Jurnal Aplikasi Sains Data
Journal homepage: https://jasid.upnjatim.ac.id/index.php/jasid

Comparative Analysis of Stochastic Gradient Descent Optimization and Adaptive Moment Estimation in Emotion Classification from Audio Using Convolutional Neural Network

Aldelia J. Tutuhatunewa[1], Dorteus L. Rahakbauw [2], and Zeth A. Leleury [3,*]

[1]Department of Mathematics, Pattimura University, [2] Department of Mathematics, Pattimura University, [3] Department of Mathematics, Pattimura University

[1]aldeliajoe@gmail.com, [2]dorteus.rahakbauw@lecturer.unpatti.ac.id, [3,*]zeth.arthur@lecturer.unpatti.ac.id

ABSTRACT

Abstract— Emotion is a fundamental aspect of human life that profoundly shapes behavior, social interactions, and decision-making processes. The ability to effectively communicate and foster mutual understanding between individuals relies heavily on accurately recognizing and expressing emotions. Among various channels of emotional expression, sound stands out as a powerful and direct medium that reflects and conveys human emotional states. This makes audio-based emotion recognition a critical and rapidly evolving field of study. With the rapid advancements in information technology and artificial intelligence, research focused on recognizing emotions through sound signals has gained significant momentum. Machine learning algorithms, particularly deep learning models like neural networks, have demonstrated remarkable capabilities in identifying and classifying emotions expressed through multiple modalities such as text, images, videos, and especially audio signals. Within the family of neural networks, Convolutional Neural Networks (CNNs) have been especially effective for audio emotion classification, due to their strength in extracting hierarchical and spatial features directly from raw input data. This study specifically investigates the comparative effectiveness of two popular optimization algorithms—Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam)—in training CNN models for emotion classification from audio recordings. Utilizing the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset, experimental results indicate that CNNs trained with the SGD optimizer achieve an overall accuracy of 53%, surpassing the 48% accuracy achieved by Adam. These results underscore the potential advantages of SGD in fine-tuning deep learning models for audio-based emotion recognition. Consequently, researchers and practitioners are encouraged to consider SGD optimization to improve the performance and robustness of emotion classification systems based on audio data.

Keywords: Convolutional Neural Network (CNN), Audio classification, Mel Frequency Cepstral Coefficients (MFCC), Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam)

I. INTRODUCTION

Emotions are a fundamental aspect of human life that influence behavior, social interactions and decision-making. Successful communication and understanding between individuals largely depend on our ability to recognize and express emotions. In this domain, voice or audio plays a key role as a medium that reflects and expresses human emotions.

* *Corresponding author.*
E-mail address: zeth.arthur@lecturer.unpatti.ac.id
https://doi.org/10.33005/jasid.v1i1.5

In the era of information technology and artificial intelligence, emotion recognition through voice has become a growing research focus. Machines and machine learning models, especially neural networks, can be taught to understand and classify emotions contained in text, images, video, and audio. Three algorithms commonly used for image classification include Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Convolutional Neural Network (CNN). Among the three, CNN has the highest accuracy in model performance with parameter values Accuracy 0.942, Precision 0.943, Recall 0.942, and F1-Score 0.942 [1].

In building CNN models, the Optimizer function is often used to maximize model performance. An optimizer is an algorithm or method used to minimize or maximize the objective function during the training process. Many optimizers are used to manage the learning rate which is a parameter that regulates how much change will be applied to the weights during the training process. Some optimizers that are often used for CNN algorithms include Adam [2], Stochastic Gradient Descent (SGD) [3] [4], and Root Mean Square Propagation (RMSProp) [5]. Emotion classification in audio has been done using the Multilayer Perceptron method which classifies 8 types of emotions achieving an average accuracy of 96% [6]. However, this research only uses the Adam Optimizer.

Based on the background above, in this final project research, the researcher takes the research title “Comparative Analysis of Stochastic Gradient Descent Optimization and Adaptive Moment Estimation in Emotion Classification from Audio Using Convolutional Neural Network

II. RESEARCH METHODS

A. Data

This study utilizes two main data sources, namely tourist attraction data in Central Java, and User data that has provided reviews of tourist attractions in Central Java obtained through the Google Places API, which is an Application Programming Interface that provides location-based geographic data via the internet using HTTP [4]. Tourist attraction data in Central Java was collected using the midpoint coordinates of each city/district as a search reference. The search was conducted within a radius of 100,000 meters from the midpoint with search parameters set to obtain places with the category "tourist_attraction". The data taken initially amounted to 1400 tourist attractions, then filtered into 432 tourist attractions in Central Java based on 13 attributes used. The tourist attraction attributes used in this study can be seen in Table I.

TABLE I. DATA ATRIBUTE

ID	Attribute	Description	Data Type
1	Place_id	Unique ID of Tourist Attraction	*Numeric*
2	Place Name	Name of Tourist Attraction	*String*
3	formatted_phone_number	Phone Number of Tourist Attraction	*String*
4	formatted_address	Address of Tourist Attraction	*String*
5	city	City	*String*
6	website	Website	*String*
7	rating	Rating of Tourist Attraction	*Numeric*
8	user_ratings_total	Total Ratings	*Numeric*
9	photo_preference	Photo of Tourist	*String*
10	lng	Longitude	*Numeric*
11	url	Google Maps Website	*String*
12	url_photo	Google Maps Photo	*String*
13	type_tourist	Type of Tourist Attraction	*String*

B. Research Stagees

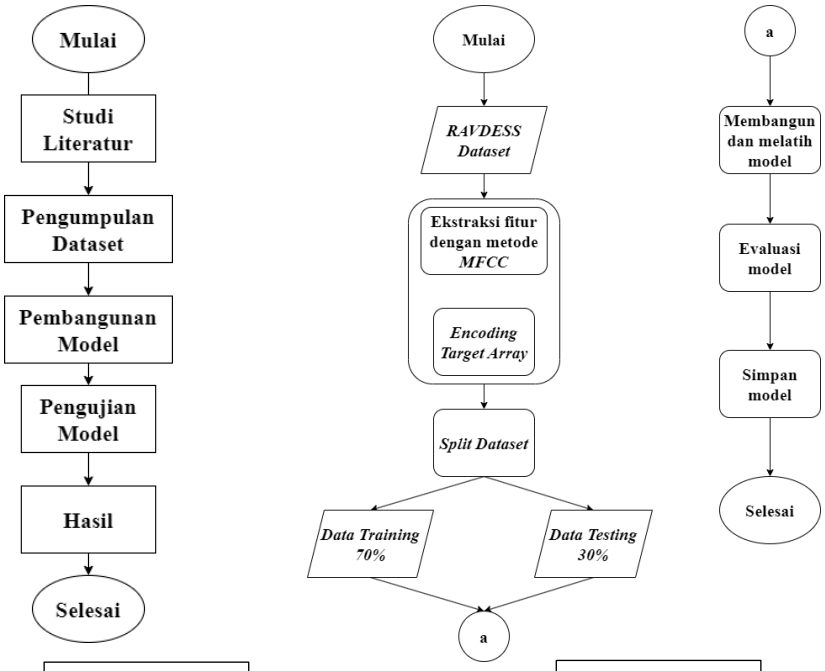

Figure 1. Flowchart of Research: (1a) Flowchart of Research Stages (1b) Flowchart of Model Building

This research is experimental research that compares two types of optimizers. Experimental research is used to test causal hypotheses about the relationship between two or more variables. The researcher will conduct a series of experiments to collect data, implement the model, and analyze the experimental results to draw conclusions about the effectiveness of two optimization techniques in each context.

The research phase begins with a data collection process conducted through the open-source platform Kaggle. The dataset used is RAVDESS [7], which is a collection of audio data containing expressions of emotion in the form of speech. This dataset is downloaded from the link available at kaggle.com.

Data pre-processing is performed to prepare the data for use by the model. This includes audio feature extraction using the Mel-Frequency Cepstral Coefficients (MFCCs) technique to convert the audio signal into a numerical representation that can be processed by the Convolutional Neural Network (CNN) model. Each audio data is also assigned a corresponding emotion label, and a label encoding process is performed to convert categorical labels into numerical form, namely 0 for neutral, 1 for calm, 2 for happy, 3 for sad, 4 for angry, 5 for fear, 6 for disgust, and 7 for surprise. The dataset is then divided into two subsets, namely 70% training data and 30% testing data.

CNN architecture specifically designed for emotion classification from audio data was built. This architecture consists of several layers, including convolution, pooling, and fully connected layers. The model training process is performed using a subset of training data and two types of optimizers, namely Stochastic Gradient Descent (SGD) and Adam. Training parameters such as learning rate and batch size are also adjusted to optimize model performance.

After the training process is complete, the model is evaluated using a subset of testing data with evaluation metrics such as training accuracy and validation accuracy. The last stage is result analysis, which compares the model performance between the use of SGD and Adam optimizers. Prior to the entire process, the dataset is first cleaned through incremental pre-processing, where data is selected and truncated using the MATLAB application. Data that is corrupted or defective after this process will be

eliminated and not used in the study.

Analysis of the results is done by evaluating the model to determine the level of performance of the model in classifying various classes. One of the model evaluation techniques is to use confusion matrix. Confusion matrix is a matrix that shows and compares the actual or true value with the predicted value of the model that can be used to generate evaluation metrics such as Accuracy, Precision, Recall, and F1-Score.

There are 4 values generated in the confusion matrix table, namely True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TN is the number of negative data detected correctly. FP is negative data but detected as positive data. TP is positive data that is detected correctly. FN is the opposite of True Positive, so it is positive data, but detected as negative data.

TABLE II. TABLE CONFUSION MATRIX

		True Values	
		True	***False***
Prediction	***True***	*TP* *Correct result*	*FP* *Unexpected result*
	False	*FN* *Missing result*	*TN* *Correct absence of result*

Precision is data that is retrieved based on less information. In binary classification, precision can be made equal to the positive predictive value. Precision is a measure of the model's accuracy in identifying positive samples. Precision shows how often the model's positive predictions are correct. The formula for finding precision [8]:

$$Precision = \left(\frac{TP}{(TP+FP}\right) \times 100\% \quad (1)$$

Recall is the successful removal of data relevant to the query. In binary classification, recall is known as sensitivity. The appearance that the retrieved relevant data is agreeing with the query can be seen by recall. Recall is a measure of the model's sensitivity in capturing all positive samples. Recall shows how good the model is at identifying all the samples that are truly positive. The formula for finding Recall [8]:

$$Recall = \left(\frac{TP}{(TP+FN)}\right) \times 100\% \quad (2)$$

F1-Score value or also known as the F Measure is obtained from the results of precision and recall between the predicted categories and the actual categories. F1-Score is a harmonized measure of precision and recall. F1-Score provides an overall picture of the model's performance by considering both. F1-Score provides a balance between precision and recall. The formula for finding F1-Score [8]:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (3)$$

Accuracy is used to measure the performance of an algorithm in a way that can be interpreted. The accuracy of a model is usually determined after the model parameters and is calculated as a percentage. It is a measure of how accurate the model's predictions are compared to the actual data and accuracy is in train. The formula for finding accuracy [8]:

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (4)$$

II. RESULTS AND DISCUSSIONS

A. Pre-Processing Data

The data used in this study are human voice recordings in English, which have been pre-labeled. The dataset used is the Ryerson Audio-Visual Dataset of Emotional Speech and Song (RAVDESS), which was collected from the Kaggle website. The voice accents in this dataset are from North America. The dataset consists of two types of files, namely files containing individual conversations and files containing songs. In this study, 1440 audio files of recorded conversations were used. The duration of each audio in this dataset ranges from 3.5 to 5 seconds. The recorded conversations in this dataset involve male and female voices, each consisting of 12 individuals. Each audio file in this dataset has a 16 bit, 48kHz format with a .wav file extension.

TABLE III. TOTAL DATA

Emosi	**Pria**	**Wanita**
Netral	48	48
Tenang	96	96
Bahagia	96	96
Sedih	96	96
Marah	96	96
Takut	96	96
Jijik	96	96
Terkejut	96	96
Total	720	720

The audio data then goes through a silent removal process, which is the process of cutting the audio at the beginning and end to remove the silent parts before and after the actor speaks. This process is done with the help of MATLAB software. Sample data results that have been cut or have been done silent removal can be seen in Figure 2 and Figure 3 below.

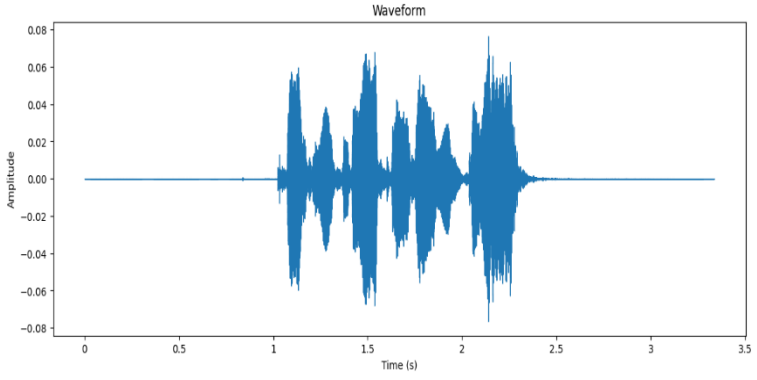

Figure 2. Happy Audio Visualization Before Silent Removal

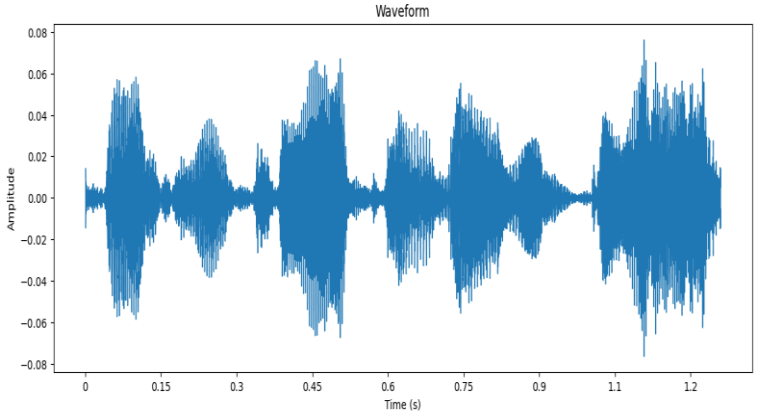

Figure 3. Happy Audio Visualization After Silent Removal

The data that has gone through the cutting process and is used as data in the study can be seen in table 3 below.

TABLE IV. AUDIO COUNT AFTER CUTTING

Kelas	Emosi	Jumlah
1	Netral	96
2	Tenang	190
3	Bahagia	191
4	Sedih	192
5	Marah	192
6	Takut	191
7	Jijik	192
8	Terkejut	191
	Total	1435

Audio signals are obtained with the help of the librosa library with python programming. Then these audio signals are used for the MFCC extraction process. After applying Discrete Cosine Transform (DCT) on the log energy of the Mel filter, each frame generates an MFCC vector. For example, if we use an MFCC count of 13 coefficients, each frame generates 13 MFCC coefficients. Since the audio signal uses MFCC for many frames, the output value of MFCC extraction is a matrix. Where each row represents one frame, and each column represents one MFCC coefficient. For example, for a signal divided into N frames and n_mfcc MFCC coefficients, the MFCC matrix has a size of $N \times n_{mfcc}$.

After the values are known, the data can be presented in the form of a table like the following table 4 and fill the non-numeric values (NaN) or empty values to 0. It can be found that NaN values occur due to the unequal length of the result vector. Replacing it with 0 will indicate that there is no information in the result to avoid loss of information [9].

TABLE V. SAMPLE MFCC EXTRACTION RESULTS FROM AUDIO

	1	**2**	**3**	**4**	**…**	**13**
1	-452,251	197,7236	-7,87114	-19,7107	…	0,542556
2	-458,808	204,3042	-23,309	-20,7977	…	-1,46148
3	-485,871	203,8673	-29,3221	-27,0689	…	-9,04032
4	-503,372	209,1617	-11,0603	-30,559	…	-7,3006
5	-533,569	196,6293	-1,95884	-20,9173	…	-5,19639
…	…	…	…	…	…	…
322	-481,685	148,3774	20,60678	2,111137	…	0,33366

Table V is a sample of data extracted from MFCC measuring 13 x 322. All data is then subjected to padding so that it has the same frame length and so that the size of each data is the same. The data is then divided into training data and testing data with 70% training data and 30% testing data [10].

TABLE VI. SPLIT DATA TRAINING AND TESTING

Kelas	**Emosi**	***Training***	***Testing***
1	Netral	67	29
2	Tenang	133	57
3	Bahagia	133	58
4	Sedih	134	58
5	Marah	134	58
6	Takut	133	58
7	Jijik	134	58
8	Terkejut	133	58
	Total	1001	434

After the data is divided, the data is normalized using the following formula (5).

$$x_{baru} = \frac{x_{lama} - \mu}{\sigma} \tag{5}$$

Where:

x_{baru} : Normalized value

x_{lama} : Old data value (not normalized)

μ : Mean

σ : Standard deviation

B. Convolutional Neural Network (CNN) Architecture Design

At this stage, the architecture design for the CNN model that will be used in the research is carried out. This research will consist of 2 convolution layers with ReLU activation function and followed by a fully connected layer consisting of a flatten layer. After that there is a hidden layer and an output layer. It is in this output layer where the Softmax activation function is used to classify the audio into the 8 existing emotion classes.

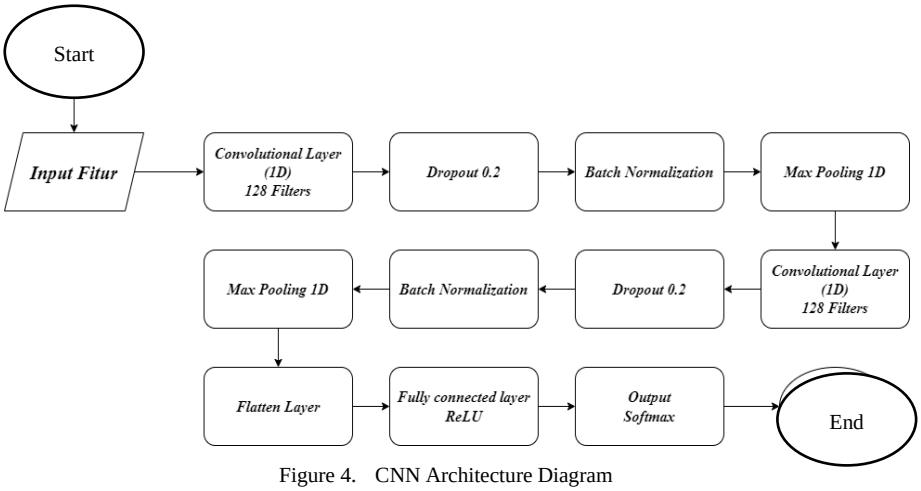

Figure 4. CNN Architecture Diagram

TABLE VII. SUMMARY MODEL CONVOLUTIONAL NEURAL NETWORK (CNN)

Model Sequential			
No.	***Layer (type)***	***Output shape***	***Param#***
1.	Conv1d_128 (Conv1D)	*(None, 4148, 128)*	*128*
2.	Dropout_128 (Dropout)	*(None, 4148, 128)*	*0*
3.	Batch_normalization_128 (BatchNormalization)	*(None, 4148, 128)*	*128*
4.	Max_pooling_128 (MaxPooling1D)	*(None, 2092, 128)*	*0*
5.	Conv1d_129 (Conv1D)	*(None, 2090, 128)*	*3104*
6.	Dropout_129 (Dropout)	*(None, 2090, 128)*	*0*
7.	Batch_normalization_128 (BatchNormalization)	*(None, 2090, 128)*	*128*
8.	Max_pooling_128 (MaxPooling1D)	*(None, 1045, 128)*	*0*
9.	Flatten_64 (Flatten)	*(None, 33440)*	*0*
10	Dense_128 (Dense)	*(None, 64)*	*2140224*
11.	Dense_129 (Dense)	*(None, 8)*	*520*
Total Params	2144232		
Trainable Params	2144104		
Non-trainable Params	128		

The next stage is training the convolutional neural network model with training data. Before the training process is carried out, it is necessary to initialize the parameters that will be used such as loss function, optimizer, number of epochs, number of batches, and callback. The model will go through the training process by being processed in google colab. In addition, 8 emotion classes have been represented into 8 labels representing 8 emotions. 0 = Neutral, 1 = Calm, 2 = Happy, 3 = Sad, 4 = Anger, 5 = Fear, 7 = Disgust, and 8 = Surprise. The following table 7 initializes the parameters for the CNN model.

TABLE VIII. PARAMETER INITIALIZATION

Parameter	Input
Output class	8
Output layer activation	Softmax
Hidden layer activation	ReLU
Epoch	50
Batch size	10 / 32
Optimizer function	SGD / Adam
Loss function	Categorical cross entropy

TABLE IX. ACCURACY DATA OF TRAINING RESULTS AND MODEL VALIDATION WITH SGD OPTIMIZER

Learning Rate	Batch_size = 10		Batch_size = 32	
	Accuracy	Val_ Accuracy	Accuracy	Val_ Accuracy
0.001	0.93	0.47	0.73	0.42
0.002	0.96	0.53	0.84	0.46
0.003	0.97	0.44	0.91	0.42
0.004	0.98	0.47	0.92	0.44
0.005	0.97	0.45	0.90	0.44
0.006	0.97	0.47	0.91	0.45
0.007	0.98	0.48	0.96	0.47
0.008	0.98	0.45	0.95	0.48
0.001	0.99	0.34	0.90	0.47

TABLE X. ACCURACY DATA OF TRAINING RESULTS AND MODEL VALIDATION WITH ADAM OPTIMIZER

Learning Rate	Batch_size = 10		Batch_size = 32	
	Accuracy	Val_ Accuracy	Accuracy	Val_ Accuracy
0.0001	0.91	0.43	0.93	0.44
0.0002	0.89	0.44	0.89	0.48
0.0003	0.96	0.45	0.83	0.40
0.0004	0.85	0.45	0.59	0.32
0.0005	0.49	0.35	0.61	0.32
0.0006	0.45	0.37	0.88	0.43
0.0007	0.92	0.42	0.77	0.38
0.0008	0.81	0.41	0.71	0.39
0.0009	0.73	0.36	0.73	0.40

Based on Table IX and Table X, the yellow cells indicate the optimal parameters for batch size and learning rate for the model used. The basis for this selection is to see the highest validation accuracy (Val_accuracy). For the SGD optimizer, the model works optimally with a learning rate of 0.007 and a batch size of 10. For the adam optimizer, the model works optimally with a learning rate of 0.0002 and a batch size of 10. However, due to the low validation accuracy, the model still needs further review. One of the symptoms of overfitting is the difference between training accuracy and validation accuracy, which

can be seen in Table IX and Table X.

C. Model Analysis with SGD Optimizer

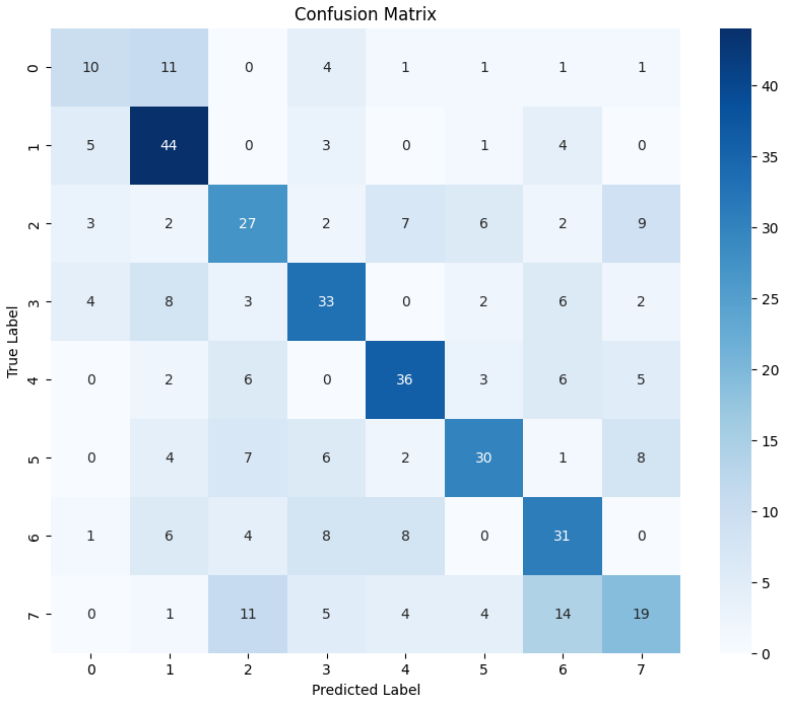

Figure 5. Confusion Matrix Model CNN Optimizer SGD

Based on Figure 4.6 True Labels are the rows of the matrix showing the original labels of the dataset and Predicted Labels are the columns showing the predicted labels made by the model. Diagonal elements (True Positives) (e.g., (0,0), (1,1), (2,2), etc.) represent the number of correct predictions for each class. For example, the cell at position (1,1) indicates that the model correctly predicted class 1 (the second class) 44 times. Then the Non-diagonal Elements (False Positives and False Negatives) represent incorrect predictions. For example, the cell at (1,2) shows the number of instances where class 1 was incorrectly predicted as class 2. The model correctly predicted class 0 for 10 times, but the model incorrectly predicted class 0 as class 1 for 11 times, class 3 for 4 times, and so on. The model correctly predicted class 1 for 44 times, but the model incorrectly predicted class 1 as class 0 for 5 times, class 3 for 3 times, and so on. The same goes for the other rows. Darker colors indicate larger numbers, while lighter colors indicate smaller numbers. The class with the darkest color on the diagonal is class 1 with 44 correct predictions. Next, we will calculate the precision, recall, and f1_score values. From the data above, it is known:

TABLE XI. TP, FP, FN, AND FN VALUES OF 8 EMOTION CLASSES MODEL WITH SGD OPTIMIZER

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
TP	10	44	27	33	36	30	31	19
FP	13	34	31	27	22	17	34	25
FN	19	13	31	25	22	28	27	39
TN	392	343	345	357	354	359	342	351

TABLE XII. PRECISION, RECALL, AND F1_SCORE VALUES WITH SGD OPTIMIZER

Label	Precision	Recall	F1-Score
0	43,47%	52,63%	0,38
1	56,41%	77,19%	0,65
2	46,55%	46,65%	0,47
3	55%	42,3%	0,55
4	62,06%	62,06%	0,62
5	63,82%	51,72%	0,57
6	47,69%	53,44%	0,5
7	43,18%	32,75%	0,37

For the model using the SGD optimizer, Label 1 performed best with an F1-Score of 0.65, precision of 56.41%, and recall of 77.19%. This shows that the model can identify most of the Label 1 samples accurately and consistently.Meanwhile, Label 7 has the lowest performance with F1-Score 0.37, precision 43.18%, and recall 32.75%. This indicates the model struggled to identify and capture samples for this category.

D. Model Analysis with ADAM Optimizer

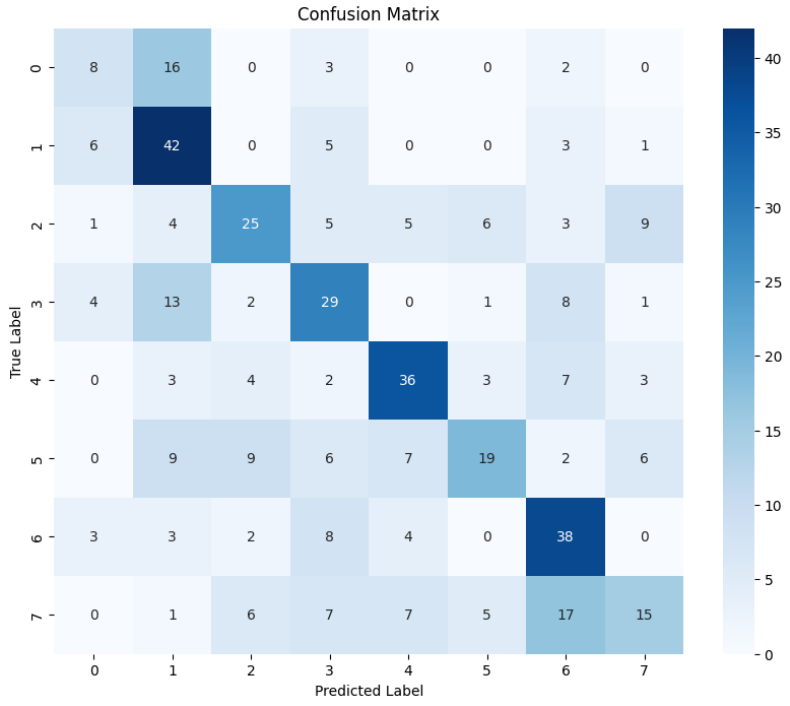

Figure 6. Confusion Matrix Model CNN Optimizer Adam

The cell at position (1,1) shows that the model correctly predicted class 1 (second class) 42 times. Then the Non-diagonal Elements (False Positives and False Negatives) represent incorrect predictions. In row 0 (Original Label = 0) the model correctly predicted class 0 for 8 times, but incorrectly predicted class 0 as class 1 for 16 times, class 3 for 3 times, and so on. Then in row 1 (Original Label = 1), the model correctly predicted class 1 for 42 times, but the model incorrectly predicted class 1 as class 0 for 6 times, class 3 for 5 times, and so on. The model still predicts class 1 more with 42 correct predictions.

Next, the precision, recall, and f1_score values will be calculated. From the data above, it is known:

TABLE XIII. TP, FP, FN, AND FN VALUES OF 8 EMOTION CLASSES MODEL WITH ADAM OPTIMIZER

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
TP	8	42	25	29	36	19	38	15
FP	14	49	23	36	23	15	42	20
FN	21	15	33	29	22	39	20	43
TN	391	328	353	340	353	367	334	356

TABLE XIV. PRECISION, RECALL, AND F1_SCORE VALUES WITH ADAM OPTIMIZER

Label	**Precision**	**Recall**	**F1-Score**
0	36,36%	27,58%	0,31
1	46,15%	73,68%	0,57
2	52,08%	43,1%	0,47
3	44,61%	50%	0,47
4	61,01%	62,06%	0,62
5	55,88%	32,75%	0,41
6	47,5%	65,51%	0,55
7	42,85%	25,06%	0,32

The model using Adam's optimizer shows the best performance in detecting Anger emotion (Label 4), which is characterized by high precision, recall, and F1-Score.This means that the model can well identify and predict this emotion with good accuracy. In contrast, the model had the greatest difficulty in detecting the Neutral emotion (Label 0), which is characterized by the lowest precision and F1-Score values, and the Surprised emotion (Label 7), which has the lowest recall value. This suggests that the model needs further improvement to increase accuracy and consistency in detecting these emotions.

CONCLUSION

Based on the results of the research that has been done, using the Convolutional Neural Network (CNN) model which uses 9 types of learning rates and 2 types of batch sizes, it can be concluded that,

1. Convolutional Neural Network (CNN) training results using the SGD optimizer, have a validation accuracy value of 53% using a learning rate of 0.002 and batch size 10.
2. As for the Adam optimizer, the best training results are at a learning rate of 0.0002 and a batch size of 32. Where the validation accuracy obtained is 48%.
3. The results of the confusion matrix analysis of the CNN model with the SGD optimizer have a higher general accuracy, which is 53% compared to the Adam optimizer which is 48%.

In addition, the results of the training also show that the model is easier to recognize emotions labeled 1, namely audio with calm emotions and the most difficult to recognize audio labeled 0, namely audio neutral emotions. So, in this case of emotion classification from the RAVDESS audio dataset, the better optimizer is the Stochastic Gradient Descent (SGD) optimizer.

REFERENCES

[1] M. F. Naufal, "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 8(2), pp. 311-317, 2021.

[2] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014.

[3] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," arXiv, 2016.

[4] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradients Descent," In Proceedings of COMPSTAT'2010, pp. 177-186, 2010.

[5] F. Zou, L. Shen, Z. Jie, W. Zhang and W. Liu, "A Sufficient Condition for COnvergences of Adam and RMSProp," In Proceedings of the IEEE/CVF Confrence on computer vision and pattern recognition, pp. 11127-11135, 2019.

[6] D. Ardiyansyah and J. Jayanta, "Model Klasifikasi Emosi Berdasarkan Suara Manusia dengan Metode Multilayer Perceptron," Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya, vol. 2, no. 1, pp. 689-702, 2021.

[7] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English.," 2018. [Online]. Available: www.kaggle.com. [Accessed April 2024].

[8] E. Alpaydin, Introduction to Machine Learning, Cambridge, Massachusetts: The MIT Press, 2014.

[9] S. Patil and D. G. K. Kharate, "Implementation of SVM with SMO for Identifying Speech Emotions Using FFT and Source Features," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, 2021. S. T. Alexander, "The Mean Squared Error (MSE) Performance Criteria," in *Adaptive Signal Processing*, Texts and Monographs in Computer Science, Springer, New York, NY, 1986.

[10] A. Muhaimin, D. D. Prastyo and H. Horng-Shing Lu, "Forecasting with Recurrent Neural Network in Intermittent Demand Data," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 802-809, doi: 10.1109/Confluence51648.2021.9376880.