Speech-to-Text Conversion in Indonesian Language Using a Deep Bidirectional Long Short-Term Memory Algorithm

Suci Dwijayanti[1], Muhammad Abid Tami[2], Bhakti Yudho Suprapto[3]
Department of Electrical Engineering, Universitas Sriwijaya, Indralaya, Indonesia

***Abstract*—Now-a-days, speech is used also for communication between humans and computers, which requires conversion from speech to text. Nevertheless, few studies have been performed on speech-to-text conversion in Indonesian language, and most studies on speech-to-text conversion were limited to the conversion of speech datasets with incomplete sentences. In this study, speech-to-text conversion of complete sentences in Indonesian language is performed using the deep bidirectional long short-term memory (LSTM) algorithm. Spectrograms and Mel frequency cepstral coefficients (MFCCs) were utilized as features of a total of 5000 speech data spoken by ten subjects (five males and five females). The results showed that the deep bidirectional LSTM algorithm successfully converted speech to text in Indonesian. The accuracy achieved by the MFCC features was higher than that achieved with the spectrograms; the MFCC obtained the best accuracy with a word error rate value of 0.2745% while the spectrograms were 2.0784%. Thus, MFCCs are more suitable than spectrograms as feature for speech-to-text conversion in Indonesian. The results of this study will help in the implementation of communication tools in Indonesian and other languages.**

***Keywords*—Speech-to-text; Deep Bidirectional Long Short-Term Memory (LSTM); spectrogram; Mel frequency cepstral coefficients (MFCC); word error rate**

I. Introduction

Speech is a longitudinal wave that propagated through a medium, which can be solid, liquid, or gaseous [1]. Humans utilize speech as a primary component of communication to exchange information. Today, humans communicate also with computers; generally, this communication requires the conversion of speech into text [2]. This process involves various stages of conversion and outputs data consisting of numbers that can be processed by a computer into text [3]. Speech-to-text conversion can be implemented in various applications, such as communication tools for deaf people [2], smart homes [4], and translators [5].

Some studies have investigated speech-to-text conversion in various languages. Ahmed et al. utilized a hidden Markov model (HMM) for English and Arabic speech recognition [6]. Hotta [7] and Othman [8] performed speech-to-text conversion using neural networks in Japanese and Jawi, respectively. Kumar et al [9] used a recurrent neural network (RNN) for speech-to-text conversion in Hindi, and Laksono et al. [10] used connectionist temporal classification (CTC), which is usually applied on top of an RNN, for speech-to-text conversion in Indonesian and Javanese. Abidin et al. presented an approach to obtain Indonesian voice-to-text data set using Time Delay Neural Network Factorization (TDNNF) [11].

Mon and Tun [12] proposed the HMM method, which uses Mel frequency cepstral coefficients (MFCCs) as features. Because they used a large dataset of English words, the HMM was ineffective owing to the high probability of similarity between words. Zhang [13] used a combination of the deep neural network (DNN) and HMM model for English speech recognition and showed that DNN-HMM was superior to the traditional Gaussian mixture model (GMM)-HMM method. Nevertheless, it still had low accuracy. Liu et al. [14] had shown that the RNN together with Long Short Term Memory (LSTM) improved the performance of speech recognition on the ChiME-5 dataset. Meanwhile, Wu et al. [15] and He [16] utilized RNN-LSTM for Chinese dataset, and the accuracy of speech recognition was improved.

Most studies on speech-to-text conversion were limited to the conversion of words or incomplete sentences from a dataset, and very few studies considered speech-to-text conversion in Indonesian. Laksono et al. [10] used DNN and CTC with MFCCs as the features for speech-to-text conversion in Indonesian and Javanese with a small number of Indonesian and Javanese words. However, the result showed low accuracy for both Indonesian and Javanese; thus, they might not be suitable for speech-to-text conversion.

In this study, we perform speech-to-text conversion in Indonesian using a deep bidirectional long short-term memory (LSTM) algorithm. We determine the features suitable for the deep bidirectional LSTM and consider complete sentences consisting of subject, predicate, object, and adverb spoken by some respondents.

The rest of this paper is organized as follows. In Section 2, the research method used in this study is presented. Section 3 reports and discusses the results. Finally, the paper is concluded in Section 4.

II. Materials and Methods

A. Data Collection

The speech data were obtained from ten speakers (five males and five females). Every speaker uttered ten sentences in Indonesian consisting of a subject, predicate, object, and adverb, as presented in Table I. Each sentence was uttered 50 times; thus, a total of 5000 sentences were recorded. Data

were manually divided as follows: 70% for training, 20% for validation, and 10% for testing. Thus, 3500 training, 1000 validations, and 500 testing data were obtained. The data were recorded in the Control and Robotics Laboratory, Universitas Sriwijaya.

B. Proposed Speech-to-Text Conversion Process

Fig. 1 shows a block diagram of the proposed speech-to-text conversion process. The speech is recorded using a FIFINE K669B microphone with a sampling frequency of 16 kHz. Speech data undergo the preprocessing stage, which involves normalization, silence removal, and pre-emphasis, to correct the speech signal by reducing noise and removing the silence area on the speech signal. Then, the speech features are extracted into spectrograms and MFCCs. The features are fed to the deep bidirectional LSTM to determine the probability of each label. In the deep bidirectional LSTM training process, CTC is used to determine the loss. Subsequently, the network performs the decoding for the process of labeling from the output of the deep bidirectional LSTM network and language model obtained from the Kompas newspaperFinally, text if obtained as the output.

TABLE I. SENTENCES UTTERED BY THE SPEAKERS

No.	Sentence
1	saya bermain bola di lapangan
2	ayah membaca buku di ruang tamu
3	nenek memasak sayur di dapur.
4	kakak bermain sepeda di halaman
5	paman menggembala sapi di kebun.
6	bibi mengantar tas ke sekolah.
7	dia membaca buku di rumah.
8	adik memakai sepeda ke sekolah
9	kakek menanam padi di sawah
10	ibu menonton tv di kamar

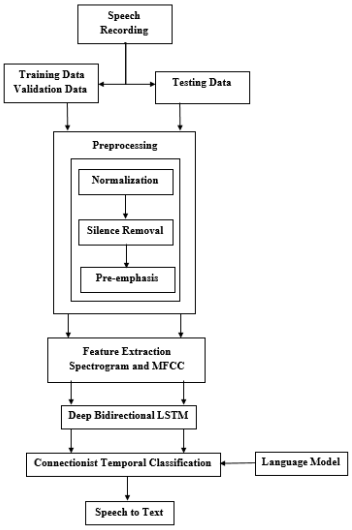

Fig. 1. Block Diagram of Speech-to-Text Conversion Process.

C. Evaluation

The word error rate (WER) is used to determine the percentage of success of speech-to-text conversion. It is determined by calculating the number of insertions, subtractions, and substitutions of the word used to convert speech into text as follows:

$$WER = \frac{1}{z}\sum ED\big(h(x)\big) \quad (1)$$

where $ED(h(x))$ is the number of insertions, subtractions, and substitutions of the word in the target sentence and z represents the total words in the reference which were actually said [17].

III. RESULTS AND DISCUSSION

A. Preprocessing Signal

The preprocessing stage involved normalization, silence removal, and pre-emphasis, which were performed using the Python library Pyrus. Normalization is performed by dividing the data in the speech signal by the maximum value of the amplitude to equate the amplitude of the speech signals. Owing to the recording process, the speech signal may have different intensities and consequently, different amplitude values. Silence removal is performed to determine the silence area to be erased on the speech signal. Finally, pre-emphasis is important to remove the noise while maintaining the frequency of the speech signal.

Fig. 2(a) and (b) show the speech signal before and after the preprocessing stage, respectively, displayed using the audacity software. From the figure, it can be seen that before preprocessing, the speech signal has an amplitude of less than 0.5 and silence areas are at the beginning and end of the speech. On the other hand, after preprocessing (Fig. 2(b), the amplitude of the signal is approximately 0.5, which is the ideal value for speech signals [18], and the silence areas are smaller than those before the preprocessing stage; the duration of the speech signal changes from 3 to 2.4 s. Furthermore, the noise in the speech signal was reduced by using a high pass filter to eliminate speech signals with frequency below 250 Hz.

B. Feature Extraction

Using the contrib audio library of TensorFlow, we extracted the log power spectra, i.e., spectrograms, and MFCCs as features and determined which is more suitable as input to the deep bidirectional LSTM.

To obtain the spectrograms, the preprocessed speech signal was divided into sections with window lengths of 32 ms and window steps of 16 ms. A fast Fourier transform (FFT) was performed to convert the speech signal from the time domain to the frequency domain. 512 frequency bins were used, and only half of the frequency bins plus one (257 bins) were used. Then, the log power spectra, which were the density of the FFT spectra, were used as input for the training process. The visual representation of log power spectra is known as spectrogram. Fig. 3 shows an example spectrogram. As shown in the figure, an x-axis shows the time length and a y-axis is the power spectrum of the speech signals.

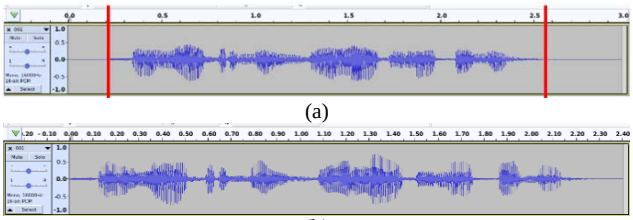

Fig. 2. Speech Signal (a) before and (b) after Preprocessing.

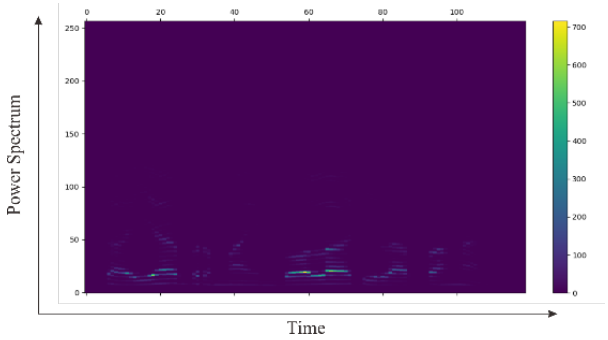

Fig. 3. Example of a Spectrogram.

The MFCCs are generated using the result of the spectral density, which is filtered with a Mel scale and filter bank to obtain the energy at each point. The resolution of the Mel filter bank was 40 with lowest and highest frequencies of 0 Hz and 8 kHz, respectively. The Mel spectrum, which is the output from the Mel filter bank, is converted into the time domain using a discrete cosine transform (DCT) with a coefficient of 13. The output from the DCT process is called an MFCC plot. Fig. 4 shows an example MFCC plot.

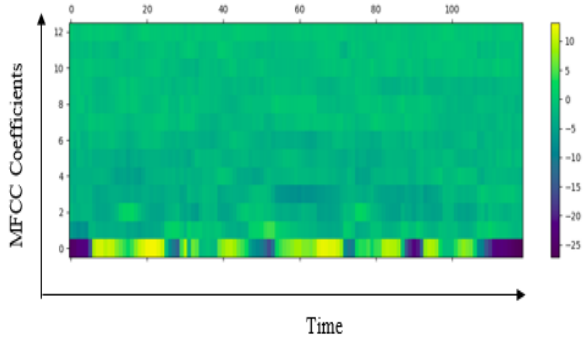

Fig. 4. Example MFCC Plot.

C. Language Model

A language model is used to perform the decoding process on the network output for the speech-to-text conversion process. The decoding process is performed by calculating the probability of appearance of each word based on the exact word order [19]. This probability is calculated from the word chunks based on word order in the N-grams model. The language model can be built using Corpus text with a large amount of data, e.g., words and sentences in a newspaper. Accordingly, in this study, Corpus text in .txt format derived from the Kompas newspaper [20] was used. The Kompas newspaper has published more than 5000 articles, which were merged to create the language model. Before using, the Corpus text is sorted in alphabetical order from A to Z, and invalid data such as space and blank () are removed.

A 5-gram language model was built with outputs in lm.arpa format using the KENLM library. The lm.arpa output was transformed into binary format to be processed and read by the computer. Then, a trie that works by tracking the minimum probability for the word prefix was created as a data structure to assist in using the memory to construct the language model.

D. Training Dataset

The DeepSpeech library was used in the training process of the deep bidirectional LSTM algorithm. This algorithm consists of a combination of bidirectional RNN and LSTM, commonly known as bidirectional LSTM. The bidirectional LSTM exploits long-range context dependencies in the past (t − 1) steps and future (t + 1) steps. This algorithm has a deep architecture, which can performe high-level representations of acoustic data [21]. The DeepSpeech architecture consists of the input layer derived from the extracted features, i.e., spectrograms and MFCCs. Then, there are five stacked hidden layers: three linear hidden layers, one LSTM hidden layer, and one linear hidden layer. The last layer is an output layer that uses the Softmax activation function to determine the probability of a transcript label. Fig. 5 shows a schematic of the DeepSpeech architecture.

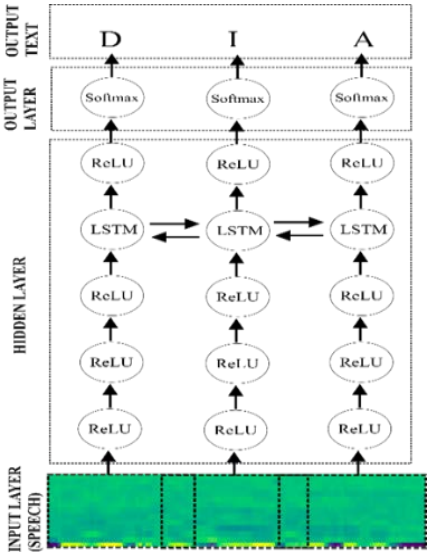

Fig. 5. Schematic of DeepSpeech Architecture.

The training performance can be determined from the loss values on the network. To prevent overfitting on the network, we use the early stopping technique, which involves the comparison of the loss value of the network during validation. The output of the network will be processed by CTC to perform the decoding process with the prefix beam search method according to the probability generated by the SoftMax layer and language model. CTC is used to model the training results obtained by a network. Because it can classify labels without having to know the alignment given, it is suitable for the deep bidirectional LSTM [22]. The CTC value decreased as the number of layers used increased, a phenomenon known as the CTC loss. The CTC loss value is a representation of the

accuracy of the training results; a smaller CTC loss value indicates a higher accuracy. Nevertheless, an excessively small CTC loss value leads to overfitting. In the training, we used five scenarios denoted as A, B, C, D, and E, as presented in Table II, to find the most suitable parameters.

In scenarios B, D, and E, the training process is continued until the epoch is ended, while in scenarios A and C, it is stopped early. We used 3500 speech data for training and a batch size of 70; thus, the number of steps in each epoch was 50. These steps were repeated for all data in each epoch. The training process resulted in a model called output graph, which could be stored and used to transcribe the data. Table III presents the results of each training scenario.

From Table III, it can be seen that when spectrograms were used as the input to the deep bidirectional LSTM and early stopping was activated (scenario A), training ended after 12 epochs within 5 h, 13 min, and 23 s with training and validation loss values of 0.746044 and 4.590043, respectively. When MFCC were used as the input and early stopping was activated (scenario C), training ended after 9 epochs within 2 h, 53 min, and 5 s with training and validation loss values of 0.424811 and 0.914198, respectively. These results indicate that MFCCs are more suitable as input to the deep bidirectional LSTM than spectrograms. These results may also imply that the MFCC features are better in terms of the computation time and loss value, and provide more useful information for the classifier.

On the other hand, when spectrograms were used as the input to the deep bidirectional LSTM and early stopping was not activated (scenario B), the loss values were lower than in scenario A. Furthermore, when MFCCs were used without early stopping (training scenario D), the loss values were lower than in scenarios A, B, and C. In particular, in scenario B, the training and validation loss values were 0.084932 and 2.765626, respectively, and training was completed within 16 h, 6 min, and 40 s; in scenario D, the training and validation loss values were 0.016846 and 0.505358, respectively, and training was completed within 15 h, 13 min, and 14 s. Therefore, when MFCCs are inputted, the training and validation loss values are smaller than when spectrograms are inputted. Although overfitting occurred in training scenario D, the training process could be re-adapted as shown in Fig. 6. Overfitting may have occurred owing to the presence of noise.

To prevent overfitting, we reduced the number of epochs before the occurrence of overfitting (scenario E). The process of training in scenario E is performed to determine the best training results before overfitting. Training ended at epoch of 24 within 7 h and 23 min with training and validation loss values of 0.077836 and 0.494393, respectively. Fig. 7 shows the plot of the loss values in scenario E. Compared to training scenario D, scenario E has higher training loss value but lower validation loss value. In scenario E, the training process took only 24 epochs, which is less than in scenario D.

E. Testing Model

The model obtained from the training results was tested. The test involved speech-to-text conversion of 500 speech data samples not included in the training process. As a measure of the accuracy, we considered the WER, which has a range of 0–1; a smaller WER value indicates a higher testing accuracy. Table IV shows the results obtained from a different model of training scenarios.

TABLE II. SCENARIOS USED FOR TRAINING

Parameter	**Spectrogram**		**MFCC**		
Training scenario	A	B	C	D	E
Train Batch Size	70	70	70	70	70
Validation Batch Size	4	4	4	4	4
Test Batch Size	1	1	1	1	1
Learning Rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Epoch	50	50	50	50	24
Early Stopping	Yes	No	Yes	No	No

TABLE III. RESULTS OF THE TRAINING SCENARIOS

Parameter	**Spectrogram**		**MFCC**		
Training Scenario	A	B	C	D	E
Actual Epoch	12	50	9	50	24
Time (h)	5:13:23	16:06:40	2:53:05	15:13:14	7:23:00
Training Loss	0.746044	0.084932	0.424811	0.016846	0.077836
Validation Loss	4.590043	2.765626	0.914198	0.505358	0.494393

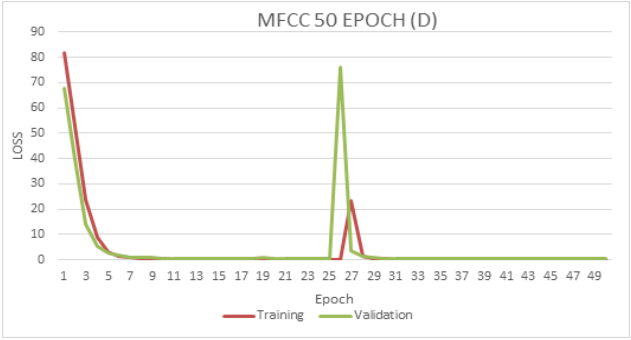

Fig. 6. Training and Validation Loss Values in Scenario D (using MFCC and 50 Epochs).

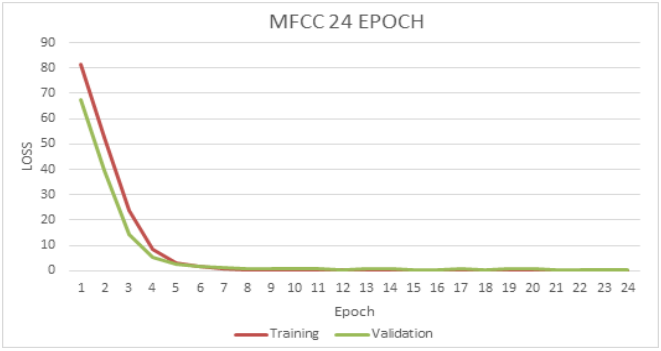

Fig. 7. Training and Validation Loss Values in Scenario E (using MFCCs and 24 Epochs).

TABLE IV. RESULTS OF TESTING USING DIFFERENT MODELS

Parameter	Spectrogram		MFCC		
Training scenario	A	B	C	D	E
WER	0.035686	0.020784	0.004706	0.002745	0.002745
% WER	3.5686	2.0784	0.4706	0.2745	0.2745

From Table IV results with a WER of 2.0784% in training scenario B, testing with MFCCs yielded the best results with a WER of 0.2745% in training scenarios D and E. These results indicate that the MFCCs are more suitable than spectrograms for speech-to-text conversion with the deep bidirectional LSTM algorithm. Training with MFCCs can identify linguistic content and remove unimportant parts of speeches, which may contain noise. Furthermore, training with MFCCs can demonstrate the vocal tract of human speech in the form of a power spectrum. Fig. 8 shows the accuracy of each model.

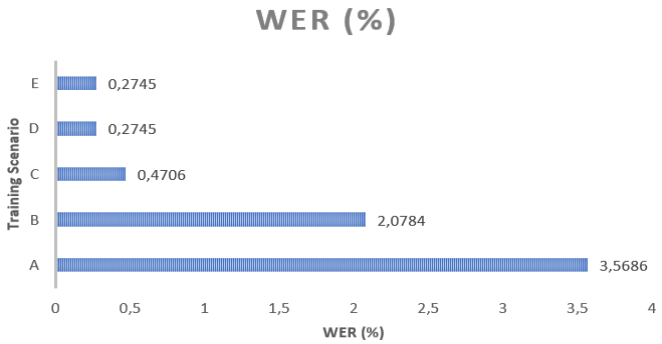

Fig. 8. WER Graph for each Training Scenario Model.

F. Testing Process using Speech Variations

Testing was also performed using five speech variations, namely regular (normal) conversation, speech with high intonation, speech with low intonation, speech with fast rhythm, and speech with slow rhythm. The test involved two speakers: a male and a female. The speakers uttered sentences 1–5 (see Table I) with the five styles listed in Table V. We tested the model obtained from training scenario E (using MFCCs as features) because it yielded the best results among the five training scenarios. Table V shows the WER obtained from the testing using different speech styles.

The results reported in Table V indicate that the model detected normal speech effectively. It can be seen that the intonation of the speech did not significantly affect the results. The model also detected the speech with fast rhythm well. However, the WER value obtained with the slow-rhythm speech is 40%, which indicates that many errors occurred during speech-to-text conversion. This may be due to the time-lapse between the words spoken and the large number of silence areas on the speech signal. These results indicate that the rhythm of the speech tends to affect the speech-to-text conversion process, while the intonation does not.

G. Testing Process with Secondary Datasets from TITML-IDN

The model obtained with training scenario E was also tested using five sentences obtained from the TITML-IDN dataset [23]. Table VI presents the results of speech-to-text conversion using speech corpus data for males and females.

TABLE V. RESULTS OF THE TESTING USING DIFFERENT SPEECH STYLES

Variation	WER (%)
Normal	0
High Intonation	9.3
Low Intonation	1.7
Fast Rhythm	6
Slow Rhythm	40

TABLE VI. RESULTS OF SPEECH-TO-TEXT CONVERSION USING THE TITML-IDN DATASET

No.	Target Transcript	Speech Detected from Male Speaker	Speech Detected from Female Speaker
1	dia tidak datang **ke sekolah**	diadi kakak **ke sekolah**	diadi eka **ke sekolah** eu
2	ketika kuatrianus hendak menelepon di wartel **di halaman** parkir bandara penjahat radin melarikan tas dengan mobil feroza.	kakak kakek muka selaku depkeu abu **di halaman** take pea asapa kakak di menanak ka sea budi rusun ke	kakak kakak sae aka di mana **di halaman** taeuk di aibak adi naik asean men sakka
3	kalian boleh lihat **saya** tidak apa apa padahal saya juga mengonsumsi produk transgenik	ada media **saya** ia kakak kakak saudi kamanan ke kekakuan	ayah menamai sa **saya** ie kakak kakak da sau iea bermasa ke sekaa sea edi
4	**adi** pandai **bermain** alat musik keyboard	aibak **bermain** skea babibu mena	**adi** aak benenain kaka aib mena
5	minggu depan ada **main bola** bareng anak kelas dua f	ibu mentan kakak **main** telapak kakak kakak	nenek kakak ber**main bola** aur aa seka

The results indicate that the model could detect the words in the TITML-IDN. For example, the target transcript “dia tidak datang ke sekolah” spoken by the male and female speakers is converted to “diadi kakak ke sekolah” and “diadi eka ke sekolah eu,” respectively. Therefore, the model successfully detected the word "ke sekolah" and it detected the word “dia” as “diadi.” However, the model did not recognize all words successfully because they were not in the transcript of the training data. Besides, the number of words in the model was much smaller (only 39 different words) than that in the TITML-IDN dataset. In addition, speech data from TITML-IDN have a different structure of sentences used in our primary data. ITITML-IDN either contains complete sentences (subject-predicate-object-adverb) or non-complete sentences or only phrases, as these data were obtained from

the text corpus. Meanwhile, the speech in our primary data recorded from 10 respondents consists of complete sentences, as described in the section Data Collection.

IV. Conclusions

This study tested the performance of a deep bidirectional LSTM algorithm on speech-to-text conversion in Indonesian using MFCCs and spectrograms as features. The data used were complete sentences consisting of subject, predicate, object, and adverb spoken by some respondents. With the MFCCs and spectrograms, the algorithm achieved the highest WER of 0.2745% and 2.0784%, respectively, indicating the higher performance of the MFCCs on speech-to-text in the Indonesian language.

The algorithm was shown to successfully convert speech with different intonation and rhythm and achieved reasonable accuracy when applied to the TITML-IDN dataset.

However, the variation of words used in this study is still limited. Thus, in the future, the algorithm should be tested with speech with the higher variation of words and rhythms to increase its universality.

References

[1] B. Gold, N. Morgan, and D. Ellis, Speech and Audio Signal Processing. John Wiley & Sons, Inc., 2011.

[2] P. Khilari and Bhope V. P., “A review on speech to text conversion,” Int. J. Adv. Res. Comput. Eng. Technol., vol. 4, no. 7, pp. 3067–3072, 2015.

[3] L. Deng et al., “Recent advances in deep learning for speech research at microsoft,” in 2013 IEEE Internatsional Conference on Acoustics, Speech and Signal Processing, 2013, vol. 26, no. 64, pp. 8604–8608.

[4] A. Munir, S. Kashif Ehsan, S. M. Mohsin Raza, and M. Mudassir, “Face and speech recognition based smart home,” in 2019 Int. Conf. Eng. Emerg. Technol. ICEET 2019, pp. 1–5, 2019.

[5] C. Jeyalakshml, “Speech recognition of deaf and hard of hearing people using hybrid neural network,” in 2010 2nd Int. Conf. Mech. Electron. Eng., vol. I, pp. 83–87, 2010.

[6] B. H. A. Ahmed and A. S.Ghabayen, “Arabic automatic speech recognition enhancement,” in Palest. Int. Conf. Inf. Commun. Technol. Arab., pp. 98–102, 2017.

[7] H. Hotta, “Japanese speaker-independent homonyms speech recognition,” Procedia - Soc. Behav. Sci., vol. 27, pp. 306–313, 2011.

[8] Z. A. Othman, Z. Razak, N. A. Abdullah, M. Yakub, and Z. Bin Zulkifli, “Jawi character speech-to-text engine using linear predictive and neural network for effective reading,” Proc. - 2009 3rd Asia Int. Conf. Model. Simulation, AMS 2009, pp. 348–352, 2009.

[9] A. Kumar, M. Dua, and T. Choudhary, “Continuous hindi speech recognition using monophone based acoustic modeling,” in Int. Conf. Adv. Comput. Eng. Appl., pp. 1–5, 2014.

[10] T. P. Laksono, A.F. Hidayatullah, and C.I. Ratnasari, "Speech to text of patient complaints for Bahasa Indonesia," in 2018 International Conference on Asian Languange Processing (IALP), pp. 79-84, 2018.

[11] T. F. Abidin, A. Misbullah, R. Ferdhiana, M. Z. Aksana, and L. Farsiah, “Deep Neural Network for Automatic Speech Recognition from Indonesian Audio using Several Lexicon Types,” Proc. Int. Conf. Electr. Eng. Informatics, vol. 2020-Octob, 2020.

[12] S. M. Mon and H. M. Tun, “Speech-to-text conversion (STT) system using Hidden Markov Model (HMM),” Int. J. Sci. Technol. Res., vol. 4, no. 06, pp. 349–352, 2015.

[13] L. Zhang, “An acoustic model for english speech recognition based on deep learning,” Proc. - 2019 11th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2019, pp. 610–614, 2019.

[14] C. R. Liu, D. Qu, and X. K. Yang, “Long short term memory networks weighted prediction error for far-field speech recognition,” Proc. 2019 IEEE 8th Jt. Int. Inf. Technol. Artif. Intell. Conf. ITAIC 2019, no. Itaic, pp. 200–203, 2019.

[15] D. Wu, L. DIng, S. Deng, and S. Lu, “Research on speech recognition acceleration technology based on embedded platform,” Chinese Control Conf. CCC, vol. 2019-July, no. 1, pp. 3663–3668, 2019.

[16] Z. He, “Improving LSTM Based Acoustic Model with Dropout Method,” Proc. - 2019 Int. Conf. Artif. Intell. Adv. Manuf. AIAM 2019, pp. 27–30, 2019.

[17] A. Ahmed and S. Renals, “Word error rate estimation for speech recognition:e-WER, ” In Proc. of the 56th Annual Meeting of the Association for Computational Lingustics, vol 2 short paper, pp. 20-24, 2018.

[18] Audacity, Guide to Using Audacity. 2018.

[19] M. Suzuki, N. Itoh, T. Nagano, G. Kurata, and S. Thomas, “Improvements To N -Gram Language Model Using Text Generated From Neural Language Model,” ICASSP 2019 - 2019 IEEE Int. Conf. Acoust. Speech Signal Process., pp. 7245–7249, 2019.

[20] K. Kurniawan, “Indonesian NLP resources,” 2018. [Online]. Available: https://github.com/kmkurn/id-nlp-resource. [Accessed: Feb 2020]

[21] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with transfer learning,” Int. Conf. Mach. Learn., vol. 32, 2014.

[22] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” Proc. 23rd Int. Conf. Mach. Learn., pp. 369–376, 2006.

[23] Speech Resources Consortium [Online] Available: http://research.nii.ac.jp/src/en/TITML-IDN.html.