DEEP NEURAL NETWORKS FOR AUTOMATIC DETECTION OF SCREAMS AND SHOUTED SPEECH IN SUBWAY TRAINS

Pierre Laffitte, David Sodoyer, Charles Tatkeu

Univ Lille Nord de France, Lille
IFSTTAR, COSYS, LEOST, Villeneuve d'Ascq

Laurent Girin

GIPSA Lab, Univ. Grenoble Alpes
INRIA Grenoble Rhône-Alpes, France

ABSTRACT

Deep Neural Networks (DNNs) have recently become a popular technique for regression and classification problems. Their capacity to learn high-order correlations between input and output data proves to be very powerful for automatic speech recognition. In this paper we investigate the use of DNNs for automatic scream and shouted speech detection, within the framework of surveillance systems in public transportation. We recorded a database of sounds occurring in subway trains in real conditions of exploitation and used DNNs to classify the sounds into screams, shouts and other categories. We report encouraging results, given the difficulty of the task, especially when a high level of surrounding noise is present.

Index Terms— scream detection, audio event detection, transport environment, Deep Belief Networks.

1. INTRODUCTION

The automatic analysis of auditory scenes, i.e. detection and classification of audio events or audio context, is a growing topic of research [1, 2, 3, 4, 5, 6]. For example, smart house concepts are currently being developed, involving automatic systems for domestic events detection using audio and video data streams [7, 8]. In humanoid robotics, an audition model is also a prerequisite for natural human-robot interaction [8, 9]. As for the detection techniques, the state of the art is rich, with many different combinations of features and classifiers: MFCC classified with GMM [10], with SVM [8, 11, 12], with HMM [13], or with multiclass Adaboost [14]; MFCC and other spectral features classified with GMM [15, 1], with kNN [1], and more recently with random forest [16]; Gabor features classified with GMM [15, 17] and with SVM [7] ; all-pole group delay features classified with DNN [6]; Gammatone-Wavelet features classified with SVM [18]; spectrogram image features classified with SVM and DNN [5], or with kNN [3].

The proposed study is part of a research project on auditory scene analysis in the embedded transport environment, for instance subway trains. This kind of context has already lead to different studies, e.g. [19] for bus transportation, [20] for trains, and [21] for subway. In the present paper, the task focuses on automatic detection of screams and shouted speech, produced in abnormal situations in the subway train: a person in physical difficulty, a quarrel between two or more persons, panic situations, calls for help, etc. Although such kind of alert signals are quite specific, this task remains challenging, for several reasons. First, there exists a large variability between "speakers", and also between different realizations of screams and shouts, depending on the causing event, the emotional state, etc. Second, the embedded transport environment is in general very noisy, rich, and strongly variable. In the present case, the acoustic scene includes noise coming from the vehicle itself (e.g., motor noise, boogie-rails frictions), noise coming from the surrounding environment (e.g., railway traffic, station noise, loud-speaker announcements), and noise produced by the passengers.

In order to design a realistic system, a dedicated database was designed and recorded. This database consists of real signals recorded in the Paris metro. A metro line was booked for the recording sessions, thanks to the Paris metro company (the RATP) being a partner of the project. Abnormal situations were enacted by actors, including extra participants representing a crowd. As a consequence all the recordings used in the present study are real and not derived from synthetic signals or simulated acoustic mixes. As for the classifiers, we used state-of-the-art Deep Neural Networks (DNNs), for instance a combination of Restricted Boltzman Machines (RBMs) and Deep Belief Networks (DBNs), applied on acoustic MFCC features. We set the task as a 4-class classification problem into screams, shouted speech, conversational speech and noise environment.

The remaining of this paper is organized as follows: Section 2 introduces the basic concepts of RBMs and DBNs. Section 3 gives a detailed description of the database recording. Section 4 reports our experiments and results. Finally Section 5 draws some conclusions and perspectives.

2. DEEP NEURAL NETWORKS

Deep Neural Networks intend to reproduce the mechanism with which the human brain processes information. It involves a network of individuals cells, called units. The term

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

"deep" owes to the fact that the cells are organized in multiple layers stacked onto each other, forming a deep architecture [22]. Conceptually, the units represent hidden causes or factors to the input data. Their output represents the probability of the associated factor to have caused the occurrence of the input data. In the present study, we use RBMs stacked into DBNs, and turned into DNNs, as described in detail in [23]. The main characteristics of such DNN are briefly given below.

2.1. Restricted Boltzman Machines

RBMs are stochastic neural networks with one layer of hidden units which have undirected connections with visible units. The interaction between visible and hidden units is modelled through an Energy function, which associate a scalar energy to each configuration of visible-hidden variables, given by:

$$E(v,h;\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_ih_j + \frac{1}{2}\sum_{i=1}^{V} b_iv_i - \sum_{j=1}^{H} a_jh_j \quad (1)$$

where w_{ij} are the weights between the visible v_i and the hidden h_j units, V is the number of visible units and H is the number of hidden units, b_i and h_j are bias terms, and θ represents the parameters of our model. The joint distribution over the visible units v and the hidden units h is given by:

$$p(v,h;\theta) = \frac{exp(-E(v,h;\theta))}{Z} \quad (2)$$

where Z is a normalization factor. The probability over the visible units is obtained by marginalizing out the hidden units:

$$p(v;\theta) = \frac{\Sigma_h exp(-e(v,h;\theta))}{Z} \quad (3)$$

Because the hidden units are defined to be independent in the posterior, the posterior distribution in an RBM is factorial and can be calculated easily:

$$P(h_j = 1 \mid v) = \frac{1}{1 + exp(\Sigma_{i=1}^{V} v_iw_{ij} + a_j)}, \quad (4)$$

Similarly, since the visible and the hidden units play a symetric role in the energy function, the posterior probability can be derived for Gaussian visible units:

$$p(v_i = 1 \mid h) = \mathcal{N}(\Sigma_{j=1}^{H} w_{ij}h_j + b_i, 1). \quad (5)$$

In order to learn the parameters θ of the model, we adopt the greedy learning method proposed in [24] where the layers of the network are learned one by one, freezing the weights of all the layers that have already been learned.

2.2. Deep Belief Networks

DBNs are composed of stacked RBMs which learn in turn different levels of correlation between the input data. The first RBM learns a hidden representation of the data, and each RBM after that one takes the hidden layer of the previous one and learns a hidden representation of it. In the end, the deeper layers represent more abstract concepts, or features, associated with the input data.

2.3. Discriminative fine-tuning

Our goal is to classify sounds in order to detect screams and shouts, which means we need to create a model that performs well at classifying. Therefore, one solution is to turn our DBN, after we have learn its parameters, into a DNN. Because DNNs are discriminative classifiers, they learn to model the frontier between classes, through a discriminative rule based on gradient descent and error propagation. The most common algorithm used in DNN training is called the "back-propagation" algorithm introduced in [25]. Practically, a last layer of N sigmoid units is added to perform the calculation of the probability for the input to belong to a class, N being the number of classes we want to classify. Then we feed our generatively initialized network with labelled data, and use the back-propagation algorithm to minimize the classification error.

3. DATABASE RECORDING

Our study addresses a specific detection task in a specific environment, thus we collected audio data recorded directly in that environment. Within the framework of the research project DéGIV [21], a subway train from the automatic line 14 of the Paris metropolitan was reserved for the recording sessions. We did several sessions between 10am and 4pm while the train was running its usual course, among other trains from the same subway line, running between different stations and stopping at all of them. For security matters, the train did not allow any regular passengers in, and 3 sets of actors played several pre-defined scenarios displaying a situation of security matter (robbery, assault, fight). Numerous extra actors simulated regular passengers. For data recording, 2 microphones were placed on the ceiling about 10cm from one another, along with a video camera. Different settings were defined in which each scenario was played. Each scene was played in 2 different zones which were defined according to their distance to the microphone: close-distance (1m to 1.5m) and far-distance (3m to 4m). 2 crowd densities were used: heavy density (between 12 and 17 people involved in the scene) and low density (between 5 and 7 people). Every alert scenario was repeated for each combination of settings. A large amount of sequences of chatter among passengers were captured. Scenes were captured while the train was either accelerating, moving at stationary high speed, or braking, and when it was stopped at a station. Sequences of door openings and closings were captured as well. Therefore our database contains all the noises induced by the train activity (door opening, brakes compressor draining air, doors closing signal, etc.), and is very noisy. For example the level of the noise created by the air flow inside the tunnel is comparable to the level of a loud conversation. To display this, we calculated signal power ratios. We observed a typical level difference between sequences with shouts and sequences without shouts (noise only) of about 8dB when the train was in full motion, and about 62dB when it was stopped. The noise level difference between those two conditions is thus about 54dB. In such a noisy environment any type of classification is difficult to perform. Finally, we believe that the recorded database exhibits a realistic diversity of signal occurrences for the considered application (different scenarios, different source to microphone positions, different noises, etc.).

4. EXPERIMENTS

4.1. Data

The data was manually cross-labelled by two different persons (i.e. the first person labelled the entire set and the second did a verification-labelling) distinguishing between 4 different categories of sound:

- Scream: very loud (and often high-pitched) vocalizations, with no lexical content;
- Shout: loud speech signal occurring in security alert situations (fight, assault, accident, etc);
- Conversation: speech signals from normal conversation between people;
- Noise: all noises related to the train and its motion (bell ringing, brakes squeaking, etc), without speech content.

It is important to note that the Conversation, Scream and Shout classes can contain a lot of noise. The difference between these classes and the Noise class is thus the absence of speech/vocal signals for the latter, besides that the background is often similar.

We split our database in two different sets: a training-validation set and a test set. All following scores are calculated on the test set. At the time we ran the experiments, the training set was composed of 33s of class Scream, 292s of Shout, 997s of Conversation and 1429s of Noise. The test set was composed of 17s of class Scream, 92s of Shout, 207s of Conversation and 396s of Noise. One could observe that our dataset displays a fairly high disproportion between classes; Scream and Shout are quite under-represented compared to other classes. Such a large difference in the number of data points of each class in the learning dataset is a known issue in the machine learning literature where it is referred to as class imbalance. However our situation is a very specific case scenario, wherein the classes are naturally imbalanced. In fact, Scream and Shout occurrences are even more imbalanced (under-represented) in real life than in our dataset. We haven't conducted experiments to address this problem extensively in our study. Simply, we purposely decided to create a dataset with more occurrences of the rare classes (Shout and Scream) than would be found in a real environment in order to improve the modeling abilities of our system. Indeed, the more occurrences the more accurate the modeling. In short, we tried to find a trade-off between (limiting the) over-representation of Shout and Scream in our dataset w.r.t. reality, and (limiting) their under-representation w.r.t. other classes. Since Neural Networks do not embed information about the frequency of occurrence of classes in the learning dataset as HMMs would do for instance through the prior probability of each class, we believe that having such controled unrealistic amount of data points for one class does not affect much the recognition process.

4.2. Features

We used 12-th order MFCC coefficients plus an energy term. The MFCC coefficients were calculated every 10ms with a 25ms window. To account for the temporality of the data, we concatenated every 10 consecutive frames of MFCC + energy vectors to form an input vector to our network, It should be noted that only the left channel from the stereo recordings was used to calculate the features and process classification.

4.3. Classifier

We used the Python machine learning library 'pdnn' [26] which is itself based on the 'theano' framework [27], to implement our DBN-DNN networks. This library allows one to learn the parameters of a DBN on a dataset, then turn this DBN into a DNN to discriminatively fine-tune those parameters, as briefly described in Section 2. The DNN is trained discriminatively with the same data used to train the DBN. We ran experiments with a network of three 512-unit layers, with 300 epochs for the DBN and 200 for the DNN, which is the configuration that yielded the best classification results on our data. The number of epoch was chosen after running multiple tests while increasing it, stopping when the delta on the validation error was close to zero.

4.4. Tests configuration

Preliminary tests have revealed that screams are easier to detect than shouts. Our main goal being to detect situations of danger, we thus first report classification between two classes: one comprised of the union of Scream and Shout, representing abnormal situations, and the other comprised of the union of two other categories (Conversation and Noise), representing normal situations. Then, to challenge our model, we run classification between Shout and Noise (still considering that screams are easier to detect than shouts). Finally, we report the results for a 3-class run, Shout vs. Conversation vs. Noise.

	Everything else	*Shout + Scream*
Everything else	**97.0**	17.8
Shout + Scream	3.00	**82.2**

Table 1. Confusion Matrix. Shout+Scream vs. Everyth. else.

	Noise	*Shout*
Noise	**96.8**	20.8
Shout	3.20	**79.2**

Table 2. Confusion Matrix. Noise vs. Shout.

4.5. Results

The results for the recognition of Scream and Shout vs. all other categories were quite good: we achieved an error rate of 6.2% (see the confusion matrix in Table 1). This error rate takes into account all errors made during the classification process, that is, every data which was classified as something else than its label counts as an error, and it also takes into account the difference of cardinal of occurrences across classes. For the case Shout vs. Noise, we achieved an error rate of 6.5%, with proper detection of Noise as high as 96.8% and proper detection of Shout of 79.2% (see the confusion matrix in Table 2). Even though they seem to be harder to detect than screams, shouts are quite well discriminated in noise. Assuming that Shout occurrences are more similar to other classes such as Conversation than Scream occurrences, we decided to see how it would compare against it. For the 3-class Shout-Conversation-Noise problem, we obtained an error rate of 28.7%, (see the confusion matrix in Table 3. We notice that 21.6% of Conversation occurrences are classified as Noise, and 34.8% of Shout occurrences are classified as Conversation. This can be explained by two different reasons. First, about one third to one half of the occurrences in our dataset contain a quite high level of noise, which makes the classes distribution overlap in the feature space. Also, some of our data are mixed: some occurrences can contain both shouts and conversation. The difference between the Shout and Conversation classes actually lies on a conceptual level, and is sometimes difficult to tell even for a Human expert, which can explain why those two classes are relatively confused by our classifier.

5. CONCLUSION AND PERSPECTIVES

As mentioned just above, one of the main problems in the present application is that our data is mixed. It contains occurrences belonging to different classes at the same time. This could be seen as a source separation problem. Therefore integrating some sort of pre-processing source separation algorithm could be helpful. Our next steps may include increasing the number of features in a way that can help our network in its classification task. Then we should further improve the way temporal structure of the data is dealt with in our model. One idea is to interface Hidden Markov Models (HMMs) with our neural networks. Finally, we will try to increase the number of classes to detect a larger and more specific set of events (siren, doors closing, etc.)

	Noise	*Conversation*	*Shout*
Noise	**77.0**	21.6	7.20
Conversation	19.5	**66.1**	34.8
Shout	3.50	12.3	**58.0**

Table 3. Confusion Matrix. Noise vs. Conversation vs. Shout.

6. ACKNOWLEDGMENTS

The recording sessions has been supported by the research project DéGIV [21] co-funded by ”FUI-BPI France” and ”Conseil Général de l’Essonne” and by the research project Secur-ED funded by European Commissions FP7 program (GA no.261605).

7. REFERENCES

[1] S. Chu, S. Narayanan, and C.-C.J. Kuo, “Environmental sound recognition with time-frequency audio features,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[2] M. Crocco, M. Cristani, A. Trucco, and V. Murino, “Audio surveillance: a systematic review,” *arXiv preprint arXiv:1409.7787*, Oct. 2014.

[3] J. Dennis, Huy D. T., and Eng S. C., “Image feature representation of the subband power distribution for robust sound event classification,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 367–377, Feb. 2013.

[4] M. Fernández-Delgado, E. Cernadas, S. Barro, and D., “Do we need hundreds of classifiers to solve real world classification ?,” *J. of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[5] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, “Robust sound event classification using deep neural networks,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, Mar. 2015.

[6] A. Diment, E. Cakir, T. Heittola, and T. Virtanen, “Automatic recognition of environmental sound events using all-pole group delay features,” in *European Signal Processing Conference*, Nice, France, Aug. 30 - Sept. 4 2015, pp. 734–738.

[7] J. Wang, C. Lin, B. Chen, and M. Tsai, "Gabor-based nonuniform scale-frequency map for environmental sound classification in home automation," *IEEE Trans. on Automation Science and Engineering*, vol. 11, no. 2, pp. 607–613, Apr. 2014.

[8] X. Wu, H. Gong, P. Chen, Z. Zhong, and Y. Xu, "Surveillance robot utilizing video and audio information," *J. of Intelligent and Robotic Systems*, vol. 55, no. 4, pp. 403–421, 2009.

[9] M. Janvier, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound-event recognition with a companion humanoid," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 104–111.

[10] J. Pohjalainen, T. Raitio, and P. Alku, "Detection of shouted speech in the presence of ambient noise," in *Interspeech*, Florence, Italy, Aug. 28-31 2011, pp. 2621–2624.

[11] W. Huang, T.-K. Chiew, H. Li, T. S. Kok, and J. Biswas, "Scream detection for home applications," in *IEEE Conference on Industrial Electronics and Applications*, Taichung, Taiwan, 2010, pp. 2115–2120.

[12] B. Lei and M. Mak, "Sound-event partitioning and feature normalization for robust sound-event detection," in *Int. Conf. on Digital Signal. Processing*, Hong Kong, Aug. 20-23 2014, pp. 389–394.

[13] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "On acoustic surveillance of hazardous situations," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 19-24 2009, pp. 165–168.

[14] Y. Lee, D. Han, and H. Ko, "Acoustic Signal Based Abnormal Event Detection in Indoor Environment using Multiclass Adaboost," *IEEE Trans. on Consumer Electronics*, vol. 59, no. 3, pp. 615–622, Aug. 2013.

[15] L. Gerosa, G. Valenzise, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and Gunshot detection in noisy environments," in *European Signal Processing Conference*, Poznan, Poland, Sep. 3-7 2007.

[16] H. Phan, M. Maas, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, Jan. 2015.

[17] J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using gabor filterbank features," in *European Signal Processing Conference*, Nice, France, Aug. 30 - Sept. 4 2015, pp. 719–723.

[18] X. Valero and F. Alas, "Gammatone wavelet features for sound classification in surveillance applications," in *European Signal Processing Conference*, Bucharest, Romania., Aug. 27-31 2012, pp. 1658–1662.

[19] J.-L. Rouas, J. Louradour, and S. Ambellouis, "Audio events detection in public transport vehicle," in *Intelligent Transportation Systems Conference*, Sept. 2006, pp. 733–738.

[20] F. Ganansia, V. Delcourt, QC. Pham, A. Lapeyronnie, C. Baudry, L. Lucat, P. Sayd, S. Ambellouis, D. Sodoyer, AC. Barcelo, and F. Heer, "Audio-video surveillance system for public transportation," in *World Congress on Railway Research*, Lille, 2011, France, May 22-26 2011.

[21] R. Zouaoui, R. Audigier, S. Ambellouis, F. Capman, H. Benhadda, S. Joudrier, D. Sodoyer, and T. Lamarque, "Embedded security system for multi-modal surveillance in a railway carriage," in *SPIE security and defence*, Toulouse, France, Sept. 21-24 2015, vol. 9652, pp. Paper N9652–11.

[22] C. M. Bishop, *Neural Networks for Pattern Recognition*, chapter The Multi-layer Perceptron, pp. 116–161, Oxford University Press Inc., New York, 1995.

[23] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. issue 3-4, pp. 197–387, 2014.

[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[26] Y. Miao, "Kaldi+PDNN: Building DNN-based ASR Systems with Kaldi and PDNN," *arXiv:1401.6984*, 2014.

[27] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2012.