Feature Extraction of Library Noise with SpecAugment Data Augmentation for Classification Model in Embedded Machine Learning Using Convolutional Neural Network

Adian Fatchur Rochim, Dania Eridani, Alvin Zulham Firdananta

Department of Computer Engineering, Diponegoro University, Semarang, Indonesia

Environmental Noise is an unwanted sound created by human activity. In Library, some unwanted noise has bad impact to visitors who want still focus reading and learning something in library. Noise Classification using embedded system should have additional abilities to classify sound, in order to provide more accurate warnings to visitors. To make the sound classification in embedded system more accurate, the creation of classification models must also be done as much as possible. One of the steps in making the classification model is feature extraction. Feature extraction from audio signal become essential phase to make machine learning who classifying audio have better accuracy. But, in embedded machine learning or known as TinyML who have certain limitation especially in performance capabilities, the selection of methods must be done in more detail to produce good abilities. Spectrogram, Mel Frequency Cepstral Coefficients (MFCC), Mel-Filterbank Energy (MFE) using K-Nearest Neighbor algorithm will be compared and the implementation of SpecAugment data augmentation to find the best method for audio feature extraction to make classification model created using Convolutional Neural Network. The device build based on Arduino Nano 33 BLE Sense. MFE become the best method for feature extraction environmental noise with 86.4%, followed by Spectrogram with 81.3% and last MFCC with 78.8% and data augmentation using SpecAugment has been shown to improve accuracy for noise classification.

Keywords: Noise Classification, Embedded Machine Learning (TinyML), Feature Extraction, Spectrogram, Mel Frequency Cepstral Coefficients (MFCC), Mel-Filterbank Energy (MFE)

1. Introduction

Noise became the most complain conveyed by visitors to admin in Library [1]. Alerting visitors using a tools become a way to reduce noise in the library is to use tools that can warn library visitors when they reach a certain level of noise [2]. The tools should have some ability to classify the noise produced around it. Adding artificial intelligence to tools can make the tool can recognize the type of sound. Adding Artificial Intelligence to tools can make the tool could recognize the type of sound. One of subsection of Artificial Intelligence (AI) is Machine Learning, which have ability to automatically learn from given concept or data without directly programmed [3]. Machine Learning has successfully entered in several applications of technology, including embedded system [4]. The challenges implementing machine learning in an embedded

system/microcontroller is the limitation of device processing power [5]. Embedded Machine Learning also known as TinyML can perform well under 128-512KB SRAM, 0.5MB-2MB eFlash, and 0.1-0.3W [4].

Environmental Noise can be defined as unwanted sound created by human activity. Environmental noise classification in embedded system has some challenges such as how to make accurate classification in a short amount of time, and how to create the classification model can work in devices with limited capabilities [4]. Environmental Noise Classification can be divided into 2 stages, feature extraction and classification [6].

Convolution Neural Network. Convolution Neural Network is a classifier algorithm which can be implemented into TinyML. Convolution Neural Network (CNN) become the most used algorithm for machine learning in fields of computer image recognition, video analysis, drug analysis, natural language processing and other fields [7].

Many researchers have research about Environmental Noise Classification, but we haven't found research about the best feature extraction method in environmental noise classification using Convolution Neural Network in the library environment. In this paper will focused on what is the best feature extraction from environmental noise using Spectrogram, Mel Frequency Cepstral Coefficients (MFCC), and Mel-Filterbank Energy (MFE) for environmental noise classification using Convolution Neural Network in Library environment.

2. Literature Review

J. Nordby in 2019 was build classification system using embedded device for classifying urban sound around it and he use Convolutional Neural Network algorithm for the model creation and spectrogram for the feature extraction with dataset from UrbanSound8K which containing 10 class (air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music) with overall accuracy is 72% [8].

J. Alves, P. Guerreiro, G. Marques and G Marques have similar research building environmental noise classification for smart city using Raspberry Pi and Pre-Amp Mics, and the classification ability was created using UrbanSound8K with CNN for the algorithm and have good performance classifying sound with overall accuracy 65% [9].

L. G. C. Vithakshana and W. G. D. M. Samankula researched abound animal classification system using CNN and they are using Mel-frequency Cepstral Coefficient (MFCC) for the feature extraction with Convolutional Neural Network as main algorithm and have overall accuracy quite high it is 91.3% [10].

João Pedro Duarte Galileu was doing reseach for MSc Thesis and take topic about sound classification for city and have using MFCC as feature extraction method and doing test with 5 machine learning algorithms (Logistic Regression, Support Vector Machines, Random Forests, Nearest Neighbor

and Artificial Neural Networks) with Artificial Neural Networks become the best algorithm compared to another algorithm he used [11].

3. Proposed Method

3.1 Feature Extraction

Human auditory system has listening range of 20 Hz-20 kHz, human can easily classify various sound without putting more effort. If we want machine have same ability as human, it will need extra effort because machine have problem called machine hearing [12]. Feature extraction is a step to extract important data from audio waveform into multiple frames [13]. Feature extraction is an implementation of Digital Signal Processing with K-Nearest Neighbor algorithm, the purpose of using feature extraction is to make simpler shape from audio waves. Feature extraction can help the machine better at distinguishing sounds. Some examples of feature extraction method are Spectrogram, Mel Frequency Cepstral Coefficients (MFCC), Mel-Filterbank Energy (MFE). Result from feature extraction will be transferred as training in classification stages. The detailed explanation about feature extraction method will be explained in next section.

3.2 Spectrogram

Spectrogram is a method transform signal record into image of time-frequency by using the Short Time Fourier Transform (STFT) [14]. Spectrograms will extract audio waveform into image which is two-dimensional graph. X axis will be represented as time and Y axis will be represented as frequency and have color to indicates the amplitudes of those points [15]. The result for audio feature extraction from raw audio data can be seen in fig 1 and fig 2.

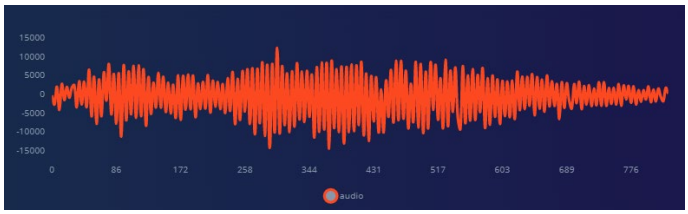


Fig 1 Raw audio data

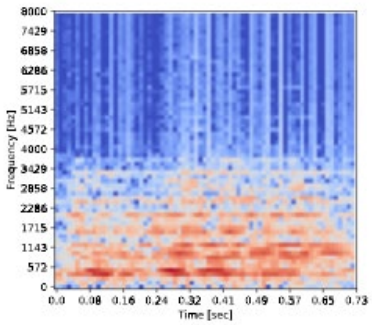


Fig 2 Feature extracted from audio data in fig 1 using Spectrogram

Spectrogram have 4 parameters to extract audio data into simpler version. They are frame length, frame stride, frequency bands and normalization noise floor. Frame length is the length of frame in seconds. Frame stride is the step between successive frames in seconds. Frequency bands is number of frequency bands from power of two (64, 128, 256). Normalization noise floor is ignored sound when less than parameter value in dB.

3.3 Mel-Filterbank Energy (MFE)

Mel-Filterbank Energy (MFE) have similarity with Spectrogram which extract frequency and time from an audio signal, however MFE used nonlinear scale called Mel-scale defined as

$$f_{mel} = 2595 log_{10}(1 + \frac{f}{700})$$

where f is the original frequency in Hz [16]. Triangular filters are applied on a Mel-scale to extract frequency band and number of frequency feature. The result for audio feature extraction using MFE from raw audio as shown in fig 1 can be seen in fig 3.

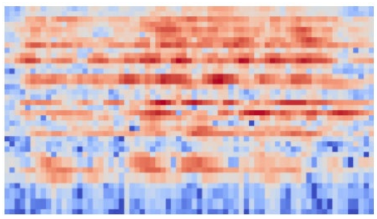


Fig 3 Feature extracted from audio data in fig 1 using MFE

MFE have 7 parameters and 3 of them are similar with spectrogram, they are frame length, frame stride, and normalization noise floor. The other 4 parameters are filter number, FFT length, low frequency, and high frequency. Filter number is the number of filters in filter bank. FFT length is number of FFT points. Low frequency is lowest band of Mel filters in Hz. High frequency is highest band edge of Mel filters in Hz.

3.4 Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) is method to extracts coefficient from an audio signal using non-linear scale based on human hearing perceptions [17]. MFCC will includes windowing the signal, applying DFT, and taking the log of magnitude, then warping the frequencies on a Mel scale [18]. The result for audio feature extraction using MFCC from raw audio as shown in fig 1 can be seen in fig 4.

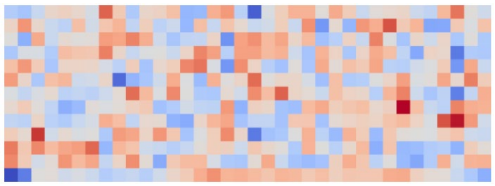


Fig 4 Feature extracted from audio data in fig 1 using MFCC

MFCC have 8 parameters, 2 of them are similar with Spectrogram and MFE, 4 of them are similar with MFE parameters. The 2 new parameters are normalization window size and number of coefficients. Number of coefficients is the number of cepstral coefficients. Normalization window size is the size of sliding window for local normalization.

3.5 Data Augmentation

Data Augmentation is a technique to randomly transform data and will increasing variations of existing dataset [19]. Data Augmentation method will be use in this research is SpecAugment. SpecAugment is a simple data augmentation which directly applied to the feature extraction of a neural network [20]. The example of data augmentation using SpecAugment is shown in fig 5.

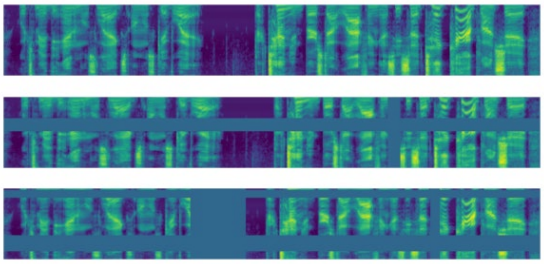


Fig 5 Example of SpecAugment Data Augmentation [20]

SpecAugment have following augmentation policy [20]:

1. Time warping is applied via the function sparse_image_warp of tensorflow
2. Frequency masking is applied so that f consecutive mel frequency channels [f_0, f_0+f] are masked
3. Time masking is applied so that t consecutive time steps [t_0, t_0+t] are masked

3.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) are a subset of machine learning and become the core of deep learning algorithm. CNN contain some layer such as input layer, some hidden layer, and output layer. Each layer node is connected to another node [21]. For CNN implementation in this study, will be using Keras. Keras is open-source library for neural networks, or it could be defined as backbone of neural network [22]. Because using raw audio are not recommended for machine learning especially using CNN algorithm [23], the input layer of CNN will be feature that extracted using one of 3 methods mentioned above. Here the example of schematic diagram basic Convolutional Neural Network.

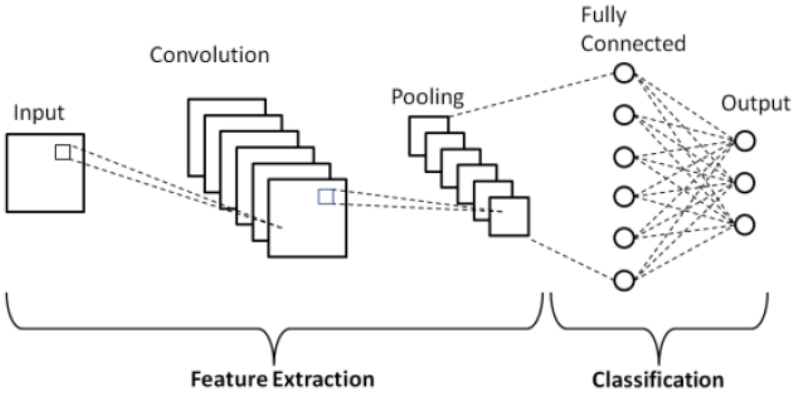


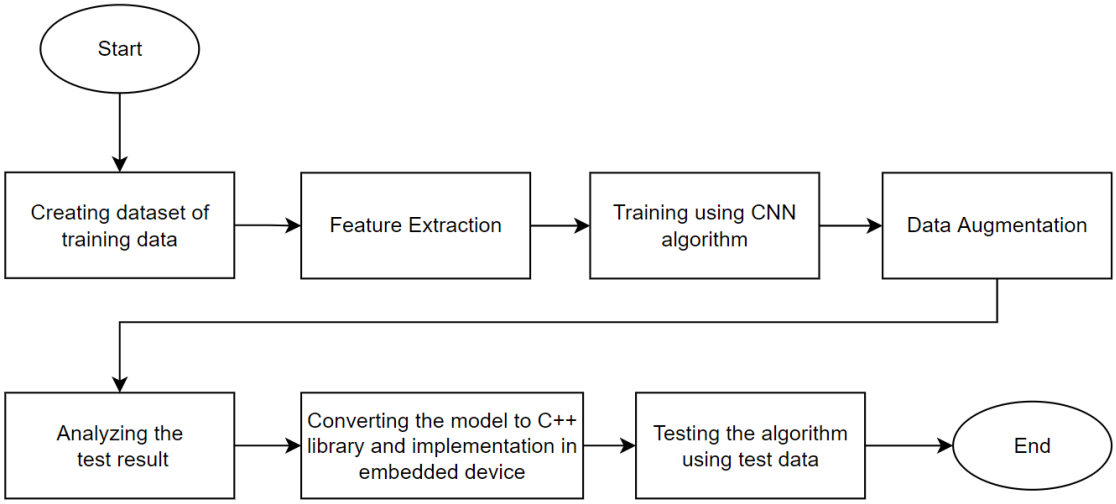
Fig 6 Basic CNN schematic diagram [24]

The hidden layer used in this study are reshape layer, conv / pool layer, flatten layer, dropout and dense layer. Reshape layer is layer that reshape layer into given shape. Conv / pool layer is related layer contain Conv1D layer and MaxPooling1D layer. Conv1D layer will create convolution kernel that convolved with input layer over single spatial or temporal dimension to create tensor output. Then, MaxPooling1D layer use for down samples representation of the input using maximum value over spatial window size. Flatten layer is a layer who flatten the input. Dropout layer will randomly dropout to the input units to 0 with a frequency of rate during training time. Last, it is dense layer which NN layer densely connected. All those layer's definition is from official Keras website [25].

In this paper, we will use 3 feature extraction method, so the CNN hidden layers are slightly different. For the first method, Spectrogram the hidden layers are reshape layer, 4 conv/pool layer, flatten layer, dropout, dense layer, and dropout again. The second method, MFE the hidden layers are reshaped layer, 2 conv / pool layer, flatten layer, dropout, dense layer and dropout again. Last, for the MFCC the hidden layers are reshape layer, 3 conv / pool layer, flatten layer and dropout.

3.7 Research Procedure

The research procedure of this study can be seen in fig 6. First, dataset which contain sound sample from 5 categories are converted into simpler form using feature extraction. The result from feature extraction will be used in training step using Convolutional Neural Network (CNN) algorithm. After the model created successfully, the model will be converted into C++ library for implemented on embedded devices. Then, the implemented classification in embedded device will be testes using the test data.



The dataset consists of 1071 sound sample and divided into 5 classes, namely fallingObj (216 sound sample), horn (198 sound sample), human (246 sound sample), phone (225 sound sample), siren (186 sound sample). Each class have 15%-18% testing data in this test. The sound sample data obtained from UrbanSound8K dataset, and the others are obtained own data gathering and sample from YouTube.

The first class named fallingObj is a represent from noise caused by something that fell. Horn is a class that represent sounds generated from the vehicle horn. Human is a class which contains noise sound produced by human activity like speaking too loudly and laugh. Phone class contains some noise sample caused by such as incoming messages, app notifications, and phones that are too loud. Last, siren class is a class contains noise generated by the sound of ambulance sirens, police siren or firefighter siren.

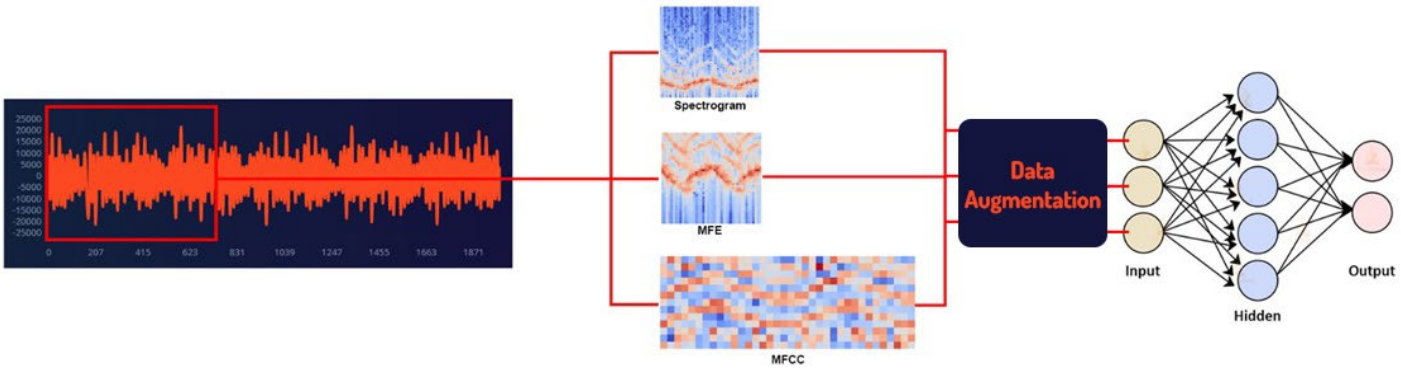


Fig 7 Workflow of the learning process

The process started with processing raw audio data into selected feature extraction. Each feature extraction will used in each different learning process. Total, we're doing 3 learning process for testing 3 feature extraction method. After doing the feature extraction, the feature will be processed in data augmentation. Ten, the result from data augmentation will be used as input layer for training in Convolutional Neural Network Algorithm. Last, each model from 3 feature extraction will be converted into C++ Library to be used in microcontrollers. We will test the classification performance without data

augmentation and with data augmentation. Then the result will be compared which one have better performance.

The device will be used in this research is Arduino Nano 33 BLE Sense which have built in microphone sensor MP34DT05 with additional DFRobot Analog Sound Level Meter as decibel sensor. The devices will be placed in a library and have alert system using LED rotator which will light up when detecting human noise and phone noise and will ignore other noise class. System will run classification when the decibel sensor reached noise level 60 dB.

4. Result and Discussion

Because raw audio data is not recommended as an input to a classification system [23], the next step after importing training and testing dataset will be converting audio waves using feature extraction. Feature extractions used in this test using following parameters with the same parameters in different methods having the same value.

Table 1 Parameters used in each method

Parameters	Spectrogram	MFE	MFCC
Frame length	0.02	0.02	0.02
Frame stride	0.01	0.01	0.01
Frequency bands	128	-	-
Normalization noise floor (dB)	-52	-52	-
Filter number	-	40	40
FFT length	-	256	256
Low frequency	-	300	300
High frequency	-	Not set	Not Set
Normalization window size	-	-	101
Number of Coefficients	-	-	13

Frame length is the length of frame in seconds. Frame stride is the step between successive frames in seconds. Frequency bands is number of frequency bands from power of two (64, 128, 256). Normalization noise floor is ignored sound when less than parameter value in dB. Filter number is the number of filters in filterbank. FFT length is number of FFT points. Low frequency is lowest band of mel filters in Hz. High frequency is highest band edge of mel filters in Hz. Number of coefficients is the number of cepstral coefficients. Normalization window size is the size of sliding window for local normalization.

We attempt to equalize the parameters of each method used except for some parameters that are not present in other methods. Then, after all feature extraction have been created, all those features will be used for creating models. The models will be created with Convolutional Neural Network with following parameters.

Table 2 CNN Parameters

1st 1DConv	8 Neurons, 3 kernel size, 1 layer
1st Dropout	0.25
2nd 1DConv	16 neurons, 3 kernel size, 1 layer
2nd Dropout	0.25
Flatter layer	True
Epoch	1000
Learning rate	0.005

After model for classification created, then the model will be converted into C++ Library. Using Edge Impulse can make the model creation and converting into C++ Library much easier. From the test, the accuracy, F1 score, precision and recall will be calculated. Accuracy is number of correct predictions divided by number of predictions. Here is the formula of accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

F1 Score is harmonic mean between recall and precision. They are formulated as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Where: TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

Result of live test before data augmentation applied is shown in fig 8 – fig 11 below, with DA as result after Data Augmentation

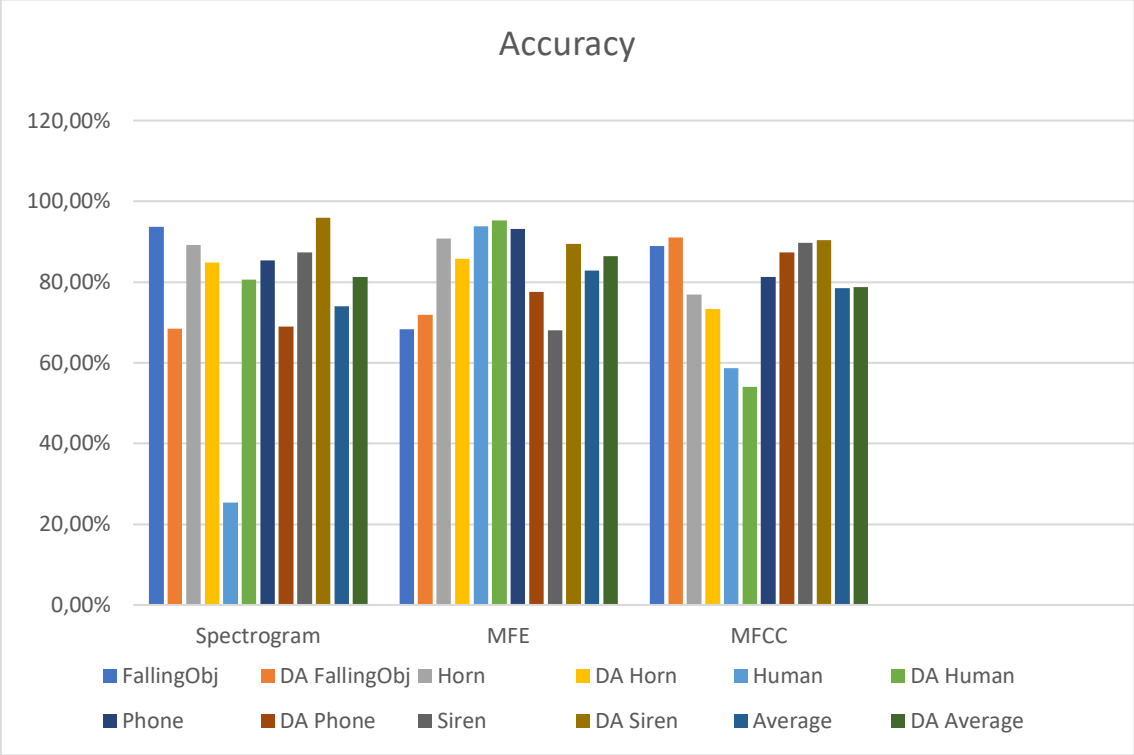


Fig 8 Result from accuracy for each class and average before and after processing by data augmentation

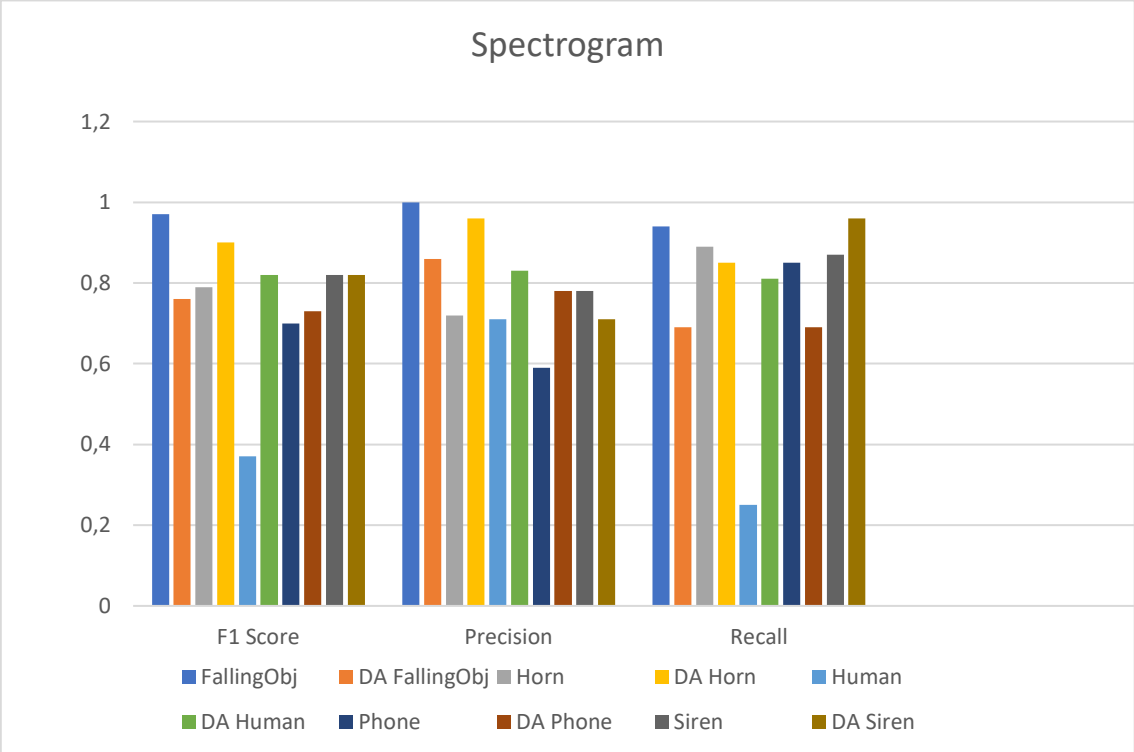


Fig 9 F1 score, precision, and recall from Spectrogram before and after processing by data augmentation

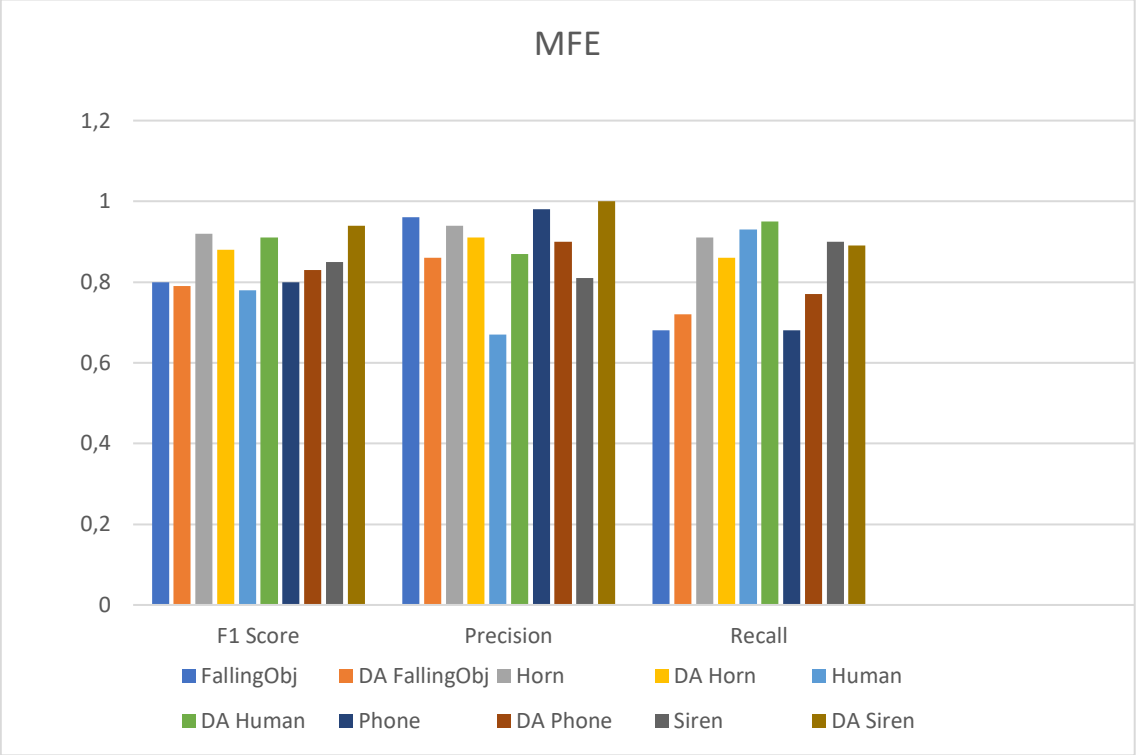


Fig 10 F1 score, precision, and recall from MFE before and after processing by data augmentation

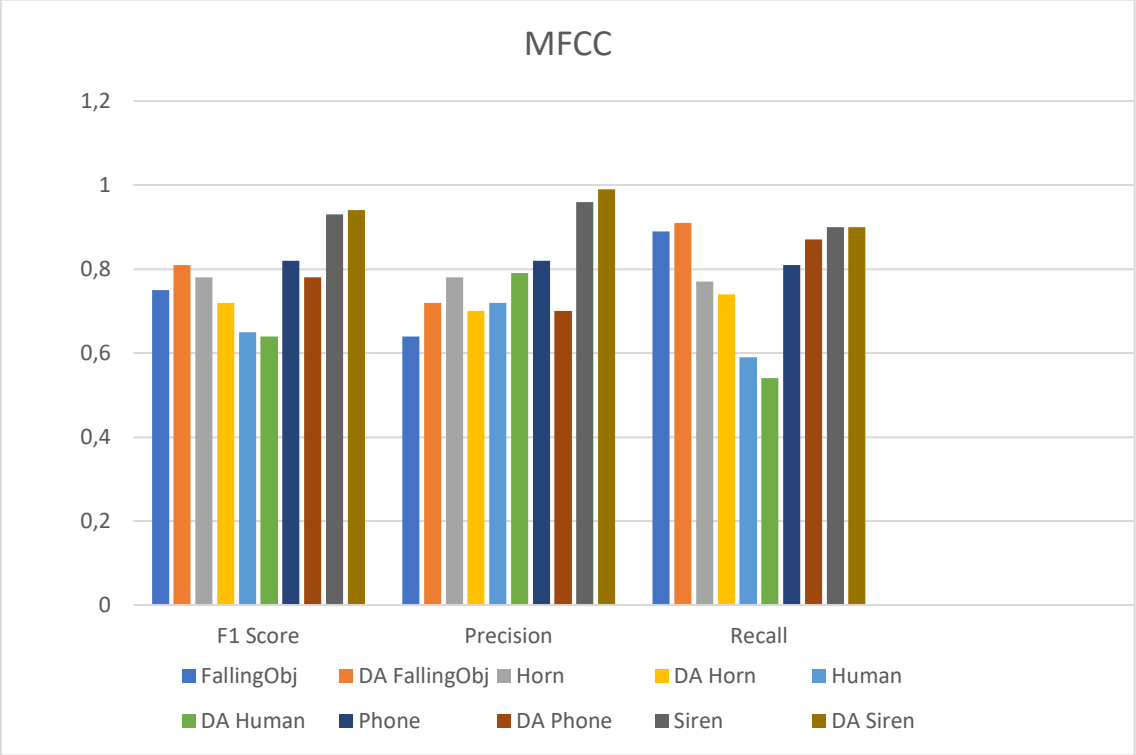


Fig 11 F1 score, precision, and recall from MFCC before and after processing by data augmentation

Model created by spectrogram before applied data augmentation have lowest performance because only have 74.0% accuracy. But, after applied data augmentation the accuracy increased significantly into 81.3%. Spectrogram become the most increased average accuracy after data augmentation applied because it has 7.3% difference.

Model created from Mel-Filterbank Energy (MFE) before data augmentation have good performance, it has 82.8% accuracy. After data augmentation, the accuracy increased into 86.4%. MFE still be the best method because it still has good accuracy before and after data augmentation.

Model created by Mel Frequency Cepstral Coefficients have good performance with 78.5% accuracy. But, after data augmentation applied, the average accuracy of MFCC not increased significantly like other method. After data augmentation, the accuracy increased 0.3% become 78.8% and have lowest accuracy compared to other method.

For fallingObj class, the highest accuracy is from spectrogram before data augmentation with 93.7% accuracy. For horn class, the best method is MFE before data augmentation with 90.8% accuracy. For Human class, the highest accuracy is MFE after data augmentation with 95.2%. The best method for phone class is MFE before data augmentation with 93.3%. Last, for the siren class, the best method is Spectrogram after data augmentation with 95.9% accuracy.

5. Conclusions

5.1 Conclusion

From this study, found the best method for extracting environmental noise feature in the creation of noise classification models using CNN algorithm it is Mel-Filterbank Energy. MFE has highest accuracy after data augmentation with 86.4% accuracy. Spectrogram is not good in case of noise feature extraction before data augmentation because only have 74.0% accuracy but, after data augmentation, the accuracy of Spectrogram increased become 81.3%. MFCC is the lowest method with 78.8% accuracy after data augmentation applied. From the test, data augmentation using SpecAugment proven to be a way to improve the accuracy of classifications performed

From the test FallingObj class become the class which have lowest accuracy. We presume this is caused by this class have very short noise time, and device only capture little sample, compared to other class which have longer sample to capture. With this result, we hope monitoring device in library can perform better to alerting visitors around it because the device can know what the source of noise is, not only alerting visitors around it when the decibel sensor reaches certain level. We also hope this device can be implemented in library and can reduce noise produced by visitors and library become more conducive to study, discussing, or learning.

5.2 Recommendations

This research still uses very simple implementation in noise classification with hand of Edge Impulse Studio. Further research should have detailed implementation in CNN with feature extraction for classifying

noise to produce better result. Dataset used in this research also not obtained from real noise around library, further improvement we recommended to use own dataset which get from real library environments. So, the result can be more accurate compared from this result.

Acknowledgment

This research was financially supported by The Faculty of Engineering, Diponegoro University, Indonesia through Strategic Research Grant 2021 number: 3178/S/komputer/4/UN7.5.3.2/PP/2021.

References

[1] J. Lange, A. Miller-Nesbitt and S. Severson, "Reducing noise in the academic library: the effectiveness of installing noise meters," *Library Hi Tech,* vol. 34, no. 1, pp. 45-63, 2016.

[2] N. David, A. C. V. Nina, E. IfeyinwaNwamaka and A. AyodejiOpeyemi, "Library Sound Level Meter," *Quest Journal of Electronics and Communication Engineering Research(JECER),* vol. 1, no. 1, pp. 20-29, 2013.

[3] S. Dargan, M. Kumar, M. R. Ayyagari and G. Kumar, "A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning," in *Archives of Computational Methods in Engineering*, Barcelona, Spain, 2019.

[4] C. Banbury, "Micronets: Neural Network Architectires for Deploying TinyML Applications on Commodity Microcontrollers," in *MLSys Conference*, San Jose, CA, USA, 2021.

[5] M. Z. H. Zim, "TinyML: Analysis of Xtensa LX6 microprocessor for Neural Network Applications by ESP32 SoC," in *Daffodil International University*, Dhaka, Bangladesh, 2021.

[6] S. Vishnupriya and K. Meenakshi, "Automatic Music Genre Classification using Convolution Neural Network," in *2018 International Conference on Computer Communication and Informatics*, Coimbatore, INDIA, 2018.

[7] G. Li, H. Tang and Y. Sun, "Hand gesture recognition based on convolution neural network," in *Springer Science+Business Media, LLC*, 2017.

[8] J. Nordby, "Environmental Sound Classification on Microcontrollers using Convolutional Neural Networks," *Faculty of Science and Technology Norwegian University of Life Science,* 2019.

[9] J. Alves, P. Guerreiro, G. Marques and G. Marques, "A Low-Cost Sound Event Detection and Identification System for Urban Environments," *ISEL Academic Journal of Electronics, Telecommunications and Computers,* vol. 6, no. 1, 2020.

[10] L. G. C. Vithakshana and W. G. D. M. Samankula, "IoT based animal classification system using convolutional neural network," in *International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2020.

[11] J. P. D. Galileu, "Urban Sound Event Classification for Audio-Based Surveillance Systems," Universidade Do Porto, Porto, 2020.

[12] G. Sharma, K. Umapathy and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics,* vol. 158, 2019.

[13] B. Ghoraani and S. Krishnan, "Time–Frequency Matrix Feature Extraction and Classification of Environmental Audio Signals," in *IEEE Transaction on Audio, Speech, and Language Processing*, 2021.

[14] J. Huang, B. Chen and B. Yao, "ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network," *National Natural Science Foundation of China,* vol. 7, pp. 92871-92880, 2019.

[15] Y. Zeng, H. Mao, D. Peng and Z. Yi, "Spectrogram based multi-task audio classification," *Multimed Tools Appl,* no. 78, p. 3705–3722, 2017.

[16] Y. Jung, Y. Kim, H. Lim and H. Kim, "Linear-Scale Filterbank for Deep Neural Network-Based Voice Activiry Detection," in *Conference of The Oriental Chapter of International Committee*, Seoul, Korea, 2017.

[17] C. Cooney, R. Folli and D. Coyle, "Mel Frequency Cepstral Coefficients Enhance Imagined Speech Decoding Accuracy from EEG," in *29th Irish Signals and Systems Conference (ISSC)*, Belfast, UK, 2018.

[18] K. Rao and M. K.E., "Speech Recognition Using Articulatory," *SpringerBriefs,* pp. 85-88, 2017.

[19] D. Hana, Q. Liu and W. Fan, "A New Image Classification Method Using CNN transfer learning and Web Data Augmentation," *Expert Systems With Applications,* 2017.

[20] D. S. Park, W. Chan, Y. Zhang and C.-C. Chiu, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Google Brain*, 2019.

[21] I. C. Education, "Convolutional Neural Networks," IBM, 20 October 2020. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Accessed 5 January 2022].

[22] A. Nandy and M. Biswas, "Reinforcement learning with keras, tensorflow, and chainerrl," in *Reinforcement Learning*, Berkeley, CA, Apress, 2018, pp. 129-153.

[23] D. Gaspon and e. al, "Deep Learning For Natural Sound Classification," in *Inter Noise 2019*, Madrid, Spain, 2019.

[24] V. H. Phung and E. J. R. 2, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," *Applied Sciences,* vol. 9, no. 21, p. 4500, 2019.

[25] F. Chollet, "Keras," [Online]. Available: https://keras.io/api/layers. [Accessed 5 January 2022].