Sound Class Classification Using The CNN Model Based on The NIGENS Dataset

1st Cristian Agusta
School of Computer Science
Bina Nusantara University
Bandung, Indonesia
cristian.agusta@binus.ac.id

2nd Chaelvyn Hindrawan
School of Computer Science
Bina Nusantara University
Bandung, Indonesia
chaelvyn.hindrawan@binus.ac.id

3rd Tasya Aulianissa
School of Computer Science
Bina Nusantara University
Bandung, Indonesia
tasya.aulianissa@binus.ac.id

4th Putri Ireine Rambi
School of Computer Science
Bina Nusantara University
Bandung, Indonesia
putri.rambi@binus.ac.id

5th Abdiel Ivan Rivandi
School of Computer Science
Bina Nusantara University
Bandung, Indonesia
abdiel.rivandi@binus.ac.id

Abstract**—**Research in this field responds to the increasing interest in computer-based sound analysis, especially for identifying events in general purpose audio. Due to the limited availability of comprehensive databases, this research introduces NIGENS, a database consisting of 714 WAV files, presenting high-quality sound events from 14 different categories. The research methodology starts with the extraction and assembly of the NIGENS dataset from official sources, followed by the development of a deep learning model based on Convolutional Neural Networks (CNN) in the Jupyter Notebook environment under Anaconda with the use of Python programming language. The design of the model involves stages such as data extraction, feature extraction (Mel-frequency Cepstral Coefficients, MFCC), dataset labeling, and data division into Training Data (90%) and Testing Data (10%).In the development of the model, variations in epoch and batch_size values are tested for assessing the model's sensitivity. The aim of the research is to classify sound event types with maximum accuracy. The conclusion of the research highlights the importance of configuring the model to achieve optimal performance in the task of sound event recognition, while also providing a deep understanding of the potential for the development of sound recognition technology in the future.

keyword : *NIGENS database, Mel-frequency Cepstral Coefficients (MFCC), Convolutional Neural Networks (CNN), Sound event recognition, Sound Class Types.* **(***Abstract***)** .

I. INTRODUCTION

Artificial intelligence serves as a fundamental basis for technology and is currently a pivotal factor in the advancement of scientific systems. Similar to humans, who possess five senses to perceive and interpret information, artificial intelligence is likewise advancing in the realm of sensor processing. This enables AI to analyze information in a more advanced manner, thereby aiding human productivity. In the age of expanding information technology, the application of artificial intelligence is highly pertinent, particularly in the domain of voice processing due to its significant potential for usage in diverse sectors.

This project intends to investigate the possible application of artificial intelligence in categorizing different speech classes, utilizing the NIGENS dataset. Subsequently, the outcomes of the voice class classification might be employed by implementing it in the realm of security. The research is motivated by the constraints of technology in effectively processing specific types of data, such as image data. Additionally, human capacity to detect potential hazards by auditory means in a single location is limited. Therefore, the utilization of artificial intelligence is crucial in order to optimize human productivity and access cutting-edge technology.

The application of artificial intelligence technology in sound processing has evolved throughout time. This phenomenon is evident through the utilization of technology in the domains of music production and speech recognition. Music creation involves considerable data processing, which entails the application of signal processing techniques to turn analogue signals into digital signals. In the field of sound processing, specifically in voice recognition, the focus is mostly on applying speech processing techniques that involve digital signal processing and the classification of sounds using artificial neural network (ANN) algorithms. The speaker identification and verification approach in speech recognition utilizes Mel Frequency Cepstrum Coefficients (MFCC). The application of technology in society is commonly seen in daily use, particularly in the form of virtual assistants.

This research uses digital signal processing techniques, namely Mel-frequency cepstral coefficients (MFCC), to categorize and differentiate sounds produced by various sources, including both natural and human sounds, for the purpose of categorization. The research methodology entails extracting data from the NIGENS dataset, which comprises categorized noises based on certain sound classifications. The voice classification method employed in this study utilizes the Convolutional Neural Network (CNN) algorithm, which involves processing voices across many layers and training the model with relevant data. This approach aims to reach the highest possible accuracy in the provided results.

This research aims to utilize the findings to advance sound processing technology, particularly in the security sector. By incorporating voice identification, the technology can effectively identify potential threats. This is especially valuable as sound sensors can overcome the limitations of image-based sensors.

II. METHOD

In research related to the process of classifying the types of voice classes that we do, data collection is done by downloading the dataset that has been provided on the NIGENS website. The NIGENS dataset contains a data set in the form of sounds that are classified into 15 types of voice classes. After downloading and extracting the dataset, the next step is to develop a deep learning model that will be trained using the previously extracted dataset. The modeling process is carried out in the Jupyter Notebook environment, which is one of Anaconda's ecosystems to assist developers in developing source code scripts using python. The deep learning modeling process will utilize the python language due to the flexible nature of the language and is more advanced in terms of artificial intelligence development than other languages. In addition, the reason for using python is also due to the large number of libraries that already exist in python allowing more functions that can be used creatively. The design step of the deep learning model starts by creating a function to extract MFCC features from each sound file that has been extracted previously. Since computers can only process things in binary numbers, the use of MFCC features is to convert files that were once voices recognized by humans into collections of numbers through the MFCC equation so that these numbers can be recognized and processed by computers. After creating a function to extract MFCC features, the next is to create a function to label each MFCC feature according to the type of voice class it has, usually labeling is denoted by an index number starting from 0. Next is to create a function to separate the dataset into data to be trained (Training Data) and data for accuracy testing (Testing Data) with a division of the amount of Training Data is 90% and Testing Data is 10%. After dividing the data for Training Data and Testing Data, the next step is to build a CNN model for deep learning. In making the CNN model, we will create several layers where the data is processed. In each layer, there are several attributes that will determine how accurate the data we have. Then in the training process, there are also attributes such as epoch and batch_size that can affect the accuracy that can be learned by the model related to the data used for Training Data. Epoch is a calculation of how much the training process will be repeated or iterated, while batch_size is a measure of how much data is trained from the total data used in one epoch round before finally iterating to the next epoch. Therefore, experiments will be conducted on 4 different possibilities, namely when the epoch value is 10 and batch_size is 32, epoch value is 10 and batch_size is 10, epoch value is 100 and batch_size is 32, and epoch value is 100 and batch_size is 10. Furthermore, in addition to configuring the training process such as epoch and batch_size, it is also necessary to configure the basic architecture of the CNN model created. Previously, we know that the CNN model consists of several layers that are interrelated in processing data to produce the expected output. In each layer, there are hyperparameters whose values can affect the model's ability to train such as filters and kernel_size in the convolution layer, pool_size in the pooling layer, and units in the dense layer.

III. RESEARCH AND DISCUSSION

TABLE 1. Data Training’s Epoch and Batch Size Configuration

Epoch	Batch Size	Accuracy
100	32	90%
100	10	91%

10	32	89%
10	10	89%

As stated on the research table above, the combination of epoch and batch_size values can influence the accuracy value of a model. The combination of training data with an epoch value of 100 and a batch size of 10 has the highest level of accuracy, namely 91%, then a combination with an epoch value of 100 and a batch_size of 32 has an accuracy of 90%. Next, for training data with an epoch value of 10 and a batch_size of 32, the accuracy value obtained was 89% and finally for training data with an epoch value of 10 and a batch_size of 10, the accuracy value obtained was also the same, namely 89%. Based on the facts above, it can be seen that the epoch quantity has a value that is quite influential in the process of forming the CNN model. The greater the epoch, the greater the update that occurs in the CNN model being built, causing greater data to be studied and smaller errors or losses that occur, so that the accuracy value increases. However, there is a maximum limit to the epoch value at which a model can learn completely, causing several epochs to be carried out causing updates to the model which results in reduced accuracy of the model, for example when training data is carried out with 150 epochs and 10 batch_size the accuracy value drops to 81%. After getting the most accurate division of epoch and batch_size, the next step is to configure the hyperparameters for certain layers in the CNN model.

The premise of the results above occurs by considering the following hyperparameters: filter value of 128 and kernel_size of 7 on the convolutional layer, pool_size value of 2 on the pooling layer and unit value of 64 on the dense layer. If other forms of configuration are carried out, different levels of accuracy will be produced for each configuration as shown in the following table:

TABLE 2. Layer Hyperparameter's Configuration on Model's Accuracy

Filter	Kernel Size	Pool Size	Units	Accuracy
128	7	2	128	92%
128	7	4	128	91.8%
128	3	2	128	89.2%
128	7	2	64	91%
128	3	2	64	87%
64	3	2	128	90%
64	7	2	128	92%
64	7	4	128	90.4%
64	3	2	64	90.4%
64	7	2	64	92.3%
64	7	4	64	90.16%
32	3	2	32	87%
32	7	2	32	91.6%

As stated on the research results in the table above, hyperparameter configuration at a particular layer affects the performance of the CNN model created, causing differences in the level of accuracy for each configuration. It can be seen that the more proportional the values for the filter and unit, the higher the accuracy value of the model. Each different dataset has a different level of data complexity to be learned by the model, therefore, the use of hyperparameters cannot be used as a definite reference that is the same for every case, settings are needed that are appropriate to the data type to be able to achieve a model with a high level of accuracy. As in this research, which divides sound into 14 class labels, of course a complex pattern is needed from each data so that each class can be differentiated by surviving one another. Also look at a fact, namely that the greater the value of pool_size, the smaller the accuracy for a model with the same configuration. Inversely proportional to the kernel_size, for the same configuration, the larger the kernel_size, the more accurate the model. Therefore, 64 filters are needed and a kernel size of 7, pool_size of 2, and 64 units to get the highest accuracy of 92.3%.

IV. CONCLUSION

In this research, the exploration of voice class type classification unveils the crucial role of hyperparameter configurations in the development of deep learning models. Leveraging the NIGENS dataset and utilizing the Python programming language along with the Jupyter Notebook environment, the study underscores the flexibility and advancements facilitated by Python in the realm of artificial intelligence. Notably, the adoption of Mel-Frequency Cepstral Coefficients (MFCC) features, transformed into binary numbers through the MFCC equation, emerges as a pivotal strategy for enhancing sound processing and recognition by computers.

The experimental findings shed light on the delicate balance between epoch values and batch sizes, revealing that a combination of 100 epochs and a batch size of 10 yields the highest accuracy rate at 91%. This highlights the model's increased efficacy with more iterations and a larger dataset. However, caution is warranted, as excessively high epoch values could lead to a decline in accuracy, exposing limitations in the model's learning process. The study further delves into the impact of hyperparameter configurations, particularly in layers such as filters, kernel size, and pool size, underscoring their significant influence on model accuracy. The overarching conclusion emphasizes the success of deep learning models in voice class type classification and provides valuable insights into the intricate trade-off between computation time and accuracy when selecting optimal configurations.

REFERENCES

[1] Chilukuri, N., 2020. AudioCNN: Audio Event Classification with Deep Learning Based Multi-Channel Fusion Networks. University of Missouri-Kansas City.

[2] Demir, F., Turkoglu, M., Aslan, M., & Şengur, A., 2020. A new pyramidal concatenated CNN approach for environmental sound classification. *Applied Acoustics*, 170, pp. 107520. https://doi.org/10.1016/j.apacoust.2020.107520.

[3] Mariostrbac. 2021. Environmental Sound Classification Using Deep Learning. GitHub. https://github.com/mariostrbac/environmental-sound-classification

[4] Massoudi, M., Verma, S. and Jain, R., 2021, January. Urban sound classification using CNN. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 583-589). IEEE.

[5] Moorer, J.A., 1977. Signal processing aspects of computer music: A survey. *Proceedings of the IEEE*, 65(8), pp.1108-1137. https://doi.org/10.1109/PROC.1977.10660.

[6] Nandyal, S., Wali, S.S. and Hatture, S.M., 2015. MFCC based text-dependent speaker identification using BPNN. *International Journal of Signal Processing Systems*, 3(1), pp.30-34.

[7] Nanni, L., Costa, Y.M., Aguiar, R.L., Mangolin, R.B., Brahnam, S. and Silla, C.N., 2020. Ensemble of convolutional neural networks to improve animal audio classification. *EURASIP Journal on Audio, Speech, and Music Processing, 2020(1)*, pp.1-14.

[8] Nogueira, A.F.R., Oliveira, H.S., Machado, J.J. and Tavares, J.M.R., 2022. Sound Classification and Processing of Urban Environments: A Systematic Literature Review. Sensors, 22(22), p.8608.

[9] Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, 24(3), 279-283. https://doi.org/10.1109/LSP.2017.2657381

[10] Serizel, R., Bisot, V., Essid, S. and Richard, G., 2018. Acoustic features for environmental sound analysis. *Computational analysis of sound scenes and events*, pp.71-101.https://doi.org/10.1007/978-3-319-63450-0_4.

[11] Zwan, P. and Czyzewski, A., 2010. Verification of the parameterization methods in the context of automatic recognition of sounds related to danger. *Journal of Digital Forensic Practice*, 3(1), pp.33-45.