Dysarthric Speech Recognition Using a Convolutive Bottleneck Network

Toru Nakashika*, Toshiya Yoshioka*, Tetsuya Takiguchi*, Yasuo Ariki*, Stefan Duffner† and Christophe Garcia†
*Graduate School of System Informatics
Kobe University, JAPAN
nakashika@me.cs.scitec.kobe-u.ac.jp, takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp
†Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR 5205, FRANCE

***Abstract*—In this paper, we investigate the recognition of speech produced by a person with an articulation disorder resulting from athetoid cerebral palsy. The articulation of the first spoken words tends to become unstable due to strain on speech muscles, and that causes a degradation of traditional speech recognition systems. Therefore, we propose a robust feature extraction method using a convolutive bottleneck network (CBN) instead of the well-known MFCC. The CBN stacks multiple various types of layers, such as a convolution layer, a subsampling layer, and a bottleneck layer, forming a deep network. Applying the CBN to feature extraction for dysarthric speech, we expect that the CBN will reduce the influence of the unstable speaking style caused by the athetoid symptoms. We confirmed its effectiveness through word-recognition experiments, where the CBN-based feature extraction method outperformed the conventional feature extraction method.**

Index terms—Articulation disorders, feature extraction, convolutional neural network, bottleneck feature, dysarthric speech.

I. Introduction

Recently, the importance of information technology in the welfare-related fields has increased. For example, sign language recognition using image recognition technology [1], text reading systems from natural scene images [2], and the design of wearable speech synthesizers for voice disorders [3], [4] have been studied.

As for speech recognition technology, the opportunities in various environments and situations have increased (e.g., operation of a car navigation system, lecture transcription during meetings, etc.). However, degradation can be observed in the case of children [5], persons with a speech impediment, and so on, and there has been very little research on orally-challenged people, such as those with speech impediments. It is hoped that speech recognition systems will one day be able to recognize their voices.

One of the causes of speech impediments is cerebral palsy. Cerebral palsy results from damage to the central nervous system, and the damage causes movement disorders. Three general times are given for the onset of the disorder: before birth, at the time of delivery, and after birth. Cerebral palsy is classified as follows: 1) spastic type 2) athetoid type 3) ataxic type 4) atonic type 5) rigid type, and a mixture of types [6].

In this paper, we focused on a person with an articulation disorder resulting from the athetoid type of cerebral palsy as in [7]. Athetoid symptoms develop in about 10-15% of cerebral palsy sufferers. In the case of a person with this type of articulation disorder, the first movements are sometimes more unstable than usual. That means, the case of movements related to speaking, the first utterance is often unstable or unclear due to the athetoid symptoms. Therefore, we recorded speech data for a person with a speech impediment who uttered a given word several times, and we investigated the influence of the unstable speaking style caused by the athetoid symptoms.

In speech recognition technology, frame-wise features such as mel-frequency cepstral coefficients (MFCC), linear predictive coding (LPC), and an autoregressive model (AR) have been widely used so far. However, these features do not capture the temporal information unless delta features are used. Especially for dysarthric speech, where the signals fluctuate more obviously than the signals uttered by a physically unimpaired person, spectral transition in the short term is considered to be an important factor in capturing the local temporal-dimensional characteristics. In this paper, we employ an approach based on convolutional neural networks (ConvNets) [8], [9]) to extract disorder-dependent features from a segment MFCC map. The ConvNet is regarded as a successful tool and has been widely used in recent years for various tasks, such as image analysis [10], [11], [12], spoken language identification [13], and music recognition [14]. A ConvNet consists of a pipeline of convolution and pooling operations followed by a multi-layer perceptron. In dysarthric speech, the key points in time-spectral local areas of an input feature map are often shifted slightly due to the fluctuation of the speech uttered by a person with an articulation disorder. However, thanks to the convolution and pooling operations, we can train the ConvNet robustly to deal with the small local fluctuations. Furthermore, we expect that the ConvNet extracts specific features associated with the articulation disorder we are targeting when we train the network using only the speech data of the articulation disorder.

For the research described in this paper, we used a convolutive bottleneck network (CBN) [15], which is an extension of a ConvNet, to extract disorder-specific features. A CBN stacks a bottleneck layer, where the number of units is extremely small compared with the adjacent layers, following the ConvNet layers. Due to the bottleneck layer having a small number of units, it is expected that it can aggregate the propagated information and automatically learn sparse feature representations.

This paper is organized as follows: we briefly review the fundamental method, ConvNet in Section II. The proposed

978-1-4799-2186-7/14/$31.00

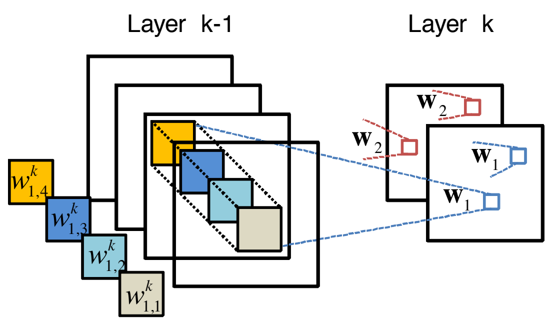

Fig. 1. A feature map in a convolutional layer (right) activated by its corresponding input feature maps (left).

feature extraction method and the structure of the CBN used in the experiments are presented in Section III. In Section IV, we show the experimental results, and we give our conclusions regarding in Section V.

II. Convolutional Neural Network

In this section, we describe a convolution layer and a pooling layer, which are fundamental components of a ConvNet (convolutional neural network).

A. Convolution layer

Assuming that we have a two-dimensional input feature map $\boldsymbol{x} \in \mathbb{R}^{N_n^x \times N_m^x}$ and a convolutive filter $\boldsymbol{w} \in \mathbb{R}^{N_n^w \times N_m^w}$, the output of a convolutive operation $\boldsymbol{h} = \boldsymbol{x} * \boldsymbol{w}$ also becomes a two-dimensional feature with the size of $N_n^h \times N_m^h$ ($N_n^h \equiv N_n^x - N_n^w + 1$ and $N_m^h \equiv N_m^x - N_m^w + 1$). A ConvNet generally has a number of such filters $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_L\}$ in a convolutive layer, and feeds an input $\boldsymbol{x}$ using each filter to create the corresponding outputs $\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_L\}$, which is referred to as a feature map.

Given all of the feature maps in the $(k-1)$th layer $\{\boldsymbol{h}_1^{k-1}, \cdots, \boldsymbol{h}_i^{k-1}, \cdots, \boldsymbol{h}_I^{k-1}\}$, the jth feature map $\boldsymbol{h}_j^k \in \mathbb{R}^{N_h \times N_h}$ in the kth (convolution) layer can be calculated as

$$\boldsymbol{h}_j^k = f\left(\sum_{i \in I} \boldsymbol{w}_{j,i}^k * \boldsymbol{h}_i^{k-1} + \boldsymbol{b}_j^k\right), \tag{1}$$

where $\boldsymbol{w}_{j,i}^k$ and $\boldsymbol{b}_j^k$ indicate a predictable filter from the ith feature map in the $(k-1)$th layer to the jth map in the kth layer and a bias map of the jth map in the kth layer, respectively. In this paper, we used an element-wise sigmoid function for the activation function f as follows

$$f(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1} + e^{-\boldsymbol{x}}}, \tag{2}$$

where the fraction bar indicates element-wise division.

Each unit in a convolution layer is connected to the units in the corresponding local area of size $N_n^w \times N_m^w$ in the previous layer (local receptive field). In other words, the convolution layer in a ConvNet captures local patterns in an input map using various filters (Figure 1).

B. Pooling layer

Followed by the convolution layer, a pooling procedure is generally used in a ConvNet, creating what is called a pooling layer. Each unit in the pooling layer aggregates responses in the local subregion $\mathcal{P}(M \times M)$ in the previous convolution layer. As a result, a feature map in the pooling layer has the size of $N_n^h/M \times N_m^h/M$.

There exist various pooling methods (e.g. max-pooling), but we use average-pooling in this paper, calculated as follows

$$\boldsymbol{h}_j^{k+1} = f\left(w_j^{k+1} \cdot \frac{1}{M^2} \sum_{(u,v) \in \mathcal{P}} \boldsymbol{h}_{j,u,v}^k + b_j^{k+1}\right), \tag{3}$$

where w_j^{k+1} and b_j^{k+1} are a weight parameter and a bias parameter of the jth feature map in the pooling layer ($(k+1)$th layer), respectively. $\boldsymbol{h}_{j,u,v}^k$ represents the unit in the corresponding subregion identified with (u, v) in the feature map in the kth layer.

This pooling process enables the network to ignore small position shifts of a key point in the input feature map since it aggregates information in the local area.

III. Proposed method

In our approach, we use a convolutional neural network (ConvNet) that has a bottleneck layer in the network, referred to as a convolutive bottleneck network (CBN) [15], for capturing speaker-dependent features from a dysarthric speech signal.

A. Convolutive Bottleneck Network

A CBN consists of an input layer, convolution layer and pooling layer pairs, fully-connected MLPs (multi-layer perceptrons) with a bottleneck structure, and an output layer in the order shown in Figure 2. In our approach, the CBN receives a mel map (two-dimensional acoustic features in time-mel-frequency) and outputs 54 phoneme labels. The MLP shown in Figure 2 stacks three layers (m_1, m_2, m_3), where we give 108 units, 30 bottleneck units[1], and 108 units in each layer, respectively. The filter sizes in the convolution layer and the pooling layer will be discussed in the experimental section. Since the bottleneck layer has reduced the number of units for the adjacent layers, we can expect that each unit in the bottleneck layer aggregates information and behaves as a compact feature descriptor that represents an input with a small number of bases, similar to other feature descriptors, such as MFCC, LDA (linear discriminant analysis) or PCA (principal component analysis). It is reported that a feature extraction method using bottleneck features improved speech recognition accuracy from the well-known ABN (American Broad News) corpus [16]. In this paper, we also use such features (the unit values in the bottleneck layer) for speech recognition, instead of using MFCC. The extracted features are obtained from the statistically-trained speaker-dependent CBN; hence, it is expected that it better represents characteristics in the speech of the target articulation disordered speech than MFCC does.

[1] We also tested other configurations as discussed later.

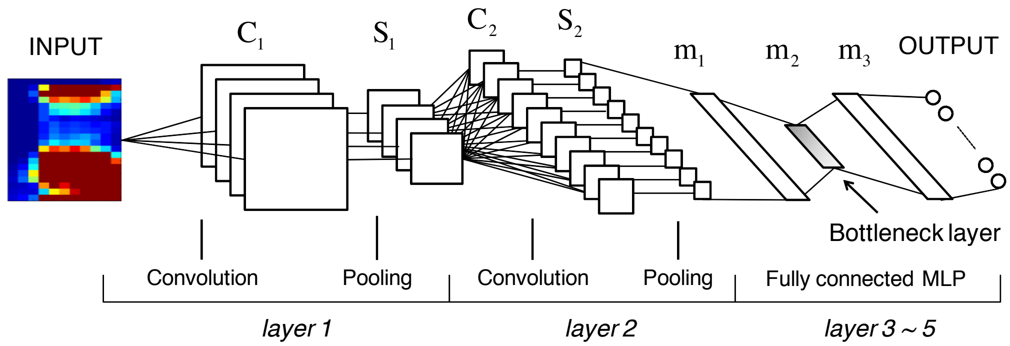

Fig. 2. Convolutive Bottleneck Network (CBN).

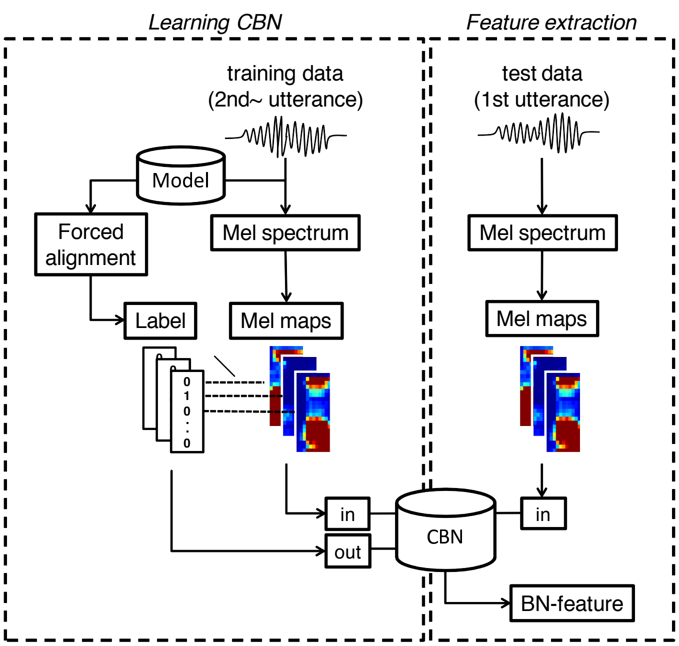

Fig. 3. Flow of our feature extraction method for dysarthric speech using a convolutive bottleneck network.

B. Bottleneck feature extraction from dysarthric speech

Figure 3 shows a system flowchart of our method, where speaker-specific features are extracted using a CBN. First, we prepare the input features for training a CBN from a speech signal uttered by a person with an articulation disorder. After calculating short-term mel spectra from the signal, we obtain mel maps by dividing the mel spectra into segments with several frames (13 frames in our experiments) allowing overlaps. For the output units of the CBN, we use phoneme labels that correspond to the input mel-map. For example, when we have a mel map with the label /i/, only the unit corresponding to the label /i/ is set to 1, and the others are set to 0 in the output layer. The label data is obtained by forced alignment using a hidden Markov model (HMM) from the speech data. The parameters of the CBN are trained by back-propagation with stochastic gradient descent, starting from random values. We used a softmax activate function for the output layer, and a sigmoid activate function for the other layers. The bottle-neck (BN) features in the trained CBN are then used in the training of another HMM for speech recognition.

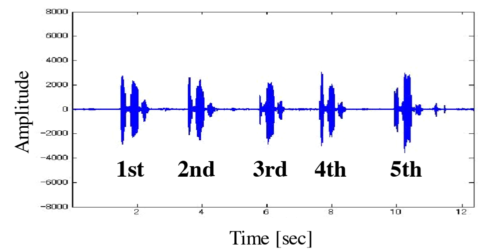

Fig. 4. Example of recorded speech data.

In the test stage ("Feature extraction" in Figure 3), we extract features using the CBN, which recieves the mel maps obtained from test data and tries to produce the appropriate phoneme labels in the output layer. Again, note that we do not use the output (estimated) labels for the following procedure, but we use the BN features in the middle layer, which contain a compact representation of the (raw) input data. Finally, the system recognizes dysarthric speech by feeding these extracted BN features into the trained HMMs because the extracted features are frame-based.

IV. Experiments

A. Experimental conditions

Our feature extraction method was evaluated on word recognition tasks for one person (male) with an articulation disorder. We recorded 216 words included in the ATR Japanese speech database [17] repeating each word five times (Figure 5). The utterance signal was sampled at 16 kHz and windowed with a 25-msec Hamming window every 10 msec. Then we clipped each utterance manually. In our experiments, the first utterances of each word were used for evaluation, and the other utterances (the 2nd through 5th utterances) were used for the training of both a CBN and an acoustic model. We used the HMMs (54 context-independent phonemes) with 5 states and 8 mixtures of Gaussians for the acoustic model.

We trained and evaluated three CBNs: 28 units, 30 units, and 32 units in the bottleneck (BN) layer. The extracted BN features from each CBN were compared with 28-, 30-, and 32-dimensional MFCC+ΔMFCC, respectively.

TABLE I. FILTER SIZE AND NUMBER OF FEATURE MAPS FOR EACH ARCHITECTURE. THE VALUES FOR C1 (AND C2) INDICATE FILTER SIZE OF THE FIRST (AND THE SECOND) CONVOLUTION LAYER THAT HAS #1 MAPS (AND #2 MAPS), RESPECTIVELY. EACH CONVOLUTION LAYER IS ASSOCIATED WITH THE POOLING LAYER (S1 AND S2). THE VALUES FOR S1 AND S2 MEAN THE POOLING FACTOR M).

	C1	S1	#1	C2	S2	#2
Arc1	4×2	3	13	4×2	3	27
Arc2	10×4	2	13	10×4	2	27
Arc3	4×2	3	18	4×2	3	36
Arc4	4×2	4	13	4×2	2	27
Arc5	8×6	2	13	8×2	3	27

B. Training of CBN

In this section, we explain the training conditions of the CBN in detail. We trained the network using pairs of a mel map from the 2nd through 5th utterances and the label, as shown in Figure 3.

The values of each convolutive filter $\boldsymbol{w}_{j,i}^k$ (and the weight w_j^{k+1}) are initialized by [18] as follows:

$$\boldsymbol{w}_{j,i}^k \ni w \sim \mathcal{U}\left(-\sqrt{\frac{6}{N_j + N_{j+1}}}, \sqrt{\frac{6}{N_j + N_{j+1}}}\right), \tag{4}$$

where $\mathcal{U}(a, b)$ denotes uniform distribution at the interval from a to b. N_j and N_{j+1} indicate the numbers of input dimensions and output dimensions at the jth layer, respectively. The bias parameters $\boldsymbol{b}_j^k$ and b_j^{k+1} were set to 0 as initial values.

These parameters were trained so as to minimize the errors (MSE) between the target labels and the output values using a back-propagation algorithm. We iterated batch-based training with 50 frames in a mini-batch 100 times, with a fixed learning rate of 0.1.

C. CBN architectures

In this section, we discuss our first experiment in which we change the architecture of a CBN (such as the number of feature maps and the size of a convolution filter) as shown in Table I. In this experiment, we will see which architecture produced the best recognition accuracy. All the architectures have two pairs of a convolution layer and a pooling layer followed by three-layer MLPs with a bottleneck layer, forming a nine-layer network in total. We used 108, 30 and 108 units in the MLP part, in this order, for all the architectures (the bottleneck feature had 30 dimensions in this experiment). For the input layer, we used a 39-dimensional mel map with 13 frames without overlapping. The size of the map was 39×13. When we use ‘Arc1’ in Table I, for example, the feature maps in each convolution and pooling layer have 36×12, 12×4, 9×3, and 3×1 sizes, in this order. In this case, $3 \times 27 (= 81)$ units are fully connected to the first layer of the MLP part.

Table II shows recognition accuracies obtained from each architecture. As shown in Table II, we obtained the best word recognition accuracies from ‘Arc1’, although it did not always outperform the other architectures with respect to the MSE and label classification accuracy. This is considered to be due to the fact that the extracted bottleneck features of ‘Arc1’ were more abstract and suited to the acoustic model in word recognition.

TABLE II. WORD RECOGNITION ACCURACIES USING THE BOTTLENECK FEATURES (WORD-ACC.) OBTAINED FROM EACH ARCHITECTURE, ALONG WITH THE MEAN SQUARED ERROR OF THE CLOSED DATA (MSE), AND THE OPEN CLASSIFICATION ACCURACIES OF THE PHONEME LABELS USING A CBN ONLY (WITHOUT USING THE ACOUSTIC MODELS) (PHONEME-ACC.).

	Arc1	Arc2	Arc3	Arc4	Arc5
MSE ($\times 10^{-1}$)	2.42	2.01	2.32	2.43	2.09
Phoneme-Acc. (%)	48.7	49.3	47.9	49.4	49.5
Word-Acc. (%)	**88.0**	87.7	82.4	84.3	83.8

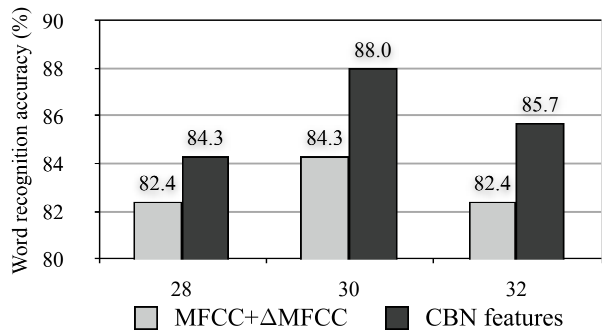

Fig. 5. Word recognition accuracies (%) for the utterances of a person with an articulation disorder using bottleneck features in a convolutional network.

For the remaining experiments, we used ‘Arc1’ for the CBN architecture.

D. Evaluation results and discussion

Figure 5 shows the word recognition accuracies comparing our CBN features with the conventional MFCC features, when changing the number of feature dimensions (for our method, the number of units in the bottleneck layer). As shown in Figure 5, the use of bottleneck features in a convolutive network improved the accuracies from 84.3% to 88.0% (30 dimensions). This is due to the robustness of the CBN features to small local fluctuations in a time-melfrequency map, caused by the dysarthric speech.

We further investigated our method to check the effectiveness of the convolution layer (and pooling layer). In this evaluation, we replaced some convolution layers (here we refer to the pair of a convolution layer and a pooling layer as simply a “convolution layer”) with fully-connected layers in the network (‘Arc1’). First, we replaced the second convolution layer (‘Layer 2’ in Figure 2) with a fully-connected layer with 108 units. Second, we alternated two convolution layers with two fully-connected layers with 108 units, which is regarded as a 7-layer DBN (deep bottleneck network). The results are summarized in Table III. From Table III, we notice that the more convolution layers the network had, the better the performance of the system. Again, we consider that this is because the convolution filter captured characteristics in the input maps, making it robust to local fluctuations. When we compare the fully-connected model (DBN) with the MFCC, we see that the DBN performs poorly since it tends to suffer from over-fitting and a lack of robustness to the open data, especially in dysarthric speech, which fluctuates every time the speaker begins speaking.

TABLE III. WORD RECOGNITION ACCURACIES AS THE NUMBER OF CONVOLUTION LAYERS CHANGED.

# of conv v.s. full layers	Acc. (%)
No conv, 5 full (DBN)	83.3
1 conv, 4 full (CBN)	84.7
2 conv, 3 full (CBN)	88.0
Baseline (MFCCs)	84.3

V. CONCLUSIONS

The articulation of speech uttered by persons with speech disorders tends to become unstable due to strain on their speech-related muscles. This paper described a robust feature extraction method using a convolutive bottleneck network (CBN). In word recognition experiments, our method achieved an approximately 4-point improvement compared with the conventional MFCC features. In this study, only one subject person was evaluated. In future experiments, we will increase the number of subjects and further examine the effectiveness of our method, although it is difficult to obtain adequate amount of dysarthric speech data due to their physical problems.

REFERENCES

[1] S. Cox and S. Dasmahapatra, "High-Level Approaches to Confidence Estimation in Speech Recognition," *IEEE Trans. on SAP*, vol. 10, No. 7, pp. 460–471, 2002.

[2] M. K. Bashar, T. Matsumoto, Y. Takeuchi, H. Kudo and N. Ohnishi, "Unsupervised Texture Segmentation via Wavelet-based Locally Orderless Images (WLOIs) and SOM," *6th IASTED International Conference Computer Graphics and Imaging*, 2003.

[3] T. Ohsuga, Y. Horiuchi and A. Ichikawa, "Estimating Syntactic Structure from Prosody in Japanese Speech," *IEICE Transactions on Information and Systems*, vol. 86, no. 3, pp. 558–564, 2003.

[4] K. Nakamura, T. Toda, H. Saruwatari and K. Shikano, "Speaking Aid System for Total Laryngectomees Using Voice Conversion of Body Transmitted Artificial Speech," *Interspeech 2006*, pp. 1395–1398, 2006.

[5] D. Giuliani and M. Gerosa, "Investigating recognition of children's speech," *ICASSP2003*, pp. 137–140, 2003.

[6] S. T. Canale and W. C. Campbell, "Campbell's Operative Orthopaedics," Mosby-Year Book, 2002.

[7] H. Matsumasa, T. Takiguchi, Y. Ariki, I. Li and T. Nakabayashi, "Integration of Metamodel and Acoustic Model for Speech Recognition," *Interspeech 2008*, pp.2234–2237, 2008.

[8] Y. Lecun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE*, pp. 2278–2324, 1998.

[9] H. Lee et al., "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in Neural Information Processing Systems 22*, pp. 1096–1104, 2004.

[10] C. Garcia and M. Delakis, "Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection," *Pattern Analysis and Machine Intelligence*, 2004.

[11] M. Delakis and C. Garcia, "Text detection with Convolutional Neural Networks," *International Conference on Computer Vision Theory and Applications*, 2008.

[12] R. Hadsell et el., "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, 2009.

[13] G. Montavon, "Deep learning for spoken language identification," *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[14] T. Nakashika, C. Garcia and T. Takiguchi, "Local-feature-map Integration Using Convolutional Neural Networks for Music Genre Classication," *Interspeech 2012*, 2012.

[15] K. Vesely et al. "Convolutive bottleneck network features for LVCSR," *ASRU*, pp. 42–47, 2011.

[16] C. Plahl et al., "Hierarchical bottle neck features for LVCSR," *Interspeech 2010*, pp. 1197–1200, 2010.

[17] A. Kurematsu et al., "ATR japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, no.4, pp.357–363, 1990.

[18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.