LEARNING ENVIRONMENTAL SOUNDS WITH END-TO-END CONVOLUTIONAL NEURAL NETWORK

Yuji Tokozume, Tatsuya Harada

The University of Tokyo, Japan

ABSTRACT

Environmental sound classification (ESC) is usually conducted based on handcrafted features such as the log-mel feature. Meanwhile, end-to-end classification systems perform feature extraction jointly with classification and have achieved success particularly in image classification. In the same manner, if environmental sounds could be directly learned from the raw waveforms, we would be able to extract a new feature effective for classification that could not have been designed by humans, and this new feature could improve the classification performance. In this paper, we propose a novel end-to-end ESC system using a convolutional neural network (CNN). The classification accuracy of our system on ESC-50 is 5.1% higher than that achieved when using logmel-CNN with the static log-mel feature. Moreover, we achieve a 6.5% improvement in classification accuracy over the state-of-the-art logmel-CNN with the static and delta log-mel feature, simply by combining our system and logmel-CNN.

Index Terms— Environmental sound classification, convolutional neural network, end-to-end system, feature learning

1. INTRODUCTION

Environmental sounds are a very diverse group of everyday audio events that can neither be described as speech nor as music [1], such as the sounding of a car horn or a knock on a door. Environmental sound classification (ESC) is typically conducted based on handcrafted features [2, 3]. One of the most powerful features for audio recognition tasks is the log-mel feature [4, 5]. This feature is calculated for each frame of sound, and represents the magnitude of each frequency area, considering human auditory perception [6]. However, the log-mel feature is designed by humans separately from other parts of the system, and was originally designed for automatic speech recognition (ASR). This suggests that there could be other effective features of ESC that humans would not be able to design.

On the other hand, end-to-end systems perform feature extraction jointly with classification, and they have achieved success particularly in image classification [7, 8, 9]. These systems automatically optimize the design of the feature extractor as connection weights of neurons; therefore, they can extract a new discriminative feature that humans are unable to design. In the same manner, if environmental sounds could be directly learned from the raw waveform, we would be able to extract a new feature representing information other than the log-mel feature, and this new feature could contribute to the improvement of classification performance.

We propose a novel end-to-end ESC system that can extract a feature that is discriminative and complementary to the log-mel feature. We evaluate the performance using ESC-50 dataset [1]. We show that the classification accuracy of our system exceeds that of logmel-CNN with the static log-mel feature by 5.1%. Moreover, we achieve a 6.5% improvement in classification accuracy over the state-of-the-art logmel-CNN with the static and delta log-mel feature, simply by combining our system and logmel-CNN. To our knowledge, this is the first work in which an end-to-end ESC system is shown to be capable of contributing to the improvement of the classification performance. Finally, we analyze the learned feature and reveal that the feature-map obtained with our system has a frequency response similar to that of human perception, with the filters ordered in a manner different from that of the log-mel feature.

2. RELATED WORK

Recently, researchers have demonstrated that it is possible to apply convolutional neural networks (CNNs) not only to image recognition tasks but also to audio recognition tasks, such as ASR [10, 11], music analysis [12], and ESC [4]. In audio recognition tasks, a CNN is applied to a two-dimensional feature-map created by arranging the log-mel features of each frame along the time axis. This feature-map exhibits locality in both the time and frequency domains [10]; therefore, we can treat this feature-map as an image and classify it accurately with a CNN in a similar way to image classification. In addition, the delta log-mel feature, the first temporal derivative of the (static) log-mel feature, is often added as the second channel of the input. Furthermore, the second temporal derivative can also be added as the third channel. These two or three inputs of static and delta log-mel features can be treated in quite a similar manner to the RGB inputs of an image [10]. We refer to the method in which a CNN is applied to a log-mel feature-map as *logmel-CNN*. One of the state-of-the-art methods of ESC is logmel-CNN with the static and delta log-mel feature, which was proposed by Piczak [4].

In ASR, an end-to-end system was proposed by Sainath et al. [5]. They showed that a raw waveform feature extracted with a convolutional layer matches the performance of the log-mel features when trained with more than $2,000$ hours of speech. Our research is highly motivated by this work, but there are some differences. The input length of our network is $24,000$ (1.5 s), which is much longer than that of Sainath et al.'s network for ASR (560 (35 ms)). Such a short duration would be appropriate for phoneme type [13], but not meaningful for environmental sound because there are much more various types of sound. Our network architecture and learning method enable to learn a long duration of sound without overfitting. Furthermore, we investigate both the most appropriate number of convolutional layers and their optimal filter size for raw feature extraction, and demonstrate that multi-convolutional layers with a small filter size are more effective than single-convolutional layer with a large filter size which Sainath et al. applied. As a result, the classification accuracy of our system exceeds that of logmel-CNN with the static log-mel feature by 5.1%, whereas Sainath et al.'s system did not exceed the system using the static log-mel features.

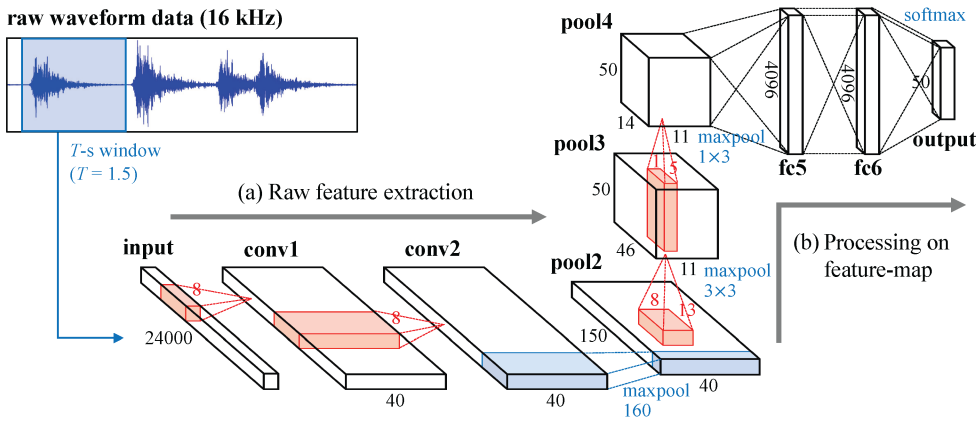

Fig. 1. **EnvNet**: End-to-end convolutional neural network for environmental sound classification.

3. END-TO-END ESC SYSTEM

In this section, we describe our novel end-to-end ESC system. In section 3.1, we present an overview of our system. In section 3.2, we show the detailed architecture of the CNN used in our system. In section 3.3, we provide the learning method of our CNN.

3.1. Overview

We use an end-to-end CNN to classify environmental sounds. We refer to our CNN as *EnvNet*. EnvNet classifies a fixed T-s section such as 1 or 2 s. When we train EnvNet, we select a T-s section randomly from the original training raw waveform data and input it into EnvNet. The selected section is different in each epoch, and we use the same training label regardless of the selected section. When we test EnvNet, we classify testing data based on probability-voting [4]. That is, we create a T-s sliding window on the testing data with a stride of 0.2 s. We input each window into EnvNet and obtain the softmax output. We take the sum of all the softmax outputs and use it to classify the testing data. Note that we do not input a silent window if its maximum amplitude is smaller than 0.2.

We assume that 1 or 2 s is a necessary and sufficient length to classify environmental sounds, regardless of the actual sound length. This assumption comes from the hypothesis that environmental sounds can be categorized into three groups: single sounds such as a mouse-click, repeated discrete sounds such as clapping hands or typing on a keyboard, and steady continuous sounds such as the sound of a vacuum cleaner or engine. If the trimmed length is too short, it becomes difficult to distinguish between single sounds and repeated sounds. If the trimmed length is too long, the ratio of silent or repeated areas, which can be thought of as not effective to the classification, becomes large and repetitive or steady sounds become redundant. We compare the performance for different T and find the best value in section 4.1.

This method makes it possible to classify various lengths of sounds universally. Furthermore, as the training data is augmented at the sample level, the system would be able to learn the raw feature without overfitting, even if the amount of training data is small.

3.2. Network architecture

The detailed architecture of EnvNet is shown in Fig. 1. ReLU is applied to each layer. In this figure, T is set to 1.5. We use a sampling rate of 16 kHz; thus, the input dimension of EnvNet is $24,000$.

3.2.1. *Raw feature extraction*

First, we apply two time-convolutional layers with a small filter size to the input raw waveform in order to extract local features, as shown in Fig. 1(a). Each convolutional layer has 40 filters, which is the same as the typical dimension of log-mel features. The filter size is 8 in all two of the layers, and we stride the filter by 1. We apply non-overlapping max pooling to the output of the convolutional layers with a pooling size of 160, which corresponds to 10 ms. The output pool2 in Fig. 1 is a two-dimensional matrix with a size of 40×150.

Pool2 is a time-series of 40-dimensional vectors, and each 40-dimensional vector can be thought of as representing frequency-like features of the corresponding 10-ms area, because of time convolution and pooling. Apart from the components of the log-mel feature being arranged according to the frequency, we assume that the components of the vector are learned to be arranged according to some type of law. Since we apply a convolutional layer in the direction of the components in the next step, we assume that the order of the components of the 40-dimensional vector will be optimized to maximize the classification performance. In this manner, pool2 has a locality, and we can treat pool2 as an image.

Contrary to the approach of Sainath et al. for ASR [5], we applied multi-convolutional layers with a very small filter size, whereas Sainath et al. applied only one large layer. We assume that our multi-convolutional layers can extract local features of various time scales hierarchically and that it is effective to classify various types of environmental sounds. We demonstrate the effectiveness of multi-convolutional layers in section 4.2.

3.2.2. *Processing on feature-map*

Next, we apply two convolutional layers and three fully connected layers to pool2 as shown in Fig. 1(b) to classify the feature-map, treating it as an image. This idea is quite similar to that of logmel-CNN [4, 10]. We need to change the direction of convolution in order to convolve in both time and frequency. We realize this process simply by reshaping pool2 from $40 \times 1 \times 150$ to $1 \times 40 \times 150$ in channel $\times$ frequency $\times$ time.

The first convolutional layer has 50 filters with a size of 8×13 in frequency $\times$ time, and we stride the filter by 1×1. We apply non-overlapping max pooling to the output with a pooling size of 3×3. The second convolutional layer has 50 filters with a size of 1×5, and we stride the filter by 1×1. We apply non-overlapping max pooling to the output with a pooling size of 1×3. The output pool4

has a size of $50 \times 11 \times 14$ and can be thought of as representing the whole feature of the sound. Finally, we apply three fully connected layers to pool4 to classify the input sound. Each fully connected layer has 4096 neurons, except for the output layer, which has as many neurons as the number of classes.

3.3. Learning method

EnvNets are trained with the cross-entropy criterion using momentum stochastic gradient descent (momentum SGD). Training is terminated after 150 epochs. We use a learning rate of 10^{-2} for the first 80 epochs, 10^{-3} for the next 20 epochs, 10^{-4} for the next 20 epochs, and 10^{-5} for the last 30 epochs.

We initialized the weights of EnvNets randomly. This is partially because it is reported that handcrafted weight initialization such as gammatone [14] initialization does not notably improve the classification performance [15, 16]; however, the main purpose is to learn another feature representation that complements handcrafted features such as the log-mel feature.

We apply 50% of dropout [17] to the fully connected layers to prevent overfitting. In addition, we apply batch normalization [18] to all the convolutional layers to accelerate the learning.

4. EXPERIMENTS

In this section, we show the result of some experiments and demonstrate the effectiveness of our method. In section 4.1, we compare the performance for different parameters to determine the best ones and demonstrate their effectiveness. In section 4.2, we compare the performance of our method to logmel-CNN. In addition, we create a new ESC system by combining our EnvNet and logmel-CNN, and show an improvement in classification performance. Finally, we analyze the feature learned with our system in section 4.3.

We evaluate the performance of the system using an environmental sound dataset ESC-50 [1]. This dataset contains a total of 2,000 samples (40 samples $\times$ 50 classes). Each sample is a monaural 5-s sound recorded with a sampling rate of 44.1 kHz. The 50 classes can be divided into 5 categories: animal sounds, natural soundscapes and water sounds, human (non-speech) sounds, interior/domestic sounds, and exterior/urban sounds. We downsample all the sound data to 16 kHz, and regularize the input vector into the range from -1 to 1. The model was evaluated with a 5-fold cross-validation scheme with a single training fold used as the validation set; thus, each model is trained with 1,200 samples. We use the fold decided by Piczak, the proposer of ESC-50.

4.1. Initial experiments

First, we conduct experiments with the aim of finding the best parameters for our system and to demonstrate their effectiveness. The performance is evaluated on the validation set.

4.1.1. Input length of EnvNet

We compare the accuracy for different values of the input length T [s] of EnvNet. The candidates of T are 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, and 5.0. We simplify the network architecture for raw feature extraction in Fig. 1(a) to only one convolutional layer with a filter size of 64. The network architecture is the same for these conditions, except for the pooling size of pool4 in Fig. 1(b). We specify this pooling size as $1 \times 2T$.

As shown in Fig. 2, the accuracy is at a high level when $T = 1.0 \sim 2.5$. There is no significant difference within that range, but the accuracy is highest at $T = 1.5$ (61.5%), which is higher than accuracy at $T = 4.0$ (57.2%) by 4.3%. We assume it is because the shorter the input length is, the denser the information of the input becomes. Moreover, when we input the whole 5-s sound without random selection ($T = 5.0$), the accuracy is only 42.1%. This suggests that our sample-level random selection is essential for the system to learn a discriminative raw feature without overfitting. However, the accuracy at $T = 0.5$ is 57.6%, which is lower than accuracy at $T = 1.0 \sim 2.5$. We assume that if the input length is too short, the input would lack the information that is needed for the classification. According to this result, we determine the value of T as 1.5.

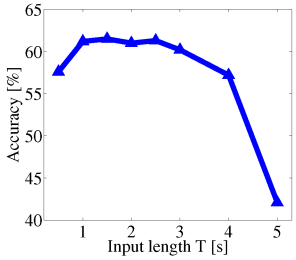

Fig. 2. Accuracy as a function of the input length T.

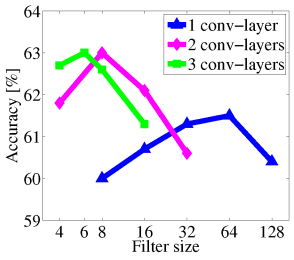

Fig. 3. Accuracy of different network architectures as a function of the filter size.

4.1.2. Architecture for raw feature extraction

We investigate both the most appropriate number of convolutional layers and their optimal filter size for raw feature extraction in Fig. 1(a). The candidates for the number of layers are 1, 2, and 3, and those for the filter size are chosen from 4, 6, 8, 16, 32, 64, and 128. The input length is 24,000 (which corresponds to 1.5 s) under all conditions. We stride the filters of each layer by 1, and apply a non-overlapping max pooling with a pooling size of 160. As the size of the feature-map created by this manipulation is unified to 40×150, the following network is completely the same under all conditions.

We summarize the result in Fig. 3. In the case of one convolutional layer, the accuracy is highest when the filter size is 64 (61.5%). In the case of two and three convolutional layers, the accuracy is highest when the filter size is 8 (63.0%) and 6 (63.0%), respectively. As indicated in this figure, multi-convolutional layers perform more accurately than single-convolutional layer. Moreover, when we increase the number of convolutional layers, the filter size should be decreased. This result is similar to that obtained by others [7, 9] for image classification. We assume that our multi-convolutional layers with a small filter size are able to extract various features hierarchically. With this experiment, we decide to use two convolutional layers for raw feature extraction, with a filter size of 8.

4.2. Results

With these two experiments, we determine the detailed parameters of our system. Now, we compare the performance of our system to logmel-CNN on the testing set. We prepare two types of logmel-CNN: one which uses the static log-mel feature as one-channel input (static logmel-CNN), and one which uses the static and delta log-mel features as two-channel input (static-delta logmel-CNN). Except for the number of input channels, these two types of logmel-CNN have completely the same architecture as that shown in Fig. 1(b). We extract a 40-dimensional static log-mel feature with a window size of 640 (40 ms) and a stride of 160 (10 ms) and then calculate the

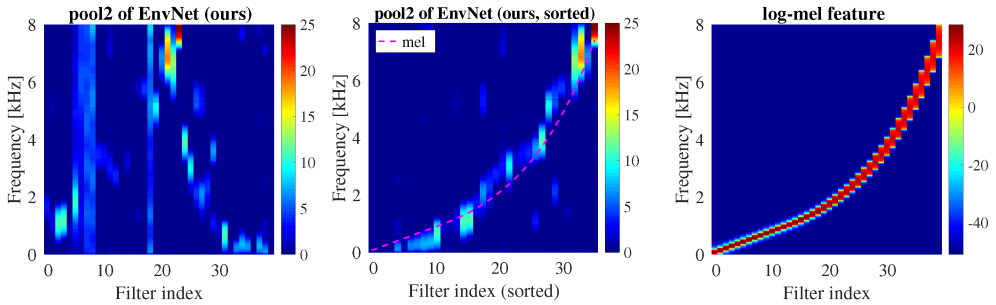

Fig. 4. Frequency response of the feature-map. **Left** shows frequency response of pool2 of EnvNet. **Middle** also shows frequency response of pool2, but the filters are sorted according to their center frequencies. **Right** shows frequency response of the log-mel feature.

Table 1. Comparison and combination with logmel-CNN. The error in this table means the standard deviation among the accuracies for the five-fold cross-validation.

logmel-CNN			
static	delta	EnvNet (ours)	Accuracy [%]
✓			58.9 ± 2.6
✓	✓		66.5 ± 2.8
		✓	64.0 ± 2.4
✓		✓	69.3 ± 2.2
✓	✓	✓	71.0 ± 3.1
Piczak logmel-CNN [4]			64.5
Human [1]			81.3

delta log-mel feature with a window size of 9. Random selection in training phase and probability-voting in testing phase are performed in all conditions.

The result is summarized in Table 1. First, the accuracy of static logmel-CNN and static-delta logmel-CNN is 58.9% and 66.5%, respectively, which is higher than the state-of-the-art static-delta logmel-CNN proposed by Piczak (64.5%) [4]. This result indicates the effectiveness of our overall learning method and network architecture for processing on feature-map shown in Fig. 1(b). Moreover, the accuracy of our system (EnvNet) is 64.0%, which is higher than static logmel-CNN by 5.1%. This result is noteworthy because, to our knowledge, there has been no end-to-end sound recognition system which exceeds the system using the static log-mel feature, including Sainath et al.'s work for ASR [5].

Next, as it is learned automatically, the feature learned with our system can be complementary to the log-mel feature. Then we combine EnvNet with logmel-CNN and investigate the classification performance. The combination method is quite simple: we use the pre-trained EnvNet and logmel-CNN, and calculate the prediction of each window for probability-voting using the average of the output of these two networks (before applying softmax).

We summarize the result again in Table 1. By combining EnvNet (64.0%) and static logmel-CNN (58.9%), we achieve an accuracy of 69.3%. Furthermore, by combining EnvNet and static-delta logmel-CNN (66.5%), we achieve an accuracy of 71.0%, which is at the state-of-the art level and constitutes an 6.5% improvement over the static-delta logmel-CNN proposed by Piczak (64.5%). To our knowledge, this is the first work in which an end-to-end ESC system is shown to be capable of contributing to the improvement of the classification performance. This result indicates that our EnvNet learns a feature capable of complementing the log-mel feature.

4.3. Analysis of learned feature

Here we present the analysis of the feature extracted with EnvNet. The magnitude of the responses of the feature-map pool2 of EnvNet (Fig. 1) are plotted in Fig. 4 (left). We created each row of this image by inputing the sine wave of corresponding frequency to EnvNet trained with all ESC-50 data. We obtain the feature-map pool2, and then the average of the feature-map along the time axis is taken.

As indicated in this figure, each of the 40 filters learns to be a band-pass filter by responding to a particular frequency area. In addition, the bandwidth of each filter increases with its center frequency. This result is similar to those obtained by other researchers [5, 15] in ASR. However, the order of the filters does not have a global regularity, which differs from the response of the log-mel feature shown in Fig. 4 (right). Instead, neighboring filters have a similar frequency response. Furthermore, as shown in Fig. 4 (middle), if we sort the filters based on their center frequency, the curve of the center frequency almost matches the mel-scale, i.e., how humans perceive the sound. This result is different from the result in ASR in which more filters are devoted to the low-frequency area [5]. Note that the filter 5, 6, 7, 8, and 18 in Fig. 4 (left), which respond to all frequency area, are removed in this figure. In this manner, as a result of optimization based on the training data, EnvNet learns a frequency response which is quite similar to human perception, but the order of the filters is optimized to maximize the classification performance in a manner that differs from the log-mel feature. We conjecture that is why our EnvNet feature is effective and has the ability to complement the log-mel feature. Hence, the classification performance is improved by combining EnvNet with logmel-CNN.

5. CONCLUSION

We proposed an end-to-end environmental sound classification system with a convolutional neural network. We achieve a 6.5% improvement in classification accuracy over the state-of-the-art logmel-CNN with the static and delta log-mel feature, simply by combining our system and logmel-CNN. Furthermore, we analyzed the feature learned with our system, and showed that our end-to-end system is capable of extracting a discriminative feature that complements the log-mel features. We assume that the application range of our system is not limited to sounds; our system could offer a solution for other signal processing tasks in the future.

6. ACKNOWLEDGEMENT

This work was funded by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan) and supported by CREST, JST.

7. REFERENCES

[1] Karol J Piczak, "ESC: Dataset for environmental sound classification," in *Proc. ACM Multimedia*, 2015.

[2] Sachin Chachada and C-C Jay Kuo, "Environmental sound recognition: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. e14, 2014.

[3] Dan Stowell and Mark D Plumbley, "Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning," *PeerJ*, vol. 2, pp. e488, 2014.

[4] Karol J Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. MLSP*, 2015.

[5] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Proc. Interspeech*, 2015.

[6] Steven B Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.

[9] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[11] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, 2015.

[12] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, "Deep content-based music recommendation," in *Proc. NIPS*, 2013.

[13] Lawrence R Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proc. IEEE*, 1989.

[14] Ralf Schluter, L Bezrukov, Hannes Wagner, and Hermann Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *Proc. ICASSP*, 2007.

[15] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proc. ICASSP*, 2015.

[16] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr.," in *Proc. Interspeech*, 2014.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.