**Playing Maze using Voice Commands**

K. Vishnu Sainadh, K. Satwik, V. Ashrith, Dr. Suja P, Dr. Peeta Basa Pati

Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru,

Amrita Vishwa Vidyapeetham, India.

[bl.en.u4aie19028@bl.students.amrita.edu](mailto:bl.en.u4aie19009@bl.students.amrita.edu), [bl.en.u4aie190034@bl.students.amrita.edu](mailto:bl.en.u4aie190016@bl.students.amrita.edu),[bl.en.u4aie19066@bl.students.amrita.edu](mailto:bl.en.u4aie19066@bl.students.amrita.edu), [p\_suja@blr.amrita.edu](mailto:p_suja@blr.amrita.edu), bp\_peeta@blr.amrita.edu,

*Abstract—* In recent years, Speech recognition is becoming increasingly well-known as a result of the numerous applications it has in virtually every industry. These applications range from wake-word recognition and emotion recognition to command recognition and interactive game play. There has been a recent uptick in interest regarding the utilization of voice in the gaming industry. The introduction of interaction based on voice control made video games available to a significantly larger demographic. However, in order to prevent an unwelcome delay, using one's voice as a controller for a game calls for real-time processing. This research paper puts forward the idea of controlling the well-known maze game with speech commands recognized by Convolutional Neural Networks (CNN). A dataset consisting of the speech commands "Left," "Right," "Up," "Down," and "Go," as well as "Stop," was prepared for training, validation, and testing. Second, in order to recognize all six speech commands, an optimal Spectrogram and CNN-based speech command recognition were suggested as possible solutions. According to the findings, the proposed algorithm was able to recognize all six commands and achieve a high recognition accuracy of 94.57 percent. In the final step, the proposed algorithm is incorporated into a maze game that is based on MATLAB.

Keywords— Speech Recognition, CNN, Maze, Spectrogram.

# INTRODUCTION

Speech Recognition is a software which enables a program to process human speech into a written format.​ The concept of speech recognition has been studied extensively and has been around for a considerable amount of time. However, its applicability in the game industry is still in its infancy, and very little research has been conducted on the topic. Speech recognition has been accomplished by the use of a wide variety of techniques, some of which include Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), and Deep Neural Networks (DNN). At least twenty years have passed since HMM was first put into practice. One of the crucial HMM parameters that are used is the distribution of the likelihood of seeing the state. For the purpose of modelling these probabilities and determining how well each frame's window matches each HMM state, GMM is utilized. DNN has lately emerged as the dominant player in the automated speech recognition industry. In order to train its models, DNN employs a number of hidden layers. It has been shown that these models perform much better than those that were trained using GMM and HMM approaches.

The experience of a player playing a game on a computer, in a virtual world, or in a mixed reality setting is enhanced by the addition of a new dimension brought about by voice interactive games. The realm of computer games is not the only possible use for them. If a mistake is discovered in the 3D design of an engine, motor, or other mechanical device while it is being inspected in virtual reality (VR), an inspector or design engineer can correct the error before the design is downloaded to a 3D printer for the purpose of producing a rapid prototype using the 3D printing technology. This technology can also be used in a variety of virtual reality (VR) applications, such as those used in the medical and manufacturing industries.

The most recent developments in neural networks make it possible for a programmer to utilize a recording of a person's voice to not only understand the words that are being uttered in real time but also to identify the person speaking and to address them by name. It is extremely conceivable, given the present level of technology, to design algorithms and computer programs that will allow the player to talk with any game character, have a conversation with them in a variety of languages, and direct them to carry out a certain sequence of activities. Several methods have been developed in order to reduce the total number of parameters utilized by neural networks. These methods include neural network connection pruning, which gets rid of "unimportant" connections between layers, sparse neural network pruning, which gets rid of "unimportant" neurons within the network, and feature channel pruning, which gets rid of "unimportant" feature channels within the network. These strategies have resulted in a compression of the parameter size of the neural networks without negatively impacting the accuracy of the recognition.

In our paper, we have our own CNN model an trained it and achieved a training accuracy of 95.34%.and then we have used this trained model to classify the voice commands and used this output as input to the maze which is built on MATLAB.

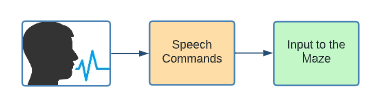


Fig 1: Aim of the paper

# LITERATURE SURVEY

In all of the work that we have evaluated, speech command recognition is performed to be used in real-life scenarios. In this paper, we try to extend the use case of speech command recognition to be applied to games to allow us to have easier control in the game. Furthermore, it would allow disabled people, children, and the elderly, to play such games. CNNs have the potential to outperform DNNs when it comes to keyword spotting tasks. This is due to the fact that CNNs are able to model correlations locally, and it is known that audio signals have high correlations between time and frequency. On the other hand, DNNs do not pay a great deal of attention to the topology of the input and instead simply resize the input into the correlating column vector.

Next, convolutional neural networks (CNNs) make use of fewer parameters than deep neural networks (DNNs), so switching to a CNN would improve performance while simultaneously reducing size.

One of the most significant discoveries in the field of artificial intelligence is the neural network, which has applications across the board in the scientific and technological communities. In recent years, with the help of ever-more-powerful computer resources, neural networks have grown both in size and in their ability to do more complex tasks. They continue to achieve ever-increasing levels of recognition accuracy, often outperforming humans in the process. Deep neural networks, on the other hand, have a great number of layers and millions of parameters. As an example, AlexNet consists of 8 layers and 62 million parameters, whereas VGG Net consists of 16 layers and 138 million parameters.

# CONCEPTS

**Spectrogram:**

A spectrogram is a visual technique of displaying the signal intensity, or “loudness”, of a signal across time at different frequencies contained in a specific waveform. In order to construct a spectrogram from a series of spectra, one must first stack them one on top of the other in time and then compress the amplitude axis.

**CNN:**

A CNN is a class of artificial neural network, most applied to analyze visual imagery.​ The acronym "CNN" refers to a "Convolutional Neural Network," which is an algorithm that belongs to the Deep Learning category. CNN is able to take an image as its input, prioritize distinct aspects of the picture, and distinguish those aspects from one another. In particular, as compared to those other classification approaches, it needs far less preparation than the other classification algorithms. The Convolutional Network is able to learn the filters or characteristics in the pictures, in contrast to the easy methods to filters, which are done manually. These simple techniques include doing things by hand. They have an input layer, an output layer and many hidden layers and millions of parameters, which allows them to learn patterns. They have five main types of layers, which are: Convolutional layer, Pooling layer, Fully-connected (FC) layer, Activation layer, Batch Normalization Layer.

* Convolution Layer: The primary component that goes into the construction of a CNN is known as a convolutional layer. It has a collection of filters, also known as kernels, the parameters of which are going to be learnt as part of the training process. In most cases, the size of the filter will be less than the actual size of the picture. Each filter performs a convolving operation on the picture, which results in the creation of an activation map.
* Batch Normalization Layer: A layer that standardizes the values of its inputs. The batch normalization process involves the application of a transformation that keeps the output mean near to the value 0 and the output standard deviation close to the value 1.
* Activation Layer: Within a convolutional neural network, a non-linearity layer is represented by an activation function. This activation function takes the feature map that was produced by the convolutional layer and uses it to build the activation map as its output.
* Pooling Layer: a layer of pooling that pools the features of the input feature map globally over time. This ensures that the input have time-translation invariance, which enables the network to perform the same classification independent of the precise location. Additionally, global pooling results in a substantial decrease in the total number of parameters included in the fully connected layer
* Fully Connected Layer: The last few levels of the network are composed of fully connected layers. The output from the final Pooling or Convolutional Layer serves as the input to the fully connected layer. This output is flattened before being fed into the fully connected layer.

# MODEL ARCHITECTURE

## Dataset:

A collection of spoken words in the form of an audio dataset that may be used to train and evaluate keyword detecting algorithms. Its primary objective is to devise a method for constructing and validating small models that can identify when a single word from a set of eight target words is spoken, with the objective of producing as few false positives as possible as a result of background noise or speech that is unrelated to the target word. The dataset contains 31 folders, the folder names represents the labels which is nothing but the speech command names in which we used only six of them which are down, go, left, right, stop, and up.

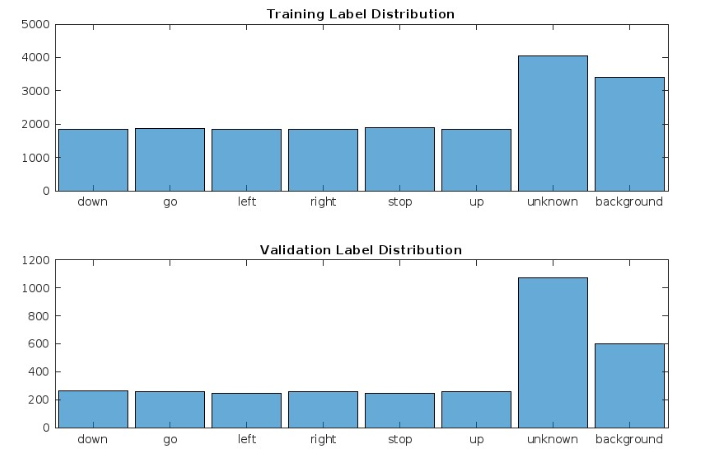


Fig 2: Distribution of training data and validation data.

## Data Preprocessing

Initially, we extracted data from the dataset. Now, pre-processing need to be done. We need to make sure that all the data is of same length. So, we made all the speech files length to one second by padding the audio file with zeros if the audio file length is less than one second. To recognize the command spoken by the user we need to get some features from the audio file. Here we extracted spectrograms for each audio file and used the generated spectrogram as the input to our neural network model.

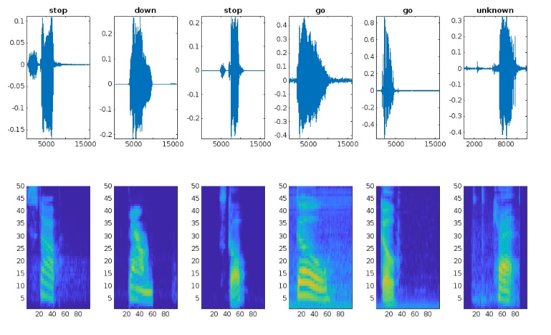


Fig 3: Wave forms and their corresponding Spectrograms of random audios taken from dataset.

## Maze:

The maze takes the number of rows and number of columns as input which will be provided by the user. The maze that is generated will be random and after starting the maze by giving the dimensions the title which is above the maze prompts to press any key to start the game. We need to start from top left in the maze and try to reach the bottom right to win the game using commands. A new figure window with maze will pop now press any key to give audio input. Start giving input when title says 'Start Speaking' the audio input is only of '1' second duration so give the desired command in '1' second and the Trained model which we saved will classify the audio recorded into a class label and performs that action if its legal on the maze. Give 'STOP' command as input audio to abort the game at any point of time. The 'Blue' diamonds tells the traversed path by us in past actions and the 'Black' diamond is our current position in the maze.

## Model:



Fig 4: The Proposed CNN Architecture

We, initially took a block of CNN which contains a convolution layer, a batch normalization layer, a activation layer with ReLU as the activation function and a max-pooling layer. We trained this network and observed the results they were not to the mark. So, we have used the above block of neural network and repeated them 2 times that is until we get a better accuracy, we still wanted to explore by adding another two similar blocks. At last, we have used five identical blocks and achieved a training accuracy of 95.34% and validation accuracy of 94.57%.

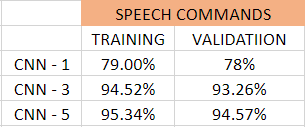


Fig 5: Accuracies of models discussed above.

In the above figure, we can see the accuracies of the models that are tried before getting to the final model. Initially, when one block of network is trained the accuracy was 79% and when 3 identical blocks are used we got 94.52% and at last we were able to get 95.34% which performed well in classifying the speech commands.

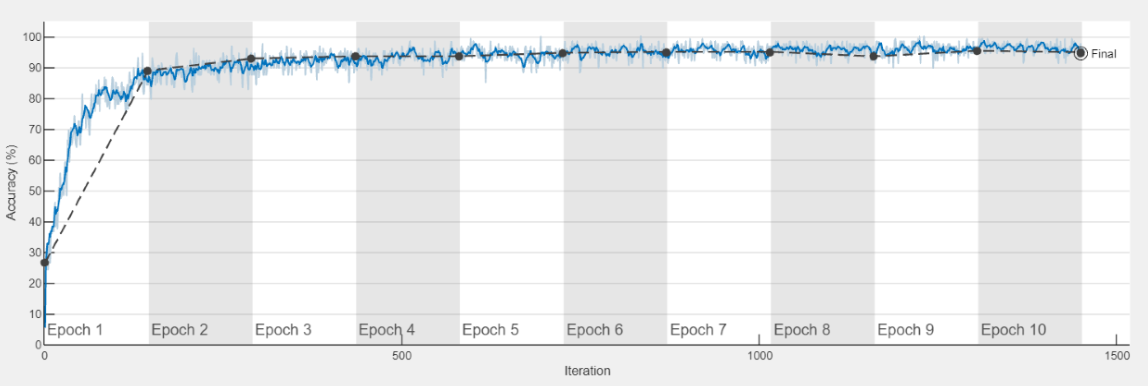


Fig 6: Training Progress

In Fig 6, the training progress has been plotted the blue line indicates the training accuracy over the specified number of epochs and the black dotted line indicates the validation accuracy. The validation accuracy is calculated after each epoch. The size of mini-batch taken is 128 and the learning rate is set to 0.003. The number of epochs used for our training is 10.

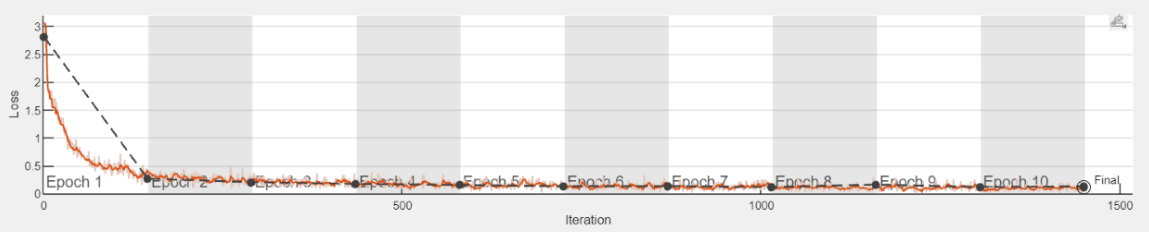


Fig 7: Loss

In Fig 7, initially the loss is high, during training the loss decreased over the specified number of epochs this indicates that the model is trying to perform better.

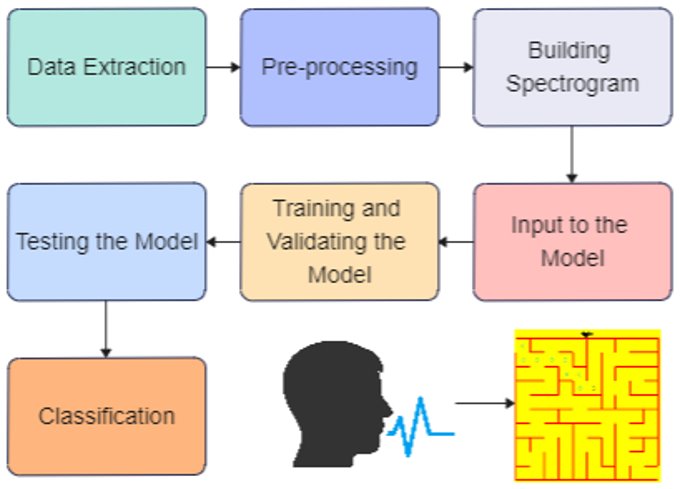


Fig 8: Methodology

# RESULTS

Performance metrics used to evaluate our model are confusion matrix, recall, precision, and F1 score. A table that provides a summary of the number of correct and incorrect predictions made by a classifier is called a confusion matrix. It is utilized in the process of evaluating the efficacy of a classification model.

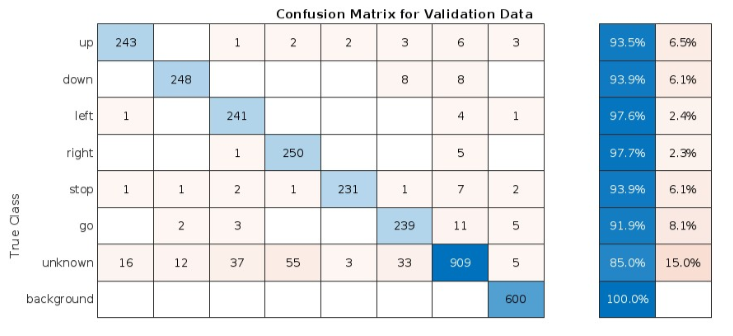


Fig 9: Confusion matrix for validation data

In Fig 9, all the diagonal values are the correctly classified and all the other values are not correctly classified. The model is performing well, it is classifying the test data to their corresponding labels correctly.

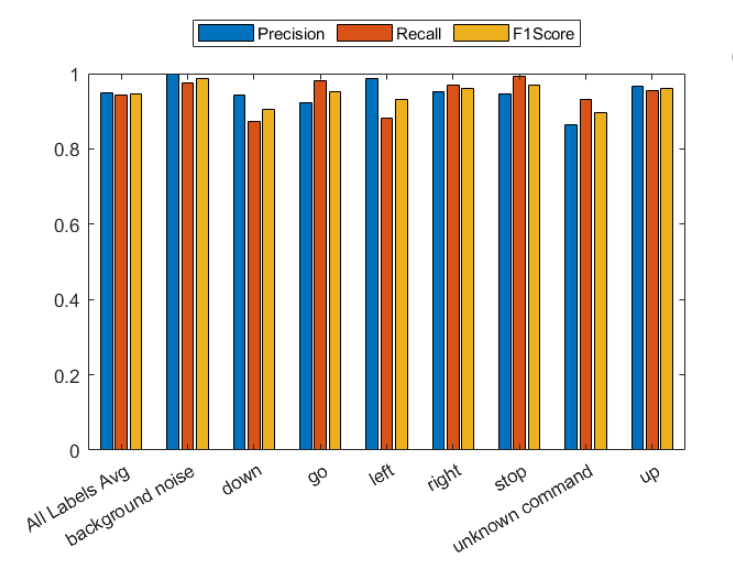


Fig 10: Recall, Precision an F1 Score for the commands

Precision is the probability of actually being positive given which we predicted as positive. Recall is the probability of predicting positive given which are actually positive.

In situations in which overlooked cases (also known as false negatives) are more expensive than false alarms, recall is more critical (False Positive). Finding the examples that fit the positive criteria is the primary emphasis of these challenges.

Where accuracy is more crucial and false alarms (also known as false positives) are more expensive than missed cases, precision is more critical (False Negatives). Eliminating the problematic instances is the primary emphasis in dealing with these issues.

From Fig 10, we can infer that we have achieved good average precision, recall and F1 Score.

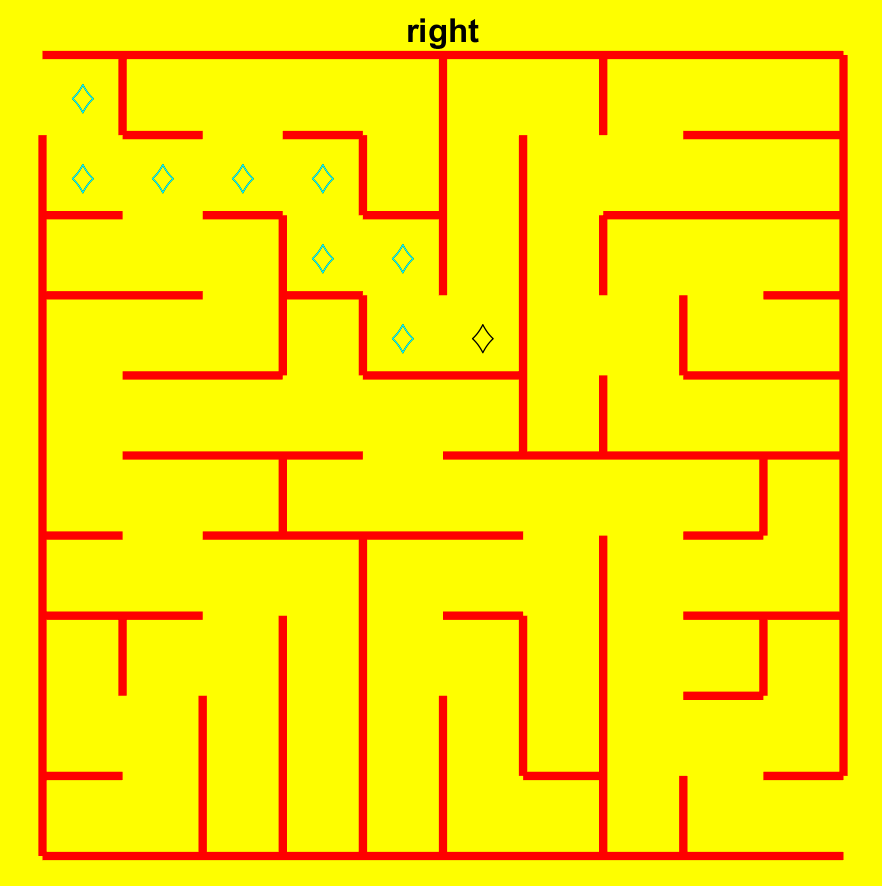


Fig 11: Snapshot of our maze game after moving some steps through commands

In Fig 11, the blue diamond’s represents the traversed path in past moves. The black diamond tells the current position where we are in the maze.

# CONCLUSION

Within the scope of this study, we classified the speech commands using CNN. We have achieved a validation accuracy of 94.57. As a result of our research, we have built the maze on MATLAB which uses commands classified by our model and completes the maze if the right set of commands are given as audio. The dimensions of the game can be changed to increase the difficulty to the player and the maze generated is random.

In the future, we could use these speech commands to play more games.​ We can also use this speech commands recognition in smart homes where appliances are controlled using voice commands.

# References

[1] D. M. Waqar, T. S. Gunawan, M. Kartiwi and R. Ahmad, "Real-Time Voice-Controlled Game Interaction using Convolutional Neural Networks," 2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 2021, pp. 76-81, doi: 10.1109/ICSIMA50015.2021.9526318.

[2] J. P. Piotrowski, E. A. Yfantis, A. Campagna, Q. Cornu and G. M. Gallitano, "Voice Interactive Games," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 0780-0784, doi: 10.1109/UEMCON47517.2019.8993015.

[3] N. Nasiri, S. Shirmohammadi and A. Rashed, "A serious game for children with speech disorders and hearing problems," 2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH), 2017, pp. 1-7, doi: 10.1109/SeGAH.2017.7939296.

[4] T. Athanaselis, S. Bakamidis, G. Giannopoulos, I. Dologlou and E. Fotinea, "Robust speech recognition in the presence of noise using medical data," 2008 IEEE International Workshop on Imaging Systems and Techniques, 2008, pp. 349-352, doi: 10.1109/IST.2008.4659999.

[5] N. M R and S. Mohan B S, "Music Genre Classification using Spectrograms," 2020 International Conference on Power, Instrumentation, Control and Computing (PICC), 2020, pp. 1-5, doi: 10.1109/PICC51425.2020.9362364.

[6] S. Hershey et al., "CNN architectures for large-scale audio classification," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 131-135, doi: 10.1109/ICASSP.2017.7952132.

[7] M. Massoudi, S. Verma and R. Jain, "Urban Sound Classification using CNN," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 583-589, doi: 10.1109/ICICT50816.2021.9358621.

[8] A. S. B. Wazir et al., "Spectrogram-Based Classification Of Spoken Foul Language Using Deep CNN," 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020, pp. 1-6, doi: 10.1109/MMSP48831.2020.9287133.

[9] A. Kaiyr, S. Kadyrov and A. Bogdanchikov, "Automatic Language Identification from Spectorgam Images," 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), 2021, pp. 1-4, doi: 10.1109/SIST50301.2021.9465996.

[10] S. Amiriparian, N. Cummins, M. Gerczuk, S. Pugachevskiy, S. Ottl and B. Schuller, "“Are You Playing a Shooter Again?!” Deep Representation Learning for Audio-Based Video Game Genre Recognition," in IEEE Transactions on Games, vol. 12, no. 2, pp. 145-154, June 2020, doi: 10.1109/TG.2019.2894532.

[11] V. Bansal, G. Pahwa and N. Kannan, "Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), 2020, pp. 604-608, doi: 10.1109/GUCON48875.2020.9231094.

[12] A. Saidi, S. B. Othman and S. B. Saoud, "Hybrid CNN-SVM classifier for efficient depression detection system," 2020 4th International Conference on Advanced Systems and Emergent Technologies (IC\_ASET), 2020, pp. 229-234, doi: 10.1109/IC\_ASET49463.2020.9318302.

[13] T. Toshniwal, P. Tandon and N. P, "Music Genre Recognition Using Short Time Fourier Tranform And CNN," 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1-4, doi: 10.1109/ICCCI54379.2022.9740939.

[14] R. V. Sharan, S. Berkovsky and S. Liu, "Voice Command Recognition Using Biologically Inspired Time-Frequency Representation and Convolutional Neural Networks," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 998-1001, doi: 10.1109/EMBC44109.2020.9176006.

[15] S. Amiriparian, N. Cummins, S. Ottl, M. Gerczuk and B. Schuller, "Sentiment analysis using image-based deep spectrum features," 2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), 2017, pp. 26-29, doi: 10.1109/ACIIW.2017.8272618.

[16] V. -S. Doan, T. Huynh-The and D. -S. Kim, "Underwater Acoustic Target Classification Based on Dense Convolutional Neural Network," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 1500905, doi: 10.1109/LGRS.2020.3029584.