DCASE REPORT FOR TASK 3: SOUND EVENT DETECTION IN REAL LIFE AUDIO

Ying-Hui Lai[1,2], *Chun-Hao Wang*[3], *Shi-Yan Hou*[3], *Bang-Yin Chen*[3], *Yu Tsao*[1], *Yi-Wen Liu*[3]

[1]Research Center for Information Technology, Academia Sinica, Taipei, Taiwan
[2]Department of Electrical Engineering, Yuan Ze University, Taoyuan City, Taiwan
[3]National Tsing Hua University, Hsinchu City, Taiwan

ABSTRACT

Our team has built an acoustic event classifier solely using short-time features. Signals were first de-noised by a log minimum square error (logMMSE) procedure. Then, Mel-frequency cepstral coefficients (MFCCs) extracted from the de-noised signal at every 20 ms were used to train two classifiers based on support vector machine (SVMs) and neural networks (NN), respectively. Optimal parameters for the classifiers were exhaustively searched to maximize the frame-wise recognition accuracy in cross validation. Frame-wise recognition rates of 93.0% and 91.8% were thus obtained from the SVM and NN, respectively, for the home events (and 86.2% and 85.7% respectively for the residential events). To process the evaluation data, the same signal processing procedures were applied so both classifiers produce their classification result for every frame. Whenever SVM and NN gives different answers, we resort to the confusion matrices obtained during the supervised learning phase so a final answer could be produced based on a maximal a posteriori (MAP) principle. Finally, a heuristic smoothing procedure was applied to the *jointly decided* recognition results so the event onsets and offsets could be determined.[i]

Index Terms— Support Vector Machines, Neural Networks, Noise Reduction, Decision Fusion

1. SYSTEM OVERVIEW

We first converted the audio files provided by DCASE 2016 from stereo to mono, then used the log minimum mean square error (logMMSE)[9] method to reduce the background noise. Afterwards, we extracted two sets of short-time features: 20 Mel-Frequency Cepstrum Coefficients (MFCC) plus their 1st and 2nd derivatives, and 72 bandpass-filtered output energy based on a time domain loudness model (TDLM)[15]. To prepare for supervised learning, we excluded so-called "silent frames" if the total energy of a frame was below a dynamically determined threshold. After silence removal, acoustically active frames were separated into the group of "target sounds" (consisting of 11 categories for the home database and 7 categories for the residential database) versus the group of "background sounds" based on their labels. A neural network was trained to distinguish background sounds from the target sounds.

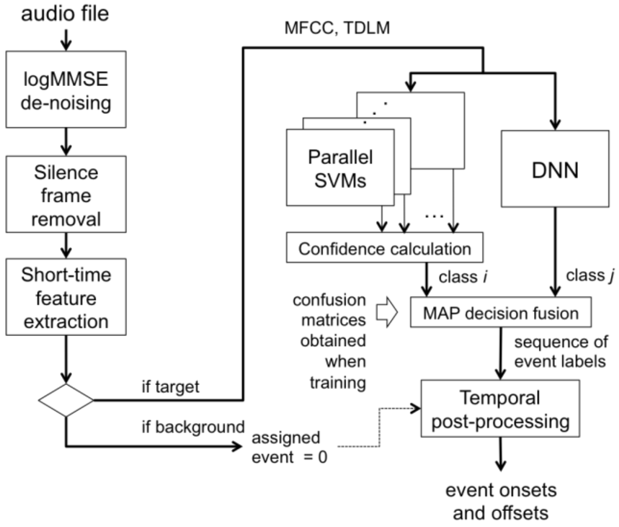


Fig. 1: System diagram

For the target sounds, 30 frames from each class were randomly selected to be included in the testing database, and all the remaining frames were utilized for training and cross validating SVM and NN-based event classifiers.

After the classifiers were trained individually, they are connected as shown in the block diagram in Fig. 1 to recognize sound events from raw audio files. Details of each block are described next.

2. FURTHER DESCRIPTION OF METHODS

This section describes further details of the signal processing and machine learning methods.

2.1 Signal pre-processing

Previous studies showed that ambient background noise deteriorates the recognition performance of a classifier in an event detection task, with approximately 10% every 5 dB [8]. Therefore, noise reduction (NR) processing is necessary to improve the recognition rates. In our system, we adopted the logMMSE [9] NR algorithm to remove the ambient background noises for both training and testing sounds. The logMMSE algorithm is a statistical-model-based NR method that aims to minimize the mean-square error between the clean and estimated magnitude spectra. Previous experiment results indicated that the logMMSE NR

[i] Y.W.L. thanks the Ministry of Science and Technology of Taiwan for supporting this research.

approach reduces the residual noise, and most importantly, without affecting the speech signals itself, that is, without introducing much speech distortion [9,10]. For more details technologies of logMMSE please refer to [9,10].

2.2 MFCC settings

When extracting MFCC features, we set the following parameters: window length = 0.04 seconds, hop length = 0.02 seconds, number of Mel-filters 40, minimum and maximum frequency 0 and 22050 Hz for Mel-filters, number of cepstral coefficients after discrete cosine transform = 20. After we obtained the MFCCs, we also computed their delta and acceleration coefficients. Then we remove the first coefficient, which represents the sum of log energy. Although the total energy is not considered informative in the classification task, we did remove the so-called silent frames based on frame energy.

2.3 Silence removing method based on frame energy

Silence removal was performed based on frame energy. For each testing file, we first computed the mean of the frame energy sequence E_1=mean$\{E(i)\}$. Then, we applied a moving average filter of length 25 to $\{E(i)\}$, resulting a filtered sequence $E_2(i)$, i being the frame index. Whether a frame was counted as silence or not would depend on the ratios between $E(i)$, $E_2(i)$ and E_1; several weighting parameters were tuned to reach the best performance according to the author's subjective evaluation. These parameters are denoted as w_1, w_2, and w_3 respectively and their definition are given as follows: first, frames satisfying $E(i) < w_1E_1$ would be considered-silent and would not be passed on further; then, the remaining frames with $E(i) > w_2E_1$ or $E(i) > w_3\,E_2(i)$ would be regarded as not silent. The main reason to base the decision on moving average filtered frame energy is because it helps detecting transient events, and the majority of the target events in task3 are transient. The first weighted mean energy w_1E_1 is used to exclude the inaudible noise to be regarded as events, and the second weighted mean energy w_2E_1 is used to ensure those long term events could be correctly kept for subsequent model training. Parameters w_1, w_2, and w_3 were set to 0.05, 0.5, and 0.5, respectively.

2.4 Building the background model

A binary background vs. targeted sound classifier was built using a neural network. Details are described in Sec. 2.6.

2.5 Training the SVMs

A support vector machine is a model that seeks to split two classes by a hyperplane in a certain feature space. Here, we adopted "libsvm"[16] to develop our classifiers. For each class of events, a binary SVM was built to detect whether a frame belongs to this class or not, based on short-time features. Therefore, 11 and 7 binary classifiers were trained for the home and residential databases, respectively. Before the training started, 30 frames were randomly selected for each targeted class to be included in the testing dataset, and the rest of frames were used to train and validate the SVMs. For each binary classifier, the training set also included "negative data" that were evenly fetched from the all the others (10 or 6) target classes. We set the number of points in the training dataset and the testing dataset to be 1:1. Empirically, we found that the radial basis function-based kernel worked well for the current task, and the parameters for the kernel functions, including an exponent coefficient gamma and a fault tolerance factor, were fined-tuned by grid searching so as to obtain the highest accuracy during cross validation.

The libsvm toolkit also calculates a conditional probability (regarded as "confidence" in Fig. 1) when performing classification [17]. Technically, this can be achieved by turning on a parameter "-b" during execution. The final SVM-based label was produced by picking the maximal confidence level among all binary classifiers. All confidence values were also saved for event label smoothing, to be described in Sec. 2.8.

2.6 Training the deep neural network (DNN)

An artificial neural network (ANN) model is a mathematical model that mimics biological neural network (NN) structures and functions to enable a machine system to execute classification or regression tasks. More recently, the multiple layers NN, called deep NN (DNN), has shown well performance in a wide variety of tasks [1-5]. The concept of DNN involves using a high number of hidden layers to strengthen the classification or regression capability, using the current output layer as the input of the next hidden layer. The details of the DNN technologies can refer to [1,6].

First of all, we based on acoustic scenes of home to train a DNN binary classifier model to identify the background or target sounds in home condition. Next, we used the acoustic sounds of home and residential area, obtained from the TUT sound events 2016 acoustic dataset, to train the other two DNN models. One DNN model is used to distinguish the sound events in home, and the other is used to identify the sound events in residential. In these three DNN models training, the MFCC features were used. In addition, the DNN structures of these three models were set as a three-layer with 500 hidden neurons, which has been demonstrated to achieve the best performance for the current task. The details performance of these three DNN models is summarized in Table 1.

2.7 Combining the decisions made by SVM and NN

SVM and NN-based event classifiers inevitably gave different results from time to time. When this happen, we relied on a maximal a priori (MAP) hypothesis-testing rule to determine whether SVM or NN was right. Let i and j be the label determined by SVM and NN, respectively, and assume that $i \neq j$. The following decision had to be made,

$$P(H_1|i,j) \gtrless P(H_2|i,j), \tag{1}$$

where H_1 and H_2 denote the hypotheses that the ground truth being i or j, respectively, and the inequality is basically a comparison between two posterior probabilities given the present judgments (i,j) from both classifiers. Using the Bayes rule, the comparison could be shown to be equivalent to the following decision to be made,

$$P(H_1)P_{\mathrm{SVM}}(i|H_1)P_{\mathrm{NN}}(j|H_1) \gtrless P(H_2)P_{\mathrm{SVM}}(i|H_2)P_{\mathrm{NN}}(j|H_2), \tag{2}$$

where $P_{\mathrm{SVM}}(l|H_k)$ denotes the conditional probability that the SVM classifier determines the label is l while the ground truth is H_k (and P_{NN} is similarly defined), and $P(H_k)$'s denote the prior probabilities. We estimated $P_{\mathrm{SVM}}(l|H_k)$ and $P_{\mathrm{NN}}(m|H_k)$ by calculating the confusion matrices during the training phase.

Note that derivation from (1) to (2) is valid only if the two classifiers are assumed to give independent results.

2.8 Rules for determining event onsets and offsets

After performing the decision fusion, the sequence of event labels were smoothed according to their run lengths. Based on the training data, we categorized the target events into two different kinds: *transient* or *sustaining*. The sustaining kind included *rustling*, *people walking*, *washing dishes*, and *water tap running* from the home database, and *car passing by*, *people speaking*, *people walking*, and *wind blowing* from the residential database. All the other events were regarded as transient.

A *minimum event gap* (MEG) of 0.5 sec was defined so that intermittent labels could be merged if they are separated by duration shorter than the MEG. For events belonging to the transient kind, if multiple event classes simultaneously claim for a segment of the event sequence, we assign the segment to the class that possesses the highest total confidence. Finally, we check whether the length of the event falls between the minimum and maximum duration as it has ever occurred in the training set. If not, the event would be discarded. For events belonging to the sustaining kind, an event label would be kept as long as its duration is above the minimal length in the training set.

These post-processing rules were designed based on a few assumptions: first, transient events rarely overlap. Secondly, sustaining events can overlap with other events. Finally, the length of any event should roughly match with what was determined by human annotators.

3. CLASSIFICATION PERFORMANCE

Since the event labels for the evaluation dataset are not available yet, here we only summarize the classification performance using the training set. Results reported here were all obtained in cross validation.

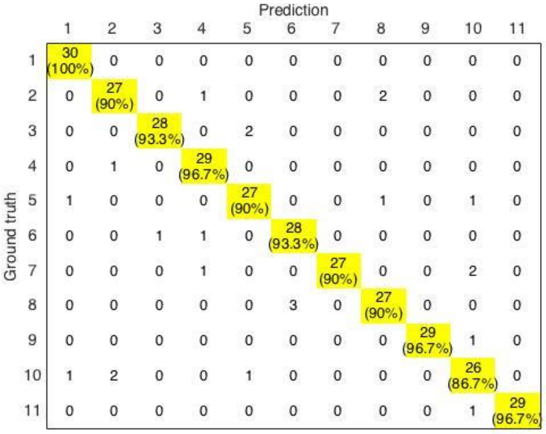
3.1 Background vs. Target (BvT) classification

The frame-wise BvT recognition rates, as the network depth increases, are listed Table 1. Note that BvT classification was only performed for the home database because labeled non-target sound events were too scarce in the residential database. Consequently, we decided not to build a BvT classifier for residential sounds.

From the results in Table 1, we note that for the task of binary decisions on background and target events, a one layer ANN can already achieve satisfactory performance while DNNs with more layers do not yield further improvements. Similar results have been reported in the previous studies [18] that when dealing with a simple classification task, NNs with deep structures might not attain additional gains when compared to an ANN with a single hidden layer.

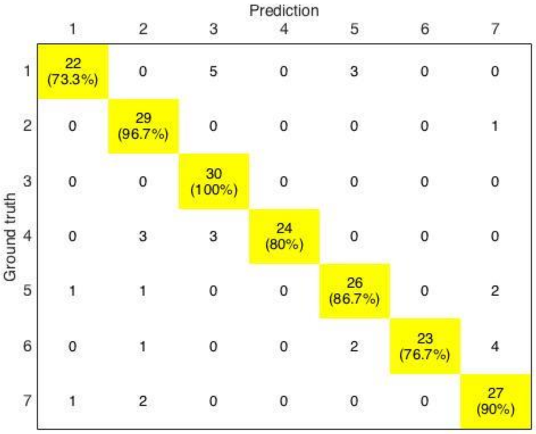
Table 1. BvT classification accuracy using different configurations of NN. Results obtained from the home database. 500 is the number of nodes in each layer (not including the input and the output layer), and the depth varied from 1 to 5.

	500×1	**500×2**	**500×3**	**500×4**	**500×5**
Home (binary)	**75.9%**	73.4%	70.8%	72.8%	72.7%


Prediction
Ground truth

	1	2	3	4	5	6	7	8	9	10	11
1	30 (100%)	0	0	0	0	0	0	0	0	0	0
2	0	27 (90%)	0	1	0	0	0	2	0	0	0
3	0	0	28 (93.3%)	0	2	0	0	0	0	0	0
4	0	1	0	29 (96.7%)	0	0	0	0	0	0	0
5	1	0	0	0	27 (90%)	0	0	1	0	1	0
6	0	0	1	1	0	28 (93.3%)	0	0	0	0	0
7	0	0	0	1	0	0	27 (90%)	0	0	2	0
8	0	0	0	0	0	3	0	27 (90%)	0	0	0
9	0	0	0	0	0	0	0	0	29 (96.7%)	1	0
10	1	2	0	0	1	0	0	0	0	26 (86.7%)	0
11	0	0	0	0	0	0	0	0	0	1	29 (96.7%)

Fig. 2: The confusion matrix of SVM-based event classification for the home events during the training phase.


Prediction
Ground truth

	1	2	3	4	5	6	7
1	22 (73.3%)	0	5	0	3	0	0
2	0	29 (96.7%)	0	0	0	0	1
3	0	0	30 (100%)	0	0	0	0
4	0	3	3	24 (80%)	0	0	0
5	1	1	0	0	26 (86.7%)	0	2
6	0	1	0	0	2	23 (76.7%)	4
7	1	2	0	0	0	0	27 (90%)

Fig. 3: Similar to Fig. 2, for the residential events.

3.2 SVM-based event classification

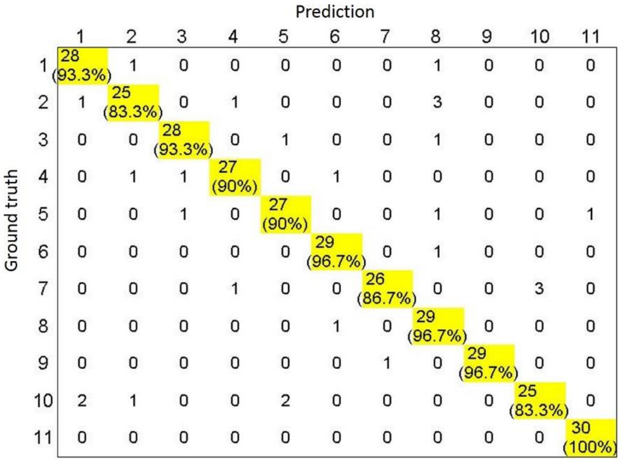
Confusion matrices for SVM-based event classification are given in Fig. 2 and 3 for home and residential database, respectively. Here, the SVM parameters, gamma and fault tolerance for every binary classifier, have been extensively searched and only the best results are shown.

3.3 NN-based event classification

A suitable configuration of DNN (i.e., units and layers) can provide optimal recognitions performance. In most deep learning studies, the general conclusion is that increasing the depth of the NN always helps in performance either for pattern classification, encoding and noise reduction tasks [11-14]. Similarly, we increase the depth of the network to test the recognition performance in our pilot study. More specifically, we carried out experiments by setting the number of hidden units to 500, and increased the depth from one to five. Table 2 show the experimental results for the recognitions of home (11 categories) and residential (7 categories) sound events, respectively. The results showed that the 500×3 (i.e., 500 units and three layers) configuration of DNN model can provide best performance in our pilot study. More specifically, on average, the best performance of DNN model provided 91.8% and 85.7% for home and residential testing conditions.

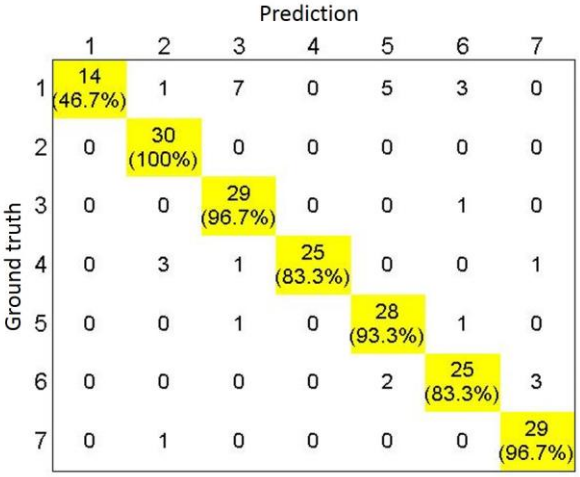
Table 2. The results of different configurations of DNN in home and residential testing conditions.

	500×1	**500×2**	**500×3**	**500×4**	**500×5**
Home	90.0%	91.2%	**91.8%**	88.8%	85.5%
Residential area	84.8%	85.2%	**85.7%**	80.5%	79.5%


Prediction
Ground truth

	1	2	3	4	5	6	7	8	9	10	11
1	28 (93.3%)	1	0	0	0	0	0	1	0	0	0
2	1	25 (83.3%)	0	1	0	0	0	3	0	0	0
3	0	0	28 (93.3%)	0	1	0	0	1	0	0	0
4	0	1	1	27 (90%)	0	1	0	0	0	0	0
5	0	0	1	0	27 (90%)	0	0	1	0	0	1
6	0	0	0	0	0	29 (96.7%)	0	1	0	0	0
7	0	0	0	1	0	0	26 (86.7%)	0	0	3	0
8	0	0	0	0	0	1	0	29 (96.7%)	0	0	0
9	0	0	0	0	0	0	1	0	29 (96.7%)	0	0
10	2	1	0	0	2	0	0	0	0	25 (83.3%)	0
11	0	0	0	0	0	0	0	0	0	0	30 (100%)

Fig. 4: The confusion matrix of DNN-based event classification for the home events during the training phase.


Prediction
Ground truth

	1	2	3	4	5	6	7
1	14 (46.7%)	1	7	0	5	3	0
2	0	30 (100%)	0	0	0	0	0
3	0	0	29 (96.7%)	0	0	1	0
4	0	3	1	25 (83.3%)	0	0	1
5	0	0	1	0	28 (93.3%)	1	0
6	0	0	0	0	2	25 (83.3%)	3
7	0	1	0	0	0	0	29 (96.7%)

Fig 5: Similar to Fig. 4, for residential events.

4. PERFORMANCE

The performance of evaluation datasets is given by dCASE. It's calculated by F1-score (F1) and error rate (ER). Table 3 shows overall system performance. Table 4 includes home and residential area segment-based performance respectively.

Table 3. Overall system performance evaluated by ER and F1 of segment-based and event-based.

	Segment-based		**Event-based**	
	ER	**F1**	**ER**	**F1**
Overall	0.9287	34.5%	2.4283	8.1%

Table 4. Performance in home and residential area evaluated by segment-based ER and F1

	Segment-based	
	ER	**F1**
Home	1.2249	17%
Residential area	1.7328	14.9%

5. DISCUSSIONS

From Table 2, we can note that an ANNs with two and three layers achieve better performance than ANN with a single hidden layer, suggesting that a deep structure may possess better detection capability than a shallow structure. Meanwhile, we note that when there are four and five hidden layers, the performance of ANN decreases, showing that the training data may be insufficient to accurately train the parameters in a DNN with too many layers.

When comparing the SVM and DNN confusion matrices as shown in Figs. 2 to 5, these two classifiers provide their advantages in different events. More specifically, DNN can more accurately detect on drawer, object impact, water tap running, bird singing, children shouting, people speaking, people walking and wind blowing events while less accurately detect (object) rustling, (object) snapping, cupboard, cutlery, dishes, glass jingling, people walking, washing dishes, (object) banging, car passing by events when compared with SVM. The results show that these two types of classifiers perform differently and provide complimentary information, which can be integrated using Eqs. (1) and (2).

6. INTENDED FUTURE WORK

The present confusion matrices were not obtained by 4-fold, file-wise cross validation as recommended by the dCASE organizer. Therefore, we admit that the recognition accuracy might be over optimistic in this regard. Nevertheless, we believe that our performance for the evaluation set could be improved by replacing the confusion matrices with more realistic ones at the MAP decision fusion step (Sec. 2.7).

By participating in this competition, we have also conducted a pilot study which used a multi-task training strategy to build the

DNN classifier by integrating heterogeneous features, namely MFCC and of TDLM [15]. The results show that the DNN models with multi-task training can improve the performance than original DNN classifier with 2% and 1.5% absolute detection rate improvements in BvT and outdoor test conditions, respectively. Due to the limited space, we did not include this part of results in this paper. In the future, we will conduct detailed investigations on finding optimal strategies to combine acoustic features with complimentary information to further improve the performance of DNN classifier for event detection. Meanwhile, we note that the amount of training data significantly affect the achievable detection performance. In the future, we will also explore suitable methods, such as transfer learning [7] or pretraining techniques [19], to effectively utilize the available training data to effectively train the DNN models.

7. ACKNOWLEDGMENT

The authors thank Dr. Ankh Li for insightful comments and critiques during the early stage of this competition.

8. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, pp. 82-97, 2012.

[2] Y. Xu, J. Du, L. R. Dai and C. H. Lee, “A regression approach to speech enhancement based on deep neural networks,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 23, pp. 7-19, 2015.

[3] X. Lu, Y. Tsao, S. Matsuda and C. Hori, “Speech enhancement based on deep denoising autoencoder,” in *Proc. Interspeech*, 2013.

[4] P. Y. Simard, D. Steinkraus and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Proc. ICDAR*, pp. 958–963, 2003.

[5] T. E. Chen, S.-I Yang, L. T. Ho, K.-H. Tsai, Y. H. Chen, Y.-F. Chang, Y.-H. Lai, S.-S. Wang, Y. Tsao, and C.-C. Wu, "S1 and S2 heart sound recognition using deep neural networks," to appear in *IEEE Transactions on Biomedical Engineering*.

[6] H. Larochelle, Y. Bengio, J. Louradour and P. Lamblin, “Exploring strategies for training deep neural networks,” *Machine Learning*, vol. 10, pp. 1-40, 2009.

[7] L. Deng, X. Li, ”Machine learning paradigms for speech recognition: An overview,” *IEEE Trans. Audio Speech Lang. Process.*, 21(5), 1060-1089, 2013.

[8] A. Mesaros, T. Heittola, A. Eronen, & T. Virtanen, “Acoustic event detection in real life recordings,” *Proc. IEEE Signal Processing Conference, 2010 18th European*, pp. 1267-1271, Aug. 2010.

[9] Y. Ephraim, and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(2), 443-445, 1985.

[10] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press. 2013.

[11] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, 2(1): 1-127, 2009.

[12] G. E. Hinton, and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, 313: 504-507, 2006.

[13] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. E. Hinton, “Binary coding of speech spectrograms using a deep autoencoder,” in *Proc. of Interspeech*, 1692-1695, 2010.

[14] X. Lu *et al*. (2013). "Speech enhancement based on deep denoising autoencoder." in *Proc. Interspeech*, pp. 436-440.

[15] S. T. Neely, J. Rodriguez, Y.-W. Liu, and M. P. Gorga, “A computational model of loudness density,” *unpublished article*, online available at researchgate.com upon request. DOI: 10.13140/RG.2.1.5044.4640, July 2015.

[16] C.-C. Chang and C.-J. Lin, “LIBSVM : a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[17] T.-K. Huang, R. C. Weng, and C.-J. Lin. “Generalized bradley-terry models and multi-class probability estimates,” *Journal of Machine Learning Research*, 7, pp. 85-115, 2006.

[18] P. Lin, S.-W. Fu, S.-S.Wang, Y.-H. Lai, and Y. Tsao, "Maximum entropy learning with deep belief networks," to appear in *Entropy*.

[19] G. E. Hinton, S. Osindero, Y.W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, 334 18, 1527–1554, 2006.