Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2024, Article ID 6479124, 13 pages
https://doi.org/10.1155/2024/6479124

Research Article

Indonesian Lip-Reading Detection and Recognition Based on Lip Shape Using Face Mesh and Long-Term Recurrent Convolutional Network

Aripin[1] **and Abas Setiawan**[2]

[1]*Department of Biomedical Engineering, Universitas Dian Nuswantoro Semarang, Semarang, Indonesia*
[2]*Department of Computer Science, Universitas Negeri Semarang, Semarang 50229, Indonesia*

Correspondence should be addressed to Aripin; arifin@dsn.dinus.ac.id

Received 29 November 2023; Revised 1 April 2024; Accepted 9 April 2024; Published 18 April 2024

Academic Editor: Ridha Ejbali

Communication through speech can be hindered by environmental noise, prompting the need for alternative methods such as lip reading, which bypasses auditory challenges. However, the accurate interpretation of lip movements is impeded by the uniqueness of individual lip shapes, necessitating detailed analysis. In addition, the development of an Indonesian dataset addresses the lack of diversity in existing datasets, predominantly in English, fostering more inclusive research. This study proposes an enhanced lip-reading system trained using the long-term recurrent convolutional network (LRCN) considering eight different types of lip shapes. MediaPipe Face Mesh precisely detects lip landmarks, enabling the LRCN model to recognize Indonesian utterances. Experimental results demonstrate the effectiveness of the approach, with the LRCN model with three convolutional layers (LRCN-3Conv) achieving 95.42% accuracy for word test data and 95.63% for phrases, outperforming the convolutional long short-term memory (Conv-LSTM) method. The proposed approach outperforms Conv-LSTM in terms of accuracy. Furthermore, the evaluation of the original MIRACL-VC1 dataset also produced a best accuracy of 90.67% on LRCN-3Conv compared to previous studies in the word-labeled class. The success is attributed to MediaPipe Face Mesh detection, which facilitates the accurate detection of the lip region. Leveraging advanced deep learning techniques and precise landmark detection, these findings promise improved communication accessibility for individuals facing auditory challenges.

1. Introduction

Speech is the most fundamental type of human communication that uses both visual and auditory elements. Vocalizations in the audio signal are represented by lip movements in speech. Although audio signals typically do a good job of conveying information, lip reading could be necessary in some circumstances, particularly in noisy areas where audio understanding might be compromised. Lip reading can interpret speech based only on visual cues and has recently attracted significant attention due to its possible uses in language identification [1], emotion recognition [2], and human-computer interaction [3].

Lip-reading applications that only identify lip movements are considered more respectful of individual privacy by not including speech during communication [4]. The impact may reduce concerns about the misuse of speech datasets. In addition to that, it is also very useable for deaf people during communication [5]. However, it may be difficult to interpret lip movements effectively, especially when there are similarities in the forms of lips of distinct words or when there are outside influences such as background noise [6].

In this regard, advances in technology present viable ways to interpret lips using visual information recorded by a camera [7, 8]. The goal of conventional machine learning methods for lip reading is to identify temporal patterns in data streams. Many researchers employ deep learning models for lip reading in addition to advancing machine learning into deep learning [9]. The key to achieving

a successful recognition system in lip reading is the precise detection of the lip region and the subsequent classification of utterances, a task influenced by factors such as language, dialect, and individual lip structure. Although some studies have developed language-specific lip-reading systems tailored to distinct regions [7, 10–14], the diversity of lip shapes and external noise poses challenges to effective detection and classification algorithms.

This study addresses these challenges by proposing an Indonesian lip-reading dataset, as well as a detection and recognition system based on the diversity of lip shapes considering the unique characteristics of eight types of lip shapes [15]. The MediaPipe face mesh [16] was used for the detection and elimination of surrounding noise, and the long-term recurrent convolutional network (LRCN) [17] for the classification of utterances.

The structure of this paper is organized as follows. Section 1 provides an introduction to the background and challenges of lip reading. Section 2 reviews related works and highlights the contributions of our research. Section 3 describes materials and methods, including data acquisition, detection algorithms, preprocessing techniques, and model development. Section 4 presents the experimental results and a discussion of various scenarios. Finally, Section 5 concludes our findings and discusses future directions.

2. Related Works

Lip-reading applications utilize image processing, machine learning, and deep learning to understand spoken words through lip movements. An eigenlip model has been proposed that calculates the Euclidean distance between the upper and lower lips, along with the hidden Markov model (HMM), for word prediction [18]. In addition, neural network models have been developed for the classification of laughter speech, using limited audiovisual mapping [19]. Lin et al. achieved an accuracy rate of 80% in predicting vowel utterances [20], while bidirectional long short-term memory (Bi-LSTM) models were used for visual speech recognition [21]. However, distinguishing silent speech from whispered speech remains a challenge. The bidirectional gated recurrent unit (Bi-GRU) extracts features for audiovisual recognition but struggles in noisy environments [22]. Convolutional neural networks (CNNs) with the pretrained VGG-16 model [23] and LSTM combinations with a histogram of oriented gradients and a support vector machine (HOG + SVM) have been proposed for spoken word recognition [23].

Recent advancements in lip reading involve deep learning algorithms, such as convolutional neural networks (CNNs). Martinez et al. improved word-by-word lip reading using multiscale temporal convolutional networks (MSTCNs) [24]. Koumparoulis and Potamianos introduced efficient networks for lip reading, achieving high accuracy levels in the lip-reading in the wild (LRW) dataset [25]. Visual speech recognition (VSR) models have also been developed, surpassing previous methods in accuracy [26]. However, these studies predominantly focus on English-language datasets.

Language-specific datasets are crucial for accurate lip reading. Recent efforts include German, Mandarin, Turkish, and Indonesian lip-reading systems. German lip-reading system achieved an accuracy rate of 88% [27], while the Mandarin system reached 61.18% accuracy using 3D-CNN with DenseNet + LSTM model [28]. Atila and Sabaz developed a Turkish lip-reading system with a Bi-LSTM model that achieves 85% accuracy for words and 91% for sentences [29]. Indonesian lip-reading system, although limited to only 50 sentences, reached an accuracy rate of 80% for syllable classification using 3D-CNN and Bi-GRU models [30].

Along with the related work, the lack of Indonesian lip-reading datasets, especially for word and phrase levels, encourages this study to be able to make datasets open publicly available datasets for researchers. The new dataset consisted of 10 words and four phrases considering eight different lip shapes. To improve the detection and classification accuracy, this study also proposed state-of-the-art detection algorithms and deep learning models to close the gap. The MIRACL-VC1 dataset [7] with word samples is also considered to test our algorithm framework. In summary, this research has the following contributions:

(1) This study presents the first open-ended dataset for lip reading called IndoLR with an Indonesian language data sample consisting of several words and phrases, considering eight different types of lip shapes.

(2) The MediaPipe Face Mesh [16] is used to obtain lip ROI, which is then trained with the long-term recurrent convolutional network (LRCN) in the Indonesian lip-reading dataset, which produces an accuracy of more than 94% compared to the convolutional LSTM (Conv-LSTM) model.

(3) The proposed framework has also been applied to an available public dataset called MIRACL-VC1 [31], achieving an accuracy of 90.67 and an $F1$ score of 91 with the best LRCN model in the word-labeled class. This performance showed a good result compared to previous studies.

A preprint has previously been published and has not yet been peer-reviewed [32]. The updated work used MediaPipe Face Mesh to detect the lip and the LRCN architecture. Subsequently, the proposed method was also evaluated in the MIRACL-VC1 dataset [7] in this study.

3. Materials and Methods

In general, the proposed system is presented in Figure 1. First, the data acquisition is carried out to collect the isolated video data. Every video will be captured in a close-up fashion (frontal). Second, lip detection is performed by using MediaPipe Face Mesh [16]. Third, the video is preprocessed by extracting the videos into image frames for the training process. Fourth, building the LRCN model and training were conducted to recognize the utterance visually.

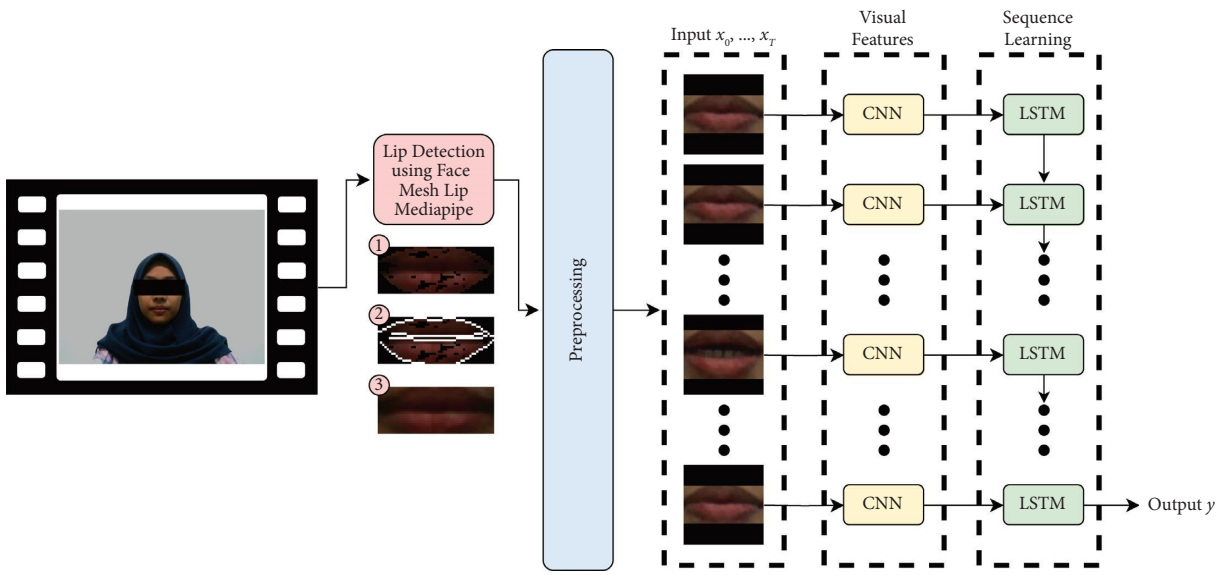


Figure 1: Proposed system.

3.1. Data Acquisition. There are different types of human lips, so it is necessary to have a reference for the types of lip shapes common to humans. Reference to the shape of the lip from [15] has been adapted as shown in Figure 2. From the types of lip shapes, it is hoped that it can represent the overall shape of the human lips. Five women and three men participated in the production of these data. Each person represents each type of suitable lip shape. These types of lips are neutral, pointly neutral, thin, cupid's bow, uni-lip, beestung, smear, and glamour.

Each person speaks ten words and four phrases. The ten words are "maaf" (sorry), "tolong" (please), "permisi" (excuse me), "halo" (hello), "mulai" (start), "berhenti" (stop), "lanjut" (next), "sakit" (hurt), "kembali" (back), and "awas" (be careful). Meanwhile, the four phrases spoken are "terima kasih" (thank you), "minta tolong" (please help), "saya minta maaf" (I am sorry), and "saya minta tolong" (I am asking for help). These words and phrases were chosen because Indonesians often use them. These words and phrases were chosen because they are often used in Indonesian language communication and reflect politeness.

Every word or phrase is recorded using a Logitech C525 camera with an 8-megapixel resolution and a standard PC to process the recording. The video captured is in MP4 format with a resolution of 480p (640×480) with a total frame rate of 30 FPS for the ten words and four phrases collected. The different settings were made due to the limitation of the machine to process each video. For every word sample, it takes 30 videos per person. Thus, the total data collected for the word dataset is 2400. In the phrases dataset only contains four phrases category, then the additional samples are gathered to 50 videos for each person. The total data collected for the phrase dataset are 1600. All these collected video samples are then called IndoLR (Indonesian lip-reading dataset).

The study of lip-reading was developed not only in one language. Each language has a different way of pronouncing the other, leading to further variations. So, some countries build their datasets, as shown in Table 1. The dataset taken from this investigation is also compared with another available dataset. Compared to some publicly available (or with limited access) datasets, IndoLR is the only publicly available dataset with the most data in Indonesia. In the research by Kurniawan and Suyanto [30], there are very few data samples due to the focus of the classification on syllables. In addition, the resolution provided in our dataset is also quite large compared to other studies. Although the number of data samples is not as large as in most recent studies (LRW [10], LRS2 [33], LRS3-TED [11], GLips [13], Turkish [29], CMLR [34], CN-CVS/Speech [35], and OLKAVS [14]), this study considers the shape of the lip type depicted in Figure 2. All speakers in IndoLR have sample representations of the previously mentioned lip-type shapes, with each type represented by one speaker.

In this study, the MIRACL-VC1 dataset [7] with word samples is considered to test our algorithm framework. MIRACL-VC1 is an openly available dataset with two sample types: color and depth. In this study, the total number of word data is 1500 utterances with word labels such as begin, choose, connection, navigation, next, previous, start, stop, hello, and web. Several researchers have also benchmarked the dataset to compare it with lip-reading studies.

3.2. Detection and Preprocessing. Detecting the position of the lip on the face of a person using computer technology is not easy. This difficulty occurs because the human lip is a small part of the human face that is considered to resemble the eyes and nose. There are many ways to detect faces, such as traditional machine learning [36] and deep learning [16, 37], which can detect human faces effectively. This study tried to use one of the most effective methods, the Haar cascade, HOG-SVM, or MediaPipe, to recognize the lip. In early-stage experiments, using the Haar cascade method, the intention was to detect the lip but sometimes not only that

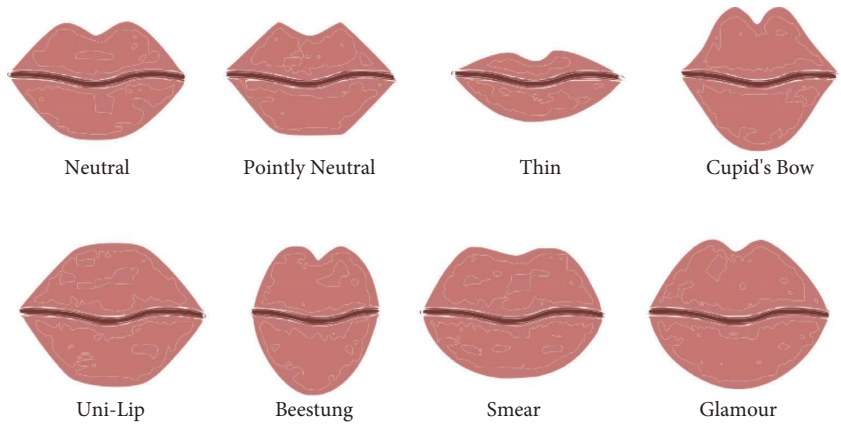


Figure 2: Types of lip shape [20].

Table 1: Multilingual dataset compared to IndoLR.

Dataset	Language	Year	Isolated	Form segment	Speakers	Classes	Total data	Resolution	Pose
MIRACL-VC1 [7]	English	2014	v	Words	15	10	1500	640 × 480	Frontal
MIRACL-VC1 [7]	English	2014	v	Sentences	15	10	1500	640 × 480	Frontal
OuluVS2 [12]	English	2015	v	Sentences	20	10	1000	720 × 576	Frontal
LRW [10]	English	2017	x	Words	>1000	500	400000	256 × 256	−30~30
LRS2 [33]	English	2017	v	Sentences	>1000	17428	118116	160 × 160	−30~30
LRS3-TED [11]	English	2018	v	Sentences	>1000	70000	165000	224 × 224	−90~90
GLips [13]	German	2022	x	Words	100	500	250000	256 × 256	Frontal
Turkish [29]	Turkish	2022	v	Words	Unspecified	111	39960	60 × 35 (30–60 FPS)	Frontal (10 rot)
Turkish [29]	Turkish	2022	v	Sentences	Unspecified	113	27120	60 × 35 (30–60 FPS)	Frontal (10 rot)
CMLR [34]	Mandarin	2020	v	Sentences	11	9	102076	64 × 128	Frontal
CN-CVS/Speech [35]	Mandarin	2023	x	Sentences	2529	~75	193,329	640 × 480	Natural
OLKAVS [14]	Korean	2023	v	Sentences	1107	>100	250000	1920 × 1080	0,45,90
Indo [30]	Indonesia	2020	v	Sentences	10	5	50	Unspecified	Frontal
IndoLR	Indonesia	2023	v	Words	8	10	2400	640 × 480 (30 FPS)	Frontal
IndoLR	Indonesia	2023	v	Sentences	8	4	1600	640 × 480 (30 FPS)	Frontal

Iso, isolated; v, isolated; x, continuous.

region but also small objects such as eyes, nose, and neck folds. Haar cascades can recognize faces effectively, but small things, such as the lip, are difficult to detect. Error detection was also proven in the study by [38], where by using the Haar cascade method, it was difficult to find and obtain ROI from the eyes. However, when it is collected in the dataset, it will cause noise.

There are other methods for detecting the lip more accurately: King [36] and MediaPipe [16]. These two methods can provide information on lip landmarks taken from facial landmarks. The Dlib uses the HOG-SVM algorithm to provide 68 landmark points in the facial image. In addition, the MediaPipe Face Mesh can estimate 468 3D landmark points on the face. Moreover, MediaPipe performance is better than Dlib when it detects local or small features of the face, including the lip. MediaPipe is also faster than Dlib in detecting the landmark of a facial image [39]. Moreover, in the study by Ishmam et al. [40], MediaPipe has better performance than Dlib in the isolation of lip from various face conditions such as angle, appearance, and lighting. In this case, the MediaPipe Face Mesh was considered as a method to detect the lip region.

The MediaPipe Face Mesh can track the lips and details of the tongue, teeth, and gums. The final image was cropped only for the lip region because there are some noises such as whiskers, chin, beaver, and nose which are close to the lip. There are three steps to detect the lip. The first is collecting the 40 landmark points from the 68 facial landmarks. Every landmark point $\mathrm{LP}_{x,y}$ has the x and y positions in 2D space. It is associated with another landmark point to create the line between the two points. Unfortunately, the detected landmark points are not ordered and must be ordered.

The second is finding the coordinate points within the dimension of the image. The calculation of the relative source point is $\mathrm{RPs}_{x,y} = \mathrm{LPs}_x * \mathrm{Img}_{\mathrm{width}}, \mathrm{LPs}_y * \mathrm{Img}_{\mathrm{height}}$, where the LPs_x and LPs_y are the landmark source point as well as $\mathrm{Img}_{\mathrm{width}}$ and $\mathrm{Img}_{\mathrm{height}}$ are the width and height of the image, respectively. Subsequently, a similar calculation is also measured for the relative target point $\mathrm{RPt}_{x,y} = \mathrm{LPt}_x * \mathrm{Img}_{\mathrm{width}}, \mathrm{LPt}_y * \mathrm{Img}_{\mathrm{height}}$. Thus, the routes between the source and the target point can be stored to find the edge of the lip. The third step is to extract the region of interest on the lip by creating a boundary box around the border. The boundary box can be calculated by using the

minimum and maximum indexes of the route to mask the lip. Since it is spatially impossible for a lip region image to produce the exact width and height dimensions, the gaps will be filled with black color.

The preprocessing is performed before the data enter as input to the network. First, each frame image is resized to 80×80. Second, the sequence length is determined. The sequence length determined for the word dataset is 30 frames, while for the phrase dataset, it is 40 frames. The image frames are not taken from frame index 0 but from the middle. Clipping of frame images from the middle index in isolated videos is carried out by considering the presence of stillness at the beginning of the video and at the end of the video, which can cause bias in the training process. If the number of frames in a video is more than the sequence length, then the silence at the beginning and end can be eliminated so that the focus is on the situation where the lips move to speak. Meanwhile, if the speaker speaks too fast, the number of frames will be less than the sequence length, resulting in a lack of image samples. This can be circumvented by adding an image that contains a fully black-colored or a black-padded image. Applying black-padded images as blank images preserves the temporal structure of the original sequence of frames and prevents information loss during training. It also ensures that all sequences are of the same length, maintaining uniformity in sequence processing. This straightforward process does not require complex processing steps (which is possibly computationally inefficient), making it accessible to operate in a fixed-length sequence when trained later on. Figure 3 depicts an illustration of the frames with the clipping sequence in the middle. It is hoped that this strategy can focus more data on the situation when the lips are speaking and will not be too affected by the speed of speech. Third, pixel frame normalization is performed to reduce the computation. Normalization produces a range of 0–1, dividing each value of pixels by 255.

After the data were preprocessed, it was split into three parts, training set, validation set, and test set for words and phrases. The compositions for each part are 80 : 10 : 10 percent. Every single class in the word or phrase datasets for each part contains every person sample. This is necessary so that the data can be distributed evenly.

3.3. Building the LRCN Model. Machine learning and deep learning are suitable methods that can be used as modern lip-reading techniques. In this study, the long-term recurrent convolutional network (LRCN) was used to train data on lip reading in the Indonesian lip-reading dataset. LRCN has been used successfully in action recognition, where each frame-frame video sequence used as a network input can be appropriately identified with the output activity associated with the video [17]. Related to this, we used LRCN to recognize what words or phrases are spoken, obtained from frame-to-frame sequence data from an uttered speech video. In LRCN, CNN and LSTM layers are combined in a single model. In this case, CNN will act as a spatial feature extraction from the frame, which will then be fed to the LSTM at each time step for temporal sequence modeling. Thus, direct training can be conducted to study spatiotemporal features end-to-end by producing a robust model.

Previously, the video data have been transformed into a sequence of image frames containing the lip area by cropping based on landmark detection using MediaPipe. After that, the image frames will be preprocessed to be a ready-to-train dataset. Every sequence of image samples with it is labeled and then collected into the words and phrases datasets. The LRCN model aims to bring those sequential inputs to static outputs that represent the word or phrase label $\langle x_1, x_2, x_3, \ldots, x_T, y\rangle \longmapsto y$. Any data that have gone through preprocessing at a specific time frame x_t will be trained up to the length of the T frame of the time sequence, which is then considered as input. The static output is a single y label that contains the word class $y \in \mathbb{R}^3$ or the sentence $y \in \mathbb{R}^4$.

Each input x_t, trained in CNN with three convolution layers, max pooling, dropout, and flatten, is wrapped by a distributed temporal layer. The isolation of each sequence of video frames is passed through the feature transformation $\phi_V(x_t)$. The details of the first convolution layer have 16 feature maps with 3×3 kernels and rectified linear unit (ReLU) activation functions [41, 42]. The ReLU function can produce nonlinear constraints on the input $\max(0, f_z(x_t, W))$, where $f_z(.)$ is a linear function resulting from the input x_t, and the weight parameter is W. The first layer of the pool is the max pool with 2×2, followed by a dropout with a dropout rate of 0.25. The second convolution layer has 32 feature maps with configurations and is accompanied by the same pooling and dropout layers as the first. Likewise, for the third layer, it is also the same. The only difference is the number of feature maps in the third convolution layer, that is, 64. Subsequently, before entering the LSTM layer, there is a flatten layer to convert the spatial output values into vectors.

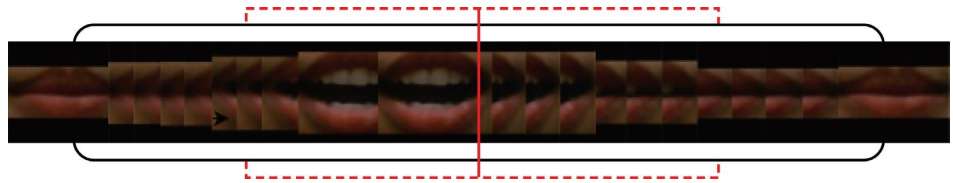
Then, there is one layer of LSTM with a total of 16 cells. Usually, in its general form, the LSTM model has a weight parameter W by mapping the input x_t and the previous time-step hidden state h_{t-1} with two outputs, namely, the nonlinear calculation output z_t and the updated hidden state h_t. In Figure 1, the LSTM sequence learning will be carried out by passing the h_{t-1} output to h_t. We calculate the first hidden layer sequence $h_1 = f_w(x_1, 0)$, where $h_0 = 0$ because there is no previous hidden layer output. Then, we calculate the second hidden layer sequence $h_2 = f_w(x_2, h_1)$ and so on, so the hidden layer output at the current time step is $h_t = f_w(x_t, h_{t-1})$. On LSTM to produce h_t, it needs to calculate the following:

$$i_t = \sigma\left(W_{\mathrm{xi}} x_t + W_{hi} h_{t-1} + b_i\right), \tag{1}$$

$$f_t = \sigma\left(W_{\mathrm{xf}} x_t + W_{\mathrm{hf}} h_{t-1} + b_f\right), \tag{2}$$

$$o_t = \sigma\left(W_{\mathrm{xo}} x_t + W_{\mathrm{ho}} h_{t-1} + b_o\right), \tag{3}$$

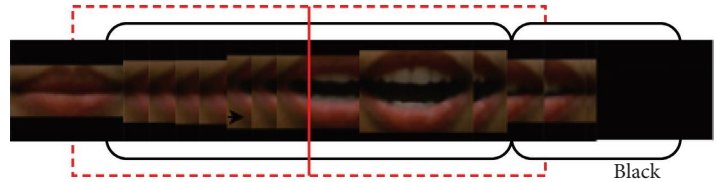
$$g_t = \tan h\left(W_{\mathrm{xc}} x_t + W_{\mathrm{hc}} h_{t-1} + b_c\right), \tag{4}$$



If no. of frames is more than the sequence length

Actual Frame Index: 0 1 2 3 4 5 6 7 8 9 .... 33 34 35 36 37 38 39

Selected Frame Index: 6 7 8 9 .... 33 34



If no. of frames is less than the sequence length

Actual Frame Index: 0 1 2 3 4 .... 23 24

Selected Frame Index: 0 1 2 3 4 .... 23 24 25 26 27 28 29

Figure 3: Clipping of the middle frame to select the sequence of data.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{5}$$

$$h_t = o_t \odot \tan h(c_t). \tag{6}$$

LSTM consists of an input gate $i_t \in \mathbb{R}^N$, forget gate $f_t \in \mathbb{R}^N$, output gate $o_t \in \mathbb{R}^N$, input modulation gate $g_t \in \mathbb{R}^N$, and the memory cell $c_t \in \mathbb{R}^N$, where N is the hidden units. The input gate, the forget gate, and the output gate use the nonlinear sigmoid nonlinear function $\sigma(x) = (1 + e^{-x})^{-1}$. Meanwhile, in the modulation gate input and updated hidden state calculations, there is a nonlinear function hyperbolic tangent $\tan h(x) = e^x - e^{-x}/e^x + e^{-x}$. The $\odot$ operator symbol is the elementwise product of two vectors. Then, the dropout was performed [43] to minimize overfitting gaps that may occur during training. To classify the distribution of $P(y_t)$ in the output layer to the desired label results ($D \in \mathbb{R}^3$ on word labels and $D \in \mathbb{R}^4$ in phrase labels), many classes are classified, and the predicted distribution $P(y_t = d)$ uses the softmax function as

$$P(y_t = d) = \text{softmax}(\hat{y}_t) = \frac{\exp(\hat{y}_t, d)}{\sum_{d' \epsilon D} \exp(\hat{y}_t, d')}. \tag{7}$$

The number of units in the output layer corresponds to the number of word and phrase labels in the two datasets. Therefore, there will be two LRCN architectures that have a different number of output layers. The word dataset has three units in the output layer, whereas the phrase dataset has four. The loss function used to evaluate this network is the categorical cross entropy L with the calculation as follows:

$$L = -\sum_{i=1}^{M}\sum_{d=1}^{D}(y_{\text{id}} * \log(P(y_{\text{it}} = d))), \tag{8}$$

where M is the number of data samples and y_{id} is the output corresponding to the current data label. Adam optimizer [44] updates the weight parameter that stores the classification pattern. The standard neural network training cycle is used to perform forward propagation, calculate the loss function and backpropagation over time, and update the weight parameters.

4. Results and Discussion

The deep learning model used to test the IndoLR and MIRACL-VC1 dataset is not only compared with LRCN but also compared with convolutional LSTM (Conv-LSTM) [45]. Table 2 shows the architectural details of the three neural network models for the experimental scenarios. The three architectures are compared using the softmax activation function in the output layer and the Adam optimizer. The testing will be carried out on a test set that previously went through the same data acquisition process as the training set. The machine used to carry out the training is a consumer-grade CPU with an Intel I5 10th-gen processor with an RTX 3060 Ti GPU and 32 GB RAM.

Every video sample in the IndoLR dataset with the three applied network architecture scenarios was trained in 100 epochs. The hyperparameter settings for the overall experiments are the learning rate of 0.0005 and the batch size of 4. Since the training set is not large and there is a limitation on the consumer-grade computer to perform the training

Table 2: Neural network architecture.

Model	Detailed architecture	
	Words	Phrase
Conv-LSTM	ConvLSTM2D (8) MaxPooling3D () ConvLSTM2D (16) MaxPooling3D () Flatten () Dense (10)	ConvLSTM2D (8) MaxPooling3D () ConvLSTM2D (16) MaxPooling3D () Flatten () Dense (4)
LRCN-2Conv	TimeDistributed (Conv2D (16)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (32)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Flatten ()) LSTM (64) Dropout () Dense (10)	TimeDistributed (Conv2D (16)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (32)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Flatten ()) LSTM (64) Dropout () Dense (4)
LRCN-3Conv	TimeDistributed (Conv2D (16)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (32)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (64)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Flatten ()) LSTM (64) Dropout () Dense (10)	TimeDistributed (Conv2D (16)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (32)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Conv2D (64)) TimeDistributed (MaxPooling2D ()) TimeDistributed (Dropout ()) TimeDistributed (Flatten ()) LSTM (64) Dropout () Dense (4)

process, the batch size of 4 was chosen. In Conv-LSTM scenarios, max pooling 3D is added to reduce the complexity of the model. The number of convolutional layers in LRCN is limited to up to three layers. The consumer-grade GPU has 8–32 GB memory, but, in this case, it only used 8 GB on RTX 3060 Ti.

In terms of time complexity, LRCN is based on the operations performed in its convolutional and recurrent layers during both training and inference. Convolutional layers typically have a time complexity of $O(N^2 * C^2 * X_{\text{in}} * X_{\text{out}})$, where $N \times N$ is the image dimension, C is the size of the convolutional kernel, X_{in} is the number of input channels, and X_{out} is the number of output channels [46]. In the recurrent layer using the LSTM, it has a time complexity of $O(T * M^2)$, where T is the number of frame sequences in a video and M is the hidden state size. LSTM uses the parameters associated with each gate operation that typically include weight matrices of dimension $N \times N$ (or $M \times 4M$ in the case of all gates combined). During the computation of each gate, these weight matrices are multiplied by the input or hidden state vectors, resulting in a computational complexity proportional to M^2 for each gate operation. Consequently, the time complexity of LRCN is dominated by the sequential processing of frames through the convolutional layers followed by the LSTM layers, resulting in a combined time complexity of $O(N^2 * C^2 * X_{\text{in}} * X_{\text{out}} + T * M^2)$.

Unlike LRCN, Conv-LSTM time complexity is determined by the operations performed within its convolutional and recurrent layers. At each time step, Conv-LSTM involves convolutional operations followed by recurrent operations. The time complexity of the convolutional layers in Conv-LSTM is the same as that of LRCN. Recurrent layers within Conv-LSTM must be LSTM units with the time complexity of $O(M^2)$. The overall time complexity of Conv-LSTM for processing a sequence of length T is represented as $O(T * (N^2 * C^2 * X_{\text{in}} * X_{\text{out}} + M^2))$. As both architectures share similarities due to their integration of convolutional and recurrent layers, Conv-LSTM focuses on capturing spatial and temporal dependencies simultaneously within each time step, while LRCN typically processes sequences through separate convolutional and recurrent stages. Therefore, the exact time complexity may vary depending on factors, namely, the design of architecture, the characteristics of data, and implementation details.

Of the three architectures, Conv-LSTM requires a longer training time than the other two architectures. Conv-LSTM uses a special architecture to combine CNN and LSTM in recurrent steps. In LRCN, there is a TimeDistributed layer performed on every time slice for a warped certain layer. No recurrence process is going on in a TimeDistributed layer. Overfitting occurs in the word dataset with Conv-LSTM, where there is a significant gap between the accuracy of the

training data and the accuracy of the validation data, as shown in Figure 4(a). Meanwhile, the two LRCN architectures, namely, LRCN-2Conv and LRCN-3Conv, look more stable compared to Conv-LSTM. These models are shown in Figure 4(b) for the LRCN-2Conv model and Figure 4(c) for the LRCN-3Conv model. Furthermore, the training and validation data accuracy gap of the LRCN-3Conv model is smaller than that of the LRCN-2Conv model. The result shows that the accuracy of the LRCN-3Conv model is more stable than that of the LRCN-2Conv model.

Overfitting also occurs in the phrase dataset starting at the 10th epoch in the Conv-LSTM architecture. There is a significant gap between the accuracy of the training data and the validation data for the Conv-LSTM model shown in Figure 5(a). The gap between the accuracy of the training data and the validation data in the LRCN model, namely, LRCN-2Conv and LRCN-3Conv, appears to be more stable, as shown in Figures 5(b) and 5(c). In general, the gap between the validation accuracy of the training set and the validation set in the word dataset is better than the gap between the accuracy of the training data and the validation data in the phrase dataset, as shown in Figures 4 and 5.

When applied to the test set, the performance results of the deep learning models are shown in Table 3. The two LRCN models produce higher accuracy than Conv-LSTM, with a difference of 2.5–5% for the word dataset and 4.38–5.01% for the phrase dataset. LRCN achieved the highest accuracy with three convolution layers in the word and phrase datasets (LRCN-3Conv). The training time of the Conv-LSTM model is longer than that of the LRCN. It is around 10 times longer. It is because of the involvement of convolutional and recurrent operations at each time step rather than passing the convolution operation first and then followed by the recurrent operation. It also affects the recognition time for each video sample, where it needs a longer time than LRCN even in real-time situations. The best accuracy was achieved by LRCN with three convolution layers in both the word and phrase datasets better than LRCN with two convolutional layers and Conv-LSTM.

The receiver operating characteristic (ROC) and area under the ROC curve (AUC) were provided to evaluate the performance of the model. Due to the unknown cost of misclassification and class distribution during the training phase, this statistical metric is preferred. In this case, the ROC and AUC are applied to the multiclass classification problem. Therefore, the one-vs-rest (OvR) approach was used to distinguish between one class and other classes. The evaluation of ROC and AUC for each model scenario is presented in Figures 6 and 7 for the datasets of words and phrases, respectively. In either the word or phrase dataset, the model scenarios show a good performance of AUC, which is close to 1. There are no classes in all scenarios near 0.5, which means poor separability between classes. However, the LRCN performs better on separability than Conv-LSTM in both word and phrase datasets. After looking at the AUC result of the LRCN-3Conv in a word dataset, there is an interesting finding that in LRCN-3Conv, the words "permisi" and "berhenti" are successfully distinguished, although in the first sequence of image frames, it has a similar vowel sound of "*p*" and "*b*." By conducting full training on the word instead of per syllable, the algorithm focuses not only on a certain frame but on the overall sequence of frames.

The performance of the Conv-LSTM, LRCN-2Conv, and LRCN-3Conv models was also evaluated using precision, recall, and *F*1 score. Formulas to calculate precision, recall, and *F*1 score are presented in equations (9)–(11). Finally, precision, recall, and *F*1 score measurements are applied to the word and phrase datasets. A test result that accurately detects the existence of a condition is called a true positive (TP). True negative (TN) is an alternative term for a test result that correctly foretells the absence of a circumstance. A test result that falsely suggests the presence of an attribute is known as a false positive (FP). The result of a test that incorrectly implies that a particular circumstance is not present is called a false negative (FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$F1\ \text{score} = 2 \cdot \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{11}$$

Precision is used to measure the level of accuracy between the actual value and the predicted value. Then, recall aims to calculate the ratio between TP and TP + FN. Meanwhile, the *F*1 score is used to calculate the average precision and recall. These additional performance calculations ensure that the model has feasible accuracy and sensitivity and are presented in Table 4 for the word dataset and Table 5 for the phrase dataset.

Table 5 shows that the average *F*1 score values for the LRCN-3Conv, LRCN-2Conv, and Conv-LSTM models are 91%, 95%, and 96%, respectively. The LRCN-3Conv model performs better than the LRCN-2Conv and Conv-LSTM models. Based on the comparison of Tables 4 and 5, in general, the model performs better on the phrase dataset than on the word dataset. The number of classes is taking a role because, in the phrase dataset, only four classes are compared with 10 classes in the word dataset.

The algorithm method proposed in this study is also applied to the MIRACL-VC1 public dataset only with the word-labeled data. The preprocessing stage and the LRCN use the same approach as applied in IndoLR. The MIRACL-VC1 dataset has fewer images for each class, as it is captured at 15 frames per second with a sequence length range of 4–27 image frames. In the experiments carried out, the length of the sequence frame determined is 12 frames.

ROC and AUC were also evaluated for each class in the MIRACL-VC1 dataset with two different LRCN models as shown in Figure 8. There is no significant difference between LRCN with two convolutional layers and LRCN with three convolutional layers. However again, the increased number of convolutional layers in LRCN has better separability as proved by the better result of AUC in most labels of classes. The accuracy performance results

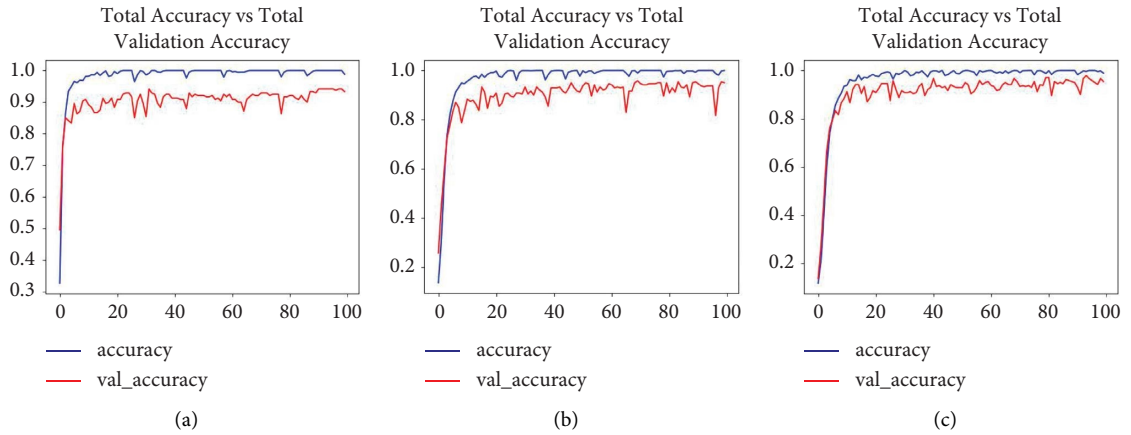


Figure 4: Results of the training accuracy of the word dataset. (a) Conv-LSTM model. (b) LRCN-2Conv model. (c) LRCN-3Conv model.

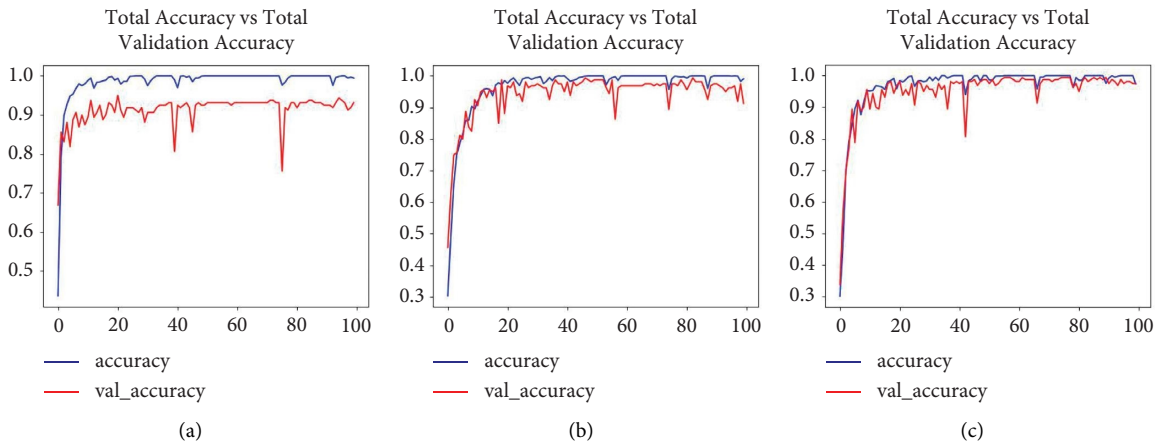


Figure 5: Training accuracy result of the phrase dataset. (a) Conv-LSTM model. (b) LRCN-2Conv model. (c) LRCN-3Conv model.

Table 3: Performance results of IndoLR.

Model	Words				Phrase			
	Val. acc. (%)	Test acc. (%)	Tt (s)	Rt (ms)	Val. acc. (%)	Test acc. (%)	Tt (s)	Rt (ms)
Conv-LSTM	94.7	90.42	7996.6	±88	95	90.62	9509.8	±105
LRCN-2Conv	95.83	92.92	600.3	±64	99.37	95.00	539.6	±68
LRCN-3Conv	97.92	95.42	727.3	±62	99.37	95.63	585.9	±66

*Tt, training time in seconds; Rt, average recognition time for each video sample in milliseconds.

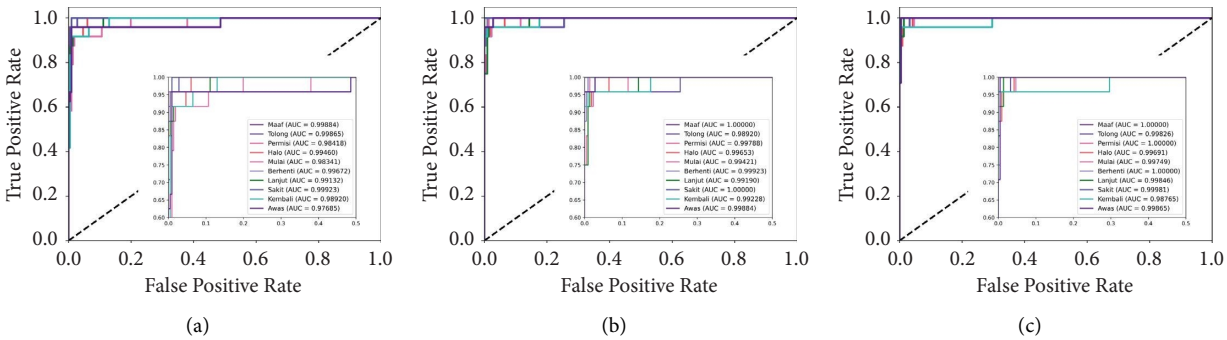
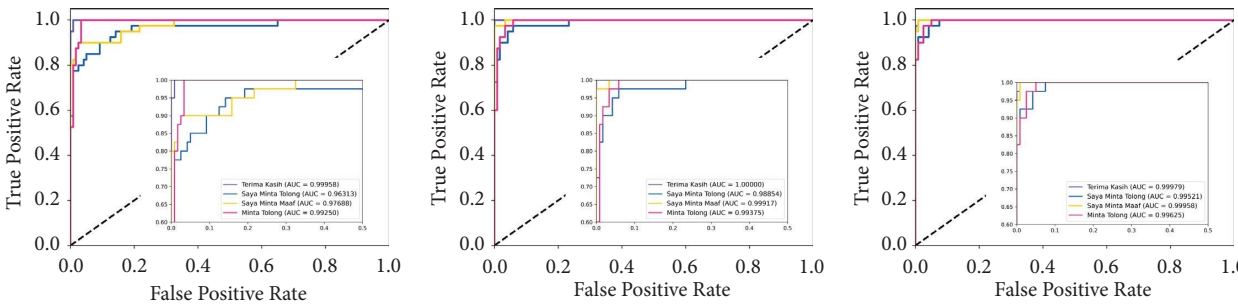


Figure 6: The ROC and AUC for each class in the word dataset. (a) Conv-LSTM model. (b) LRCN-2Conv model. (c) LRCN-3Conv model.



(a) (b) (c)

Figure 7: The ROC and AUC for each class in the phrase dataset. (a) Conv-LSTM model. (b) LRCN-2Conv model. (c) LRCN-3Conv model.

Table 4: Results of the performance evaluation (precision, recall, and *F*1 score) on the word dataset.

Words	Conv-LSTM			LRCN-2Conv			LRCN-3Conv		
	Precision (%)	Recall (%)	*F*1 score (%)	Precision (%)	Recall (%)	*F*1 score (%)	Precision (%)	Recall (%)	*F*1 score (%)
Maaf	96	96	96	100	100	100	100	100	100
Tolong	85	96	90	88	96	92	96	96	96
Permisi	96	96	96	92	96	94	100	96	98
Halo	91	88	89	96	92	94	91	88	89
Mulai	89	71	79	100	83	91	96	92	94
Berhenti	89	100	94	96	96	96	96	100	98
Lanjut	95	75	84	90	79	84	88	92	90
Sakit	92	96	94	92	100	96	96	96	96
Kembali	88	92	90	96	92	94	96	96	96
Awas	85	96	90	82	96	88	96	100	98
Accuracy			90			93			95
Macro avg	91	90	90	93	93	93	95	95	95
Weighted avg	91	90	90	93	93	93	95	95	95

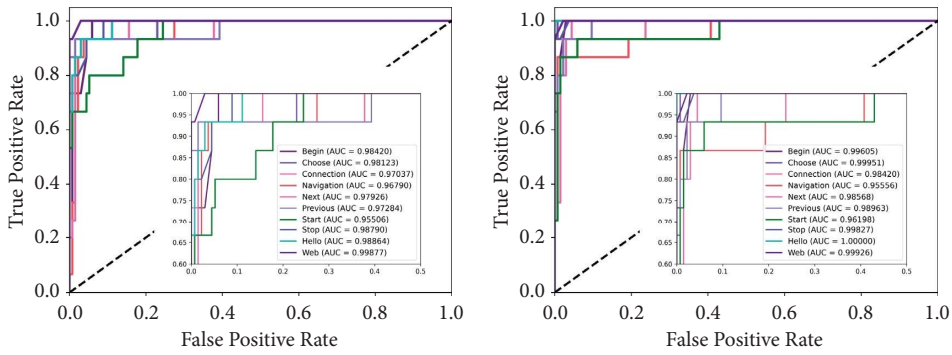
Table 5: Results of the performance evaluation (precision, recall, and *F*1 Score) on the phrase dataset.

Phrase	Conv-LSTM			LRCN-2Conv			LRCN-3Conv		
	Precision (%)	Recall (%)	*F*1score (%)	Precision (%)	Recall (%)	*F*1score (%)	Precision (%)	Recall (%)	*F*1score (%)
Terima kasih	98	100	99	100	97	99	100	97	99
Saya minta tolong	86	80	83	95	88	91	93	93	93
Saya minta maaf	88	90	89	97	97	97	97	97	97
Minta tolong	90	93	91	89	97	93	93	95	94
Accuracy			91			95			96
Macro avg	91	91	91	95	95	95	96	96	96
Weighted avg	91	91	91	95	95	95	96	96	96

obtained are also excellent compared to previous studies, which can be seen in Table 6.

The detection method with MediaPipe Face Mesh is also the only one used to compare it with other studies. Detection with this method proved to be more robust, as it can more effectively and efficiently localize the lips compared to HOG + SVM and Dlib. The consistency of the results between accuracy and the *F*1 score is also not far away. For the original dataset, MediaPipe + LRCN with 3 CNN layers has superior results (87%) compared to Inception V3 (86.6%) [48], CNN (52.9%) [48], VGG-16+LSTM [47] (59%), 3D-CNN [51] (70.2), and 3D-CNN + LSTM [52] (85%).

The MobileNet + LSTM and VGG-16+LSTM architectures [50] have an accuracy of more than 90%, which exceeds this study in the modified MIRACL-VC1 dataset. Modified means that the MIRACL-VC1 dataset is producing a new dataset which is similar to MIRACL-VC1. This is performed because the original MIRACL-VC1 has a lot of noise, such as part of the nose detected as a background, which can interfere with the training process. However, the value of the *F*1 score in this study, compared to the results of the MobileNet + LSTM *F*1 score, is 3% higher than the original MIRACL-VC1 [7]. This achievement is also inseparable from using MediaPipe Face Mesh to detect lips very well to avoid false information. Moreover, LRCN with three convolutional layers also gave



(a) (b)

Figure 8: The ROC and AUC for each class in the MIRACL-VC1 dataset. (a) LRCN-2Conv model. (b) LRCN-3Conv model.

Table 6: Comparison of the results of the proposed method with previous work in MIRACL-VC1.

Recognition methods	Detection methods	Dataset	Accuracy (%)	*F*1 score (%)
Fine-tune VGG-16 + LSTM [47]	Dlib	MIRACL-VC1	59	Unspecified
Inception V3 [48]	Dlib	MIRACL-VC1	86.6	Unspecified
CNN + Batch normalization [49]	Haar cascade	MIRACL-VC1	52.9	Unspecified
MobileNet + LSTM [50]	HOG + SVM	Modified MIRACL-VC1	94	84
VGG-16 + LSTM [50]	HOG + SVM	Modified MIRACL-VC1	96	70
3D-CNN [51]	Dlib	MIRACL-VC1	70.2	Unspecified
LSTM [52]	Dlib	MIRACL-VC1	66	66
3D-CNN + LSTM [52]	Dlib	MIRACL-VC1	85	Unspecified
LRCN-2Conv	MediaPipe	MIRACL-VC1	81.33	81
LRCN-3Conv	MediaPipe	MIRACL-VC1	90.67	91

a significant result in accuracy. Therefore, the findings of this investigation can be accepted and used as a reference by looking at previous studies. The use of MediaPipe and LRCN is a potentially robust detection algorithm to support the performance of the neural network training model to detect lips correctly.

5. Conclusion

IndoLR has been successfully built for Indonesian lip-reading benchmarking. In this study, the LRCN architecture and MediaPipe Face Mesh have been proposed to recognize lip reading. The performance of the LRCN model has been tested under various conditions. Testing has been carried out on two types of test data, namely, the word and phrase datasets. The experimental results show that the LRCN with three convolutional models produces the highest accuracy in the word dataset and the phrase dataset than the LRCN with two convolutional layers and convolutional LSTM. Adding more convolutional layers can improve the performance of the algorithm.

The average *F*1 score values of the LRCN-3Conv, LRCN-2Conv, and Conv-LSTM models for the word dataset are 90%, 93%, and 95%, respectively. Meanwhile, the average *F*1 score values of the LRCN-3Conv, LRCN-2Conv, and Conv-LSTM models for the phrase dataset are 91%, 95%, and 96%, respectively. In addition, testing was also conducted on an open dataset available called MIRACL-VC1 in word-labeled classes. The LRCN with three convolutional layers also outperforms previous studies in the *F*1 score. The findings of this study show that it is possible to use MediaPipe to get lip ROI without any noise and implement LRCN in the frame-to-frame data. The types of lip shapes are also necessary, which may not be considered in other research studies.

For future work, it will be more robust if the dataset is enriched by involving more people with various lip types, poses, angles, and lighting conditions. The diversity of datasets can support emerging classification algorithms such as transformer or attention-based models because it has a "data-hungry" behavior. The larger data obtained can provide a better model performance. This study also has limitations, which are related to the efficiency of the computational time and memory used without losing the correlation between every frame sequence of the sample video. A more effective and efficient method is still needed, which can make lip-reading recognition perform better in real-world applications.

Data Availability

The dataset in MP4 format used to support the findings is deposited in the Kaggle repository and is available at https://www.kaggle.com/datasets/abasset/indolr. The working source code experiment is published at https://github.com/sukasenyumm/IndoLR.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the Ministry of Education, Culture, Research, and Technology of Republic Indonesia, through the Directorate of Research and Community Service, for supporting the funding of this research through a competitive grant program with the Higher Education Excellence Applied Research scheme (Grant No. 182/E5/PG.02.00.PL/2023).

References

[1] G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, and M. Masud, "Spoken Language identification using deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, 2021.

[2] M. J. Al-Dujaili and A. Ebrahimi-Moghadam, "Speech emotion recognition: a comprehensive survey," *Wireless Personal Communications*, vol. 129, no. 4, pp. 2525–2561, 2023.

[3] W. Alsabhan, "Human–computer interaction with a real-time speech emotion recognition with ensembling techniques 1D convolution neural network and attention," *Sensors*, vol. 23, no. 3, p. 1386, 2023.

[4] J. S. Chung and A. Zisserman, "Learning to lip read words by watching videos," *Computer Vision and Image Understanding*, vol. 173, pp. 76–85, 2018.

[5] S. Rudregowda, S. Patil Kulkarni, G. H L, V. Ravi, and M. Krichen, "Visual speech recognition for Kannada language using VGG16 convolutional neural network," *Acoustics*, vol. 5, no. 1, pp. 343–353, 2023.

[6] R. El-Bialy, D. Chen, S. Fenghour et al., "Developing phoneme-based lip-reading sentences system for silent speech recognition," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 129–138, 2023.

[7] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "A new visual speech recognition approach for RGB-D cameras," *Lecture Notes in Computer Science*, vol. 21, pp. 21–28, 2014.

[8] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: end-to-end sentence-level lipreading," 2016, https://arxiv.org/abs/1611.01599.

[9] A. Fernandez-Lopez and F. M. Sukno, "Survey on automatic lip-reading in the era of deep learning," *Image and Vision Computing*, vol. 78, pp. 53–72, 2018.

[10] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*, University of Oxford, Oxford, UK, 2016.

[11] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," 2018, http://arxiv.org/abs/1809.00496.

[12] I. Anina, Z. Zhou, G. Zhao, and M. Pietikainen, "OuluVS2: a multi-view audiovisual database for non-rigid mouth motion analysis," in *Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–5, IEEE, Ljubljana, Slovenia, May 2015.

[13] G. Schwiebert, C. Weber, L. Qu, H. Siqueira, and S. Wermter, "A multimodal German dataset for automatic lip reading systems and transfer learning," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6829–6836, Marseille, France, February 2022.

[14] J. Park, "OLKAVS: an open large-scale Korean audio-visual speech dataset," 2023, https://arxiv.org/abs/2301.06375.

[15] R. Kosif, M. Diramali, and S. Yilmaz, "Investigation on the relationship between personal characteristics with lip, jaw and philtrum dimensions," *Int J Res Med Sci*, vol. 6, no. 9, p. 2911, 2018.

[16] C. Lugaresi, "MediaPipe: a framework for building perception pipelines," 2019, https://arxiv.org/abs/1906.08172.

[17] J. Donahue, L. A. Hendricks, M. Rohrbach et al., "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.

[18] C. Bregler and Y. Konig, "Eigenlips for robust speech recognition," *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, in *Proceedings of the ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE*, pp. II/669–II/672, Seoul, Korea, April 1994.

[19] S. Petridis, A. Asghar, and M. Pantic, "Classifying laughter and speech using audio-visual feature prediction," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5254–5257, Dallas, TX, USA, March 2010.

[20] B. S. Lin, Y. H. Yao, C. F. Liu, C. F. Lien, and B. S. Lin, "Development of novel lip-reading recognition algorithm," *IEEE Access*, vol. 5, pp. 794–801, 2017.

[21] S. Petridis, J. Shen, D. Cetin, and M. Pantic, "Visual-only recognition of normal, whispered and silent speech," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6219–6223, IEEE, Calgary, AB, Canada, April 2018.

[22] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-End audiovisual speech recognition," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6548–6552, IEEE, Calgary, AB, Canada, April 2018.

[23] R. Shashidhar and S. Patilkulkarni, "Visual speech recognition for small scale dataset using VGG16 convolution neural network," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 28941–28952, 2021.

[24] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading using temporal convolutional networks," in *Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6319–6323, IEEE, Barcelona, Spain, May 2020.

[25] A. Koumparoulis and G. Potamianos, "Accurate and resource-efficient lipreading with Efficientnetv2 and transformers," in *Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8467–8471, IEEE, Singapore, May 2022.

[26] P. Ma, S. Petridis, and M. Pantic, "Visual speech recognition for multiple languages in the wild," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 930–939, 2022.

[27] C. T. Huyen, *German Word Level Lip Reading with Deep Learning', Doctoral Dissertation*, Hochschule für angewandte Wissenschaften, Hamburg, 2019.

[28] X. Chen, J. Du, and H. Zhang, "Lipreading with DenseNet and resBi-LSTM," *Signal, Image and Video Processing*, vol. 14, no. 5, pp. 981–989, 2020.

[29] Ü. Atila and F. Sabaz, "Turkish lip-reading using Bi-LSTM and deep learning models," *Engineering Science and Technology, an International Journal*, vol. 35, 2022.

[30] A. Kurniawan and S. Suyanto, "Syllable-based Indonesian lip reading model," in *Proceedings of the 2020 8th International*

Conference on Information and Communication Technology (ICoICT), pp. 1–6, IEEE, Yogyakarta, Indonesia, June 2020.

[31] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "An adaptive approach for lip-reading using image and depth data," *Multimedia Tools and Applications*, vol. 75, no. 14, pp. 8609–8636, 2016.

[32] A. Aripin and A. Setiawan, "Indonesian lip-reading recognition using long-term recurrent convolutional network," *SSRN [Preprint]*, 2023.

[33] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3444–3453, IEEE, Honolulu, HI, USA, July 2017.

[34] Y. Zhao, R. Xu, X. Wang, P. Hou, H. Tang, and M. Song, "Hearing lips: improving lip reading by distilling speech recognizers," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 6917–6924, 2020.

[35] C. Chen, D. Wang, and T. F. Zheng, "CN-CVS: a Mandarin audio-visual dataset for large vocabulary continuous visual to speech synthesis," in *Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, Rhodes Island, Greece, June 2023.

[36] D. E. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[37] S. Sharma and V. Kumar, "Voxel-based 3D face reconstruction and its application to face recognition using sequential deep learning," *Multimedia Tools and Applications*, vol. 79, no. 26, pp. 17303–17330, 2020.

[38] M. E. Colak and A. Varol, "Eyematch: an eye localization method for frontal face images," in *Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–4, IEEE, Ankara, Turkey, November 2019.

[39] G. Sanil, K. Prakash, S. Prabhu, V. C. Nayak, and S. Sengupta, "2D-3D facial image analysis for identification of facial features using machine learning algorithms with hyperparameter optimization for forensics applications," *IEEE Access*, vol. 11, pp. 82521–82538, 2023.

[40] A. Ishmam, M. Hasan, M. S. Hassan Onim, K. Roy, M. A. Hoque Akif, and H. Nyeem, "Modelling lips state detection using CNN for Non-verbal communications," *Lecture Notes in Networks and Systems*, pp. 59–70, 2022.

[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the the 27th International Conference on International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, June 2010.

[42] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, http://arxiv.org/abs/1803.08375.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.

[44] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, pp. 1–15, San Diego, CA, USA, December 2015.

[45] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Proceedings of the NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 802–810, Montreal Canada, December 2015.

[46] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: a survey," *Mechanical Systems and Signal Processing*, vol. 151, 2021.

[47] A. Gutierrez and Z.-A. Robert, "Lip reading word classification," 2017, http://cs231n.stanford.edu/reports/2017/pdfs/227.pdf.

[48] P. Sindhura, S. J. Preethi, and K. B. Niranjana, "Convolutional neural networks for predicting words: a lip-reading system," in *Proceedings of the 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pp. 929–933, IEEE, Montreal Canada, December 2018.

[49] S. NadeemHashmi, H. Gupta, D. Mittal, K. Kumar, A. Nanda, and S. Gupta, "A lip reading model using CNN with batch normalization," in *Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–6, IEEE, Noida, India, August 2018.

[50] G. A. Kumar and J. H. William, "Development of visual-only speech recognition system for mute people," *Circuits, Systems, and Signal Processing*, vol. 41, no. 4, pp. 2152–2172, 2022.

[51] P. Nemani, G. S. Krishna, N. Ramisetty, B. D. S. Sai, and S. Kumar, "Deep learning-based holistic speaker independent visual speech recognition," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1705–1713, 2023.

[52] R. Shashidhar, M. P. Shashank, and B. Sahana, "Enhancing visual speech recognition for deaf individuals: a hybrid LSTM and CNN 3D model for improved accuracy," *Arabian Journal for Science and Engineering*, 2023.