Article **ECORFAN Journal Democratic Republic of Congo**
June 2023 Vol.9 No.16 1-8

Gunshot detection neural network implemented on a low-cost microcontroller

Red Neuronal para detección de disparos implementada en un microcontrolador de bajo costo

RODRÍGUEZ-PONCE, Rafael†*.

Universidad Politécnica de Guanajuato, Ingeniería en Robótica

ID 1st Author: *Rafael, Rodríguez-Ponce* / **ORC ID**: 0000-0001-5006-5580, **CVU CONAHCYT ID**: 209261

DOI: 10.35429/EJDRC.2023.16.9.1.8 Received March 26, 2023; Accepted June 30, 2023

Abstract

Nowadays, criminal activity is on the rise, and it usually involves some type of firearm. There are automated shot detection systems but in the end, they still require human intervention to decide if it is an actual gunshot. Distinguishing between two similar sounds, such as the detonation of a firearm or a firecracker, is not always possible with the naked ear. There are multiple publications on artificial intelligence to identify gunshots; however, they use convolutional neural networks, which, despite being highly effective, require a system with extensive computational resources. This document presents a fully connected neural network implemented on a microcontroller that can identify up to 90% of firearm detonations. This document will be of interest to students or researchers interested in the design of neural networks for sound recognition on embedded systems.

Artificial Intelligence, Mel frequency cepstral coefficients (MFCC), Micropython, Pytorch, Embedded systems

Resumen

Hoy en día, la actividad criminal se encuentra al alza y usualmente involucra algún tipo de arma de fuego. Existen sistemas de detección de disparos automatizados pero al final, aún requieren de la intervención humana para decidir si en verdad es la detonación de un arma de fuego. Distinguir entre dos sonidos similares, como la detonación de un arma de fuego o un petardo, no es siempre posible a simple oído. Existen múltiples publicaciones en inteligencia artificial para identificar disparos, sin embargo utilizan redes neuronales convolucionales, las cuales, a pesar de tener una alta efectividad, requieren de un sistema con amplios recursos computacionales. Este documento presenta una red neuronal multicapa implementada en un microcontrolador que puede identificar hasta un 90% de detonaciones de arma de fuego. Este documento será de interés a estudiantes o investigadores interesados en el diseño de redes neuronales para reconocimiento de sonidos en sistemas embebidos.

Inteligencia Artificial, Coeficientes cepstrales de frecuencias de Mel (MFCC), Micropython, Pytorch, Sistemas embebidos

Citation: RODRÍGUEZ-PONCE, Rafael. Gunshot detection neural network implemented on a low-cost microcontroller. ECORFAN Journal-Democratic Republic of Congo. 2023, 9-16: 1-8

* Author's Correspondence (e-mail: rrodriguez@upgto.edu.mx)
† Researcher contributing first author.

© ECORFAN Journal - Democratic Republic of Congo

Introduction

Currently in Mexico, there is a climate of fear and uncertainty, due to the constant acts of violence that occur, day by day, in different parts of the country.

It is common to hear loud noises and immediately attribute them to gunshots, when in fact it can be something as mundane as fireworks, the backfire of a car exhaust, or simply a tire explosion.

To reduce and prevent gun violence in public areas, a growing number of cities in Europe and North America are adopting gunshot detection technology. However, these systems have not been without criticism due to their high maintenance cost, false-positive activation rates and their low efficacy in reducing violence, in addition to the fact that they are not available to the public in general (Aguilar, 2018; Lawrence *et al.*, 2018).

There are multiple publications on gunshot detection projects, nevertheless most are commonly based on the use of convolutional neural networks (Katsis *et al.*, 2022; Morehead *et al.*, 2019) or recurrent neural networks (Olmos *et al.*, 2018). Despite both being highly effective, they require computer systems with extensive processing, memory and graphics card resources, since they analyze the spectrogram image of the detected audio signal (Arif *et al.*, 2021; Bajzik *et al.*, 2020).

In this paper, the author opted for implementing a Fully Connected Neural Network (FCNN) on a low-cost microcontroller with limited computational resources, to have a portable, economical, and low-power gunshot detection system.

To extract the most relevant information from the gunshot audio signal, Mel Frequency Cepstral Coefficients (MFCC) were used since they are the most appropriate digital processing technique for this type of application (Li *et al.*, 2022; Sharma *et al.*, 2019). These coefficients are obtained through a filter bank whose operation is similar to that of the human ear (Jo, Yoo and Park, 2015).

For the recognition of the detonation audio signal, a FCNN was implemented with 10 inputs, two layers, and eight neurons in the hidden layer. Firstly, the audio signal is digitally processed with MFCC and 10 coefficients are obtained. These are fed into the neural network, and at the output there is a single bit indicating on an LED if the sound detected is coming from a firearm or not. The network was trained with more than 3,000 shot sounds from internet databases and detonations recorded at a local shooting range. Similarly, locally generated sounds such as blows, balloon explosions, firecrackers, and car exhaust detonations were used for negative training of the neural network.

This work can be very interesting and useful for students, teachers or researchers interested in the implementation of neural networks to recognize unique sounds on mobile devices.

Artificial Neural Network

An artificial neuron is a computational algorithm inspired by the functioning of biological neurons in the human brain. The neuron, also known as a perceptron, has several data inputs, a processing stage, and a single output. The main benefit of the neuron is that, just like human neurons, it can be trained for various applications, primarily pattern recognition (Aggarwal, 2018).

Each of the neuron inputs has a different weight, which is adjusted during the training process based on the error detected at the output. If a large amount of data is used during training, it is possible to reduce the error in such a way that the neuron can identify a pattern quickly and efficiently.

In order for the algorithm to carry out the learning of more complex processes, multiple neurons are interconnected in multiple layers, thus obtaining a FCNN. In the network, neurons are normally ordered by layers, in such a manner that the outputs of each neuron in one layer are fed to all the neurons in the next layer. Firstly, there is the input layer, where the data to be processed is entered. Next, there can be several hidden layers, and lastly, there is the output layer. The general structure of a FCNN neural network is shown in Figure 1.

ISSN 2414-4924
ECORFAN® All rights reserved

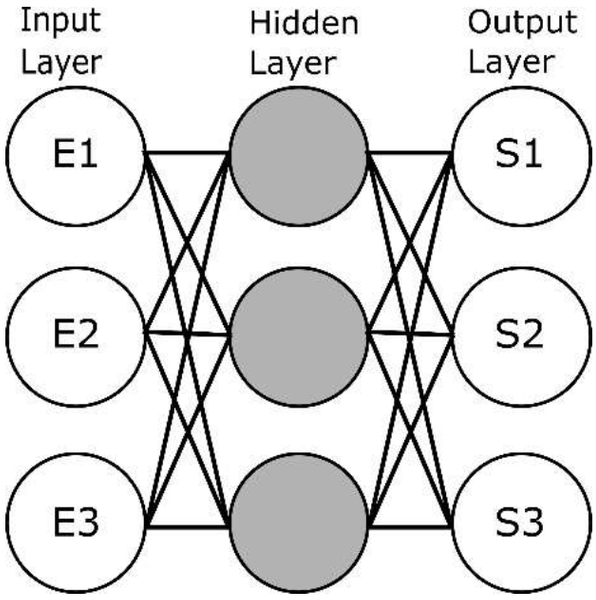

Figure 1 Structure of a fully connected neural network
Source: Self-Made

Currently, there is a great diversity of neural networks for different applications such as event prediction, pattern recognition, image generation, and industrial part design, among many others (Zu and Sutton, 2003).

Gunshot Waveform Analysis

A conventional firearm uses a gunpowder combustion to propel the bullet out of the barrel. Explosive gases expand rapidly, forcing a supersonic jet stream from the muzzle of the weapon. The flash causes an acoustic shock wave and a brief explosive sound that lasts only a few milliseconds. The maximum sound pressure level associated with the flash can exceed 150 dB in the vicinity of the firearm (Maher and Shaw, 2008). If a person is close enough to the gunshot, the sound is unique and unmistakable. Notwithstanding, at a certain distance, surrounded by buildings and mixed with ambient noise, it is not so evident.

In Figure 2 a comparison is made between the sound of a gunshot (a) and the detonation of a car backfire (b). To the naked eye, the waveform of the former does not differ much from the detonation waveform of the latter. Nevertheless, when obtaining the frequency content of both sounds, it is possible to visualize that the spectral content of the gunshot (c) is much broader than that of the exhaust detonation (d).

That is, when for the detonation of the exhaust the frequencies with the highest amplitude ranging from 1000 to 9000 Hz, for the gunshot the frequencies have high amplitude ranging from 3 to 20,000 Hz. This difference becomes even more noticeable once the MFCC coefficients are extracted; it is this difference between signals that is trained into the neural network to be able to recognize the sound of a gunshot.

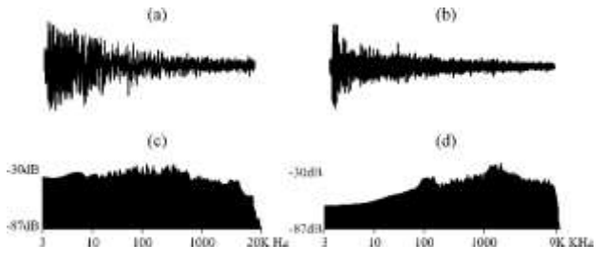

Figure 2 Waveforms for (a) a gunshot and (b) a car backfire detonation. Frequency content of (c) the gunshot and (d) the car backfire detonation
Source: Self-Made.

Audio Signal Information Extraction

A detonation audio signal is composed of numerous frequencies, and therefore cannot be fully processed due to time limitations or computational resources. For this reason, to identify it quickly and effectively, it is necessary to first extract the most relevant information so that it can be properly analyzed (Shi *et al*., 2018).

There are various techniques for extracting information from digital signals, such as the discrete Fourier transform (DFT), the discrete Wavelet transform, linear prediction coefficients, among many others; nonetheless, a technique widely used in detonation detection projects, is by means of the mel frequency cepstral coefficients, or MFCC. This is due to their simplicity and effectiveness, as well as the ease of applying them computationally (Hossan *et al.*, 2010).

To obtain the MFCC, firstly, the DFT transform is applied to extract the spectral content. Next, this information passes through a mel-scale filter bank, which extracts the most relevant data in certain frequency ranges. Finally, the logarithm of these energies is calculated, and the discrete cosine transform is applied to revert them into the time domain.

ISSN 2414-4924
ECORFAN® All rights reserved

The description of the MFCC process is shown in Figure 3.

With the MFCC technique, it is possible to change the number of coefficients necessary to be obtained from the audio signal, which can usually range between 10 and 25, depending on the application. Selecting more cepstral coefficients only results in a higher complexity in the model. In the case of this project, 10 coefficients were enough to effectively identify the sound of a gunshot.

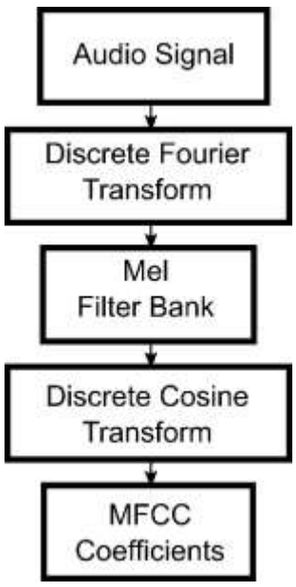

Figure 3 Block diagram for the MFCC information extraction technique
Source: Self-Made

Neural Network Design

Before implementing the gunshot identification system on the microcontroller, it was found beneficial to first design and train the neural network on a personal computer using the Python programming language and PyTorch libraries. This network would only serve as a reference for the design of the network for the microcontroller.

PyTorch is an open-source library based on Python, focused on performing numerical calculations through tensor programming, which facilitates its application in neural network projects. In addition, the simplicity of its interface and its ability to run on graphics processing units (GPUs) make it the most suitable and fastest option for creating artificial neural networks of different sizes and configurations (Chirodea *et al.*, 2021).

Due to the ease with which a neural network can be designed and trained on Pytorch, it was possible to experiment with different sizes and numbers of layers to obtain the smallest and most effective network for the identification of the available audio signals. Thus, it was proven that a neural network of 10 inputs and 2 layers is sufficient for the identification of gunshots with a success rate of up to 100%.

For the neural network training process, a database of more than 3000 gunshot sounds from different types of weapons and calibers was used. The audios were obtained free of charge from the Kaggle website (www.kaggle.com) and from the Gunshot Audio Forensics Dataset website (www.cadreforensics.com/audio/). Both sites contain an extensive database of audio shots of different calibers, including pistols, revolvers, short and long-range rifles (AK-12, AK-47, M16), machine guns, Lugers, and carbines. Similarly, a compendium of 500 shots recorded with a USB microphone at a local shooting range was created. For similar non-firearm audio, sounds such as banging, balloon popping, firecrackers, car backfires, and loud clapping sounds were used.

For the extraction of the most important audio features using MFCC, the Librosa libraries was used, which are a free Python toolkit for audio and music analysis and processing. Some common functions such as time frequency processing, feature extraction and sound graphing are available (McFee *et al.*, 2015). All the sounds to be used were processed with this toolkit, and the corresponding MFCCs were stored in a comma-separated values (CSV) file for use in Pytorch, and also in a text file for use on the microcontroller.

Figure 4 shows the neural network implementation process in Pytorch and is briefly described below.

ISSN 2414-4924
ECORFAN® All rights reserved

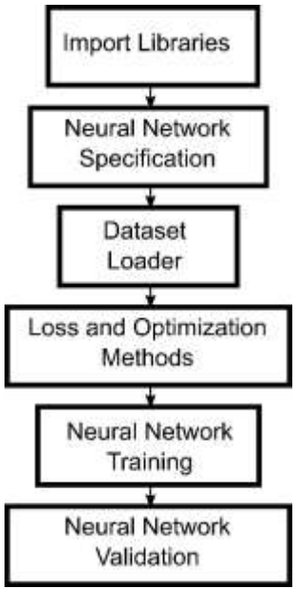

Figure 4 Block diagram for the gunshot identification system on Pytorch
Source: Self-Made.

Firstly, all the libraries required for the project are imported, in this case, the Pytorch libraries for the neural network, Numpy libraries for matrix operations and Pandas libraries for data retrieval from the CSV file. Following, the neurons and the necessary layers for the network are specified, as well as the activation function to be applied on each neuron, which in this project, both the sigmoid function and the hyperbolic tangent function, were tested. Subsequently, the audio MFCCs are retrieved from the CSV file; data and labels are separated and then loaded into the Dataloader.

This loader function is quite useful since it allows loading the data randomly so that the neuron training is not based on the order of the information. Furthermore, the error calculation methods and the optimization of the weights for the neural network are specified; in this case, the binary crossed entropy loss function (BCELoss) and the Stochastic Gradient Descent (SGD) algorithms were used, respectively. Finally, the training of the network is executed. A sample of the Pytorch code where the FCNN class is defined is shown in Figure 5. In this class, two methods must be defined. The first part is the initialization method which is used to create all the necessary layers and activation functions. The second is the forward method where all the layers and activation functions are combined together to create the desired neural network.

```
# Neural network model is defined
class gunshot_fcnn(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(gunshot_fcnn, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)   # Hidden layer
        self.fc2 = nn.Linear(hidden_size,output_size)   # Output layer
        self.sig = nn.Sigmoid()                         # Activation function

    def forward(self, x):
        out = self.sig(self.fc1(x))                     # Hidden Layer
        out = self.sig(self.fc2(out))                   # Output Layer
        return out
```

Figure 5 Sample of the code in Pytorch for the fully connected neural network definition
Source: Self-Made

For the training of the neural network, 500 epochs with a batch size of 10 audio samples were sufficient. In addition, the verification of the accuracy of the network was carried out using 300 samples that were not included in the training. Table 1 shows the success results obtained with different sizes and configurations of the network with the training data.

Number of Coefficients	Network layers	Activation function	Success rate
5	2	Sigmoid	73.2%
7	2	TanH	82.5%
10	2	Sigmoid	100%
10	2	TanH	100%
15	3	Sigmoid	100%
15	3	TanH	100%

Table 1 List of results obtained in Pytorch using different network sizes and configurations
Source: Self-Made

Gunshot Identification System on the Microcontroller

Once a general idea of the size and configuration required on the neural network was determined in Pytorch, the implementation was carried out on the Espressif Systems ESP-32 microcontroller. Table 2 presents some of the processor main components.

Components	Description
Processor	Dual-core CPU, 32 bits, 240 MHz.
Memory	4 MB SPI Flash, 8MB PSRAM.
Peripherals	ADC, DAC, PWM, Timers, 34 GPIO
Communication	Wi-Fi, Bluetooth, SPI, I2C, UART.

Table 2 List of the ESP32 microcontroller main components
Source: Self-Made

ISSN 2414-4924
ECORFAN® All rights reserved

For the implementation of the gunshot identification system on the ESP-32, Micropython programming language was used, which despite having the same syntax as Python, cannot use the Librosa libraries for obtaining the MFCC coefficients. Therefore, this extraction algorithm had to be written from the start for the ESP-32. Similarly, Numpy libraries which are used for matrix operations cannot be used on a microcontroller, for this reason, only one audio signal can be processed at a time. Thus, it was expected that the accuracy obtained previously in Pytorch would not be the same on the ESP-32.

In order to expedite the training of the neural network on the microcontroller, the audio files for the 3000 shots and 500 non-shots were previously processed on a personal PC with Librosa to obtain the MFCC coefficients, and then stored in a MicroSD memory card. Subsequently, it was inserted into the ESP-32 development board and the network training was carried out.

To verify the accuracy of the gunshot detection system, the microcontroller first uses a high-sensitivity sound detection module (KY-037). This device includes a condenser microphone to observe changes in environmental noise. When the sensor detects a sound above an adjustable threshold, it provides a high signal to the microcontroller, and an interrupt service routine (ISR) is executed. In this ISR, the microcontroller performs 10-bit analog-to-digital conversions for a duration of 200 ms while appending each sample to a list. Next, the 10 MFCC coefficients are extracted from the list, normalization is applied to the data, and is then fed into the neural network. The network output is displayed on an LED, which stays on for 2 seconds if the audio signal comes from a firearm. Otherwise, it will turn on for only 0.5 seconds. Figure 6 shows the block diagram of the gunshot identification system on the ESP-32.

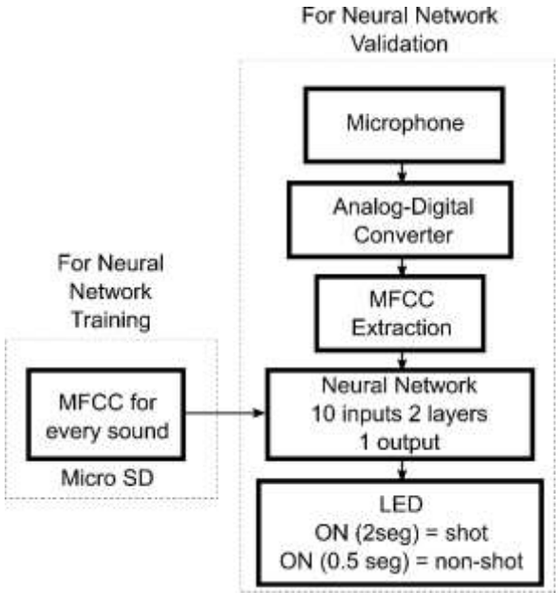

Figure 6 Block diagram for the gunshot detection system executed on the ESP-32 microcontroller
Source: Self-Made

Experimental Results

For the first series of tests of the gunshot detection system on the microcontroller, the sounds were reproduced, one by one, on a personal computer using two 150 W speakers at a volume of approximately 50 dB. The system microphone was placed at a distance of 50 cm from both speakers during the playback of shot and non-shot audios. To verify the success rate, 200 audio recordings of different caliber gunshots and 200 non-gunshots were used. The accuracy rates are shown in Table 3.

A second series of tests was carried out at a local shooting range to test the accuracy of the system with direct detonations of the weapon at different distances. The best results were obtained at a distance of 20 meters from the detonating weapon. When the distance was increased to 100 and 200 meters, the accuracy rate was significantly reduced. These results are attributed to the reduction in the volume of the shot compared to the volume of environmental noise. The results of these tests can also be seen in Table 3.

ISSN 2414-4924
ECORFAN® All rights reserved

Sounds	Emitted	Asserted	Accuracy
Shots on PC	200	175	87.5 %
Non-shots on PC	200	163	81.5 %
Gunshots at a 20 meters	20	18	90 %
Gunshots at 100 meters	20	13	75 %
Gunshots at 200 meters	20	9	70 %

Table 3 Accuracy rates for the gunshot detection system on the ESP-32 microcontroller
Source: Self-Made

Conclusion

In this work, a fully-connected neural network for the identification of gunshots was presented. This network was implemented on an ESP-32 microcontroller, which is a low-cost and low-power mobile device. With this architecture, up to 90% identification was achieved for firearm sounds and 81.5% for non-firearm sounds.

This work will be of great interest to students, teachers or researchers looking for a simple and efficient alternative for the implementation of fully connected neural networks for sound detection.

References

Aguilar, J.R. (2018). Sistemas de detección de disparos: ¿son eficaces para controlar la violencia con armas de fuego en América Latina?, *URVIO Revista Latinoamericana de Estudios de Seguridad*, *23*(1), 128-141. DOI: 10.17141/urvio.23.2018.3454.

Aggarwal, C.C. (2018). *Neural Networks and Deep Learning: A Textbook*, Ed. Springer. DOI: 10.1007/978-3-319-94463-0.

Arif, E., Shahzad, S.K., Mustafa, R., Jaffar, M.A. and Iqbal, M.W. (2021). Deep Neural Networks for Gun Detection in Public Surveillance, *Intelligent Automation and Soft Computing*, *32*(2), 909-922. DOI: 10.32604/iasc.2022.021061.

Bajzik, J., Prinosil, J. and Koniar, D. (2020). *Gunshot Detection using Convolutional Neural Networks*, 24th International Conference Electronics, Palanga, Lithuania. DOI: 10.1109/IEEECONF49502.2020.9141621

Jo, J., Yoo, H. and Park, I.C. (2015). Energy-Efficient Floating-Point MFCC Extraction Architecture for Speech Recognition Systems, *IEEE Trans. VLSI Syst*., *24*(2), 754-758. DOI: 10.1109/TVLSI.2015.2413454.

Katsis, L.K.D., Hill, A.P., Piña-Covarrubias, E., Prince, P., Rogers, A., Doncaster, C.P. and Snaddon, J.L. (2022). Automated detection of gunshots in tropical forests using convolutional neural networks, *Ecological Indicators*, *141*(1), 1-9. DOI: 10.1016/j.ecolind.2022.109128.

Lawrence, D.S., La Vigne, N.G., Goff, M. and Thompson, P.S. (2018). Lessons Learned Implementing Gunshot Detection Technology: Results of a Process Evaluation in Three Major Cities, *Justice Evaluation Journal*, *1*(2), 109-129. DOI: 10.1080/24751979.2018.1548254.

Li, J., Guo, J., Ma, M., Zeng, Y., Li, C. and Xu, J. (2022). A Gunshot Recognition Method Based on Multi-Scale Spectrum Shift Module, *Electronics*, *11*(1), 1-12. DOI: 10.3390/electronics11233859

Morehead, A., Ogden, L., Magee, G., Hosler, R., White, B. and Mohler, G. (2019). *Low Cost Gunshot Detection using Deep Learning on the Raspberry Pi*, IEEE International Conference on Big Data (Big Data), Los Angeles, California. DOI: 10.1109/BigData47090.2019.9006456.

Olmos, R., Tabik, S. and Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning, *Neuro Computing*, *275*, 66–72. DOI: 10.1016/j.neucom.2017.05.012.

Sharma, G., Umapathy, K. and Krishnan, S. (2019). Trends in audio signal feature extraction methods, *Applied Acoustics*, *158*(1), 1-19. DOI: 10.1016/j.apacoust.2019.107020.

Zu, J. and Sutton, P. (2003). *FPGA Implementations of Neural Networks - A Survey of a Decade of Progress*, Ed. Springer-Verlag, Berlin, 1062-1066. DOI: 10.1007/978-3-540-45234-8_120.

ISSN 2414-4924
ECORFAN® All rights reserved

Maher, R.C. and Shaw, S.R. (2008). *Deciphering Gunshot Recordings*, AES 33rd International Conference, Denver, Colorado, 1-8. https://www.researchgate.net/publication/228900815_Deciphering_Gunshot_Recordings.

Shi, L., Ahmad, I., He, Y. and Chang, K. (2018). Hidden Markov model based drone sound recognition using MFCC technique in practical noisy environments, *Journal of Communications and Networks*, 20(5), 509-518. DOI: 10.1109/JCN.2018.000075.

Hossan, M.A., Memon, S. and Gregory, A. (2010). *A novel approach for MFCC feature extraction*, 4th International Conference on Signal Processing and Communication Systems, Gold Coast Australia. DOI: 10.1109/ICSPCS.2010.5709752.

Chirodea, M.C., Novac, O.C., Novac, C.M., Bizon, N., Oproescu, M. and Gordan, C.E. (2021). *Comparison of Tensorflow and PyTorch in Convolutional Neural Networks - based Applications*, 13th. Int. Conf. on Electronics, Computers and Artificial Intelligence, Pitesti, Romania. DOI: 10.1109/ECAI52376.2021.9515098.

McFee, B., Raffel, C., Liang, D., Ellis, DPW., McVicar, M., Battenberg, E. and Nieto, O. (2015). *Librosa: audio and music analysis in Python*, In Proceedings of the 14th Python in science conference, 18-25. DOI: 10.5281/zenodo.6097378.

ISSN 2414-4924
ECORFAN® All rights reserved