J-COSINE (Journal of Computer Science and Informatics Engineering)
Vol. 6, No. 2, December 2022 E-ISSN:2541-0806
Accredited Sinta-3 by RISTEKDIKTI Decree No. 28/E/KPT/2019 P-ISSN:2540-8895

Performance Comparison of Deep Learning Algorithm for Speech Emotion Recognition

I Gusti Bagus Arya Pradnja Paramitha*, Hendra Budi Kusnawan, Muji Ernawati
Computer Science Program, Faculty of Information and Technology, Nusa Mandiri University
Jl.Kramat Raya No.18, RW.7, Kwitang, Kec. Senen, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta
Email: 14210149@nusamandiri.ac.id, 14210206@nusamandiri.ac.id, 14210225@nusamandiri.ac.id
Corresponding Author

Abstract **One of the problems in Speech emotion recognition is related to time series data, while the feedforward process in neural networks is unidirectional where the results from one layer are directly channeled to the next layer. This kind of feedforward process cannot store past data. Thus, if Deep Neural Network (DNN) is used for Speech emotion recognition, some problems arise, such as the speech rate of the speaker. DNN cannot analyze the existing acoustic patterns and so cannot map different levels of speech rate. Another method that can take input at once while retaining relevant data in the previous process is the Recurrent Neural Network (RNN). This paper presents the characteristics of the RNN method consisting of LSTM and GRU techniques for Speech emotion recognition using the Berlin EMODB dataset. The dataset is divided into 80% for training and 20% for testing. The feature extraction methods used are Zero crossing Rate (ZCR), Mel Frequency Cepstral Coefficients (MFCC), Root Mean Square Energy (RMSE), Mel Spectrogram, and Chroma. This study compares the CNN, LSTM, and GRU algorithms. The classification results show that the CNN algorithm gets better results, namely 79.13%. Meanwhile, LSTM and GRU only got an accuracy of 55.76% and 55.14%, respectively.**

Keywords**: Speech Emotion Recognition, Convolutional Neural Network, Long short term memory, Gated Recurrent unit, Deep learning**

I. INTRODUCTION

In speech emotion recognition neural networks have long been used, and deep learning methods are widely used to identify the relationships of many layers of cells generated by neural networks. The rapid development of technology, for example in the use of GPUs, which makes it easier to analyze large amounts of data, also encourages deep learning methods to be widely applied in various fields [1]. In speech emotion recognition, there are several applications of deep learning methods that have been carried out in previous studies, for example by using CNN which imitates the workings of the human brain [2], [3]. Problems in identifying speakers containing time series data is the feedforward process in neural networks unidirectional where the results from one layer are directly channeled to the next layer. This kind of feedforward process cannot store past data. So, if Deep Neural Network (DNN) is used for speech emotion recognition, some problems arise such as speech speed [4]. DNN cannot analyze existing acoustic patterns so it cannot map various levels of speech rate [5], [6]. Another method that can take input at a time by retaining relevant data in the previous process is the Recurrent Neural Network (RNN).

This paper presents the characteristics of the RNN method, LSTM, and GRU techniques for speech emotion recognition. In part 2 conduct a literature review on existing research. Section 3 describes the methodology of the research carried out. Chapter 4 describes the results of the tests carried out and compares the performance of each of these techniques, and ends with conclusions.

II. LITERATURE REVIEW

In speech emotion recognition, significant knowledge is required, for example, understanding a particular word, and preprocessing datasets. In some cases, the dialect used is also important, including the speaker, as well as the use of the channel, in this case, the RNN can be used to plot patterns to display vector features. LSTM is more widely used than RNN in modeling Context-Free Language (CFL) and Context-Sensitive Language (CSL) [7]. In addition, [7] proposed a deep LSTM and RNN to assess the structure of speech recognition. In the case of language processing with large data, the quality of LSTM, RNN, and DNN configurations has been analyzed and compared [8], [9]. Meanwhile, GRU has several similarities with LSTM where both are used to deal with time series problems. However, GRUs have fewer components when compared to LSTMs. This shows that GRU is also very useful like LSTM.

A. Recurrent Neural Network (RNN)

RNN is an algorithm that consists of cells, where each part and every nerve cell will use memory to keep a record of various instructions, this becomes very important when analyzing a dataset. In the RNN the results from past simulations must be identified to predict the next outcome depending on the available dataset. So that RNN is a system that can remember every sequence that has occurred so far so that the current input and the results of the previous output will be used as input in the next process [10].

B. Long short-term memory (LSTM)

In 1997, Hochreiter and Schmidhuber analyzed the backpropagation problem and introduced a new algorithm called Long-Short Term Memory (LSTM) for Neural networks. In the LSTM layer, there are 4 new hidden layers named gates [9].

A long short-term memory network (LSTM) is a development of a Recurrent Neural Network (RNN). LSTM is one of the popular developments in RNN because LSTM aims to complete the shortcomings of RNN which has a deficiency in remembering past information that is stored for a long time. LSTM is able to remember a collection of information that has been stored for a long time. In addition, LSTM is able to remove information that is no longer relevant for use. In short, LSTM is more efficient for processing, predicting, and classifying based on certain time-series data.

C. Gated Recurrent Unit (GRU)

The gated recurrent unit (GRU) was introduced by Cho, et al. in 2014 to solve the vanishing gradient problem that occurs in RNN. GRU is an extension of LSTM with the aim of making each recurrent unit adaptively capture dependencies on different time scales. The GRU is used to solve the vanishing gradient problem that comes with standard recurrent neural networks. In GRU, the information flow control component is called a gate and the GRU has 2 gates, namely the reset gate and the update gate.

The advantage of GRU is that the computation is simpler than long short term memory (LSTM), but has the same accuracy and is still quite effective. GRUs share many properties with LSTMs. Both of these algorithms use a gating mechanism to control the recall process. GRU is simpler when compared to LSTM and computationally faster process.

GRU supports gating and hidden states to control the flow of information. To overcome the problems that arise in the RNN, GRU uses two gates, the update gate, and the reset gate. While the LSTM consists of three gates, namely the input gate, forget gate, and output gate. However, unlike the LSTM, GRU does not have an output gate but combines the input gate and forget gate into one as an update gate.

III. Research Methodology

In this research, the reseach methodology can be seen on Fig. 1. The first stage is dataset collection, then the dataset is preprocessed. After that, apply the augmented data and enter the feature extraction stage. The next stage is classification using CNN, LSTM, and GRU. Model evaluation is done based on confusion matrix and accuracy.

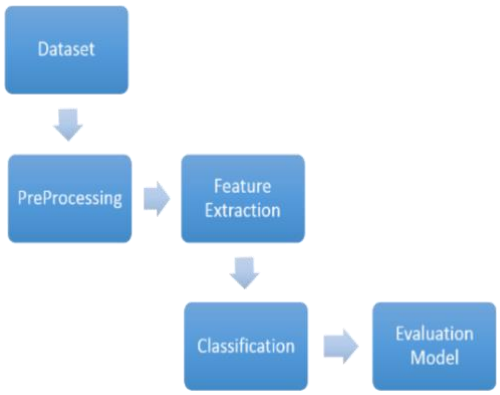

Fig. 1. Research Methodology

A. Dataset

In this study, voice samples were used from the Emo-DB dataset (Berlin Emotion Speech Database), which is a free German emotional database. The database was created by the Institute of Communication Sciences, Technical University, Berlin, Germany. Ten professional speakers (five male and five female) participated in the data recording. The database contains a total of 535 utterances. This dataset consists of seven voice emotions namely 1) anger; 2) bored; 3) anxious; 4) happy; 5) sad; 6) disgust; and 7) neutral. The dataset used for training data is 80% and testing data is 20%.

B. Preprocessing

The dataset used is still raw data that has not been categorized. At this preprocessing stage, labeling is carried out to group according to the category of emotions based on the name in the .wav file. In addition, the dataset is converted from a .wav file to a spectrogram form to simplify the feature extraction process.

C. Augmentation Data

Deep learning models perform better with large data sets. One very broad way to make data sets larger is augmentation. Data augmentation can increase the size of the data set by 10 or 20 times the original or more, which helps avoid overfitting when training very little data. This approach helps in building simpler and more robust models that can be generalized better. In this research, the augmentation techniques used are noise injection, stretching and pitching.

D. Feature Extraction

The next process is feature extraction, this is done to get the features and differences in patterns that exist in the dataset. The feature extraction method used is Zero crossing Rate (ZCR), Mel Frequency Cepstral Coefficients (MFCC), Root Mean Square Energy (RMSE), Mel Spectrogram, and Chroma. These methods are standard methods in voice emotion recognition to select and extract features. This is because these methods can represent human voice signals well.

D.1. Zero crossing Rate (ZCR)

ZCR is a basic property of audio signals that are often used in audio classification. Zero crossing allows rough estimation of dominant frequency and spectral center. One of the simplest features is ZCR, which is defined as the number of Zero crossings in the temporal domain in one second.

Zero-Crossing Rate (ZCR) an audio file frame is a rate at which the signal changes in a frame. In other words, is the number of times the signal changes value either from positive to negative or vice versa. ZCR can be interpreted as a measure of the noise of a signal. It usually shows a higher value in a noisy signal [11].

D.2. Mel Frequency Cepstral Coefficients (MFCC)

MFCC is determined by the auditory characteristics of the human ear by simulating the human auditory system using nonlinear frequency units. The fast Fourier Transform (FFT) technique is used to transform each sample frame from the time domain to the frequency domain.

While the Mel filter bank consists of an overlapping triangular filter with a frequency cut defined by the two adjacent center frequencies of the filter. The filtration has a linearly distributed center frequency with a fixed mel scale bandwidth.

D.3 Root Mean Square Energy (RMSE)

Signal energy is the total amount of the signal, which is how loud a sound signal is [12]. Since the amplitude of the oscillating signal varies over the period, it is usually unreasonable to estimate the instantaneous energy, but only average it over several windows. In the time-frequency representation, the energy of a single frequency component can be estimated over time. That is, the average energy of the frequency components can be taken through the next several frames or windows.

D.4 Mel Spectrogram

Mel Spectrogram is a spectrogram converted to a Mel scale. The spectrogram is a visualization of the frequency spectrum of a signal, where the frequency spectrum of a signal is the frequency range contained in the signal. The Mel scale mimics how the human ear works, with research showing that humans do not perceive frequencies on a linear scale. Humans are better at detecting differences at lower frequencies than at higher frequencies.

The Mel Spectrogram is related to the linear frequency of the spectrogram, i.e. the magnitude of the short-time Fourier transform (STFT). This is achieved by applying a non-linear transformation to the frequency of the STFT axis, which is inspired by the human auditory system and encapsulates the frequency content with less size. The use of such an auditory frequency scale has the effect of emphasizing detail in lower frequencies, which is critical for speech intelligibility while de-emphasizing high-frequency detail, which is dominated by fricatives and other bursts of noise and generally does not need to be modeled with great precision. [13].

D.5 Chroma Vector

A Chroma vector is usually a vector with 12 elements that represent the amount of energy of each tone class, {C, C#, D, D#, E, ..., B}, present in the signal. The Chroma vector is a motivating feature perception vector. It uses the concept of chroma in a cyclic helix, a representation of the perception of musical tones. The Chroma vector represents the magnitude of the twelve class notes on the standard chromatic scale.

The chroma feature is a descriptor, which represents the total content of a music audio signal in condensed form. Therefore, chroma features can be considered important prerequisites for high-level semantic analysis, such as chord recognition or harmonic similarity estimation. The better quality of extracted chroma features enables much better results in these high-level tasks. Short-Time Fourier Transform (STFT) and Constant Q Transform are used for chroma feature extraction. [14]

E. Classification

After the feature extraction process is complete, the next step is the classification stage using the CNN, LSTM, and GRU algorithms. Each algorithm will be used for classification with different architecture. The architectural design is described in section IV.

F. Model Evaluation

The evaluation process to measure a model is an important step. This is because the results of the measurements taken can be considered in choosing the best model. One technique to measure the performance of a model is the Confusion matrix. The confusion matrix is an evaluation method that can be used to calculate the performance or level of truth of the classification process.

There are four terms that represent the results of the classification process in the Confusion matrix, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

In accordance with the Confusion matrix table, it can be calculated the values of accuracy, precision, and

recall. In this case, the three matrices are very useful for measuring the performance of the classifier or algorithm used to make predictions on the model created.

IV. Result and Discussion

In this section, we will discuss the preprocessing results, the architecture of the algorithm used, and the classification results.

A. *Data Visualization*

The results of the labeling on the dataset as shown on Fig. 2 show that the data with angry emotions are 127 data, bored emotions are 81 data, disgusted emotions are 46 data, fear emotions are 69 data, happy emotions are 71 data, neutral emotions are 79 data, and sad emotions are 62 data.

Fig. 2. Dataset Frequency Distribution

To simplify the feature extraction process, the dataset is converted from .wav form to wave plot and spectrogram. The following is a wave plot and spectrogram for each emotion.

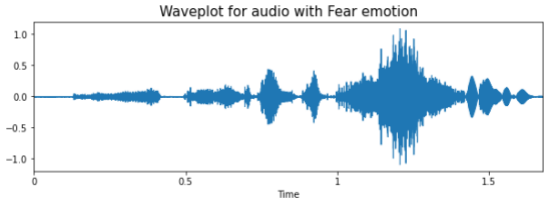

Fig. 2. Fear Waveplot

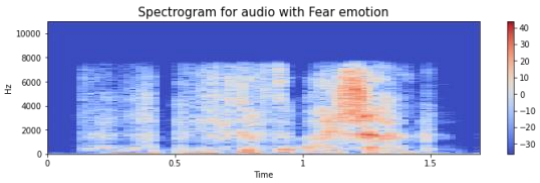

Fig. 3. Fear Spectogram

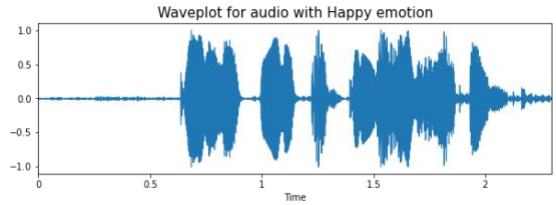

Fig. 4. Happy Waveplot

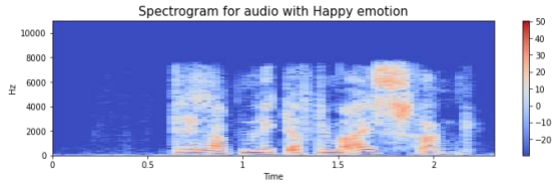

Fig. 5. Happy Spectrogram

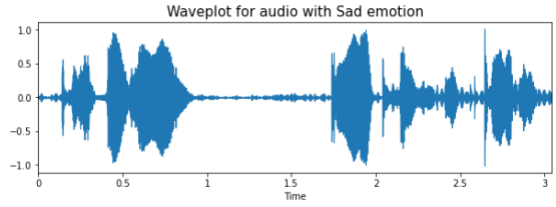

Fig. 6. Sad Waveplot

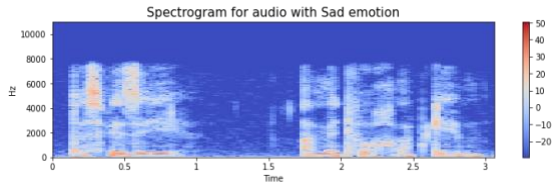

Fig. 7. Sad Spectrogram

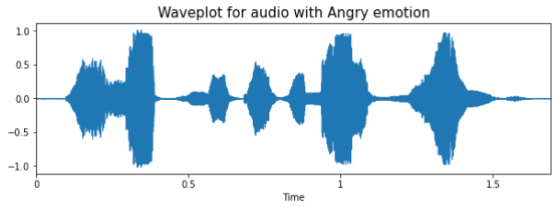

Fig. 8. Angry Waveplot

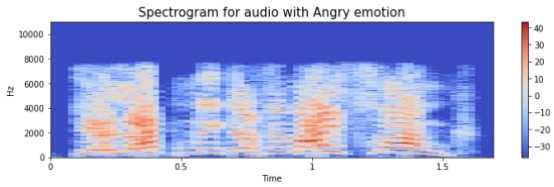

Fig. 9. Angry Spectrogram

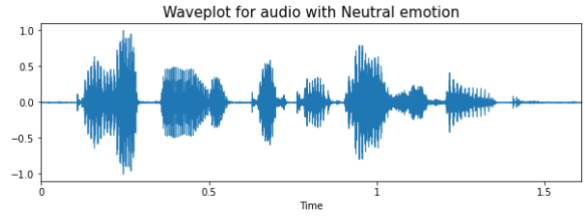

Fig. 10. Neutral Waveplot

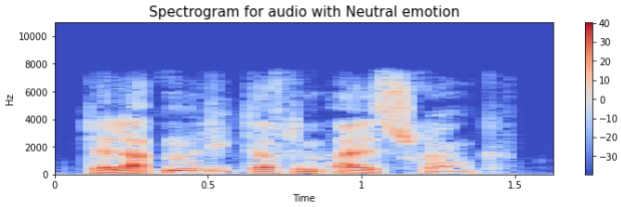

Fig. 11. Neutral Spectrogram

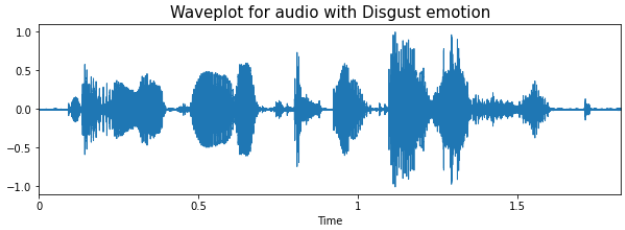

Fig. 12. Disgust Waveplot

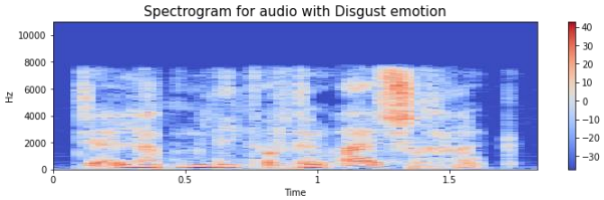

Fig. 13. Disgust Spectogram

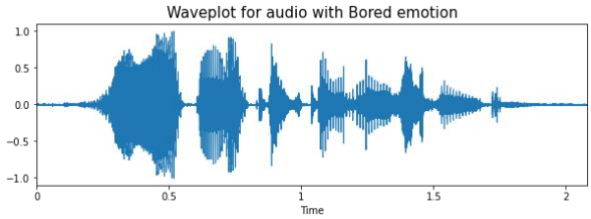

Fig. 14. Bored Waveplot

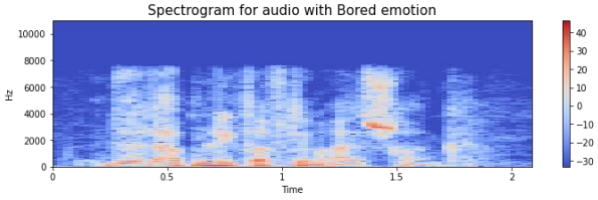

Fig. 15. Bored Spectrogram

B. Classification Architecture

The following is the classification architecture of the CNN, LSTM, and GRU algorithms

B.1. Convolutional Neural Network Architecture

TABLE 1. CNN ARCHITECTURE

Layer (type)	Output Shape	Param
conv1d (Conv1D)	(None, 162, 256	1536
max_pooling1D (MaxPooling1D)	(None, 81, 256)	0
Conv1d_1 (Conv1D)	(None, 81, 256)	327936
max_pooling1D_1 (MaxPooling1D)	(None, 41, 256)	0
conv1d_2 (Conv1D)	(None, 41, 128)	163968
max_pooling1D_2 (MaxPooling1D)	(None, 21, 128)	0
dropout (Dropout)		0
conv1d_3 (Conv1D)	(None, 21, 64)	41024
max_pooling1D_3 (MaxPooling1D)	(None, 11, 64)	0
flatten (Flatten)	(None, 704)	0
dense (Dense)	(None, 32)	22560
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 7)	231
Total params: 557,255 Trainable params: 557,225 Non-trainable params: 0		

Model: “sequential”

Convolutional neural network architecture starts from the input 162 audio data. In the first and second convolution layers with the number of feature maps 256, kernel size 5, kernel shift or can be called stride with a value of 1, with stride 1 it is not expected to reduce the number of features or features in the image, and use ReLu activation. The results of the first and second convolutions then enter the pooling layer using max pooling. The third convolution has the same architecture as the first and second convolution layers, only adding a dropout with a value of 0.2. The fourth convolution with a total of 64 feature maps, kernel 5, strides 1, and using ReLu activation. Max pooling is also used in the fourth pooling layer and then converted to vector or 1-dimensional with flattening so that it can enter the fully connected layer. In the fully connected layer, there is a neural network operation with 2 hidden layers. The first hidden layer has 32 neurons and uses ReLu activation and 0.3 dropouts. In the second hidden layer, the softmax activation function is used which is used to change the output results in a probability distribution according to the classification generated by the output. The number of output layers according to the specified category is the same as the input, namely 5 categories. The optimizer used Adam, with loss categorical cross-entropy, batch_size: 32, and epoch 100.

B.2. LSTM Architecture

TABLE 2. LSTM ARCHITECTURE

Layer (type)	Output Shape	Param
lstm (LSTM)	(None, 162, 128)	66560
lstm_1 (LSTM)	(None, 128)	131584
dense (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 7)	455
Total params: 206,855 Trainable params: 206,855 Non-trainable params: 0		

Model: "sequential"

The LSTM architecture used in this study consists of 3 hidden layers and 1 output. Where the first 2 hidden layers use LSTM with the number of neurons as many as 128 and 1 hidden layer uses dense with the number of neurons as many as 64 and dropout 0.3. the number of outputs produced is 7 categories. The optimizer used Adam, with loss categorical cross-entropy, batch_size: 32, and epoch 100.

B.3. GRU Architecture

TABLE 3. GRU ARCHITECTURE

Layer (type)	Output Shape	Param
gru_1 (GRU)	(None, 128)	50304
dense_7 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
batch_normalization_2 (BatchNormalization)	(None, 64)	256
dense_8 (Dense)	(None, 32)	2080
dropout_6 (Dropout)	(None, 32)	0
batch_normalization_3 (BatchNormalization)	(None, 32)	128
dense_9 (Dense)	(None, 7)	231
Total params: 61,255 Trainable params: 61,063 Non-trainable params: 192		

Model: "sequential_3"

The GRU architecture consists of 3 hidden layers and 1 output. The first hidden layer uses the GRU layer with 128 neurons and uses recurrent_activation sigmoid. In the second and third hidden layers using a dense layer with the number of neurons 64 and 32, using ReLu activation and dropout 0.3 which then normalizes the data with batch normalization. The resulting output is 7 categories with sigmoid activation. The optimizer used Adam, with loss categorical cross-entropy, batch_size: 32, and epoch 100.

C. Classification Result

In this study, to measure the performance of the model that has been made using the Confusion matrix and Accuracy. The confusion matrix is a summary of the predicted results of the classification. The number of correct and incorrect predictions is summed up with calculated values for each class.

C.1. CNN Classification Result

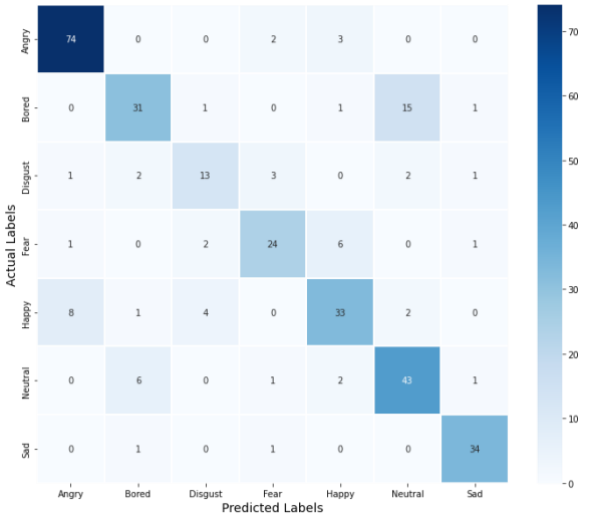

Fig. 16. CNN Confusion matrix

Based on the results shown on the confusion matrix, the prediction results built on the CNN algorithm are still not good because there is still quite a lot of loss. As shown in Fig. 16, the prediction results for neutral emotions occur as many as 19 data losses, 12 data for happy emotions, 10 data for angry emotions, 10 data for bored emotions, and there are still many losses in other emotions whose prediction results do not match. with the actual data available.

C.2. LSTM Classification Result

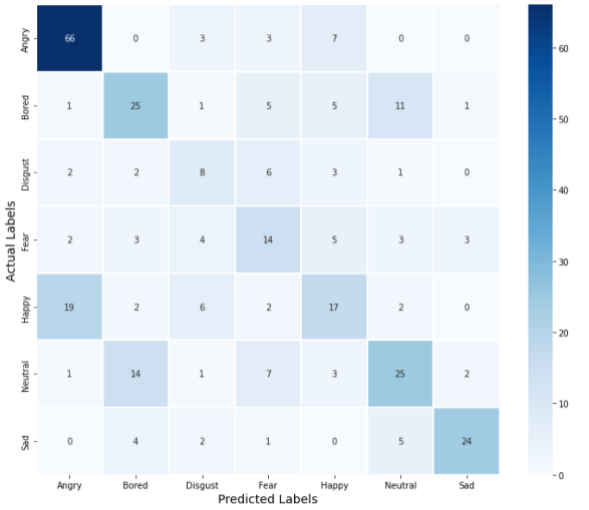

Fig. 17. LSTM Confusion matrix

Based on the performance results shown in the confusion matrix, a lot of losses are generated in the LSTM prediction model that is built. As shown in Fig. 17, the prediction results for neutral emotions occur as much as 22 data loss, then 25 data for angry emotions, 25 data for bored emotions, and there are still many

losses in other emotions whose prediction results do not match the actual data available. These results can be concluded that the prediction model is still not good.

C.3. GRU Classification Result

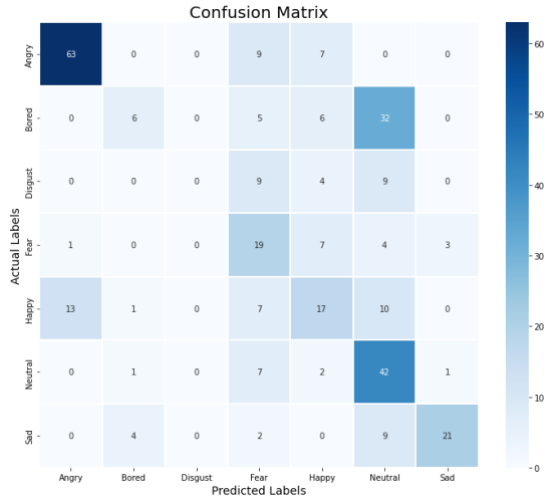

Fig. 18. GRU Confusion matrix

Based on Fig. 18 on the confusion matric, the prediction results from the architecture still produce a lot of loss. For example on neutral emotions, where the prediction results from neutral emotions resulted in a loss of 55 data, 39 data for fear emotions, 26 data for happy emotions, and other emotions still experiencing loss. It can be concluded that the model built on the GRU algorithm produces a poor model.

C.4. Accuracy Comparison

At this stage, a comparison of the performance results of each model that has been obtained is carried out as shown in table 4. below. Based on the classification results that have been carried out, the accuracy obtained in the CNN model is 79.13%, the LSTM model is 55.76% and the GRU model is 55.14%.

TABLE 4. ACCURACY COMPARISON

Model	Accuracy
CNN	79.13%
LSTM	55.76%
GRU	55.14%

From the results obtained, the accuracy is still not good, the researcher assumes that the results are influenced by the type of feature extraction used and less complex classification architecture.

V. CONCLUSION AND SUGGESTION

A. Conclusion

Based on the results of the study concluded that Deep learning algorithms such as CNN, LSTM, and GRU can be used for speech emotion recognition. Feature extraction used ZCR, MFCC, RMSV, Mel Spectrogram, and Chroma yielded an accuracy value of 79.13% on CNN, 55.76% on LSTM, and 55.14 on GRU. Of the three algorithms used, CNN produces better accuracy.

B. Suggestion

Future research is expected to improve performance by modifying the feature extraction used or by modifying the architecture of the algorithm used. And it is hoped that there will be an Indonesian dataset for speech emotion recognition.

REFERENCES

[1] I. Jinyu Li, Member, IEEE, Li Deng, Fellow, IEEE, Yifan Gong, Senior Member, IEEE, and Reinhold Haeb-Umbach, Senior Member, “An Overview of Noise-Robust Automatic Speech Recognition,” IEEE/ACM Trans. AUDIO, SPEECH, Lang. Process., vol. 22, no. C, pp. 745–777, 2014, doi: 10.1016/s0001-6918(61)80351-x.

[2] W. K. Andreas Schwarz, Christian Huemmer, Roland Maas, “SPATIAL DIFFUSENESS FEATURES FOR DNN-BASED SPEECH RECOGNITION IN NOISY AND REVERBERANT ENVIRONMENTS,” Icassp, pp. 4380–4384, 2015.

[3] Y. B. Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, “BATCH-NORMALIZED JOINT TRAINING FOR DNN-BASED DISTANT SPEECH RECOGNITION,” IEEE Spok. Lang. Technol. Work., pp. 28–34, 2016.

[4] M. S. Kasiprasad Mannepalli, Panyam Narahari Sastry, “FDBN: Design and development of Fractional Deep Belief Networks for speaker emotion recognition,” Int. J. Speech Technol., vol. 19, no. 4, pp. 779–790, 2016, doi: 10.1007/s10772-016-9368-y.

[5] and D. Y. Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, “Convolutional Neural Networks for Speech Recognition,” IEEE/ACM Trans. AUDIO, SPEECH, Lang. Process., vol. 22, no. 10, pp. 1533–1545, 2014, doi: 10.1109/ijcnn.1999.835942.

[6] H. Yao, S. Wang, X. Zhang, C. Qin, and J. Wang, “Detecting Image Splicing Based on Noise Level Inconsistency,” Multimed. Tools Appl., 2016, doi: 10.1007/s11042-016-3660-3.

[7] A. Graves, N. Jaitly, and A. Mohamed, “HYBRID SPEECH RECOGNITION WITH DEEP BIDIRECTIONAL LSTM,” IEEE Work. Autom. Speech Recognit. Underst., pp. 273–278, 2013.

[8] Z.-H. T. and J. J. Morten Kolbæk, “SPEECH ENHANCEMENT USING LONG SHORT-TERM MEMORY BASED RECURRENT NEURAL NETWORKS FOR NOISE ROBUST SPEAKER VERIFICATION,” IEEE Spok. Lang. Technol. Work., no. 1, pp. 305–311, 2016.

[9] F. Weninger et al., “Speech enhancement with LSTM recurrent neural networks and its application to noise-

robust ASR," Lect. Notes Comput. Sci. (including Bioinformatics), vol. 9237, pp. 91–99, 2015, doi: 10.1007/978-3-319-22482-4_11.

[10] Y. B. Cesar Laurent, Gabriel Pereyra, Philemon Brakel, Ying Zhang, "BATCH NORMALIZED RECURRENT NEURAL NETWORKS," Icassp 2016, pp. 2657–2661, 2016.

[11] G. F. Chen and Y. D. Wu, "Segmentation of Singing, Speech and Instruments in Kunqu Audio Based on Zero-Crossing Rate," Proc. - 2019 12th Int. Symp. Comput. Intell. Des. Isc. 2019, vol. 1, pp. 270–273, 2019, doi: 10.1109/ISCID.2019.00069.

[12] Y. Khoirotul Aini, T. Budi Santoso, and T. Dutono, "Pemodelan CNN Untuk Deteksi Emosi Berbasis Subser. Lect. Notes Artif. Intell. Lect. Notes Speech Bahasa Indonesia," J. Komput. Terap., vol. 7, no. Vol. 7 No. 1 (2021), pp. 143–152, 2021, doi: 10.35143/jkt.v7i1.4623.

[13] J. Shen et al., "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2018-April, pp. 4779–4783, 2018, doi: 10.1109/ICASSP.2018.8461368.

[14] M. Kattel, A. Nepal, A. K. Shah, and D. C. Shrestha, "Chroma Feature Extraction," Encycl. GIS, no. January, pp. 1–9, 2019.