FEATURE EXTRACTION USING PRE-TRAINED CONVOLUTIVE BOTTLENECK NETS FOR DYSARTHRIC SPEECH RECOGNITION

Yuki Takashima[1], *Toru Nakashika*[1], *Tetsuya Takiguchi*[2], *Yasuo Ariki*[2]

[1]Graduate School of System Informatics, Kobe University, Japan
[2]Organization of Advanced Science and Technology, Kobe University, Japan
1-1, Rokkodai, Nada, Kobe, 6578501, Japan

ABSTRACT

In this paper, we investigate the recognition of speech uttered by a person with an articulation disorder resulting from athetoid cerebral palsy based on a robust feature extraction method using pre-trained convolutive bottleneck networks (CBN). Generally speaking, the amount of speech data obtained from a person with an articulation disorder is limited because their burden is large due to strain on the speech muscles. Therefore, a trained CBN tends toward overfitting for a small corpus of training data. In our previous work, the experimental results showed speech recognition using features extracted from CBNs outperformed conventional features. However, the recognition accuracy strongly depends on the initial values of the convolution kernels. To prevent overfitting in the networks, we introduce in this paper a pre-training technique using a convolutional restricted Boltzmann machine (CRBM). Through word-recognition experiments, we confirmed its superiority in comparison to convolutional networks without pre-training.

Index Terms— Articulation disorders, feature extraction, convolutional neural networks, bottleneck feature, convolutional restricted Boltzmann machine

1. 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] have been studied. However, 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. There are various types of cerebral palsy. In this paper, we focused on a person with an articulation disorder resulting from the athetoid type as in [4]. 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.

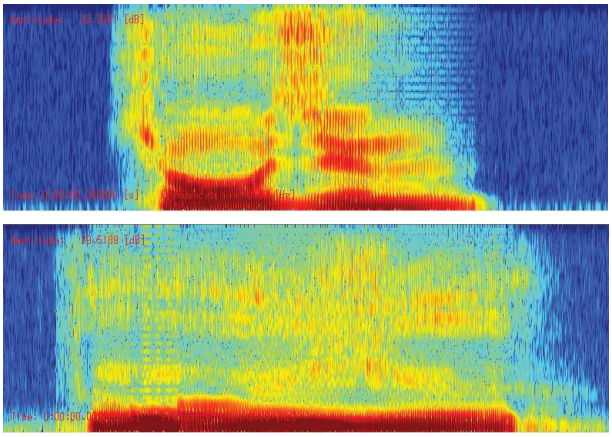

Fig. 1. Example of spectrograms for an utterance (/hyoujun/ in Japanese) spoken by a physically unimpaired person (top) and a person with a dysarthric articulation disorder (bottom)

Fig. 1 shows spectrograms for an utterance (/hyoujun/ in Japanese) spoken by a physically unimpaired person and a person with a dysarthric articulation disorder. For dysarthric speech, where the signal is not obviously more clear than the signal uttered by a physically unimpaired person, the spectral transition in the short term is considered to be an important factor in capturing the local temporal-dimensional characteristics. From this fact, it is clear that a speaker-independent acoustic model for physically unimpaired persons is not adequate. Therefore, we employ convolutional neural networks (CNN), a [5]-based approach to extract disorder-dependent features from a segment MFCC map. The CNN is regarded as a successful tool and has been widely used in recent years for various tasks, such as image analysis [6], a spoken lan-

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

guage [7], and music recognition [8]. A CNN consists of a pipeline of convolution and pooling operations followed by a multi-layer perceptron. Thanks to the convolution and pooling operations, we can train the CNN robustly to deal with the small local fluctuations associated with articulation disorders. Furthermore, we expect that the CNN extracts specific features associated with the articulation disorder that we are targeting when we train the networks using only the speech data of the articulation disorder.

In this paper, we used convolutive bottleneck networks (CBN [9]), which are an extension of CNNs, to extract disorder-specific features. CBNs stack a bottleneck layer, where the number of units is extremely small compared with the adjacent layers, following the CNN layers. Due to the bottleneck layer having a small number of units, it is expected that it can aggregate the propagated information and extract fundamental features included in an input map [10]. However, the features extracted using CBNs are sometimes inferior to conventional features, such as MFCC.

The amount of speech data obtained from a person with an articulation disorder is limited because their burden is large due to the strain placed on their speech muscles. Although the degree of difficulty experienced by such people when they speak varies depending on the person, uttering many words is often difficult for most. Therefore, the amount of recordable speech data is limited, and the trained CBNs tend to overfit. In this paper, we show the effectiveness of pre-training using Convolutional restricted Boltzmann machines (CRBM [11]) through our experiments. Recently, deep learning is researched widely by developing a fast learning algorithm for a deep belief network (DBN [12]). The DBN is a multilayer generative model that is composed of a restricted Boltzmann machine (RBM), and neural networks based on a DBN have been improving in performance dramatically. We expect that CNN can be trained efficiently using a CRBM based on a RBM that deals with two-dimensional acoustic features.

2. CONVOLUTIONAL BOTTLENECK NETWORKS

2.1. Convolutional neural networks

2.1.1. Convolutional layer

Assuming that we have a two-dimensional input feature map $\mathbf{x} \in \mathbb{R}^{N_n^x \times N_m^x}$ and a convolutive filter $\mathbf{w}^k \in \mathbb{R}^{N_n^w \times N_m^w}$, the output of a convolutive operation $\mathbf{h} = \mathbf{x} * \mathbf{w}$ 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 vice versa). CNN generally have a number of such filters $\{\mathbf{w}_1, \cdots, \mathbf{w}_L\}$ in a convolutive layer, and feeds an input $\mathbf{x}$ using each filter to create the corresponding outputs $\{\mathbf{h}_1, \cdots, \mathbf{h}_L\}$, which is referred to as a feature map.

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

$$\mathbf{h}_j^k = \mathrm{sigm}\Big(\sum_i^I \mathbf{w}_{j,i}^k * \mathbf{h}_i^{k-1} + b_j^k \mathbf{E} \Big), \tag{1}$$

where $\boldsymbol{w}_{j,i}^k$ and 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. $\mathbf{E}$ denotes a matrix whose elements are 1. In this paper, we used an element-wise sigmoid sunction for the activation function as follows:

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

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 CNN capture local patterns in an input map using various filters.

2.1.2. Pooling layer

Followed by the convolution layer, a pooling procedure is generally used in CNN, creating what is called a pooling layer. Each unit in the pooling layer aggregates responses in the local subregion $B(M \times M)$ in the previous convolution layer. As a result, a feature map in the pooling layer has the size of $N_n^h/M \times N_m^h/M$. We use average-pooling in this paper.

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.

2.2. Architecture of CBN

Convolutional bottleneck networks (CBN) consist 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 Fig. 2. In our approach, the CBN receives a mel map (two-dimensional acoustic features in time-melfrequency) and outputs 54 phone labels. We give 15 feature maps with the 7×11 kernel to convolution layer. In the pooling layer, previous feature map is contracted one third. The MLP shown in Fig. 2 stacks three layers (m1, m2 , m3), where we give 108 units, 30 bottleneck units, and 108 units in each layer, each respectively. 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.

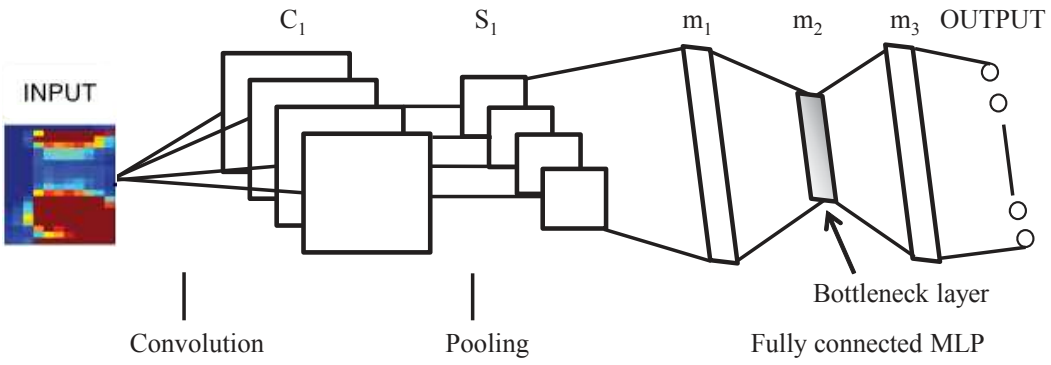

Fig. 2. Convolutive Bottleneck Networks (CBN)

2.3. Bottleneck feature extraction

First, we prepare the input features for training CBN from a speech signal. 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 phone binary labels that correspond to the input mel-map. The parameters of the CBN are trained by back-propagation with stochastic gradient descent, starting from random values. The bottleneck (BN) features in the trained CBN are then used in the training of a GMM-HMM for speech recognition.

In the test stage, we extract features using the CBN, which feed the mel maps obtained from test data and tries to produce the appropriate phone labels in the output layer. Again, we use the BN features in the middle layer, where it is considered that information in the input data is aggregated. Finally, the system recognizes dysarthric speech by feeding the extracted BN features into HMMs.

3. PRE-TRAINING OF CBN

3.1. RBM

A restricted Boltzmann machine (RBM) is an undirected graphical model that defines the distribution of visible unit with binary hidden units, the probabilistic semantics and the energy function as follows:

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\exp\Big(-E(\boldsymbol{v},\boldsymbol{h})\Big), \tag{3}$$

$$E_{\mathrm{RBM}}(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i,j} v_i W_{ij} h_j - \sum_j c_j h_j - \sum_i b_i v_i, \tag{4}$$

where Z is a normalization constant, $\boldsymbol{v} \in \mathbb{R}^{I\times 1}$ denotes binary visible states, and $\boldsymbol{h} \in \mathbb{R}^{J\times 1}$ denotes binary hidden states. i and j are the index of the number of visible units and hidden units, respectively. b_i and c_j are biases of visible and hidden units, respectively. An RBM was originally introduced as a method of representing binary valued data (Bernoulli-Bernoulli RBM; BB-RBM), and it later came to be used to deal with real-valued data known as a Gaussian-Bernoulli RBM (GB-RBM). The GB-RBM is further developed to Improved Gaussian-Bernoulli RBM (IGB-RBM [13]). The conditional probability of a BB-RBM can be written as follows:

$$P(h_j = 1|\boldsymbol{v}) = \mathrm{sigm}\Big(\sum_i W_{ij} v_i + c_j\Big), \tag{5}$$

$$P(v_i = 1|\boldsymbol{h}) = \mathrm{sigm}\Big(\sum_j W_{ij} h_j + b_i\Big). \tag{6}$$

3.2. CRBM

A Convolutional restricted Boltzmann machine (CRBM) is a probabilistic energy based model that has two layers. The difference between the standard RBM and the CRBM is that the input data of the former is a 2-D feature map against the form of vector of the latter. When the visible units are real-valued and the hidden feature map consists of binary units, the model is called a Gaussian-Bernoulli CRBM. Furthermore, when considering the variance of visible units, we call the model Improved Gaussian-Bernoulli CRBM (IGB-CRBM). The energy function of an IGB-CRBM is defined as follows:

$$E_{\mathrm{CRBM}}(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{2\sigma^2}\sum_{i,j}(v_{i,j}-b)^2 - \sum_{k=1}^{K}\sum_{i,j} h^k_{i,j}\left(\frac{1}{\sigma^2}(\tilde{\mathbf{W}}^k * \boldsymbol{v})_{i,j} + c_k\right), \tag{7}$$

where $\boldsymbol{v} \in \mathbb{R}^{N^v_n \times N^v_m}$ denotes the visible nodes, and $\boldsymbol{h} \in \mathbb{R}^{N^h_n \times N^h_m \times K}$ denotes the hidden nodes of K groups. b and c_k are biases of visible and k-th hidden units, and σ is the standard deviation associated with a Gaussian visible unit $\boldsymbol{v}$. The visible nodes and hidden nodes are related by the weight matrix $\boldsymbol{W}^k \in \mathbb{R}^{N^w_n \times N^w_m}$ that represents the connection between the visible units and the hidden units in the k-th group. (N^v_n, N^v_m), (N^h_n, N^h_m) and (N^w_n, N^w_m) refer to the size of visible layer, hidden layer and weight matrix, respectively. We

define $\tilde{\boldsymbol{W}}^k$ as the filter matrix $\boldsymbol{W}^k$ flipped horizontally and vertically.

The conditional probabilities of an IGB-CRBM can be written as follows:

$$P(\boldsymbol{v}|\boldsymbol{h}) = \mathcal{N}\Big(\boldsymbol{v}; \sum_k \mathbf{W}^k * \boldsymbol{h}^k + b\mathbf{E}, \sigma^2\mathbf{E}\Big), \quad (8)$$

$$P(h^k_{i,j} = 1|\boldsymbol{v}) = \mathrm{sigm}\Big((\tilde{\mathbf{W}}^k * \boldsymbol{v})_{i,j} + c_k\Big), \quad (9)$$

where $\mathrm{N}(\cdot|\mu, \sigma^2)$ denotes the Gaussian probability density function with mean μ and variance σ^2. In the rest of paper, we call the IGB-CRBM the CRBM simply.

Given a training data set $\{v^{(n)}\}_{n=1}^N$, the CRBM parameter is estimated to maximize the log-likelihood of the CRBM $\mathcal{L} = \log \prod_n P(\boldsymbol{v}^{(n)})$. The gradient of this log-likelihood with respective to θ is written as

$$\frac{\partial \log P(\boldsymbol{v}^{(n)})}{\partial \theta} = \Big\langle \frac{\partial E_{\mathrm{CRBM}}(\boldsymbol{v}^{(n)}, \boldsymbol{h})}{\partial \theta} \Big\rangle_{\mathrm{data}} + \Big\langle \frac{\partial E_{\mathrm{CRBM}}(\boldsymbol{v}, \boldsymbol{h})}{\partial \theta} \Big\rangle_{\mathrm{model}}, \quad (10)$$

where $\langle\cdot\rangle_{\mathrm{data}}$ and $\langle\cdot\rangle_{\mathrm{model}}$ indicate expectations of input data and the inner model, respectively. However, it is usually difficult to compute the second term in Eq. (10), we use Contrastive Divergence [12]. Each parameter is updated using stochastic gradient descent (SGD) from Eq. (10).

3.3. Pre-training

In a deep network, error signals become weaker as they are backpropagated, especially for the first layer's units. In a CBN, that corresponds to a convolution layer and the convolutional kernel is affected considerably by the initial values. Therefore, the recognition accuracy also depends on the initial values of the convolution kernels. In order to avoid this problem, we propose pre-training using a CRBM. First, we train the CRBM parameters. Then, we copy these weights $\boldsymbol{W}^k$ as the initial values of convolution weights $\mathbf{w}^l_{k,\mathrm{input}}$ in the lth layer, where $input$ is an input feature. Finally, we fine-tune CBN using backpropagation. We expect that this approach will result in efficient learning.

4. EXPERIMENTAL EVALUATION

4.1. Experimental conditions

Our feature extraction method was evaluated on a word-recognition task for one male with an articulation disorder. We recorded 216 words included in the ATR Japanese speech database [14], repeating each word three times. 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 the CBN and acoustic models. We prepare a mel map by merging mel spectra into a 2D feature with 13 frames. We used the HMMs (54 context-independent phones) with 5 states and 8 Gaussian mixtures for the acoustic model. We trained and evaluated a CBN that has 30 units in the bottleneck (BN) layer.

In the pre-training, the number of feature maps and the kernel size of the CRBM were set to 15 and 7×11, respectively (the same as the CBN parameters). We used the same dataset described above for the training of the CRBM. A feature map consisted of 28 frames. We iterated batch-based training with 50 feature maps in a mini-batch. The learning rate was set to 0.001 for the first few epochs, and then changed to 0.0001. Furthermore, the variance of visible units σ^2 was set to the average of the variance calculated from the input maps, and the hidden feature map's bias was fixed at -4 at first [15], and few epochs later, learned.

4.2. Recognition results using speaker-independent HMMs for physically unimpaired persons

At the beginning, we evaluated the recognition experiment using a speaker-independent acoustic model for physically unimpaired persons included in Julius [16]. The test dataset was the same as that mentioned in 4.1. The acoustic model consisted of a triphone HMM set with 25 dimensional MFCC features (12-order MFCCs, their delta, and energy) and 16 mixture components for each state. Each HMM had three states and three self-loops.

The obtained recognition accuracy was only 24.07%. Based on these results, it is clear that a speaker-dependent model and a robust feature are necessary for speech recognition of a person with an articulation disorder.

4.3. Results and discussion

First, we investigated the influence of randomness, where the convolution kernel is randomly initialized each time. Fig. 3 shows the average accuracy for five different initializations. The maximum recognition accuracy with the random initial kernel was 84.26% and the variance of accuracy was 5.37.

Next, we showed the effectiveness of pre-training. We obtained the parameters of the CRBM from training data. Next, these kernels were used as the initial value of the convolution layer for the CBN and we then trained the CBN with backpropagation. The maximum recognition accuracy using pre-training was 86.11% and the variance of accuracy for 5 trials was 0.83.

Fig. 3 also shows the average accuracies of word recognition experiments comparing two CBNs with the conventional MFCC feature. An average accuracy with pre-training is 3.58% and 2.84% higher than those without pre-training and the conventional MFCC feature, respectively. This result

shows that the learned kernel using a CRBM is more effective than the random value kernel as the initial value for CBN training.

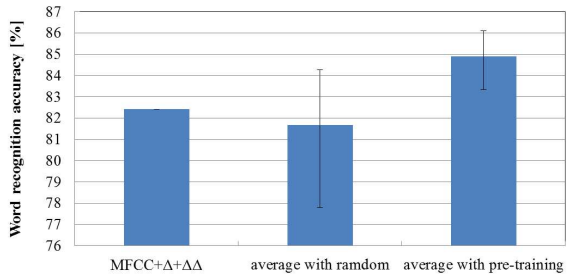

Fig. 3. Word recognition accuracy using MFCC+Δ+ΔΔ and CBN

It is thought that the reason why the recognition accuracy improved is that the network was trained adequately by pre-training using a CRBM. The backpropagation algorithm has the problem in which the error signals become weaker and weaker as they are propagated in the lower layers. Initializing the convolution kernel using a CRBM compensates for this problem. Because a CRBM is a generative model, a CRBM is learned to capture the authentic distribution of observed data. With the CBN, the convolutional kernel pre-trained by a CRBM extracts the essential information and it can propagate them to the next layer. Therefore, the network is trained efficiently compared to the case where a randomly initialized kernel is used.

5. CONCLUSION

In this paper, we presented a pre-training method for a CBN using a CRBM. Through experiments, we showed improvements in speech recognition accuracy compared with a randomly initialized convolution kernel, and the RBM pre-training helps to train networks when data are limited. In future work, we will apply pre-training to other speakers with articulation disorders and adapt a better pre-training method.

REFERENCES

[1] S. Cox and S. Dasmahapatra, "High-level approaches to confidence estimation in speech recognition," *IEEE Trans. on SAP*, vol. 10, pp. 460–471, 2002.

[2] Y. Takeuchi H. Kudo M. K. Bashar, T. Matsumoto 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] 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.

[4] H. Matsumasa, T. Takiguchi, Y. Ariki, I. LI, and T. Nakabayashi, "Integration of metamodel and acoustic model for speech recognition," in *Interspeech 2008*, 2008, pp. 2234–2237.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Intelligent Signal Processing*. 2001, pp. 306–351, IEEE Press.

[6] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *Pattern Analysis and Machine Intelligence*, 2004.

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

[8] C. Garcia T. Nakashika and T. Takiguchi, "Local-feature-map integration using convolutional neural networks for music genre classication," *Interspeech*, 2012.

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

[10] Toru Nakashika, Toshiya Yoshioka, Tetsuya Takiguchi, Yasuo Ariki, Stefan Duffner, and Christophe Garcia, "Convolutive Bottleneck Network with Dropout for Dysarthric Speech Recognition," *Transactions on Machine Learning and Artificial Intelligence*, pp. 1–15, Apr. 2014.

[11] R. Ranganath H. Lee, R. Grosse and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning, ser, ICML'09*, 2009, pp. 609–616.

[12] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[13] A. Llin K. Cho and T. Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *Artificial Neural Networks and Machine Learning*, 2011, pp. 10–17.

[14] A. Kurematsu et al., "ATR Japanese speech database as a tool of speech recognition and synthesis," in *Speech Communication*, 1990, number 4, pp. 357–363.

[15] Mohammad Norouzi, "Convolutional Restricted Boltzmann Machones for Feature Learning," 2009.

[16] "Open-Source Speech Recognition Software Julius," http://julius.sourceforge.jp/.