AUDIO BASED DANGEROUS EVENT RECOGNITION IN INDOOR ENVIRONMENT

[1]ZHANDOS DOSBAYEV, [2]PERNEKUL KOZHABEKOVA, [2,5]GULBAKHRAM BEISSENOVA, [3]ZHANAR AZHIBEKOVA, [2]ZHALGASBEK IZTAYEV, [4]VENERA NAKHIPOVA, [5]MUKHTAR KERIMBEKOV, [2]AIGERIM SEITKHANOVA, [6]NURBEK KONYRBAYEV, [7]GAUKHAR SEIDALIYEVA

[1]Satbayev University, Almaty, Kazakhstan
[2]M.Auezov South Kazakhstan University, Shymkent, Kazakhstan
[3]Asfendiyarov Kazakh National Medical University, Almaty, Kazakhstan
[4]Silkway International University, Shymkent, Kazakhstan
[5]University of friendship of people's academician A. Kuatbekov, Shymkent,Kazakhstan
[6]Korkyt Ata Kyzylorda University, Kyzylorda, Kazakhstan
[7]Kazakh National Agrarian Research University, Almaty, Kazakhstan

E-mail: zhandosdosbayev@gmail.com

ABSTRACT

In recent years, automatic systems that control the daily activities of a person are becoming more common. Their main purpose is to ensure civil security, which is achieved through surveillance in public places and the recognition of potentially dangerous situations. Research in the field of automatic surveillance systems is mainly focused on the detection of events using video analytics. In turn, acoustic monitoring can be used as an additional source of information, and being integrated with video surveillance systems, increase the efficiency of event detection. Audio analysis has features that in some situations allow you to solve monitoring tasks more efficiently than video analysis systems, such as: a) low computational requirements, b) independence from visibility conditions (for example, the presence of fog or insufficient lighting). In this paper, we propose audioevent detection system using audio analysis applying machine learning techniques.

Keywords: *Audio events, detection, classification, machine learning, security.*

1. INTRODUCTION

Modern methods of information processing have made a big step forward in various tasks of data processing and analysis. In this ever-increasing volume of digital information, audio plays a special role, since about 20% of the information a person receives through hearing [1]. There are a huge number of different streaming platforms and services that provide access to multimedia content in various forms.

All this has led to the need to develop various methods and systems for automatic analysis of such content. New techniques and approaches help to solve a wide range of tasks: speech recognition, information search based on audio files, multimodal analysis, audio file classification, segmentation, event recognition for security systems and process automation, etc.

Early works on the topic of extracting musical information used symbolic representations or notations, such as MIDI files. [2] Symbolic representations were quite easy to work with, as they do not require high performance capacity. This led to the development of tools for parsing such representations. Monophonic and polyphonic transcriptions helped to work with audio, using the analysis of symbolic representations. However, digitally distributed music is mostly in the form of unstructured audio files.

Various studies have shown that listeners pay attention not to individual notes, but to other aspects of the sound that disappear from the field of view of automatic systems [3]. None of the systems using monophonic and polyphonic transcriptions were successful enough to work with real-world signals.

The emerging interdisciplinary science of extracting information from audio has brought together various fields: computer science, machine learning, signal processing, psychology, and psychoacoustics. The discipline has many practical applications for categorizing, manipulating, and even generating new information.

Methods that are based on semantic similarity of instances are used to create recommendation systems [4]. Previously, such systems were based only on metadata – information about the artist, genre, year of release, etc. Another approach used information about other users ' auditions and made suggestions based on the corresponding collections. Modern systems allow you to look at the internal structure of the signal and analyze the audio characteristics directly [5].

The extracted characteristics help to solve the problem of separating the track without access to the original studio version [6]. The corresponding programs can recognize and divide the track into separate instruments. This is how karaoke versions of musical compositions are created, but the quality is not always perfect, because the frequency range of the vocals is within the ranges of some other instruments [7].

Some applications focus on automatic transcription of music, that is, the process of converting an audio recording to symbolic notation [8]. The result of such programs can be rhythm data, melodies, harmonic information, and final MIDI files. This task becomes more difficult with the increasing number of instruments in the mix and the high polyphony in cases where independent melodies are superimposed on each other.

2. PROBLEM STATEMENT

The increasing quantity and availability of audio data has led to the emergence of automated systems that analyze it. The objectives of the qualification work are

- Development and validation of an algorithm for classifying audio information. The created algorithm should automatically assign one or more predefined classes to any audio recording.

To achieve these goals, the following tasks were set:

1. 1. Search for a representative dataset.
2. 2. Extract the characteristics of audio files.
3. 3. Application and comparison of classification algorithms.
4. 4. Transfer of knowledge of a model trained on a large data set.

The very first step in the work is very important, because the result of the algorithms directly depends on the correct selection of the data set. As part of the work, it is necessary to consider data sets with a good sample of urban sound classes.

Audio can be stored in many formats that have different purposes. These audio data representation formats store information about the frequency and amplitude of the sound. They differ from each other in the degree of compression and focus on professional or budget sound reproduction equipment.

There are three main formats of audio files:

- Uncompressed formats, WAV, AIFF, RAW.
- Lossless compression formats, FLAC, M4A.
- Lossy audio compression formats, MP3, AAC.

Despite the fact that MP3 is a very popular format and is used everywhere in file-sharing networks, the compression principle significantly reduces the accuracy of parts of the audio stream that are considered difficult to distinguish for the human ear. The psychoacoustic model allows you to adjust the compression ratio depending on the limits of sound perception [9]. It is assumed that the lower limit of perception is 16 Hz, and the upper limit is 20,000 Hz. The absolute threshold of audibility depends on the sound pressure level, which is measured in decibels (dB), and the frequency [10].

Sound recorders are able to capture audio beyond the human hearing threshold, which allows additional information to be used in the study.

This paper uses audio files in WAV format, which was developed by Microsoft and IBM in 1991. WAV files are one example of the RIFF (Resource Interchange File Format) container format. Audio files in WAV format store audio with and without compression, although the most common is uncompressed digitization using pulse code modulation (PCM).

Linear pulse code modulation (PCM) is widely used in digital audio recording. For example, WAV, MP3, FLAC, and other formats use pulse-code modulation during the conversion of an analog signal from a CD to a digital one. The sampling rate of such files is 44100 Hz with 16 bits per sample. In professional audio recording, the WAV format with LPCM is used to achieve high-quality audio.

Most devices work with the WAV format, and files in this format are quite easy to edit and process with the help of special software.

The initial data of the study is a set of single-channel audio files converted to WAV format. Each audio file is assigned a y vector, which stores information about the classes of audio events present in the recording.

An audio event is a class or label that answers the question of what is happening on the audio recording. Audio events help you understand and describe the structure of the audio corresponding to the scene (street, small room). For example, a busy street may contain the following sound events: the sound of passing cars and horns, footsteps, and the speech of people. Each audio event can be described with varying degrees of information content. The "music" class can contain a large number of subclasses that describe a genre, a particular instrument, or even a part of the world where a particular sound event can be heard. The "speech" class can additionally describe the speaker using the subclasses "male voice", "female voice", and so on.

The original set of audio signals is used to extract the characteristics. The characteristics of an audio file can be related to time, frequency, or tempo. After combining the characteristics for each instance, a vector x is obtained, which contains numerical information and can be used as an input vector for classifiers and neural networks.

The task of classification can be gradually complicated and move from binary classification to multiclassication. To solve the first problem, you need to divide the data set into two classes, for example, sound events that are related to human speech and others. Further, new classes are added to the model, and each audio file can contain several event labels, which significantly complicates the task.

In this paper, popular classification algorithms should be considered. The resulting models are compared with each other. The weights of these models can be used as initial weights for other datasets.

The final model should classify the incoming audio signal in real time. The result of the program is a vector of classes predicted by the model. When choosing the best method, the accuracy and running time of the algorithm will have to be taken into account.

3. RELATED WORKS

A large amount of research is being conducted in the field of automatic audio generation. Just as with automatic image creation, the developed algorithms have limited success in terms of human perception and evaluation of the result. For example, in 2019, on the birthday of the German composer Johann Sebastian Bach, Google released an interactive Doodle. Users could select the notes that were used to compose a composition in the style of a famous composer. The model was trained on a set of 306 compositions by Johann Sebastian Bach, making patterns using machine learning algorithms [11].

The field of machine learning, associated with the use of deep neural networks, penetrates into all spheres. Using the MNIST (Modified National Institute of Standards and Technology database) data set for classifying handwritten digits, similar sets of audio data appear for analyzing spoken digits and the speaker's gender [12-14].

This paper focuses on solving the problem of classifying audio events. In contrast to the classification of music, the set of event classes is not limited to genres and instruments, but also takes into account audio recordings with other content. The many classes that are considered in the work range from the sounds of nature and animals to the sounds of the urban environment. Some of these classes are of great interest in the field of security [15]. For example, timely recognition of the sounds of gunfire or the noise of broken glass can help the relevant services to respond in time to an emergency situation.

In the modern world, security systems can be found everywhere. Some of these systems have the ability to record not only video, but also audio. The audio signal after processing can carry information that will help the operation of the "smart home".

Recognition of sound events has received considerable interest in recent years. Applications are emerging that solve health monitoring tasks, analyze urban sounds, and even track bird populations [16-18].

In this article [19], the authors solve the problem of classifying audio events (Sound Event Detection, SD) using convolutional Neural Networks (CNN). Neural networks help to extract higher-level characteristics that are invariant to local temporal or spectral changes. By combining convolutional neural networks with Recurrent Neural Networks (RN), the authors propose a CRNN method that significantly improves performance for four sets of everyday audio events. A camera and an equal error rate were used to evaluate the classifier. Despite reducing the error rate to 11% for the CHIMEHOME dataset, the authors point to limitations related to the amount of data available. They also believe that using the Transfer Learning method can improve the performance of algorithms. This idea is considered in this qualification paper.

This article [20] describes an open library for analyzing audio signals written in the Python programming language. The library allows you to extract audio characteristics, and provides high-level and easy-to-use wrappers for various tasks, such as classification, regression, or segmentation.

In addition, the library includes tools for visualizing audio data. The author uses this work for the task of determining emotions in speech, automatically creating previews for the track.

The characteristics extracted using pyAudioAnalysis are used in the qualification work, and the classification models based on the method of support vectors and k-nearest neighbors obtained by the author of the library are compared with the new trained neural networks.

In this paper [21], the authors describe the convolutional network architecture and the algorithms required for data preprocessing. The proposed architecture consists of 3 convolutional layers, followed by two full-weight layers. The authors use methods of increasing the data to achieve a better result on a small set of Urban-Sound 8 K. The methods of shifting the recording in time, adding noise, which are also used in this qualification work, were applied. Although the average accuracy of the algorithm was low-74%. For individual classes, the authors managed to reach 90%.

One of the most important aspects of audio signal processing is feature extraction [22]. There are several features in audio signals, although not all of them are essential for audio processing. Each function represents a vector unit in the feature space, and all classification systems use a series of features derived from the input audio signal. As a result, a variety of audio classification approaches have been proposed based on device accuracy assessment. The main differences in both methods are the classifiers used and the amount of acoustic features used. The derived features are divided into temporal, spectral, and prosodic features in terms of decomposition. Audio classification is a step in audio signal processing and pattern recognition that can be used in audio identification, recording, and event interpretation. The capacity to specifically identify chosen function vectors in appropriate ng groups is referred to as audio classification. To minimize classification issues, various classifier types are used, including time-consuming manual classification, supervised, unsupervised, and semi-supervised learning algorithms, and supervised, unsupervised, and semi-supervised learning algorithms.

4. DATASET

There are many different data sets for analyzing music, for example: Free Music Archive, Million Song Dataset [22-23]. For speech analysis, you can use the Free Spoken Digit Dataset [24], Voxceleb [25], The Spoken Wikipedia Corpora [26]. Data sets differ in size (from 10 megabytes to several terabytes), the presence of meta-information, and the type of data in which this set is represented (some sets consist only of pre-calculated characteristics).

However, the above datasets are very homogeneous, so the choice fell on the large-scale AudioSet dataset [27] provided by Google. AudioSet consists of more than 2 million manually labeled ten-second audio clips uploaded to the YouTube video hosting service. Each passage is assigned classes of audio events (music, speech, car sound, etc.). The number of unique classes is 527. The ontology covers a wide range of human and animal sounds, musical instruments and genres, as well as everyday environmental sounds.

The AudioSet dataset is divided into three disjoint samples: two sets for balanced learning and outcome evaluation, and a set for unbalanced learning. The first two data sets contain about 20,000 passages, with at least 59 instances per class. The unbalanced set represents the remainder of the dataset.

Figure 1 shows the distribution of classes by the number of instances in the AudioSet. The largest classes are related to music and speech, while the number of instances for some specific sounds does not exceed 200.

Audio Set provides data to download in two formats:

- A CSV file that describes for each instance: the ID of the YouTube video; the start and end time of the segment; the classes to which this instance belongs.
- 128-dimensional audio characteristics extracted at a sampling rate of 1 Hz. To extract them, a VGG-like model was used, which worked with YouTube-8M (a set of 6 million marked-up excerpts for video analysis).

In order to independently extract the characteristics and work directly with the audio files, it was decided to use the first option.

The model trained on the AudioSet dataset is then used to work with the Urban Sound Classification dataset [28]. This set contains 8732 audio fragments in WAV format, which belong to 10 classes.

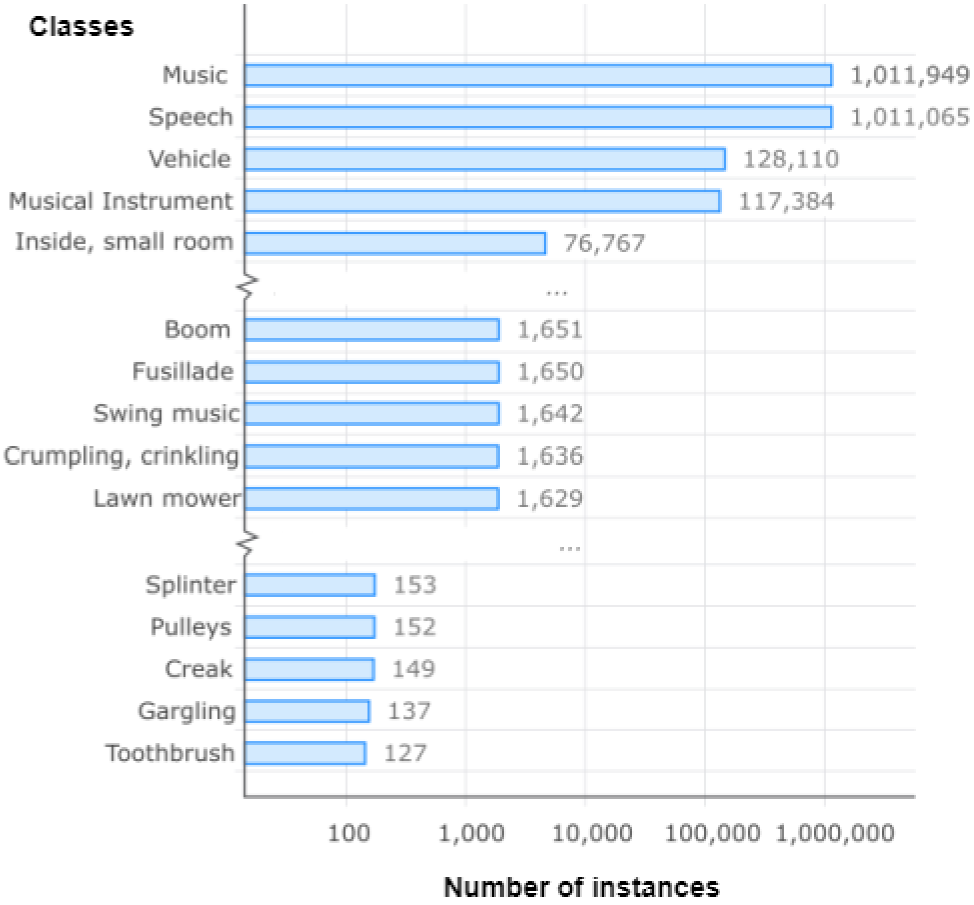


Figure 1: Instances And Classes

5. AUDIO FEATURES

There are many different characteristics for audio. They can be related to time, frequency, or tempo. Basically, the time characteristics are extracted directly from the audio signal. In this paper, the following characteristics were used.

Energy – the sum of the squares of the signal values, normalized over the length of the frame. Let $x\ i\ (\ n\ )$, n = 1,..., $W\ L$ be the sequence of audio samples of the frame$i$, where$W\ L$ is the length of the frame. Then the value of the signal energy E, can be calculated by the formula:

$$E(i) = \frac{1}{W_L} \sum_{n=1}^{W_L} |x_i(n)|^2 \qquad (1)$$

Equation (1) describes the signal strength. This characteristic tends to have large jumps in successive signal windows for certain classes. For example, speech may contain a large number of pauses and weak phonemes.

Zero crossing rate, ZCR – the number of changes in the sign of the audio signal in the interval, that is, the number of times when the signal changes its value from positive to negative or vice versa. ZCR is defined by the following equation:

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} |sgn[x_i(n)] - sqn[x_i(n-1)]|, \qquad (2)$$

Where sgn is a piecewise constant function defined by the following equation:

$$sgn[x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0, \\ -1, & x_i(n) < 0. \end{cases} \qquad (3)$$

ZCR can be interpreted as a measure of noise, and this characteristic usually takes on large values in the case of noisy signals. In addition, ZCR can express the spectral characteristics of a signal with some accuracy. These properties and low computational complexity have led to the fact that ZCR is used in many tasks from speech detection to the classification of musical genres.

Entropy of Energy – short-term entropy of energy, a measure of sudden changes in the energy level of a signal. To calculate the entropy of the energy, each frame is divided into K parts of the fixed duration. Then, for each sub-frame j, the energy is calculated $EsubFramej$ and is divided by the total energy $E\ s\ hortFramei$. The resulting sequence of energy values for the subframese j, j = 1,..., K is expressed by the relation:

$$e_j = \frac{E_{subFrame_j}}{E_{shortFrame_i}}, \tag{4}$$

Where

$$E_{shortFrame_i} = \sum_{k=1}^{K} E_{subFrame_k} \tag{5}$$

At the last step, the entropy H (i) of the sequence $e\ j$ is calculated according to equation (6):

$$H(i) = -\sum_{j=1}^{K} e_j \cdot log_2(e_j) \tag{6}$$

The resulting value decreases with sudden changes in energy. Therefore, this characteristic can be used to detect the moment of the start of a shot, explosion, or other similar sounds with large and rapid changes in energy values.

Additionally, studies show that the minimum entropy value varies depending on the musical genre. For example, the minimum entropy for classical music is greater than for electronic music. This can be explained by the fact that electronic music contains a greater number of sharp transitions and energy surges.

6. FEATURE EXTRACTION

The audio signal is first divided into short segments or frames, then the desired audio characteristics are extracted for each frame. The result is a sequence of short-term characteristic vectors. The size of such segments may vary, but windows with a duration of 20 to 100 milliseconds are widespread.

The pyAudioAnalysis library was used to extract the characteristics [29]. The resulting result can be represented as a graph shown in Figure 2. The graph shows two characteristics: ZCR and energy.

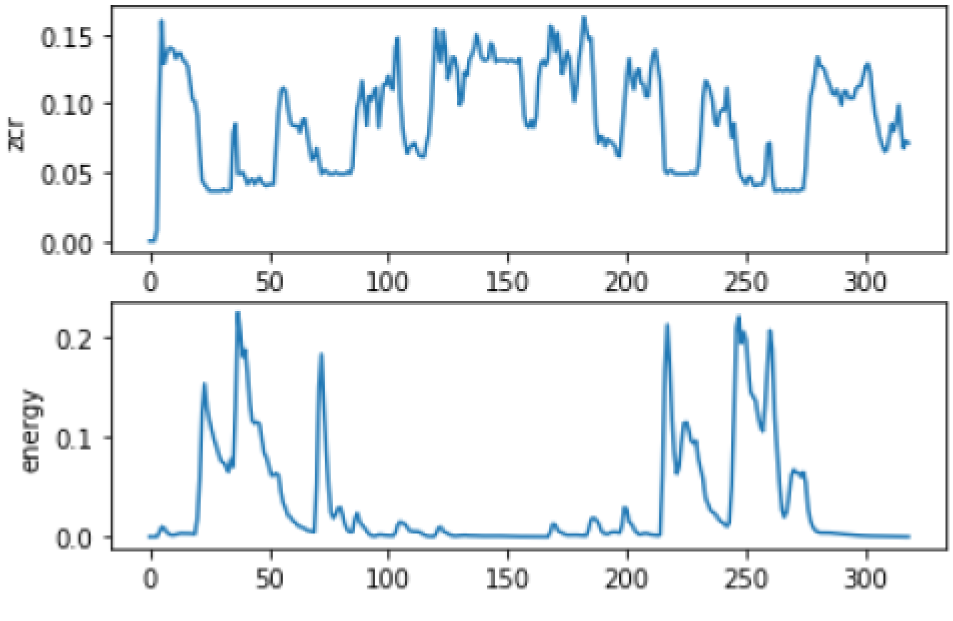


Figure 2: Zero Crossing Rate And Energy Values Depending On The Frame

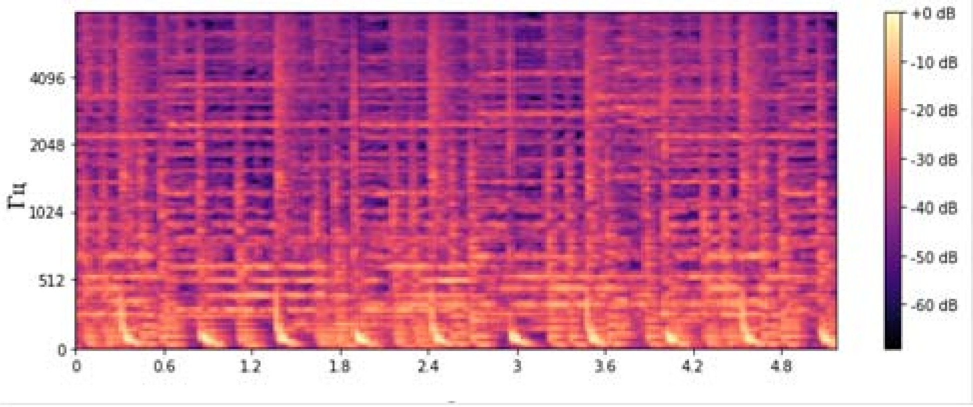


Figure 3: Audio Signal Spectrogram

The pyAudioAnalysis library also allows you to output audio signal spectrograms. Figure 3 shows an example of this representation of an audio file.

AudioAnalysis are calculated on small segments with overlapping (overlapping) or without (non-overlapping). In the first case, the step length is less than the window length, and in the second case, they are equal.

Another approach that is used in audio analysis is to process feature sequences based on medium-sized segments. These segments can also overlap. For each segment, short-term characteristics are calculated on small windows, and then the average values for each characteristic are found. Further, the resulting sets of statistics are used to represent these medium-sized segments.

Standard values for the length of the middle segment range from 1 to 10 seconds. For longer audio recordings, you can calculate long-term characteristics, that is, find the average for medium-sized segments.

For the AudioSet data set, short-term characteristics were used in this work, because the duration of audio recordings does not exceed 10 seconds. The sequence of characteristics was extracted with a frame duration of 50 milliseconds and a step length of 25 milliseconds. Thus, the percentage of overlapping segments is 50%.

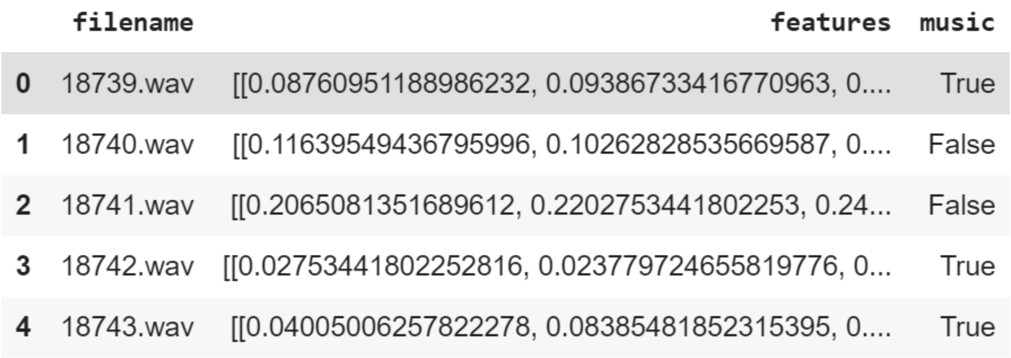
The function that extracts the characteristics in pyAudioAnalysis returns 68 characteristics. After extracting them, each audio file can be represented as a matrix. For ten-second excerpts, the matrix size is 68x400.

The speed of feature extraction is low, for example, it took about 2 hours to process 6,500 instances of the dataset. However, this speed still allows you to process audio signals in real time.

The resulting array with characteristics for all audio recordings is then stored in the Data Frame data structure from the pandas library. The first column of the table corresponds to the file name, and the second column stores the numerical values of the characteristics themselves.

The third column stores the y vectors of the correct classes that the audio recording belongs to. The ontology file is used to construct the vectors.

The table with the data and extracted characteristics can be easily saved for later work. An example of a table for binary classification of music and all other classes can be seen in Figure 4.



	filename	features	music
0	18739.wav	[[0.08760951188986232, 0.09386733416770963, 0....	True
1	18740.wav	[[0.11639549436795996, 0.10262828535669587, 0....	False
2	18741.wav	[[0.2065081351689612, 0.2202753441802253, 0.24...	False
3	18742.wav	[[0.027534418022528816, 0.023779724655819776, 0...	True
4	18743.wav	[[0.04005006257822278, 0.08385481852315395, 0....	True

Figure 4: The First Five Items In The Table For Binary Classification

In the course of the study, it was found that the parent classes were not specified in the original sample for many audio events. Thus, the child classes that are responsible for musical instruments do not belong to the parent class – music.

For the vectors to correctly describe all audio events, it is necessary that they contain information about each class to which the audio recordings belong. Each class is recursively checked for subclasses, and then a hierarchy is created.

Combining classes into larger groups allows you to simplify the task in the first step. This approach helps to reduce the task of multiclassication to binary classification, for example, to learn to distinguish between sound events related to speech and music.

7. CLASSIFICATION

With the increase in the amount of information and the popularization of machine learning methods, neural networks have become one of the most accurate algorithms. They are used to analyze text, images, find defects, create new data, etc.

Over time, the number of parameters, the depth of neural networks, and the number of hidden layers only grows. One of the most important steps was the introduction of VGG-like models, which will be described later.

How well the data set is represented is incredibly important for classifying and training the model. For example, in an Audio Set, there may be large differences within a class, while different classes may have similar characteristics. That is, there is a problem of high intra-class variation. This means that the characteristics of objects belonging to the same class can differ significantly from each other.

For example, the “music " class contains classical works, electronic remixes, and individual instrument sounds. For such audio events, the difference in the values of the frequency and time characteristics can lead to erroneous misclassification.

On the other hand, the AudioSet dataset contains instances with high inter-class similarity. For some speech-related classes, it is required to determine the gender of the speaker or even their age.

Figure 5 illustrates audio event classification pipeline.

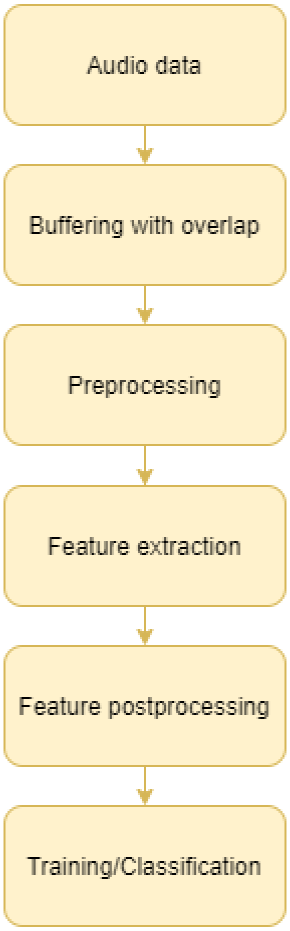


Figure 5: Audioevent Classification Overlap

In order to cope with such a task during the training of a deep neural network, it is necessary to use non-trivial methods. In particular, in this paper, three methods were used to train a small convolutional neural network, which simultaneously increased the efficiency of the algorithm and served as a tool for regularization. Then the weights were moved to work with a smaller data set.

8. DATA AUGMENTATION

One of the disadvantages of using machine learning methods is the need to work with extended data sets to better generalize the results. Working with small data sets, the model is able to quickly retrain, that is, to reach a state in which with each new epoch, the accuracy for the training set increases, and for the test set decreases.

A retrained model will produce erroneous results if all instances of the test set were obtained in the same environment. In order to prevent this, you can use the method of increasing the amount of data (Data augmentation).

There are various ways to transform audio files, including adding random noise, time shifts, changing the pitch and recording speed.

The balanced learning dataset contains only 22 thousand instances, which is significantly less than the unbalanced set, which has 2 million fragments. The above methods of increasing the amount of data can significantly improve the quality of the model.

In this work, we used the superimposition of noise on audio recordings and changing the pitch of the sound. Adding noise is done using the NumPy library [30], which generates a sequence of random values. The sequence elements are multiplied by the noise factorization parameter, which changes the force with which random noise affects the original audio recording. In order to change the pitch of the sound, the pitch_shift function from the librosa library was used [31]. The result is shown in Figure 6.

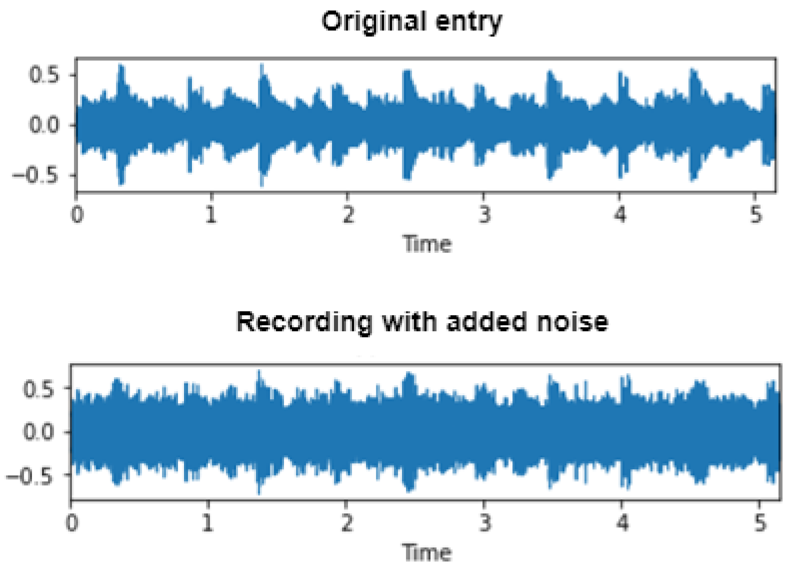


Figure 6: Comparing The Waveform Of An Audio Recording Before And After Adding Noise

This algorithm was applied to each audio recording of the original set. Thus, we managed to double the amount of data for training.

9. RESULTS

The data set was divided using the stratification function built into the scikit-learn library. 70 percent of the data was used for training, and 30 percent for testing. The batch size is 50. Hence, each epoch consists of 280 small batches. Testing was carried out on the remaining 120 packages.

The loss function was optimized using the Adam algorithm, which uses advanced methods to overcome the disadvantages of conventional stochastic gradient descent [30]. The value of 10-4 was chosen for the learning rate coefficient. ReLU was used as the activation function, as it provides better convergence and does not suffer from gradient fading.

A two-dimensional array of characteristics with a dimension of 68x400 is fed to the input of each network. The result on the output layer is different for binary classification and multi-value classification problems, where values rounded to the nearest integer are used.

All calculations were performed on the Google Colab platform [31], which provides access to the Tesla P4 GPU. Each approach has been tested on at least 40 epochs. If at some point in time the loss function started to grow over several epochs, the training stopped. Next, the approaches are analyzed and compared with each other.

A simple model trained from scratch took about 100 epochs to converge. Despite the fact that each epoch took no more than a minute, training a simple model took the most time and computing power due to the high number of epochs, (Figure 7).

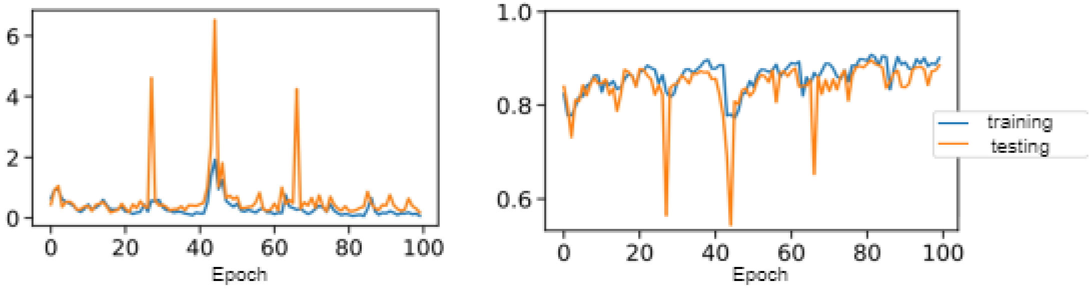


Figure 7: Plot Of The Loss Function And Accuracy Of A Simple Model For Binary Classification

Multiple outliers in the graph of the loss and accuracy function indicate that the model did not finish training and reached a local minimum during optimization. After 30 epochs without sharp outliers, the training was stopped, as the neural network showed signs of convergence. The additional epochs did not lead to an increase in accuracy, but the model showed no signs of overfitting.

As a result, after 100 epochs, the simple model achieved 92.1% accuracy for the binary classification problem, while for the multiclassication problem the accuracy was lower-85.7%, (Figure 8).

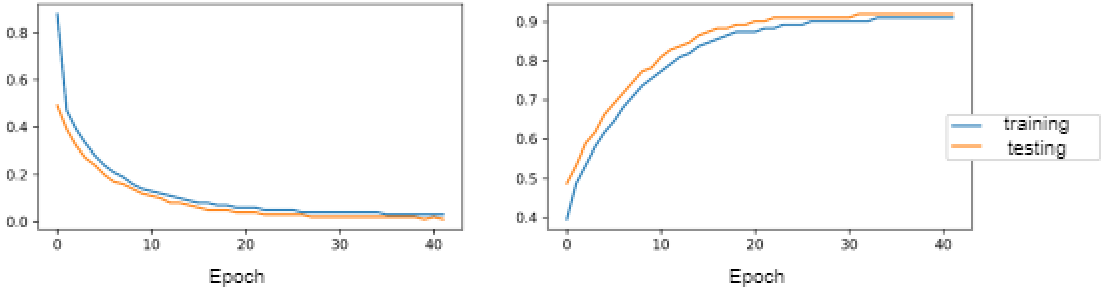


Figure 8: VGG 16 Loss And Accuracy Function Graph For Multiclassication

The learning curve for the VGG 16 model shows faster convergence. Since the number of parameters for VGG 16 is much higher than for the previous model, each training epoch took approximately 114 seconds, which is almost twice the time for a simple model. After 40 epochs, the neural network achieved 98.8% accuracy in the binary classification problem, and 92.6% for multiclassication.

After 40 epochs and about 42 minutes of real time, the simple model achieved 74% accuracy, much less than the results obtained for the AudioSet dataset.

Pre-trained VGG models trained on the extended Urban Sound Classification dataset performed slightly better, reaching 81-85%. The results of VGG 11 and VGG 16 were compared. It turned out that the final accuracy values for VGG 16 do not exceed VGG 11. Moreover, training a model with VGG 11 is less time-consuming. It took 34 minutes for VGG 16 to converge, and 28 minutes for VGG 11 to converge.

The final result tables are as in the table 1:

Table 1: Results Of Audio Event Detection.

Event type	***Accuracy***	***Precision***	***Recall***	***F1 score***	***AUC ROC***
Gunshot	0.9178	0.9245	0.9427	0.8945	0.9748
Broken glass	0.9372	0.9765	0.9215	0.9154	0.9578

Fire	0.9435	0.9346	0.9215	0.9345	0.9576
Siren	0.9537	0.9462	0.9876	0.9642	0.9623
Explosion	0.8132	0.8254	0.8352	0.8124	0.9348
Cry	0.8635	0.8524	0.8864	0.8754	0.9467
Dog barking	0.8456	0.8325	0.8571	0.8254	0.9425
Fire alarm bell	0.8654	0.8452	0.8576	0.8457	0.9472

10. CONCLUSION

In the course of the work, we managed to complete the tasks set. Algorithms for downloading data and extracting characteristics were implemented. In addition, solutions were found to increase the sample size for small data sets.

The resulting models, based on neural networks, are able to classify a wide range of sound events. They are not inferior in terms of indicators to previously developed algorithms based on the methods of k-nearest neighbors and support vectors.

Models trained on a large AudioSet set performed well for the extended Urban Sound Classification set. Previous work achieved 74% accuracy, and pre-trained VGG models managed to improve this result, bringing the accuracy to 85%.

It can be noted that the increase in the number of

Models trained on a large AudioSet set performed well for the extended Urban Sound Classification set. Previous work achieved 74% accuracy, and pre-trained VGG models managed to improve this result, bringing the accuracy to 85%.

It can be noted that increasing the number of hidden layers in VGG-like models increases the accuracy only by small values. It may be worth paying attention to simpler models with fewer parameters, because for some problems, increasing computational complexity only increases the training time.

The classification of audio recordings occurs at a speed sufficient for real-time operation. The next steps in the research will be the introduction of automatic audio capture and simultaneous analysis of the audio stream, which will allow you to classify audio events captured by the microphone on any device.

The implemented classifiers can be used in various fields. They can help inform the services in advance about an emergency situation, or about the condition of the patient who is being monitored. Some apps even include animal migration tracking based on sound analysis.

In addition, such networks can be used to automatically generate new audio recordings. Of great interest is the use of variational autoencoders and generative-adversarial networks in the field of music.

Probably, the latest algorithms based on residual learning and more fine-tuning of parameters can significantly improve the efficiency of solving problems of classification and analysis of audio data.

REFERENCES:

[1] Babaee, E., Anuar, N. B., Abdul Wahab, A. W., Shamshirband, S., & Chronopoulos, A. T. (2017). An overview of audio event detection methods from feature extraction to classification. Applied Artificial Intelligence, 31(9-10), 661-714.

[2] Alsina-Pagès, R. M., Navarro, J., Alías, F., & Hervás, M. (2017). homesound: Real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring. Sensors, 17(4), 854.

[3] Omarov, B. (2021). Electronic Stethoscope for Heartbeat Abnormality Detection. In Smart Computing and Communication: 5th International Conference, SmartCom 2020, Paris, France, December 29–31, 2020, Proceedings (p. 248). Springer Nature.

[4] Kumar, A., & Ithapu, V. K. (2019). Secost: Sequential co-supervision for weakly labeled audio event detection. arXiv preprint arXiv:1910.11789.

[5] Krstulović, S. (2018). Audio event recognition in the smart home. Computational Analysis of Sound Scenes and Events, 335-371.

[6] Dang, A., Vu, T. H., & Wang, J. C. (2017, December). A survey of deep learning for polyphonic sound event detection. In 2017 International Conference on Orange Technologies (ICOT) (pp. 75-78). IEEE.

[7] Lu, X., Shen, P., Li, S., Tsao, Y., & Kawai, H. (2018). Temporal Attentive Pooling for Acoustic Event Detection. In Interspeech (pp. 1354-1357).

[8] Adavanne, S., Politis, A., & Virtanen, T. (2018, July). Multichannel sound event detection using 3D convolutional neural networks for learning inter-channel features. In 2018 international joint conference on neural networks (IJCNN) (pp. 1-7). IEEE.

[9] Kothinti, S., Imoto, K., Chakrabarty, D., Sell, G., Watanabe, S., & Elhilali, M. (2019, May). Joint acoustic and class inference for weakly supervised sound event detection. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 36-40). IEEE.

[10] Arora, V., Sun, M., & Wang, C. (2019, May). Deep embeddings for rare audio event detection with imbalanced data. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3297-3301). IEEE.

[11] Chung, S. H., & Chung, Y. J. (2018). Comparison of Audio Event Detection Performance using DNN. The Journal of the Korea institute of electronic communication sciences, 13(3), 571-578.

[12] Takahashi, N., Gygli, M., & Van Gool, L. (2017). Aenet: Learning deep audio features for video analysis. IEEE Transactions on Multimedia, 20(3), 513-524.

[13] Wang, Y., Li, J., & Metze, F. (2019, May). A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 31-35). IEEE.

[14] Omarov, B., Baisholanova, K., Abdrakhmanov, R., Alibekova, Z., Dairabayev, M., Narykbay, R., & Omarov, B. (2017). Indoor microclimate comfort level control in residential buildings. Far East Journal of Electronics and Communications, 17(6), 1345-1352.

[15] Lim, H., Park, J., & Han, Y. (2017, November). Rare sound event detection using 1D convolutional recurrent neural networks. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017) (pp. 80-84).

[16] Omarov, B., Anarbayev, A., Turyskulov, U., Orazbayev, E., Erdenov, M., Ibrayev, A., & Kendzhaeva, B. (2020). Fuzzy-PID based self-adjusted indoor temperature control for ensuring thermal comfort in sport complexes. J. Theor. Appl. Inf. Technol, 98(11).

[17] Pham, P., Li, J., Szurley, J., & Das, S. (2018, April). EVENTNESS: Object detection on spectrograms for temporal localization of audio events. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2491-2495). IEEE.

[18] Kong, Q., Xu, Y., Sobieraj, I., Wang, W., & Plumbley, M. D. (2019). Sound event detection and time–frequency segmentation from weakly labelled data. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(4), 777-787.

[19] Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection. Emre Çakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, Tuomas Virtanen. 2017.

[20] pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. Theodoros Giannakopoulos. 2015.

[21] Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. Justin Salamon, Juan Pablo Bello. 2016.

[22] Dhanalakshmi, P., S. Palanivel, and V. Ramalingam. 2009. Classification of audio signals using SVM and RBFNN. Expert Systems with Applications 36 (3):6069–75.

[23] Adavanne, S., Pertilä, P., & Virtanen, T. (2017, March). Sound event detection using spatial features and convolutional recurrent neural network. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 771-775). IEEE.

[24] Adavanne, S., Parascandolo, G., Pertilä, P., Heittola, T., & Virtanen, T. (2017). Sound event detection in multichannel audio using spatial and harmonic features. arXiv preprint arXiv:1706.02293.

[25] Altayeva, A., Omarov, B., & Im Cho, Y. (2018, January). Towards smart city platform intelligence: PI decoupling math model for temperature and humidity control. In 2018

IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 693-696). IEEE.

[26] Omarov, B., Omarov, B., Issayev, A., Anarbayev, A., Akhmetov, B., Yessirkepov, Z., & Sabdenbekov, Y. (2020, November). Ensuring Comfort Microclimate for Sportsmen in Sport Halls: Comfort Temperature Case Study. In International Conference on Computational Collective Intelligence (pp. 626-637). Springer, Cham.

[27] Murzamadieva, M., Ivashov, A., Omarov, B., Omarov, B., Kendzhayeva, B., & Abdrakhmanov, R. (2021, January). Development of a System for Ensuring Humidity in Sport Complexes. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 530-535). IEEE.

[28] Lafay, G., Benetos, E., & Lagrange, M. (2017, October). Sound event detection in synthetic audio: Analysis of the dcase 2016 task results. In 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (pp. 11-15). IEEE.

[29] Derakhshan, M. O. R. A. D., Marvi, H. O. S. S. E. I. N., & Hassan Poor, H. (2019). Providing an Adaptive Model with two Adjustable Parameters for Audio Event Detection and Classification in Environmental Signals. TABRIZ JOURNAL OF ELECTRICAL ENGINEERING, 49(2), 565-576.

[30] Arora, P., & Haeb-Umbach, R. (2017, October). A study on transfer learning for acoustic event detection in a real life scenario. In 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP) (pp. 1-6). IEEE.

[31] Lim, H., Kim, M. J., & Kim, H. (2015). Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation. In Sixteenth Annual Conference of the International Speech Communication Association.