Gestures [9]	<ul style="list-style-type: none"> • Viola-Jones algorithm to find face [6] • Hand extraction [10] • Lucas-Kanade Hand tracking [11] • LSTM for classification [8] 	Root Word	<ul style="list-style-type: none"> • 98.214% accuracy
Recognizing the Components of Inflectional Word Gestures in Indonesian Sign System known as SIBI (Sistem Isyarat Bahasa Indonesia) by using Lip Motion [12]	<ul style="list-style-type: none"> • Sagonas technique to get 68 face points [13] • LipNet for classification [14] 	Inflectional words	<ul style="list-style-type: none"> • root words: 83,98% accuracy • prefix: 73,94% accuracy • suffixes: 82,10% accuracy
Recognizing Word Gesture in Sign System for Indonesian Language (SIBI) Sentences Using DeepCNN and BiLSTM [15]	<ul style="list-style-type: none"> • Deep Residual Network (ResNet50) [16] • MobileNetV2 [17] • LSTM for classification [8] 	SIBI Sentences	<ul style="list-style-type: none"> • 99% accuracy
Recognizing Finger spelling in SIBI (Sistem Isyarat Bahasa Indonesia) using OpenPose and Elliptical Fourier Descriptor [18]	<ul style="list-style-type: none"> • OpenPose [19] for hand tracking and Elliptical Fourier Descriptor (EFD) [7] for feature extraction • OpenPose [19] for hand tracking and MobileNetV2 [17] for feature extraction 	Alphabet and Number	<ul style="list-style-type: none"> • 67.23% accuracy • 88.41% accuracy
Improving Recognition of SIBI (Sign System for Indonesian Language) Word Gesture Performance by Combining Skeleton and Handshape Features [20]	<ul style="list-style-type: none"> • Sequence Feature Vector Concatenation technique • Two-layer Bidirectional LSTM feature extraction [8] 	SIBI Sentences	<ul style="list-style-type: none"> • 88.016% accuracy

Table 1 Continued. Summary of previous research.

Title	Method	Dataset	Best Result
<i>Feature Extraction</i>			
Audio Feature Extraction on SIBI Dataset for Speech Recognition [21]	<ul style="list-style-type: none"> Mel Frequency Cepstral Coefficients as feature extraction for Automatic Speech Recognition [22] 	SIBI Sentences	<ul style="list-style-type: none"> 4.71% WER and 10.30% SER
Recognition of Sign Language System for Indonesian Language Using Long Short-Term Memory Neural Networks [2]	<ul style="list-style-type: none"> Hidden Markov Model [24] LSTM [8] 	Inflectional words	<ul style="list-style-type: none"> 2-layer LSTM Inflectional Word: 77.4% root words: 95.4% accuracy prefix: 66.7% accuracy suffixes: 69% accuracy
Sign Language System for Bahasa Indonesia (Known as SIBI) Recognizer using TensorFlow and Long Short-Term Memory [25]	<ul style="list-style-type: none"> One-layer, Two-layer and Two-layer Bidirectional LSTM [8] 	Inflectional words	<ul style="list-style-type: none"> Inflectional Word - 2-layer LSTM - 78.4% accuracy Root Word - 2-layer LSTM - 96.2% accuracy Prefix - 2-layer LSTM - 72.3% accuracy Suffix - 1-layer LSTM - 69% accuracy
Indonesian Language Sign System (SIBI) Recognition using Threshold Conditional Random Fields [26]	<ul style="list-style-type: none"> Threshold Conditional Random Field [27] 	Inflectional words	<ul style="list-style-type: none"> Skeleton Feature - 81.5% accuracy Image Feature - 65.5% Combine Skeleton + Image feature - 81.2%
Word Recognition and Automated Epenthesis Removal for Indonesian Sign System Sentence Gestures [28]	<ul style="list-style-type: none"> MobileNetV2 [17] for feature extraction Threshold Conditional Random Field [27] for recognize gestures and non gestures Two-layer Bidirectional LSTM for classification [8] Sandwiched majority voting of the TCRF output LSTM output grouping 	SIBI Sentences	<ul style="list-style-type: none"> Word Error Rate (WER) 3.4% Sentence Accuracy (SAcc) 80.2%

The following section will discuss how to build the mobile app based on these best findings.

3 Mobile Application Overview

This section shows the specification of the devices used in this study, libraries or tools, and the mobile application design.

3.1 Smartphone and Laptop Specification

The device's specifications for running the SIBI translator are as shown in Figures 5 and 6. On smartphones, the translator is loaded into an Android application, while on a laptop it is loaded into an executable program. The data used, the method of measurement, and the evaluation carried out on the two devices are the same so that the translator's performance is comparable.

Device	Smartphone
Name	Oppo F11 Pro
Operating System	Android 9.0 (Pie)
Chipset	Mediatek MT6771 Helio P70 (12nm)
CPU	Octa-Core (4x2.1 GHz Cortex-A73 & 4x2.0 GHz Cortex-A53)
GPU	Mali-G72 MP3
RAM	4GB
Internal Storage	64GB

Figure 5 Android smartphone specification.

Device	Laptop
Name	ASUS TUF FX504GE
Operating System	Windows 10
Chipset	Intel HM370 Express
CPU	Intel Core i7-8750H (2.2 GHz up to 4.1 GHz boost clock)
GPU	Nvidia GeForce GTX 1060
RAM	8GB
Storage	1TB

Figure 6 Laptop specification.

3.2 Main Libraries

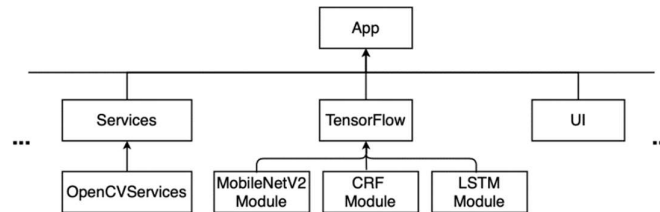
The application uses TensorFlow to run each machine learning model used in this research, JavaCV, a wrapper library commonly used in computer vision, such as OpenCV is used in the preprocessing stage and FFmpeg to extract video data into image data. ReactiveX is used in this application to support asynchronous programming. The list of main libraries used in this application is shown in Table 2.

Table 2 Main libraries used in the Android application.

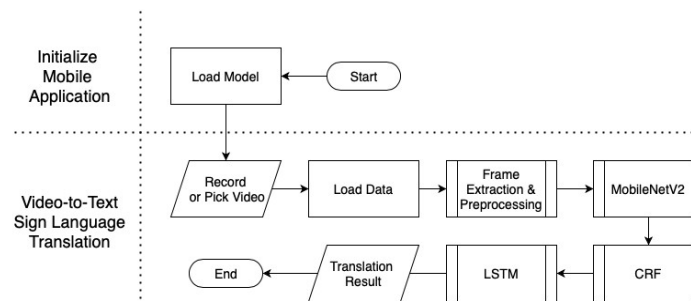
Name	Dependency		
	Group	Name	Version
TensorFlow Java (Android)	org.tensorflow	tensorflow-android	1.13.1
JavaCV	org.bytedeco	javacv-platform	1.5.3
RxJava	io.reactivex.rxjava3	rxjava	3.0.0
	io.reactivex.rxjava3	rxandroid	3.0.0

3.3 Application Structure and Flow Process

Two main modules were created in the Android application. The first module, called Service, has two main tasks, i.e., frame extraction and preprocessing. In this module, video data is converted into a sequence of image data and then prepared into data that are ready to be processed by the model. The second module is TensorFlow, which contains the series of models used in this study, whose task is to process the data and convert it into sign translation text.

**Figure 7** Android application module structure.

The flow of the translator in the application is shown in Figure 8. When the user opens the app, the TensorFlow models are initialized to be ready for use. This process only runs once, and the model can continue to be used until the application is killed. Users can then navigate to the application's camera feature. Users can record a sign language video, which will then be processed by a series of models and translated into sign language translation in text.

**Figure 8** End-to-end application process.

3.4 User Interface Design

The application was created only to have one functionality: to record a video and translate the content of the video into text. The application has one homepage that shows the currently available feature, which is gesture-to-text translation. Then the user will be directed to a page that is integrated into the smartphone's camera module, and there the user can record video through their smartphone and get the translation results after they are done recording. The application design is shown in Figure 9.

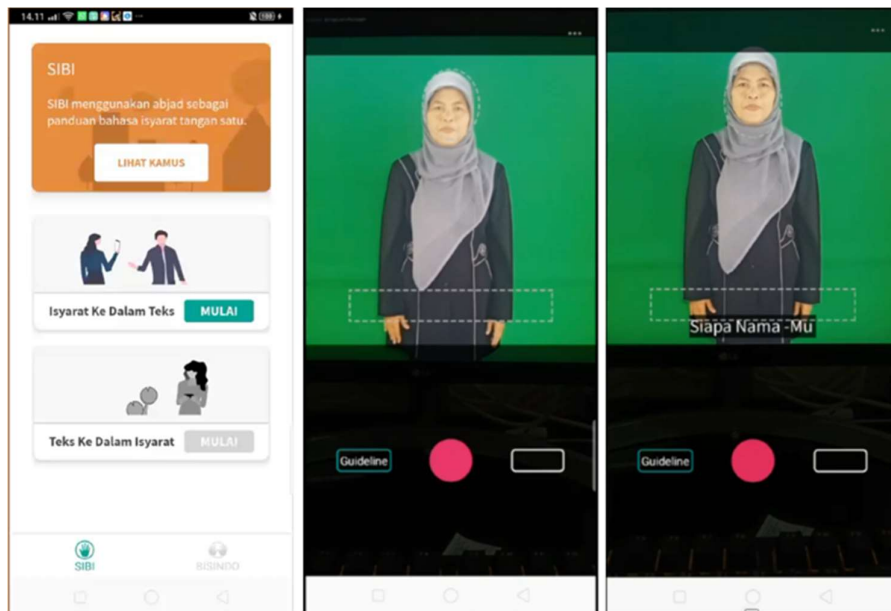


Figure 9 Application user interface design.

3.5 Dataset

This research used a sign language video dataset was collected in previous studies [14]. Sign language videos were taken using a smartphone camera and a greenscreen. The data consisted of 28 types of sentences built with 84 unique words consisting of a combination of base words and affixes.

The sign language was demonstrated by three teachers and two special school students who were fluent in SIBI and used it daily. In total, the data used in this research were 2,247 videos. The data was divided five-fold, and one of the

partitions was used as test data, i.e., 455 data. The sentences used in this study are shown in Table 3.

Table 3 Sentences, number of labels, average numbers of frames.

Sentence (in Indonesian)	Sentence (in English)	Total Labels	Average Number of Frame (30 FPS)
<i>siapa namamu?</i>	What is your name?	3	158
<i>di mana alamat rumahmu?</i>	Where do you live?	5	205
<i>di mana sekolahmu?</i>	Where is your school located?	4	180
<i>bolehkah saya minta nomor teleponmu?</i>	May I have your telephone number?	7	234
<i>film apa yang sedang diputar?</i>	What movies are playing right now?	6	247
<i>jam berapa film ini diputar?</i>	At what times will this movie be shown?	6	246
<i>berapa harga karcis film ini?</i>	How much would a ticket for this movie cost?	5	224
<i>di mana film ini diputar?</i>	Where is this movie being shown?	6	238
<i>apa nama sayuran itu?</i>	What is this vegetable called?	5	187
<i>berapa harga sayuran itu?</i>	How much does this vegetable cost?	5	192
<i>apakah harga sayuran ini boleh ditawar?</i>	Is the price of this vegetable negotiable?	9	270
<i>berapa jumlah yang harus saya bayar?</i>	So how much do I have to pay for all this?	6	230
<i>kami ingin pergi ke kota tua, naik bis apa?</i>	We would like to go to the Old Town – which bus do we have to take	9	319
<i>berapa harga karcis yang harus saya bayar?</i>	How much would the tickets cost?	7	261
<i>kami harus turun di mana?</i>	At which station should we get off	5	199
<i>adakah cara lain kita pergi ke kota tua?</i>	Is there another way we can get to the Old Town?	9	297
<i>saya ingin membuka tabungan, bagaimana caranya?</i>	How would I go about opening a savings account?	9	278
<i>bagaimana cara menabung?</i>	How would I go about saving money?	4	171
<i>di mana kami bisa mengambil tabungan?</i>	Where can we withdraw our savings?	8	247

Table 3 Continued. Sentences, number of labels, average numbers of frames.

Sentence (in Indonesian)	Sentence (in English)	Total Labels	Average Number of Frame (30 FPS)
<i>bagaimana cara mengirim uang melalui bank</i>	How can I send money through a bank?	7	280
<i>selamat natal dan tahun baru</i>	Merry Christmas and Happy New Year	5	220
<i>selamat idul fitri mohon maaf lahir dan batin</i>	Happy Eid ul-Fitr, please forgive me for my mistakes	7	274
<i>selamat ulang tahun</i>	Happy Birthday	3	155
<i>semoga panjang umur</i>	May you live for as long as you wish	3	183
<i>saya sering sakit kepala, saya harus perika ke bagian mana?</i>	I frequently get headaches, which medical specialty department should I visit?	11	315
<i>saya ingin ke dokter umum, siapa nama dokternya?</i>	I want to go to a general practitioner, could you put me in touch with one that's available?	9	272
<i>jam berapa dokter datang?</i>	At what time is the doctor expected to arrive?	4	167
<i>di apotek mana obat ini bisa dibeli?</i>	At which pharmacy can I obtain this medicine?	8	251

3.6 Evaluation and Testing Plan

The test was carried out by comparing the performance of the sign language translator system run on the Android application and the computer. Performance evaluation was carried out by measuring the processing time required to translate one sentence, the accuracy of the resulting translation, and the use of application resources such as CPU, memory, and storage.

The measurement of the processing time started when the video was loaded and ended when the sign text translation had been generated. The measurement was broken down into four stages in the translation system: the time for preprocessing, MobileNetV2, CRF, and LSTM.

The accuracy measurements were divided into two types: word accuracy and sentence accuracy in Eqs. (1) and (2) respectively. Word accuracy was calculated based on the number of correctly predicted words from sentence A to the actual

sentence B and in the correct order index named x, y. The word accuracy formula was made in such a way that it could anticipate a missing word, as illustrated in Table 4. Meanwhile, in sentence accuracy, if there was a mistake in the word prediction results or the word order in the sentence, the prediction was immediately declared wrong (Eq. (2)).

Table 4 Examples of word accuracy and sentence accuracy.

	Expected	Prediction	Accuracy
Word Accuracy	<i>Siapa Nama -Mu</i> (What is your name?)	<i>Siapa Nama -Mu</i>	3/3 = 100%
	<i>Siapa Nama -Mu</i> What is your name?)	<i>Siapa -Mu</i>	2/3 = 66%
	<i>Di Apotek Mana Obat Ini Bisa Di- Beli</i> At which pharmacy can I obtain this medicine?)	<i>Di Apotek Mana Obat Bisa Di- Beli</i>	7/8 = 87.5%
Sentence Accuracy	<i>Siapa Nama -Mu</i> (What is your name?)	<i>Siapa Nama -Mu</i>	3/3 = 100%
	<i>Siapa Nama -Mu</i> (What is your name?)	<i>Siapa -Mu</i>	2/3 = 0%
	<i>Di Apotek Mana Obat Ini Bisa Di- Beli</i> (At which pharmacy can I obtain this medicine?)	<i>Di Apotek Mana Obat Bisa Di- Beli</i>	7/8 = 0%

% i –

$$\text{th sentence} = \frac{\sum(A[x]==B[y], x==y \vee (x+1 \leq |A| \wedge x+1==y), (1 \leq x \leq |A|) \wedge (1 \leq y \leq |B|))}{|A|} \quad (1)$$

$$\% \text{ i – th sentence} = \frac{\sum(A[x]==B[y], x==y, (1 \leq x \leq |A|) \wedge (1 \leq y \leq |B|))}{|A|} == 1 \quad (2)$$

4 Results and Analysis

The test results are presented in Table 5. The experimental results showed that the average time for the smartphone to translate gesture videos was about 31 s. The portion of time required for each stage was 34.29% for preprocessing, 65.22% for MobileNetV2, and 0.12% for CRF, and 0.35% for LSTM.

The smartphone took 2.87 times longer to run the translation system than the computer. This happens because the capacity and capabilities of smartphones are limited compared to computers. In addition, there are still some parts that can be further developed and optimized on smartphones, specifically in the preprocessing and MobileNetV2 stages.

Table 5 Test results: average processing time.

No	Sentence	Average Processing Time							
		Preprocess		MobileNetV2		CRF		LSTM	
		Computer	Smartphone	Computer	Smartphone	Computer	Smartphone	Computer	Smartphone
1	siapa namamu?	4 s 905 ms	7 s 031 ms	1 s 882 ms	13 s 351 ms	22 ms	30 ms	70 ms	54 ms
2	di mana alamat rumahmu?	6 s 801 ms	9 s 343 ms	2 s 287 ms	17 s 792 ms	17 ms	35 ms	22 ms	88 ms
3	di mana sekolahmu?	6 s 347 ms	8 s 173 ms	2 s 022 ms	15 s 588 ms	17 ms	27 ms	25 ms	64 ms
4	bolehkah saya minta nomor teleponmu?	8 s 337 ms	11 s 026 ms	2 s 634 ms	20 s 817 ms	17 ms	39 ms	25 ms	123 ms
5	film apa yang sedang diputar?	8 s 640 ms	11 s 119 ms	2 s 776 ms	21 s 219 ms	17 ms	41 ms	24 ms	113 ms
6	jam berapa film ini diputar?	9 s 129 ms	11 s 500 ms	2 s 807 ms	21 s 677 ms	17 ms	41 ms	28 ms	105 ms
7	berapa harga karcis film ini?	7 s 706 ms	10 s 147 ms	2 s 542 ms	19 s 295 ms	17 ms	42 ms	23 ms	101 ms
8	di mana film ini diputar?	8 s 337 ms	10 s 708 ms	2 s 711 ms	20 s 206 ms	17 ms	37 ms	23 ms	108 ms
9	apa nama sayuran itu?	6 s 344 ms	8 s 661 ms	2 s 080 ms	16 s 231 ms	16 ms	29 ms	22 ms	83 ms
10	berapa harga sayuran itu?	6 s 100 ms	8 s 741 ms	2 s 155 ms	16 s 360 ms	17 ms	28 ms	24 ms	85 ms
11	apakah harga sayuran ini boleh ditawarkan?	9 s 312 ms	12 s 476 ms	3 s 094 ms	23 s 887 ms	18 ms	47 ms	23 ms	159 ms
12	berapa jumlah yang harus saya bayar?	8 s 464 ms	10 s 591 ms	2 s 698 ms	20 s 015 ms	18 ms	36 ms	27 ms	113 ms
13	kami ingin pergi ke kota tua, naik bis apa?	12 s 430 ms	14 s 409 ms	3 s 677 ms	27 s 890 ms	18 ms	62 ms	28 ms	171 ms
14	berapa harga karcis yang harus saya bayar?	8 s 810 ms	11 s 834 ms	2 s 972 ms	22 s 558 ms	17 ms	48 ms	27 ms	129 ms
15	kami harus turun di mana?	7 s 199 ms	9 s 252 ms	2 s 280 ms	17 s 536 ms	16 ms	33 ms	22 ms	90 ms
16	adakah cara lain kita pergi ke kota tua?	11 s 403 ms	13 s 422 ms	3 s 379 ms	25 s 232 ms	17 ms	50 ms	27 ms	164 ms
17	saya ingin membuka tabungan, bagaimana caranya?	9 s 721 ms	12 s 646 ms	3 s 099 ms	24 s 145 ms	17 ms	43 ms	27 ms	147 ms
18	bagaimana cara menabung?	5 s 930 ms	7 s 957 ms	2 s 010 ms	14 s 890 ms	17 ms	29 ms	26 ms	61 ms
19	di mana kami bisa mengambil tabungan?	8 s 418 ms	11 s 269 ms	2 s 839 ms	21 s 233 ms	17 ms	37 ms	25 ms	118 ms
20	bagaimana cara mengirim uang melalui bank?	9 s 564 ms	13 s 038 ms	3 s 211 ms	24 s 821 ms	17 ms	45 ms	23 ms	151 ms
21	selamat natal dan tahun baru	7 s 566 ms	9 s 836 ms	2 s 522 ms	18 s 885 ms	17 ms	44 ms	26 ms	95 ms
22	selamat idul fitri mohon maaf lahir dan batin	9 s 349 ms	12 s 848 ms	3 s 135 ms	24 s 373 ms	17 ms	53 ms	24 ms	129 ms
23	selamat ulang tahun	5 s 325 ms	6 s 771 ms	1 s 752 ms	13 s 025 ms	17 ms	29 ms	25 ms	55 ms
24	semoga panjang umur	6 s 105 ms	8 s 572 ms	2 s 044 ms	16 s 473 ms	17 ms	33 ms	23 ms	76 ms
25	saya sering sakit kepala, saya harus perika ke bagian mana?	10 s 912 ms	14 s 803 ms	3 s 687 ms	28 s 562 ms	17 ms	67 ms	26 ms	171 ms
26	saya ingin ke dokter umum, siapa nama dokternya?	8 s 942 ms	12 s 488 ms	3 s 038 ms	23 s 872 ms	17 ms	42 ms	24 ms	142 ms
27	jam berapa dokter datang?	5 s 737 ms	7 s 720 ms	1 s 915 ms	14 s 451 ms	17 ms	27 ms	25 ms	62 ms
28	di apotek mana obat ini bisa dibeli?	8 s 675 ms	11 s 554 ms	2 s 914 ms	22 s 247 ms	17 ms	43 ms	25 ms	127 ms
Average		8 s 089 ms	10 s 641 ms	2 s 649 ms	20 s 237 ms	17 ms	39 ms	27 ms	110 ms

As shown in Table 6, the SIBI translator running on a smartphone could produce good accuracy, slightly less than that on a computer, with a two-tailed t-test value of 0.04282. The reduced accuracy was due to minor differences in the preprocessing results between the computer, which processes data with Python, and the smartphone, which uses Java, especially during the RGB to HSV color conversion process, which could affect the final accuracy results. Besides that, some common errors occurred, as shown in Table 7, which require improvisation in the machine learning models used.

Table 6 Comparison of prediction accuracy.

Accuracy	Smartphone	Computer (Laptop)
Word accuracy	90.560%	92.064%
Sentence accuracy	64.000%	68.130%

Table 7 Common errors in prediction.

Common Error	Example		Possible Causes
	Expected	Predicted	
Missing word	<i>Berapa Harga Karcis Film Ini</i> (How much would a ticket for this movie cost?)	<i>Berapa Harga Film Ini</i>	the number of frames representing the word is too small (person's movement is too fast)
Wrong prediction	<i>Jam Berapa Dokter Datang</i> (At what time is the doctor expected to arrive?)	<i>Jam Berapa Dokter Pergi</i>	movements that look similar; the only difference is the direction of movement (MobileNetV2 does not have enough information to differentiate movement direction)
Generating additional word	<i>Berapa Harga Karcis Film Ini</i> (How much would a ticket for this movie cost?)	<i>Berapa-an Harga Karcis Film Ini</i>	non-gesture frame detected as gesture frame

5 Improvement Approach: Parallel Multi-Inference

We tried to improve the performance of MobileNetV2 by implementing parallel processing. Previously, the models used in this study (MobileNetV2, CRF, and LSTM) were all loaded into one inference. This experiment tested MobileNetV2 loaded into multiple inferences, precisely in the range of one to four inferences. The way the machine learning model works remained the same, but the processed data were divided equally among each MobileNetV2 inference as shown in Figure 10. The outputs of all these inferences were concatenated and forwarded to the next model. This approach can be used because the predicted results of MobileNetV2 from one frame to another do not affect each other, so each feature extraction can be run separately.

As shown in Figure 11, an implementation of parallel processing using two to four inferences can have a significant impact compared to MobileNetV2 using one inference. Significant changes occurred when the number of inferences was increased to two, resulting in a reduction in processing time by up to 56.64%.

The experiment was continued by choosing the two-inference parallel processing method. This method was then retested by comparing the overall time of the translation process between the SIBI translator using MobileNetV2 with one inference and the SIBI translator using two inferences, as well as the resources used for each translation.

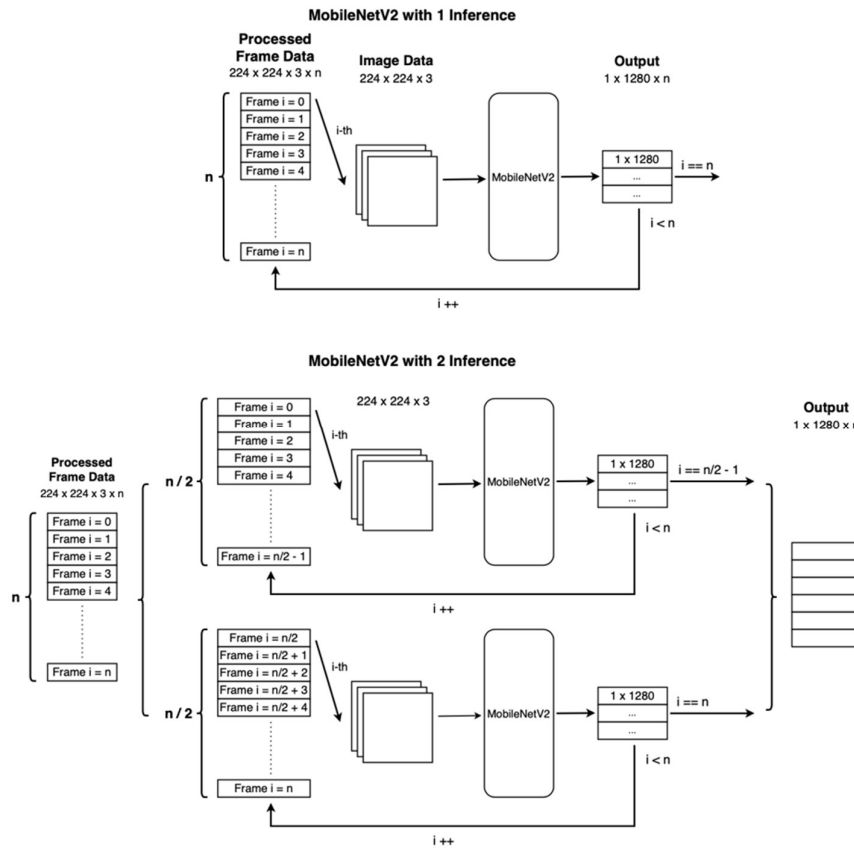


Figure 10 How MobileNetV2 multi-inference works.

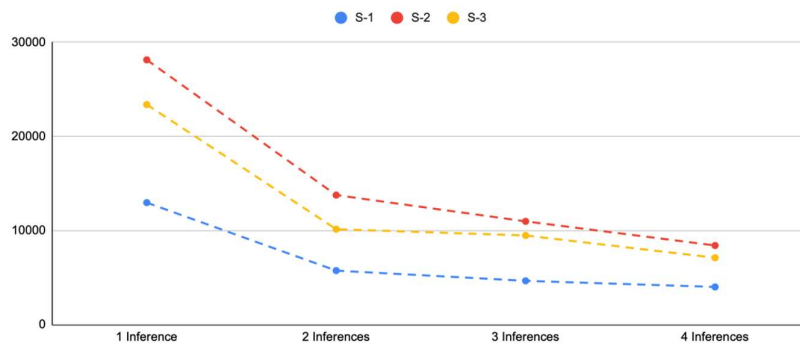


Figure 11 Number of inferences to processing time.

As shown in Table 8, the processing time is reduced by up to 9 s on average. However, CPU usage increased by about 8.4%. Memory usage does not change significantly and is still categorized as a heavy process. It can be concluded that the implementation of parallel processing can reduce the time required for translation while possessing the drawback of CPU usage.

Table 1 Translation processing time and resources usage.

Type	Sentence	Time	CPU (%)	Memory (GB)	Energy
Standard	<i>Siapa nama mu?</i> (What is your name?)	22 s 666 ms	65.1	0.627	Heavy
	<i>Saya sering sakit kepala, saya harus periksa ke bagian mana?</i> (I frequently get headaches, which medical specialty department should I visit?)	47 s 620 ms	64.5	0.693	Heavy
	<i>Di mana kami bisa mengambil tabungan?</i> (Where can we withdraw our savings?)	40 s 354 ms	64.9	0.654	Heavy
	Average	36 s 880 ms	64.8	0.658	Heavy
Enhanced MobileNetV2 (2 inference)	<i>Siapa nama mu?</i> (What is your name?)	15 s 229 ms	68.9	0.638	Heavy
	<i>Saya sering sakit kepala, saya harus periksa ke bagian mana?</i> (I frequently get headaches, which medical specialty department should I visit?)	37 s 460 ms	81	0.639	Heavy
	<i>Di mana kami bisa mengambil tabungan?</i> (Where can we withdraw our savings?)	30 s 338 ms	69.9	0.651	Heavy
	Average	27 s 675 ms	73.2	0.642	Heavy

6 Conclusion and Future Work

This study aimed to implement an SIBI translation system on a smartphone and to test several approaches for improving translation performance. This study used frozen models of three machine learning models, namely MobileNetV2, Conditional Random Field, and Long Short-Term Memory.

The SIBI translator application running on Android could translate sign language videos with 90.560% word accuracy and 64% sentence accuracy. The average time required to complete a translation was 31 s, which was 2.87 times slower compared to using translators running on a PC, with the majority of that time occurring in the preprocessing and MobileNetV2 stages.

Several experiments were conducted to improve the performance of preprocessing and MobileNetV2 in terms of processing time, namely by implementing parallel processing. Parallel processing was able to reduce the time in the MobileNetV2 stage by at least 51.08% using two inferences, which can still be increased. Parallel processing can be achieved as long as the prediction results of the model do not affect each other between one data and another.

This study made full use of the TensorFlow-Lite Java API, which is the most common and easiest to use for running machine learning models. The C++ API offers flexibility and better performance, so it could be used to speed up processing time. To use these APIs on Android, we had to create a Java Native Interface and Native Development Kit. The NDK guarantees higher performance and provides greater security because it prevents recompilation or decompilation.

The processing time can also be improved by leveraging an on-device accelerator using TensorFlow. It supports delegate such as a Graphics Processing Unit (GPU) and a Digital Signal Processor (DSP), which are already supported in some high-end devices. Power efficiency is also one of the benefits from using these delegates.

Acknowledgement

This work was supported by the Ministry of Research and Technology Research Grant PTUPT, Number NKB-258/UN2.RST/HKP.05.00/2021. This support is gratefully received and acknowledged.

References

- [1] Siswomartono, S., *Simple Way to Learn SIBI (Sign System for Indonesian Gesture)*, Jakarta: Indonesian National Federation of Welfare for the Deaf, 2007.
- [2] Rakun, E., Arymurthy, A.M., Stefanus, L.Y., Wicaksono, A.F. & Wisesa, I.W.W., *Recognition of Sign Language System for Indonesian Language using Long Short-Term Memory Neural Networks*, *Adv. Sci. Lett.*, **4**(2), pp. 400-407, 2016.
- [3] Rakun, E., *SIBI Gesture to Text Translation System on a Mobile Cellular Device*, Indonesia Patent No. IDP000078068, 2021.
- [4] Harits, M., Rakun, E. & Hardianto, D., *Feature Extraction from Smartphone Images by Using Elliptical Fourier Descriptor, Centroid and Area for Recognizing Indonesian Sign Language SIBI (Sistem Isyarat Bahasa Indonesia)*, in 2nd International Conference on Intelligent Autonomous Systems, 2019.
- [5] Shaik, K.B., Ganesan, P., Kalist, V., Sathish, B. & Jenitha, J.M.M., *Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space*, *Procedia Comput. Sci.*, **57**, pp. 41-48, 2015.
- [6] Viola, P. & Jones, M., *Rapid Object Detection Using A Boosted Cascade of Simple Features*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.
- [7] Kuhl, F.P. & Giardina, C.R., *Elliptic Fourier Features of a Closed Contour*, *Comput. Graph. Image Process.*, **18**(3), pp. 236–258, 1982.
- [8] Hochreiter, S. and Schmidhuber, J., *Long Short-Term Memory*, *Neural Comput.*, **9**(8), pp. 1735–1780, 1997.
- [9] Pratama, A., Rakun, E. & Hardianto, D., *Human Skeleton Feature Extraction from 2-Dimensional Video of Indonesian Language Sign System (SIBI [Sistem Isyarat Bahasa Indonesia]) Gestures*, in International Conference on Computing and Artificial Intelligence, 2019.
- [10] Tanibata, N., Shimada, N. & Shirai, Y., *Extraction of Hand Features for Recognition of Sign Language*, *Int. Conf. Vis. Interface*, pp. 391-398, 2002.
- [11] Lucas, B.D. & Kanade, T., *An Iterative Image Registration Technique with an Application to Stereo Vision*, in International Joint Conference on Artificial Intelligence, 1981.
- [12] Anggraini, K., Rakun, E. & Stefanus, L.Y., *Recognizing the Components of Inflectional Word Gestures in Indonesian Sign System Known as SIBI (Sistem Isyarat Bahasa Indonesia) by using Lip Motion*, in International Conference on Electrical Engineering and Informatics (ICEEI), 2019.
- [13] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S. & Pantic, M., *300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge*, in IEEE International Conference on Computer Vision Workshops, 2013.

- [14] Assael, Y.M., Shillingford, B., Whiteson, S. & Freitas, N.D., *LipNet: End-to-End Sentence-level Lipreading*, arXiv Learn., 2017.
- [15] Setyono, N. & Rakun, E., *Recognizing Word Gesture in Sign System for Indonesian Language (SIBI) Sentences Using DeepCNN and BiLSTM*, in International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2019.
- [16] He, K., Zhang, X., Ren, S. & Sun, J., *Deep Residual Learning for Image Recognition*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.
- [17] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.C., *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520, 2018.
- [18] Maulina, N. & Rakun, E., *Recognizing Finger spelling in SIBI (Sistem Isyarat Bahasa Indonesia) using OpenPose and Elliptical Fourier Descriptor*, in International Conference on Advanced Information Science and System, 2019.
- [19] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E. & Sheikh, Y., *OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*, IEEE Trans. Pattern Anal. Mach. Intell., **43**(1), pp. 172-186, 2021.
- [20] Rakun, E. & Setyono, N., *Improving Recognition of SIBI (Sign System for Indonesian Language) Word Gesture Performance by Combining Skeleton and Handshape Features*, Manusc. Submitt. Publ., 2021.
- [21] Shoalihin, R. & Rakun, E., *Audio Feature Extraction on SIBI Dataset for Speech Recognition*, in International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), 2021.
- [22] Baroi, O.L., Kabir, M.S.A., Niaz, A., Rakib, A.M., Islam, M.J. & Rahimi, M.J., *Effects of Different coefficients on MFCC and PLP for Bangla Speech Corpus using Tied-state Triphone Model*, in International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1-6, 2019.
- [23] Aulia, A., Rakun, E. & Hardianto, D., *Human Skeleton Feature Extraction from 2-Dimensional Video of Indonesian Language Sign System (SIBI [Sistem Isyarat Bahasa Indonesia]) Gestures* Title, in ACM Conference Proceedings, 2019.
- [24] Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. IEEE, **77**, pp. 257-286, 1989.
- [25] Halim, K. & Rakun, E., *Sign Language System for Bahasa Indonesia (Known as SIBI) Recognizer using TensorFlow and Long Short-Term Memory*, in International Conference on Advanced Computer Science and Information Systems ICACSIS, pp. 403-407, 2018.
- [26] Widhinugraha, I. & Rakun, E., *Indonesian Language Sign System (SIBI) Recognition Using Threshold Conditional Random Fields*, in 8th

- International Conference on Computing and Pattern Recognition, pp. 380–384, 2019.
- [27] Cho, S.S., Yang, H.D. & Lee, S.W., *Sign Language Spotting Based on Semi-Markov Conditional Random Field*, 2009 Work. Appl. Comput. Vision, WACV 2009, 2009.
- [28] Rakun, E., Widhinugraha, I. & Setyono, N., *Word Recognition and Automated Epenthesis Removal for Indonesian Sign System Sentence Gestures*, Manuscr. Submitt. Publ., 2021.