http://dx.doi.org/10.35596/1729-7648-2023-21-2-114-120

Original paper

UDC 004.912.5

VOICE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

ULADZIMIR A. VISHNIAKOU, SHAYA BAHAA H.

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Submitted 24.08.2022

Abstract. The article presents an approach, methodology, the software system based on a machine learning technologies for convolutional neural network and its use for voice (cough) recognition. Tasks of article are receiving evaluating a voice detection system with deep learning, the use of a convolutional neural network and Python language for patients with cough. The convolutional neural network has been developed, trained and tested using various datasets and Python libraries. Unlike the existing modern works related to this area, proposed system was evaluated using a real set of environmental sound data, and not only on filtered or separated voice audio tracks. The final compiled model showed a relatively high average accuracy of 85.37 %. Thus, the system is able to detect the sound of a voice in a crowded public place, and there is no need for a sound separation phase for pre-processing, as other modern systems require. Several volunteers recorded their voice sounds using microphones of their smartphones, and it was guaranteed that they would test their voices in public places to make noise, in addition to some audio files that were uploaded online. The results showed an average recognition accuracy – of 85.37 %, a minimum accuracy – of 78.8 % and a record – of 91.9 %.

Keywords: voice detection, convolution neural network, machine learning-based dataset, audio files.

Conflict of interests. The authors declare no conflict of interest.

For citation. Vishniakou U. A., Shaya B. H. (2023) Voice Detection Using Convolutional Neural Network. *Doklady BGUIR*. 21 (2), 114–120. http://dx.doi.org/10.35596/1729-7648-2023-21-2-114-120.

РАСПОЗНАВАНИЕ ГОЛОСА С ИСПОЛЬЗОВАНИЕМ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ

В. А. ВИШНЯКОВ, Б. Х. ШАЙЯ

Белорусский государственный университет информатики и радиоэлектроники (г. Минск, Республика Беларусь)

Поступила в редакцию 24.08.2022

Аннотация. Представлены подход, методология, программная система, основанные на свёрточной нейронной сети, для распознавания голоса (кашля) в условиях зашумленности с использованием технологий машинного обучения. Разработана и оценена система распознавания кашля на основе машинного обучения, использования свёрточной нейронной сети и библиотек языка Python. Свёрточная нейронная сеть протестирована с помощью различных наборов данных и библиотек. В отличие от существующих современных работ в этой области предложенная система оценивалась с применением реального набора звуковых данных окружающей среды, а не только отфильтрованных или разделенных звуковых параметров голоса. Окончательная скомпилированная модель показала относительно высокую среднюю точность – 85,37 %. Предлагаемая система способна распознавать звук голоса в многолюдном общественном месте, и нет необходимости в фазе разделения звука для предварительной обработки, как в других системах. Несколько добровольцев записали звуки своего голоса с помощью смартфонов. Затем они протестировали свои голоса в общественных местах на предмет шума в дополнение к некоторым аудиофайлам, которые были загружены онлайн. Результаты показали среднюю точность распознавания – 85,37 %, минимальную – 78,8 % и рекордную – 91,9 %.

Ключевые слова: распознавание голоса, свёрточная нейронная сеть, набор данных на основе машинного обучения, аудиофайлы.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов

Для цитирования. Вишняков, В. А. Распознавание голоса с использованием свёрточной нейронной сети / В. А. Вишняков, Б. Х. Шайя // Доклады БГУИР. 2023. Т. 21, № 2. С. 114–120. http://dx.doi.org/10.35596/1729-7648-2023-21-2-114-120.

Introduction

The integration of information technology into the healthcare domain was a point of interest, as well a sensitive domain in which people lives are involved. However, technology gave us a great advance in detecting, diagnosing, and treating diseases, as well as speeding and facilitating the huge work of people working in this domain. Patient monitoring tools, scanning tools (X-ray, MRI, etc.), electronic tools, are all examples of the advantages of this interference. But what seems interesting, and according to a study would experience, is automating the healthcare domain, by making early detection, accurate diagnosis, choosing the medicine, or sharing in surgeries [1].

There is published a study on the deadliest diseases in July 2019, and lower respiratory infections come in the third rank, which affect lungs and airways related infection. Lower respiratory infections are considered contagious such as influenza, pneumonia, bronchitis, and tuberculosis in addition to the new virus COVID-19 which affected the whole world. Early detection of such diseases can prevent the spread in a society and help in controlling the infections besides treating them. All the aforementioned diseases share coughing as a common symptom, yet, cough sound is unique for each one of these diseases and diagnosis of the infection can be done from listening to the cough sound.

COVID-19 was a motivation for researchers to involve machine learning, deep learning, and artificial intelligence in detecting infections to stop its high speed spreading around all regions, inspired by others, cough detection was also a point of interest for many researchers before COVID-19 pandemic. In article [2] it is stated that 93 papers related to cough detection and classification were found between 2012 and 2021 which indicates the huge interest of researchers in this domain and the benefits that comes from the detection of cough sound or the type of cough. Their survey is also focused on the method that are used to classify cough sounds and the obtained results of different approaches. Artificial neural networks, convolutional neural networks, k-Nearest Neighbor, and Deep neural network are an example of the methods used in the approaches either to detect coughing or to diagnose cough.

Task statement

Respiratory diseases are known as a common cause for death around the world. “The Global Impact of Respiratory Disease”, a report published by the Forum of International Respiratory Societies, shows very concerning numbers and statistics of deaths and infections caused by the respiratory and lung diseases. Around 3 million people die each year from chronic obstructive pulmonary disease (COPD) only, and asthma affects more than 334 million people, especially children, adding to them acute lower respiratory tract infections that take the third rank in death causalities, not to mention tuberculosis that has 1.4 million died, as well lung cancer the causes death of more than 1.6 million annually. All the aforementioned respiratory diseases share more than one symptom; cough is one of the mutual symptoms and an illness can be diagnosed by the cough sound.

Cough is divided into two types, acute that happens suddenly because of flu or cold, and chronic which is more serious and lasts for months. Medics depend on their diagnosis on the sound of a cough when it is related to respiratory infections, because cough sound is unique for every disease, therefore detecting the sound can prevent complications and the spread of the illness. According to what mentioned above from the serious danger of respiratory diseases and the first diagnosis step it is important to find a solution that can detect cough sound among other sounds to control the infections especially in crowded places such as train stations, airports, libraries, universities, etc. Thus taking the advantage of artificial intelligence seems to be important to detect cough sounds in an accurate and fast manner by using machine learning technologies.

Manual detection of cough sound seems to be easy when it happens in a clinic, however when infected people don’t realize their diseases or they don’t care about other’s health, they could travel in planes,

trains or share restaurants or libraries with others. So the need raises for automated cough detection in a crowd of people. The main objective of this work is to develop a well-trained machine learning model that will be able to distinguish cough sounds from many other environmental sounds, for the sake of slowing or preventing the spread of the respiratory diseases in public environments.

The first step for implementing a machine learning model is selecting the dataset in order to select the suitable method accordingly. The used dataset was Environmental sound classification (ESC) a public dataset that is labeled according to the sound type. This dataset includes cough sounds in addition to many others in the form of WAV. The model (neural network) was built and trained on this data set. Python programming language was used to generate this model with the help of deep learning and data science libraries as "Numpy" and "Keras". The results of the model study showed an improvement in performance after training and testing, recognition increased from 72 to 85 %.

Design of system for cough detection

The proposed system is designed to make classifications and detect cough sounds. There are four main stages after selecting the sound classification dataset (Fig. 1). The first stage is extracting the features from audio files such as the MFFCs, Chromagram, Mel-spectrogram, Spectral contrast, and Tonal centroid features. The second stage is labeling stage, here we categorize the sound samples into cough and non-cough, then we feed the inputs into the convolutional neural network (CNN). Next, we reach the training stage and record the results until we reach the optimal parameters according to the best results (changing epoch number, learning rate, etc.). The final stage, after generating the model, is to conduct several tests on recorded sounds from volunteers.

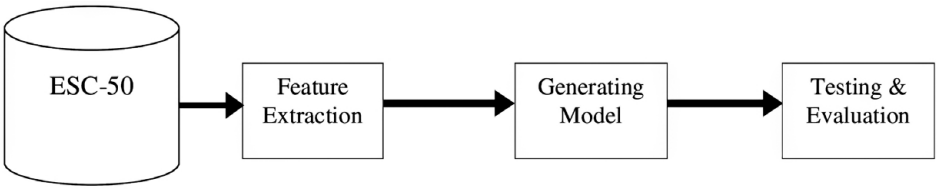

Fig. 1. Proposed system architecture

In general, a dataset is a matrix or database that contains rows and columns that defines a unit, or an object to be analyzed by computers. Columns defines the features or patterns of the object and each row represent a unique entity. Sounds can't be represented in a dataset directly since there are features to be extracted and represented numerically in order to analyze and process them. However, public sound classification datasets come in the form of audio files (WAV, MP3), and its features are extracted manually to meet data scientists and researchers requirements. Environmental sound classification dataset is a collection of 2000 sound files that represent 50 types of environmental sounds. ESC-50 was published in 2015 and used in many publications and systems. Sounds included in ESC have limited background noise since they were in the foreground. The dataset contains animal sounds, natural soundscapes and water sounds, human sounds (non-speech), urban noises, internal sounds. When the data was collected and processed the duration was fixed to 5 s for each audio event where some of them were initially less than the fixed time, they got padded with silence. The sampling frequency was unified to 44.1 KHz as well as the Ogg Vorbis compression set at 192 Kbit/s. More than 50 proposals and approaches using machine learning and signal processing are evaluated and tested on the ESC-50 [4–6]. ESC-50 was used for this research and downloaded from Kaggle website for data scientists that contains more than 50 000 datasets in addition to more than 400 000 public notebooks. Users can access Kaggle using their Google accounts to work in the cloud space offered to them.

Processing of sound in system

The first step before generating DL model, is – preprocessing. The dataset should be cleaned from noise, empty rows, and redundancy, in addition feature extraction methods are important to be done in order to get the important features. Since this dataset is an audio dataset, we used Librosa library to extract the features we mentioned before using functions included in the library. The code is designed to read files that are in *.WAV or *.OGG formats, from a specific folder, then enumerate every sound file and extract its features (STFT, MFCC, Mel-spectrogram, Chroma, Spectral contrast, and Tonnetz), and finally parse them in an array using NumPy.

Here are the functions used from the Libfor library. The first function *extracted_feature*(), used to extract characteristics, returns the average value of each of them as a floating-point number. The function *parse_audio_files()* is used to read audio files from a specific directory and extract the features for every audio file in .WAV or .OGG formats and stack the results in an array. After extracting features (function *get_ext_features*), a user can get the features for any audio file, that was extracted before. Hickle library was used to save the extracted features in order to save time and to use the saved features every time we need them. So, using dump function to save the features in data format (*.HKL) and load function to get them from the directory.

Construction of convolutional neural network system

After extracting the features from audio files and saving them, the CNN now is ready to be fed from the built dataset. The architecture of the CNN is as follows (Fig. 2).

1. The input layer that has the parameter input shape 193.1 referred to the number of features within the dataset. The input layer is convolutional 1-dimension layer with 64 filters and a window (kernel) of size 3. Using the activation function ReLU (Rectified Linear Activation Function), which is a linear function to set negative inputs to zero, whereas positive ones remains the same. And it became the default function for it easiness and high performance results obtained for models using it in neural networks.

2. Two convolutional 1-D layers are implemented after the input layer with 64 filters and kernel of size 3 using ReLU as the activation function.

3. The following layer is the max pooling layer used to down sample the input, and it is used with pool size 3 where the output result will be the input shape (193.1) subtracted from the pool size and added to 1 then divided by the number of strides which default is 1.

4. Two convolution layers with the same activation function are added with 128 filters and kernel of size 3.

5. As a replacement of the fully connected layer, global average pooling layer is added to the CNN, since using the fully connected layer could result in an over-fitted model. Global average pooling layer will generate one feature map related to each category in cough detection system and create a bridge between the previous convolutions and the normal neural network.

6. To prevent over fitting regularization methods are implemented before training the neural network. One method for regularization is to dropout the contribute in reducing interdependent learning between the neurons. During the training phase within every iteration and hidden layer the model will ignore a random fraction of nodes (in our case it is 1/2).

7. Finally, the dense layer, which is the deep neural network that is each neuron is connected to all previous neurons. The dense layer will output finally the results according to the number of classes found in the dataset which are in our case early 49. The activation function used in this network is Softmax which is a probabilistic function that turns vector of numbers to one of the probabilities. The values obtained for each class from the 49 classes will be normalized to 1 using the Softmax activation function.

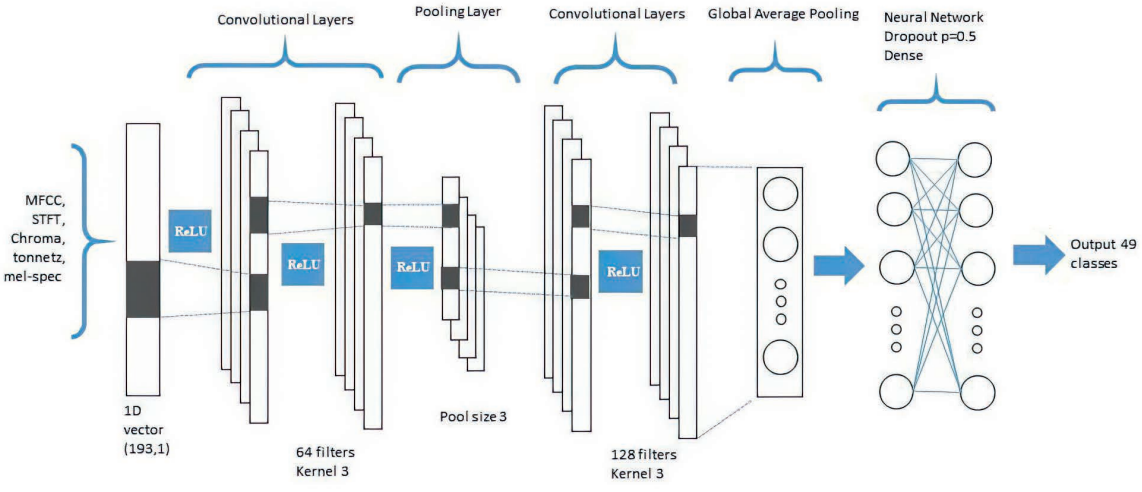

Fig. 2. Architecture of the convolutional neural network

Keras library offers training and testing functions for the given dataset. Some parameters need to be set for the neural network to get the highest performance results. The function "train" accepts the following parameters: features, labels, type, number of classes, epochs, and optimizer. Neural network adjusts its weights according to the results obtained after each iteration using different algorithms known as optimizers which implies calculations on the difference between the results. The implemented network uses the stochastic gradient decent or SGD as the optimizer for its simplicity that calculates the gradient of the network loss function. An initial learning rate for this system was 0.1 with momentum 0.9. The features are those saved using Hickle and the labels as well. The number of epochs usually needs to be changed to get the best results, and we increased the number of epochs from 300 to 500, 750, 1200, and finally 1500. After training stage ended up, the model was saved in *.h5 format, that was used for predicting new sound files. For prediction some sounds were tested and classified using the "predict" function in addition to some recorded cough sounds from volunteers. Prediction method was implemented using a loop to iterate several sound files and output the detected result accuracy for the top three classes that got the highest results.

STFT, MFCC, Mel-Spectrogram, Chromagram, Tonnetz, and Spectral contrast were the features and feature extraction methods applied to get a numerical array in order to compute the DL algorithm. 193 features as total were extracted using Librosa from all the above mentioned methods and resulted in a data-frame that contained 1977 rows and 193 columns from all the input data.

Since we are using the CNN that takes features from images these features were also displayed as images, yet, we take a cough sound as an example for demonstration to show the features as 2D images in Fig. 3. On it the *X*-axis represents the time for the voice recorded in seconds and the *Y*-axis represents frequency (the first image at the left), the second image is the MFCC where the *Y*-axis represents the Cepstral coefficients, the third image shows the Chromagram in terms of frequency and time, and the last images represent the Tonnetz and the Spectral contrast.

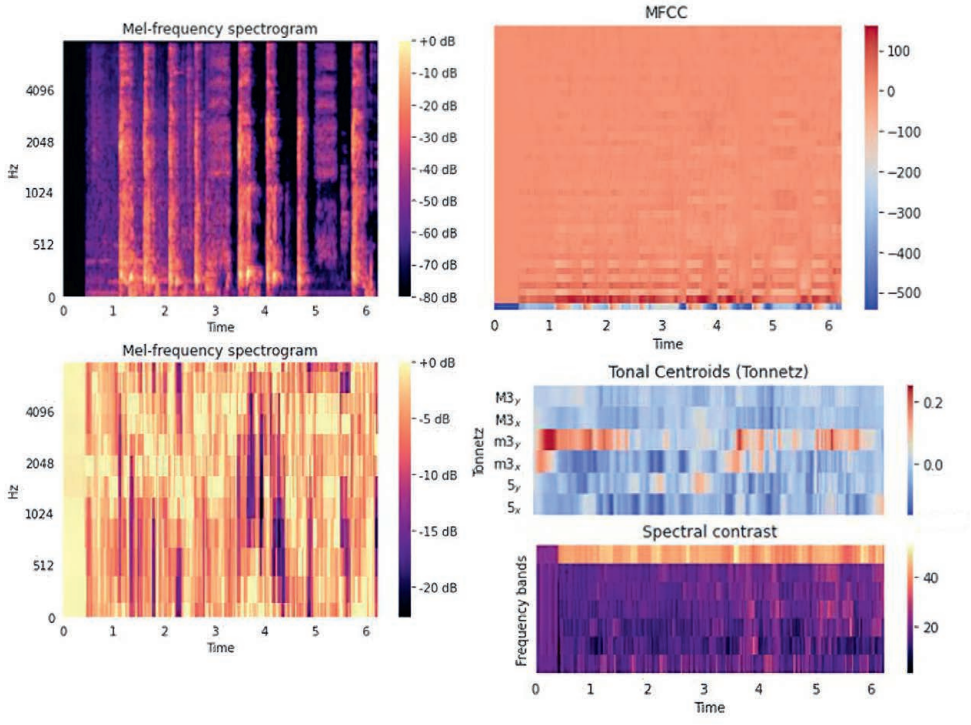

Fig. 3. 2D images representing audio features

The results obtained from the training stage varied according to the number of epoch parameters that were manually altered till the results showed no improvement. As the number of epochs was 300, it is obvious that the performance results will not be as expected and the average accuracy at this number was 43.2 %. The accuracy increased with the increase of epoch number that reached 53.06 % at 500 and 62.9 % at 700. At 1200 the results showed a good accuracy that recorded 77.95 % which also increased to reach 85.37 % at 1500 epochs and remained steady. According to the obtained result the model was saved and used for tests.

Fig. 4 shows the confusion matrix obtained when testing the algorithm on 111 sound files that were not imported to training and not seen before in addition to some recorded cough voices from volunteers using a smartphone microphone. The results showed that 93 files were classified correctly as cough sounds, 10 of them were also detected as non-cough sounds as they actually are. One of the sound files was classified as cough, however it is not and the remaining were classified as non-cough sounds however they are cough sounds. Moreover, the system was able to show the percentage of detection for the tested audio (i. e. 70 % – cough, 20 % – sneezing, 10 % – clock-tick).

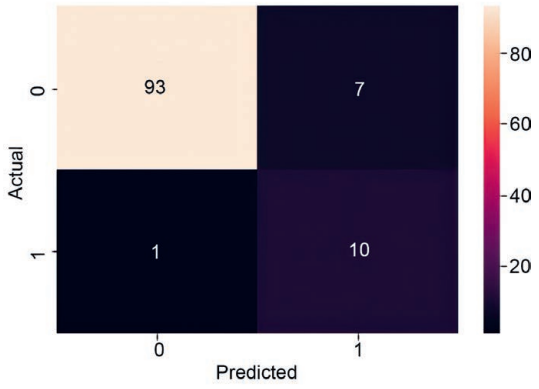

Fig. 4. Confusion matrix for cough detection system

Evaluating of machine learning model depends on some metrics known as precision and recall. Together, precision and recall should be used to know whether the system is good or not. Precision is the ratio of all true positives (classified as cough and they actually were cough) over the whole number of truly positives (both cough and non-cough). Whereas, recall or sensitivity is a ratio of all true positives over the number of true positives and false negatives (all data classified as cough). Using Sci-kit learn, both recall and precision were evaluated and showed 0.919 and 0.788 respectively, which definitely showed a very good classifier. This cough detection system can be used in the IoT network for sound environment monitoring, described in article [7].

Conclusion

1. Cough is a critical symptom associated with many different respiratory diseases, and the detection of cough can lead to the prevention of serious infections and pandemics, such as COVID-19. A lot of research has been done in this area to identify cough sounds associated with certain diseases and to identify cough in general. The completed review introduces the current state of the most relevant cough detection systems and discusses the results obtained by them. The cough detection system designed by the authors was created using the publicly available environmental sound classification 50 (ESC-50) dataset, which was used for machine learning of a convolutional neural network. The network levels as well as the training methods were described in details.

2. After creating the model, another set of sounds was tested to evaluate the model. Several volunteers recorded their voices while coughing using their smartphones, it was guaranteed that they would record their voices in public places to make some noise in the sounds, in addition to some audio files that were uploaded online. The results showed an average accuracy of 85.37 %, precision of 78.8 % and a recall record of 91.9 %.

References

1. Shakel N. V., Ablameyko M. S. (2020) *Medical Worker and Patient: Interaction in the Context of E-Health.* Minsk, Eco-Perspective Publ. (in Russian).
2. Alqudaihi K. S., Aslam N., Khan I. U. [et al.] (2021) Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities. *IEEE Public Health Emergency Collection*. 9, 102327–102344.
3. Amoh J., Odame K. (2016) Deep Neural Networks for Identifying Cough Sounds. *IEEE Transactions on Biomedical Circuits and Systems*. 10 (5), 1003–1011.

4. Gong Y., Lai C.-I. J., Chung Y.-A., Glass J. (2021) SSAST: Self-Supervised Audio Spectrogram Transformer. *Applied Science.* 570–575.
5. Nanni L., Maguolo G., Brahnam S., Paci M. (2021) An Ensemble of Convolutional Neural Networks for Audio Classification. *Applied Science.* 57–76.
6. Chowdhury A., Ross A. (2019) Fusing MFCC and LPC Features using 1D Triplet CNN for Speaker Recognition in Severely Degraded Audio Signals. *IEEE Transactions on Information Forensics and Security.* 15, 1616–1629.
7. Visniakou U. A., Shaya B. H. (2022) Implementation of the Internet of Things Network for Monitoring Audio Information on a Microprocessor and Controller. *System Analysis and Application Informatics.* (1), 39–44.

Authors' contribution

The authors contributed equally to the writing of the article.

Information about the authors

Vishniakou U. A., Dr. of Sci. (Eng.), Professor at the Department of Infocommunication Technologies of the Belarusian State University of Informatics and Radioelectroctronics

Shaya Bahaa H., Postgraduate at the Department of Infocommunication Technologies of the Belarusian State University of Informatics and Radioelectroctronics

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovki St., 6
Belarusian State University
of Informatics and Radioelectroctronics
Tel.: +375 44 486-71-82
E-mail: vish@bsuir.by
Vishniakou Uladzimir Anatolievich