Neural Networks for automatic environmental sound recognition

Svetlana Segarceanu
Beia Consult International
Bucharest, Romania
svetlana.segarceanu@beia.ro

George Suciu
Beia Consult International
Bucharest, Romania
george@beia.ro

Inge Gavat
Universitatea Politehnica
Bucuresti Bucarest, Romania
i_gavat@yahoo.com

***Abstract*—Environmental sound recognition is currently an important and valuable field of computer science and robotics, , security or environmental protection. The underlying methodology evolved from primary speech application characteristic methods to more specific approaches, and with the advent of the deep learning paradigm many attempts using these methods arose. The paper reopens the research we have started on the application of the Feed Forward Neural Networks, by exploring several configurations, and introduces the Convolutional Neural Networks in our investigation. The experiments consider three classes of forest specific sounds and meant to detect the chainsaw sounds chainsaw, vehicle, and genuine forest.**

Keywords—AESR, Deep Feed Forward Neural Networks, Convolutional Neural Networks, training

I. INTRODUCTION

In recent times, automatic environment sound recognition gains increasing attention for its multidisciplinary applications in areas like an audio surveillance system, environment monitoring, wildlife, and protection, by detection of logging activities, wildfire events, impostor in the wildlife or logging areas [1] [2]. It is more and more extensively applied in home automation by investigating domestic non-human and apart from music environmental sounds in day-to-day life like glass breaking, door knock, or floor crack, pouring of water, the sound of an engine, vehicle brakes, baby crying. It is used to detect sound-related respiratory symptoms, such as sneeze, cough, sniffle, throat clearing, related to illness, allergies, infections, commonly observed and useful in health-related research [3]. For example, by regularly collecting constantly self-reported flu symptoms from registered users, nationwide flu maps are built in [4] to illustrate how flu spreads.

It is also employed in monitoring acoustic emissions, to study or predict landslides, avalanches or other mass movement [5].

A complex field of AESR, Computational Auditory Scene Analysis (CASA) [2], concerns the recognition of combinations of sound sources using computational means that simulate human listening perception. It investigates mainly two important tasks, Environmental Audio Scene Recognition (EASR) and Sound Event Recognition (SER), for environment audio observation and surveillance. EASR refers to the recognition of indoor or outdoor acoustic scenes (e.g., cafes versus crowded or silent streets, forest soundscape, countryside). SER investigates specific environmental acoustic events, like dog barking, cat meowing, crying, gunshots, cough, laugh, whistle. Monitoring of human social activities and early detection of suspicious events are essential for public security and safety, and are usually performed using one or more video cameras, and possibly infrared cameras. When visual signals cannot recognize the environment activities and events, audio supplementary cues are valuable. For instance, when a suspicious/object is occluded, when an activity happens in the dark, or in an area beyond the coverage of video cameras.

The present paper is related to previous research concerning identification of logging activity by forest environment sound recognition by investigating the acoustics forest environment recognition. Previous research has emphasized the advantages of using Deep Feed Forward Neural Networks (DNN) over the traditional machine learning approaches, like Gaussian Mixture modelling (GMM) or Dynamic Time Warping (DTW). We have analysed in the above-mentioned context several different schemes of applying DNN. We will go further with our analysis concerning DNN, by investigating several settings in defining the network in the training process. We will also present some preliminary results obtained applying Convolutional Neural Networks (CNN).

II. STATE-OF-THE-ART

Early attempts of environmental sound recognition were made by using the traditional approaches applied in speech and voice-based applications, with the particular means in what concerns signal processing, feature extraction and modelling methods [6 7]. Gradually the catalogue of these methods widened and begun to specialize, for instance taking in account acoustic signal division into stationary and non-stationary, or according to the particular fields of applications, With the advent of the state-of-the-art modelling paradigm of deep neural networks the AESR techniques evolved by exploring the applicability of these architectures. As Deep Neural Networks are able to discover and generate themselves effective features from less processed, or even raw data, the stress is also on discovering the suitable acoustic data processing approaches.

Among the first approach using a DNN classifier for sound event recognition was proposed [8] The research evaluated both Support Vector Machines (SVM) and DNN classifiers and the results showed that the DNN classifier with simple de-noising performed well for the recognition task. Piczak in [9], used CNNs trained on Log-Mel spectrogram features to achieve a similar level of output as other deep learning methods (DNN and Recurrent Neural Network - RNN). CNN, successfully applied in image recognitional systems, were investigated in the context of the time-frequency features called Spectrogram Image Features (SIF), to reproduce the image recognition context. The use of multi-label Convolutional Recurrent Neural Network (CRNN) for polyphonic scenery in real-life recordings was proposed in [10] by Cakir et al. The researchers have shown that CRNN performance is slightly better compared to CNN and RNN for certain audio events. In [11] Su et al. also used fully

978-1-7281-0984-8/19/$31.00 ©14/06/2021 IEEE

convolutional network with log-scale Mel-spectrogram features to identify audio events using the.

The authors in [12] studied the Spectrogram Image Features (SIF) for noisy environments. In this study, the highly overlapped spectrograms were converted into linear quantized images and their dimensions were reduced by applying various image resizing methods. The feature learning and recognition was performed with the CNN approach. The work in]13] introduces an audio-visual descriptor, called the Auditory Receptive-field Binary Pattern (ARFBP), built on the SIF, the cepstral features, and the Human Auditory Receptive Field model. These features are fed to a hierarchical version of DNN, called Hierarchical-diving Deep Belief Network (HDDBN). Diverse other approaches are mentioned in [2]. [2] makes also an exhaustive and coherent analysis and categorization of the features and modelling techniques used at present in different tasks of CASA. A first class of features, include the traditional ones, in time domain (Zero-Crossing Rate, energy, etc.), frequency (Fourier coefficients, power spectrum), and quefrency domains (cepstral coefficients), perceptual features (Perceptual Linear Predictive, Mel-Frequency Features), or features obtained by other processing (e.g., Linear Predictive Coding). A second category of features are the Auditory Image-based Features, in time-frequency domain, such as spectrogram-based features, applied widely in such techniques like CNN; These features include log-spectrograms, spectrograms on perceptual scales as Mel or Bark-frequency, Gabor Filter Bank (GBFB) features [14]wavelet packets [15], SIFs [8]. GBFB features represent the spectro-temporal modulation patterns of the signal using the Gabor filter bank; the authors in [16] proposed the Local Binary Pattern (LBP) descriptor approach for capturing the temporal dynamics of MFCC features. This approach evolved into different variants such as Variable-Q Transform (VQT) and Adjacent Evaluation Completed LBP. A third class of features are learning-based, i.e., the outcome of machine learning techniques used to enhance data representations. Such examples are i-vectors (based, for instance, on MFCCs), x-vectors (the features generated at some intermediary layer of a feed-forward network), features generated by exemplar coding [17], Sparse coding [18], Bag of aural words [19], non-negative matrix factorization (NMF) [20], etc.

In [2] the modelling methods are also organized into three classes. The first class contains the Genervative Model-based Approaches, where a model is built for each environment class, using samples belonging to that class alone. However, among various generative models, Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), Vector Quantization (e.g., k-NN), and their variances are the most widely used models in the EASR tasks [20] [21]. A second class are Discriminative Model-based Approaches, such as SVM and Artificial Neural Network (ANN), focused on constructing hyper-planes between environmental audio scene classes. SVM is a kernel-based discriminative classifier that focuses on modelling the decision boundaries between classes [22]. The SVM-based classifier performs well, not only for linearly separable data, but also for non-linear data

A third class are Deep Learning Model-based Approaches, with good performance in complex recognition task with more data where conventional machine learning methods do not guarantee better performance. DNN works on an unsupervised pre-training step using probabilistic graphical models to initialize the parameters. Convolutional Neural Network is one of the widespread architectures used in deep learning approaches. The DNN-based approach for acoustic scene recognition has been proposed by Petetin in [23] using MFCC, spectral centroid, and spectral flatness features, and outperformed the classical classifiers (GMM, SVM) with the same features. The results are exceptionally good for DNN with frequency features.

Public and comparative evaluation made on benchmark datasets help studying the performance of various proposed systems. There are few datasets publicly available for use in sound event recognition containing the corresponding sound event classes. In Environmental Sound Classification (ESC)-50 dataset, 50 sound classes are grouped into 5 major categories (10 classes per category), such as animal sounds, natural soundscapes, water sounds, human (non-speech) sounds, interior/domestic sounds, and exterior/urban noises. The dataset provides a variety of sound sources, such as common sounds (laughing, mewing, barking), distinct sounds (glass breaking, teeth brushing), and noisy sounds (helicopter and airplane). The ESC-10 dataset is a subset of the ESC-50 dataset. The examples are equally distributed among the 10 classes with 40 examples per class. CICESE dataset consists of 392 examples from 7 environment sound classes. The training set was composed of 4 different subjects with 10 examples per subject for each class. The test dataset was created with the same audio source with 4 samples per class from 4 different subjects. The DCASE 2013 dataset used in [19] for event detection consists of 3 subsets (development, training, and testing datasets). The extension of the DCASE 2013 dataset are DCASE 2016 and DCASE 2017 with a large number of classes and a diversity of data. The UPC-TALP dataset was recorded in a meeting room location. This dataset is multimodal (i.e., audio and video) and contains recordings of both isolated and spontaneous audio events. The MIVIA dataset contains highly noisy environmental sounds with events superimposed at different values of the SNR, making the detection and classification of events very challenging tasks. The intensity of the background sound is modulated to obtain low levels of the SNR and simulate events that occur at various distances from the microphone.

III. METHOD

The general framework we have applied is drawn on the ideas presented in [6]. The usual pre-processing of the environmental acoustic signal includes a framing step, possibly followed by sub-framing or sequential processing. In the “framing” stage the signal is processed continuously, frame by frame. A classification decision is made for each frame and successive frames may belong to different classes. Framing can enhance the acoustic signal classification by structuring the stream into more homogeneous blocks to better catch the acoustic event. Yet, there is no way of setting an optimal frame length, as for stationary events a length of 3s is a reasonable choice, while for acoustic events like thunder or gunshots, a 3s window length might be too large, and contain other acoustic events, so that they could be associated to inappropriate classes. Due to the latest advances in instrumentation, different frame lengths are used to streamline energy consumption during a monitoring process, based on detecting energy levels of environmental sounds. Next, a sub-framing process is applied, by dividing the frame into usually overlapping, analysis subframes. The length of a subframe is explicitly set in [6] to 20-30ms. This length is suited for speech analysis, as it ensures a good resolution in time and frequency,

as 20-30ms of male speech would include about three fundamental periods of the respective voice, whereas it might contain no period for a chainsaw sound. Therefore, we considered analysis sub-frames of 44ms or 88ms as a reasonable choice for chainsaw detection. A realistic setup must consider a value convenient to all sounds in the acoustic environment. We have used the above framework and applied on each analysis frame spectral analysis to extract frequency, quefrency or time-frequency features which will be fed to feed-forward and convolutional neural networks. We will detail subsequently.

A. *Feed Forward Networks*

The artificial neural networks (ANNs), intended to simulate human associative memory, learn by processing known input samples and the corresponding expected results, creating weighted associations between input and output, stored within the network data structure. Deep feedforward networks (FFNN) or multilayer perceptrons (MLPs), are the typical deep learning models [24]. The basic unit of a FFNN is the artificial neuron, analogous to the biological concept of neuron [25]. They receive input data, combine the input through internal processing elements (weights and bias terms), and apply an optional threshold using an activation (transfer) function. Transfer functions are used to map the output values usually between 0 to 1 or -1 to 1 or between 'yes' and 'no', and provide a smooth, differentiable transition as input values change. Transfer function are linear and non-linear. Non-linear transfer functions are "S" – shaped functions like arctg, hyperbolic tangent, logistic functions, competitive, .Elliot sigmoid transfer function, positive and symmetric hard limit, Radial basis transfer function, softmax.

A feedforward network defines a mapping between the input and output $y = f(p,\theta)$ and learns the value of parameter θ that ensures the best approximation of the expected value y by the output of f, given the input p and parameters θ. FFNNs have one or more hidden layers of "S" – shaped neurons followed by an output layer of linear neurons. A layer of neurons brings together the weight vectors and biases corresponding to its neurons, so it can be expressed by a matrix of weights and bias vectors, as in figure 1.

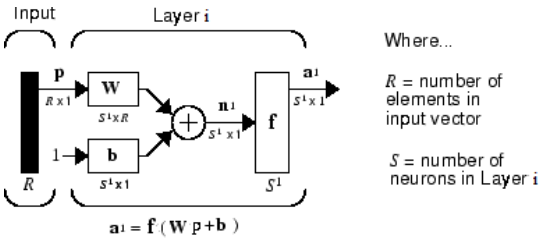

Fig. 9. Structure of a layer of neurons

The transfer function is supposed to be the same for each neuron in the layer. The general diagram of a network is shown in figure 2, where the parameters to be tuned are the weight matrices and bias terms applied at the level of each layer, so that the output of the overall system would be close to expected values. These networks are called feedforward because the information flows in one direction through intermediate computations and there is no feedback connection. The number of neurons does not necessarily decrease with the layer level as presented in figure 10, but usually the goal is to reduce the dimensionality of the input layer, a process similar to feature extraction. The computation corresponding to figure 3 can be expressed by :

$$a^k = f^k(W_k a^{k-1} + b_k) =$$
$$f^k\left(W_k\left(f^{k-1}(W_{k-1}a^{k-2} + b_{k-1})\right) + b_k\right) = \quad (8)$$
$$f^k\left(W_k\left(f^{k-1}(W_{k-1}(f^{k-2}(W_{k-2}a^{k-3} + b_{k-2})) + b_{k-1})\right) + b_k\right) = \cdots$$

In equations (1) the known information is

- The input parameters p, e.g., measurements from sensors (wind speed, temperature, humidity), parameters coming from images (matrices of colours, or grey hues), or parameters coming from acoustic signals (Fourier spectrum on an analysis window, etc.);

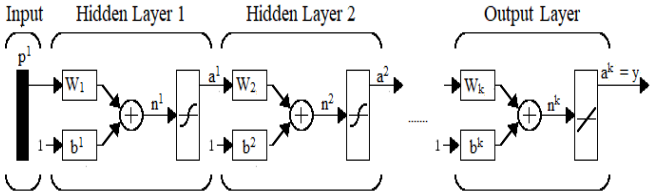

Fig. 2. Flow of data in a feedforward network

- The expected output: for instance, to solve a three classes problem the output corresponding to each class input might be defined as either unidimensional (a scalar value for each class): for instance, (-1, 0, 1) or (0, 1. 2) or multidimensional (a vector for each class):((1, 0, 0), (0, 1, 0), (0, 0, 1));
- The neural network architecture: number of hidden layers, number of neurons on each layer, transfer function.

Unknown parameters are:

- weights at layer k: W_k,
- bias terms at layer k: b_k.

Learning the unknown parameters is performed during the training process. Training of a FFNN can be made in batch mode or in incremental mode [38]. In batch mode, weights and biases are updated after all the inputs and targets are presented. Incremental networks receive the inputs one by one and adapt the weights according to each input. Usually, batch training is used. Solving equations (8), to identify weight matrices and bias terms is made by minimizing the error between the output value and expected output, minimal:

$$e(W, b) = \sum_{i=1}^{N}(y_i - a_i^k(p, W, b))^2 \quad (2)$$

where N is the number of (input, output) pair samples. To minimize the least mean square (LMS) expression in (2) several schemes based on variants of the steepest descent procedure, are used. MATLAB has implemented and supports a range of network training algorithms among which: Levenberg-Marquardt Algorithm (LMA), Bayesian Regularization (BR), BFGS Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, One Step Secant, etc. Minimization using any of these algorithms, starts an initial guess for the parameter vector θ=(W, b). The

performance of the system depends on this initial guess. Most of the above algorithms try to optimize this process.

At the end of the training process, we get a FFNN model: $net = (W_k, b_k)$, $k= 1,2...K$, where K is the number of layers in the network. To classify a vector of data $x = \{x_1, x_2, \ldots, x_d\}$, we "feed" it at the input of the network, apply the operations involving the weights and biases to the input data, and evaluate the output $score = net\ (x)$. If the output classes are $y = \{y_1, y_2, \ldots, y_C\}$, C the number of classes, a sample belongs to a certain class if its output *score* is closest to the respective class expected output. The overall decision on the 3s frame level is taken by applying one of the rules:

- Majority voting (the segment is associated with the class for which most of the samples of the segment belong to the respective class);
- Average output: the average output score of the samples on the segment is closest to the expected output of a certain class;
- Minimum distance of the segment scores to the ideal segment of expected outputs of the respective class.

B. *Convolutional Neural Networks (CNN)*

A Convolutional Neural Network (CNN) is a Deep Learning algorithm usually taking at input images, assign meaning by learnable weights and biases to various parts in the image. The pre-processing required in a CNN is lower compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNNs are able to learn these filters characteristics.

In CNNs architecture, inspired by the organization of the Visual Cortex, individual neurons respond to stimuli only in a restricted region of the visual field called the Receptive Field. A collection of overlying fields covers the whole visual area.

The CNN architecture is presented in figure 3.

The convolutional layer is represented by moving smaller size filters (kernel) K, shifting at a certain rate (stride) and performing at each pace a matrix multiplication between K and a portion of the image over which it is hovering. The objective of the convolution operation is to extract high-level features (e.g., edges), from the input image. The Convolutional Layer and the Pooling Layer, together form one layer of a CNN. Depending on the complexities in the images, the number of such layers may be increased to capture further low-level details, but at the cost of more computational power.

CNNs need not be limited to only one convolutional layer. Conventionally, the first layer is responsible for capturing the low-level features (edges, colour, gradient orientation, etc.). By adding layers, the architecture adapts to the high-level features, generating a network with wholesome understanding of images in the training set. There are two types of results of the operation: one in which the convoluted features are reduced in dimensionality as compared to the input (Valid padding), another where the dimensionality is either increased or remains the same (Same padding).

Rectified linear unit (ReLU), referred to also as activation, allows for faster and more effective training by mapping negative values to zero and sustaining positive values. Only the activated features are carried further into the next layer.

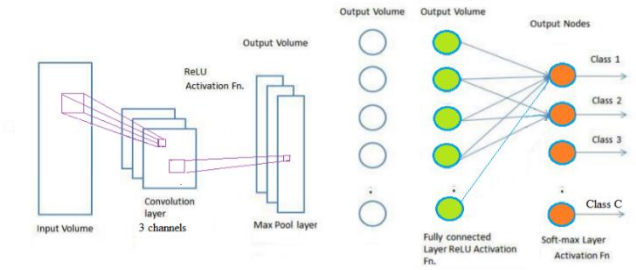

Fig. 3. Flow of data in a convolutional neural network

The pooling layer is responsible for reducing the spatial size of the convolved features. It is useful for extracting dominant features, rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of pooling: Max pooling and Average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel. On the other hand, Average pooling returns the average of all the values from the portion of the image covered by the kernel. Max pooling performs de-noising along with dimensionality reduction while average pooling simply performs dimensionality reduction as a noise controlling mechanism.

The convolutional layer and the pooling layer form the *i*-th layer of a CNN. Depending on the complexities in the images, the number of such layers may be increased for catching low-level details, but at the cost of more computational power [9] [26].

After learning features in many layers, the architecture of a CNN shifts to classification. Once the input image is converted into a suitable form for our Multi-Level Perceptron the image is flattened into a column vector, which is fed to a feed-forward neural network and backpropagation applied to every iteration of training. The last layer is a fully connected layer that outputs a vector of C dimensions where C is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified. The final layer of the CNN architecture uses a classification layer such as softmax to provide the classification output.

As this approach was initially devised for image classification, it was further adapted to be applied in acoustic signal applications by operating on a spectrogram-like data derived from the acoustic signal.

IV. Experimental Results

This section will reopen and extend the research presented in [27]. We have tested the FFNN on the acoustical material containing logging activities events in forest environment. We have tested the behaviour of FFNNs when feeding at input Mel-cepstral coefficients and Fourier power spectrum coefficients, using several sizes of the analysis frame, and different lengths for the frequency domain. The experiments employed the Matlab framework, and we have taken the advantage of the training solutions implemented by Matlab, by using two training algorithms: Bayesian Regularization (BR) and the Levenberg-Marquardt algorithm (LMA). We will extend te research by including the Broyden–Fletcher–Goldfarb–Shanno (BFGS Quasi-Newton) algorithm. We also will study the influence of the activation function on the evaluation results, by assessing the *logsig*, *tansig*, and *softmax*. For classification we used the three approaches

presented above: majority voting, average score, and distance based (equivalent to computing the distance of the scores for one segment to the ideal segment represented by ideal outputs for each class). In the pre-processing phase we evaluated segments of 3s, and used analysis windows of 44 and 88ms, as previous experiments have shown that for such analysis frames the performance is better than for 22ms windows. Regarding the type of features we fed as input to FFNNs, we only used the spectral features, as the results obtained previously are much better than using Mel-frequency cepstral features. Concerning the frequency domain, we have tested the ranges of [0, 3.7], [0,7.4], [0,10], [0,12]kHz. As the performance depends on the initial guess at training, we have performed 5 trials for each test.

In a second series of experiments, we applied the CNN framework with inputs log-Mel Spectrograms and, sheer log--spectrograms also using the Matlab framework.

The experiments considered three classes of sounds which could exhaust the specific sounds in the forest environment susceptible to illegal deforestation: chainsaw, vehicle, genuine forest. The expected values were coded 1, 0, -1 respectively.

The acoustic material contains 99 recordings of the three classes of sounds, in average about 15s each, 39 were used for training and 60 for testing. The testing set resulted in 685 segments of three seconds. The performance of each of the approaches we tested is presented subsequently. The performance was evaluated in terms of Identification rate, the ratio of numbers of correctly identified segments and the evaluated segments.

A. *Results obtained using FFNNs*

Table 1 presents the average identification rates (of the 5 trials) obtained using the three approaches for classification, *maxvote*, *average* (avg), and distance-based, for the four level (of 10, 9, 8, 7 neurons respectively) FFNNs, on log-power spectra, trained using the BR, LMA and BFGS Quasi-Newton algorithms, with the default activation function *tansig*. The results obtained on frequency domains of [0, 3.7], [0,7.4], [0,12]kHz , and analysis lengths of 44ms, and 88ms. The best results were achieved using the *maxvote* classification, with the BR initialization for frequency domains [0, 3.7] and [0,7.4] kHz, and are almost equally good for 44ms. and 88ms analysis windows. Figure 4 is the graphical representation of these results. The other 2 classification approaches generated results about 8 percent below, in any configuration.

TABLE I. AESR PERFORMANCE USING FOUR LEVEL (O10, 9, 8, 7 NEURONS RESPECTIVELY) FFNNS TRAINED WITH THE BR, LMA AND BFGS QUASI-NEWTON ALGORITHMS, THE ACTIVATION FUNCTION *TANSIG*, FREQUENCY INTERVALS AND ANALYSIS WINDOWS OF DIFFERENT LENGTHS

		Frequency domain and analysis window				
		3.7kHz -44ms	*7.4kHz -44ms*	*3.7kHz -88ms*	*7.4kHz -88ms*	*12kHz-88ms*
maxvote	*trainbr*	79.65	79.65	76.40	80.70	76.93
	trainlm	74.79	75.81	79.18	76.84	73.12
	trainbfg	71.45	73.68	76.31	74.73	73.82
avg	*trainbr*	68.46	68.52	69.96	69.19	64.60
	trainlm	69.66	68.73	72.56	72.42	65.33
	trainbfg	67.47	66.15	72.62	70.04	67.38
distance	*trainlm*	68.67	68.08	71.30	71.30	65.59
	ttainbr	68.55	69.78	70.16	70.60	66.85

Another range of experiments tried to evaluate the performance for different activation functions, available in the Matlab framework. We have evaluated the performance of 4-layer FFNNs, with 10,9,8,7 neurons respectively, using the LMA and BR training algorithms, and *tansig*, *logsig*, and *softmax* activation functions. Classification of 3s segments was made using the *maxvote* approach. The average, maximum, minimum score values obtained for some relevant settings of the analysis window and the frequency domain on which the Fourier power spectrum was calculated, are presented in figures 4 and 5, for the LMA and BR training approaches. In average *logsig* worked better with the LMA, as average performance, but a higher score variance among the 5 trials, while *tansig* was better with the BR approach. The teste *softmax* also generated very good results.

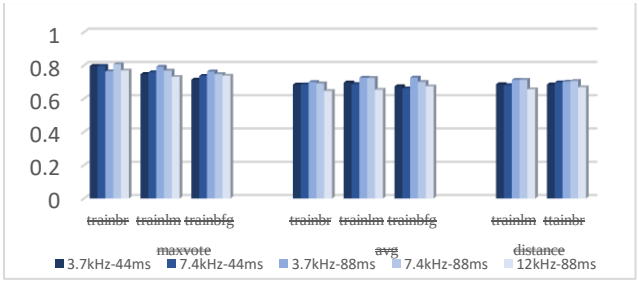

Fig. 3. Performance using FFNN with the Fourier Power Spectrum as input and different training and classification settings.

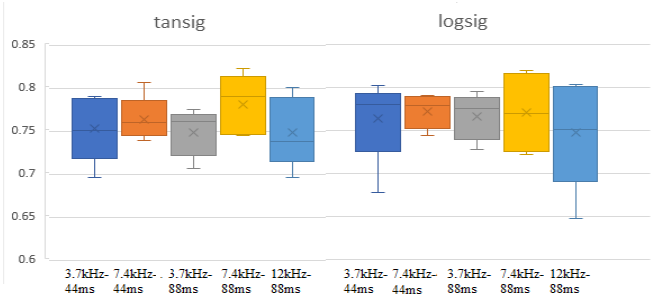

Fig. 4. Average, maximum. minimum scores, using the LMA training algorithm and *tansig*, and *logsig* activation functions.

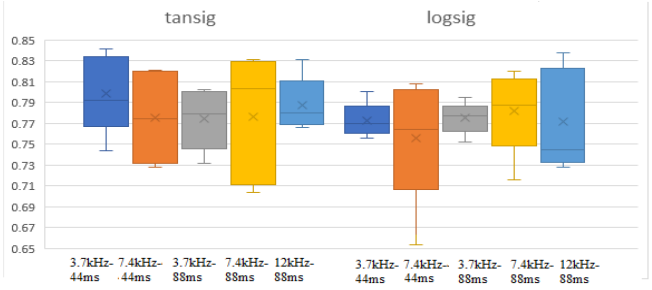

Fig. 5. Average, maximum. minimum scores, using the BR training algorithm and *tansig*, and *logsig* activation functions.

B. *Results obtained using CNNs*

We have tested the CNNs, using as input log-Mel-spectrograms, and log-spectrograms. Concerning the setting in the Matlab framework we have, on one hand, stuck to our earlier findings concerning the length of the analysis window (between 44ms and 88ms) and the frequency domain (3.7 – 7.4kHz) where the spectrograms are calculated. Concerning the network architecture, we got inspiration from the experiment description in [1] [9]. In the case of Mel-spectrograms we used 64-128 bands, depending on the analysis window length (2048-4096 samples), Given the size of the Mel-spectrogram (128·64), we set the what we considered the proper dimensions in the definition of the three-four layers. However, the best performance was under 60% (58,6).

Better identification rate was attained using Log-Spectrograms as input, 66.53%. with a three-level CNN architecture, with 16, 32, 64, 128 sized kernels, and 40 to 10 receptive fields, 2x2 Max pooling. Using more than 200 iterations seems useless as there might not be observed any ascending trend in the training process in Figure 6.

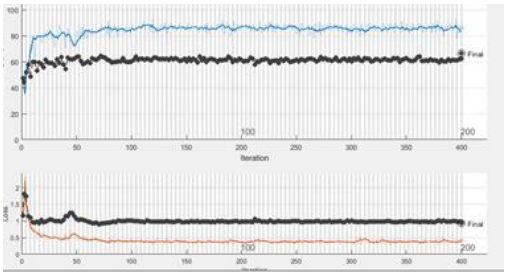

Fig. 6. Training process of a CNN with Log-spectrograms as input.

CONCLUSIONS

We have presented our research regarding the application of DNN approaches in environmental sound recognition. One part of the work reviews and deepens the previous investigation of the FFNNs facets, by examining different training approaches, activation functions, classification variants, in the context of the advantageous settings for the analysis window length and the frequency domain, established previously. The *maxvoting* classification scheme of the 3s frames was the most efficient, as for several training paradigms the attained performance is above 80%, and the average performance is in many cases 79%. Concerning the CNN experiments we have not found yet a reasonable configuration to provide satisfactory results. Another possible explanation is the fact that the CNN approach was demonstrated to provide very good results in SER experiments, i.e. acoustical event recognition, while our acoustic material contains stationary sounds.

ACKNOWLEDGMENT

This work has been supported by a grant of the Ministry of Innovation and Research, POC-5C project.

REFERENCES

[1] R. Ahmed, T. I. Robin, A. A. Shafin, "Automatic Environmental Sound Recognition (AESR) Using Convolutional Neural Network," International Journal of Modern Education and Computer Science, **12**(5), 41-54, 2020, doi: 10.5815/ijmecs.2020.05.04

[2] S.Chandrakala, S. Jayalakshmi, "Environmental Audio Scene and Sound Event Recognition for Autonomous Surveillance: A Survey and Comparative Studies," ACM Computing Surveys, 52(3), 1-34, 2019, doi: 10.1145/3322240.

[3] F. Al Hossain, A. A. Lover, G. A. Corey, N. G. Reich, T. Rahman, "FluSense: A Contactless Syndromic Surveillance Platform for Influenza-Like Illness in HospitalWaiting Areas", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 4, No. 1. March 2020.

[4] Xiao Sun, Zongqing Lu,Wenjie Hu, Guohong Cao, "SymDetector: Detecting Sound-Related Respiratory Symptoms Using Smartphones", the 2015 ACM International Joint Conference, September 2015.

[5] V. Gibiat, E. Plaza, P. De Guibert, "Acoustic emission before avalanches in granular media," The Journal of the Acoustical Society of America, **123**(5), 3270, 2008,doi: 10.1121/1.2933600

[6] S. Chachada, C.-C. Jay Kuo , "Environmental Sound Recognition: A Survey," APSIPA Transactions on Signal and Information Processing, 3, 2014, doi: 10.1109/APSIPA.2013.6694338.

[7] Cowling M, Sitte R, "Analysis of Speech Recognition Techniques for use in a Non-Speech Sound Recognition System", Advanced Signal Processing for Communication Syste ms. Springer, 2002

[8] Ian McLoughlin, H. Zhang, Z. Xie, Yan Song, Wei Xiao., "Robust sound event classification using deep neural networks". IEEE/ACM Trans. Aud., Speech, Lang. Proc. 23, 3 (2015), 540‑552.

[9] K. J. Piczak, "Environmental sound classification with convolutional neural networks", Proc. of the 25th IEEE Intern Workshop on Machine Learning for Signal Processing (MLSP'15) 1‑6.

[10] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen., "Convolutional recurrent neural networks forpolyphonic sound event detection", IEEE/ACM Trans. Aud., Speech, Lang. Proc. 25, 6 (2017), 1291‑1303

[11] Ting-Wei Su, Jen-Yu Liu, Yi-Hsuan Yang., "Weakly supervised audio event detection using event-specific Gaussian filters and fully convolutional networks", Proc. of the IEEE International Conf. on Acoustics, Speech and Signal Process/ (ICASSP'17). IEEE, 791‑795.

[12] I. Ozer, Z. Ozer, O. Findik, "Noise robust sound event classification with convolutional neural network",Neurocomp. 272 (2018), 505‑512.

[13] Chien-Yao Wang, Jia-Ching Wang, Andri Santoso, Chin-Chin Chiang, Chung-Hsien, "Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network", IEEE/ACM Trans. Aud., Speech, Lang. Proc. 26, 8 (2018), 1336‑1351.

[14] J. Schroder, N. Moritz, M. R. Schadler, B. Cauchi, K. Adiloglu, J. Anemuller, S. Doclo, B. Kollmeier, S. Goetze. "On the use of spectro-temporal features for the IEEE AASP Challenge on "Detection and Classification of Acoustic Scenes and Events." , Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'13). IEEE, 1‑4.

[15] A. Rabaoui, M. Davy, S. Rossignol, N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance", IEEE Trans. Inform. Forens. Sec. 3, 4 (2008), 763‑775.

[16] W. Yang, S. Krishnan, W. Yang, S/ Krishnan, "Combining temporal features by local binary pattern for acoustic scene classification", IEEE/ACM Trans. Aud., Speech, Lang. Proc. 25, 6 (2017), 1315‑1321.

[17] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, et al., "An exemplar-based NMF approach to audio event detection", Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'13). IEEE, 1‑4.

[18] X. Lu, Yu Tsao, S. Matsuda, C. Hori, "Sparse representation based on a bag of spectral exemplars for acoustic event detection". Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14). IEEE, 6255‑6259.

[19] A. Plinge, R. Grzeszick, G. A. Fink, "A bag-of-features approach to acoustic event detection", Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14). 3704‑3708.

[20] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, M. D. Plumbley, "Detection and classification of acoustic scenes and events", IEEE Trans. Multimed. 17, 10 (2015), 1733‑1746.

[21] S. Chandrakala, C. C. Sekhar, "Classification of varying length multivariate time series using Gaussian mixture models and support vector machines", Int. J. Data Mining, Modell. Manag. 2, 3 (2010), 268‑287.

[22] Vladimir Vapnik, "Statistical Learning Theory", Vol. 3. Wiley, New York, 1998.

[23] Y. Petetin, C. Laroche, A. Mayoue, "XDeep neural networks for audio scene recognition", Proceedings of the 23rd European Signal Processing Conference (EUSIPCO'15). 125‑129.

[24] V. Zacharias, AI for Data Science Artificial Intelligence Frameworks and Functionality for Deep Learning, Optimization, and Beyond, Technics Publications, 2018.

[25] M. H. Beale, M. T.Hagan, H. B. Demuth, Neural Network Toolbox™ User's Guide, The MathWorks, Inc., Natick, Mass., 2010.

[26] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", available at https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[27] S.Segarceanu, G.Suciu, I.Gavat, "Environmental Acoustics Modelling Techniques for Forest Monitoring", ASTESJ Journal, **6**(3), 15-26 (2021).*