Deep Convolutional Neural Network with Structured Prediction for Weakly Supervised Audio Event Detection

Inkyu Choi, Soo Hyun Bae and Nam Soo Kim*

Department of Electrical and Computer Engineering and INMC, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea; ikchoi@hi.snu.ac.kr (I.C.); shbae@hi.snu.ac.kr (S.H.B.)
* Correspondence: nkim@snu.ac.kr; Tel.: +82-2-880-8419

Received: 8 April 2019; Accepted: 3 June 2019; Published: 4 June 2019

Abstract: Audio event detection (AED) is a task of recognizing the types of audio events in an audio stream and estimating their temporal positions. AED is typically based on fully supervised approaches, requiring strong labels including both the presence and temporal position of each audio event. However, fully supervised datasets are not easily available due to the heavy cost of human annotation. Recently, weakly supervised approaches for AED have been proposed, utilizing large scale datasets with weak labels including only the occurrence of events in recordings. In this work, we introduce a deep convolutional neural network (CNN) model called DSNet based on densely connected convolution networks (DenseNets) and squeeze-and-excitation networks (SE Nets) for weakly supervised training of AED. DSNet alleviates the vanishing-gradient problem and strengthens feature propagation and models interdependencies between channels. We also propose a structured prediction method for weakly supervised AED. We apply a recurrent neural network (RNN) based framework and a prediction smoothness cost function to consider long-term contextual information with reduced error propagation. In post-processing, conditional random fields (CRFs) are applied to take into account the dependency between segments and delineate the borders of audio events precisely. We evaluated our proposed models on the DCASE 2017 task 4 dataset and obtained state-of-the-art results on both audio tagging and event detection tasks.

Keywords: audio event detection; weakly supervised learning; convolutional neural network; structured prediction; conditional random field

1. Introduction

People experience a variety of audio events with meaningful information that can be useful for human activities. Audio event detection (AED) aims to identify the occurrence of specific sounds in audio recordings. As the amount of multimedia data on the Internet is growing rapidly, analyzing audio events will help in describing and understanding environmental and social activities in video and audio contents. AED is also useful in many other applications, including surveillance, self-driving cars, healthcare, smart home systems, and military applications.

In early studies on AED, several approaches were proposed based on signal processing and machine learning techniques, and recently deep learning based methods have been widely developed. Most of these studies are based on fully supervised learning methods that require strongly labeled data. In strongly labeled data, either audio event examples are directly provided or the exact time of each audio event is given. However, building a large strongly labeled database is a time-consuming and challenging work. For these reasons, there exist only a few publicly available large-scale audio event datasets with strong labels.
Recently, there have been some studies on weakly supervised AED [1–4]. These studies focus on learning AED models based on weakly labeled data that provide only the presence or absence of events in the recording. We can obtain weakly labeled datasets much more easily than strongly labeled datasets. For example, we can collect videos uploaded on the Internet and use the tags (user-generated keywords which describe the content of videos) of the videos as weak labels. However, it is problematic to directly use these data for AED since the exact occurrence times of the events are not known, which makes it difficult to learn a model for segment-level predictions.

Most of the AED methods use spectro-temporal representations as input features. Since the spectro-temporal feature of an audio signal, such as log mel spectrogram, can be considered as a 2D image, computer vision techniques can be applied to AED. In recent works on computer vision, deep learning approaches including convolutional neural network (CNN) models such as the residual network (ResNet) [5], the densely connected convolution network (DenseNet) [6], and the squeeze-and-excitation network (SENet) [7] have shown impressive performance. In addition, many studies report that better results are obtained by using structured prediction methods, which consider dependencies of each pixel-level output [8,9].

In this paper, we propose a deep convolutional network based on DenseNet and SENet for weakly supervised AED. We take advantage of strengthening feature propagation from DenseNet and modeling channel-wise relationships by SENet. In addition, the correlations among segments in recordings are considered through a recurrent neural network (RNN) and conditional random field (CRF) [10]. We evaluated our proposed method and compared its performance with a CNN-based baseline approach. Empirical results show that the proposed method outperformed the baseline on the DCASE 2017 task 4 dataset.

The rest of the paper is organized as follows. In Section 2, we introduce previous works on AED and related studies. In Section 3, we present our proposed model for weakly supervised AED. Section 4 describes the experimental settings and performance measurement. In Section 5, we present the experimental results. We draw some conclusions in Section 6.

2. Related Work

Early works on AED focus on detecting audio events based on various machine learning techniques. Several approaches are proposed based on hidden Markov models (HMMs) [11,12]. In [12], Gaussian mixture model (GMM)-HMM-based modeling, similar to speech recognition techniques, is proposed to model audio events. Support vector machine (SVM) [13–15] and non-negative matrix factorization (NMF) [16–18] are also applied to AED in some studies. Bag of words representation is used to represent and detect audio events with various classifiers [19,20]. In [21], the use of multi label deep neural networks (DNNs) are proposed for detection of temporally overlapping audio events in realistic environments. Many works on AED have been proposed based on CNN [22–24]. RNNs have also been utilized in conjunction with DNN or CNN [25]. However, increasing the size of a fully supervised deep learning model is difficult due to a lack of large-scale strongly labeled datasets. This limitation can be somewhat alleviated by model regularization and data augmentation, but it is difficult to overcome the limitation completely.

There have been several studies on analyzing and detecting audio events in a weakly supervised scenario. Weakly supervised AED has been widely studied after the release of AudioSet [26], which contains more than two million 10-s YouTube clips with weak audio labels. In the early studies of weakly supervised AED, a multiple instance learning (MIL) [27] based approach is proposed in [1]. The authors formulated weakly supervised AED as a MIL problem and proposed MIL methods based on SVM and DNN. Although the training was done using weakly labeled data without temporal information, the authors showed that temporal localization of audio events was able to be extracted. In [2], the authors proposed a unified framework for supervised and weakly supervised learning (SWSL) using a graph-based model. The proposed model was able to be learned simultaneously from strongly and weakly labeled data.
Deep learning based methods have been widely proposed for weakly supervised AED and many of these methods have employed CNNs [3,4,28,29]. In [3], CNN is applied with an event-specific Gaussian filter layer, which is designed to improve its learning ability. McFee et al. [4] proposed a CNN structure with adaptive pooling operators to aggregate temporally dynamic predictions. Kumar and Raj [28] used CNN to scan and produce outputs at small segments and then mapped these segment-level outputs to full recording level outputs. Kumar et al. [29] used transfer learning to effectively convey knowledge from weakly labeled web audio data to the target data. In the DCASE 2017 [30], most of the top performing methods on the weakly labeled task relied on CNNs [31–33].

Recent improvements in computer hardware have enabled training very deep CNNs. However, this is not easy due to the problem of vanishing/exploding gradients particularly in lower layers. Many algorithms have been proposed to solve this problem such as ResNet [5]. ResNet introduces a residual block that sums a non-linear transformation of the input and its identity mapping. The identity mapping is implemented through a shortcut connection, which makes the networks avoid the vanishing gradient problem. The shortcut connections help to improve the performance of the networks and obtain faster convergence of training. As an extension of ResNets, a new CNN architecture, called DenseNet, is introduced in [6]. DenseNet is built from stacks of dense blocks and pooling operations. The dense blocks consist of multiple layers with direct connections from any layer to all subsequent layers to improve the information flow between layers.

In [7], the authors focused on the channel relationship and proposed a novel architectural unit, the squeeze-and-excitation (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. They proposed to squeeze global spatial information into a channel descriptor and modeled channel-wise relationships using a lightweight gating mechanism. They demonstrated that SE blocks brought significant improvements in the performance of the state-of-the-art CNNs at a minimal additional computational cost.

CRFs have been employed to enforce structure consistency in semantic segmentation. In [34], a fully connected CRF is used to consider the structural properties of the segmentation outputs. More recently, deep learning models integrating the densely connected CRF are proposed in many studies. DeepLab [8] proposes deep CNNs with atrous convolution, which is convolution with upsampled filters, and combines the responses at the final layer with a fully connected CRF. In [9], an RNN is introduced to approximate the mean-field iterations of CRF optimization, allowing for an end-to-end training of both the fully convolutional network and the RNN.

In this section, we describe our weakly supervised AED model, referred to as DSNet. The overall structure of the DSNet is depicted in Figure 1. The input of DSNet is a log mel spectrogram image \( X \in R^{N \times M} \), where \( N \) denotes the number of segments and \( M \) is the number of mel filterbanks. First, convolution is performed on the log mel spectrogram images to extract feature maps. We use four DS blocks, which consist of a dense block, an SE block, and a max-pooling layer. Two fully connected layers are applied for segment-level prediction. To detect overlapping audio events simultaneously, we define AED as a multi-label classification problem. For this, segment-level predictions are calculated using sigmoid activation functions at the final fully connected layer. A global pooling layer is applied for clip-level prediction. For structured prediction, DSNet with an RNN is proposed and a fully connected CRF is applied as a post-processing method. The detailed architectures of DSNet and DSNet-RNN are given in Section 4.
Figure 1. Overview of the proposed DSNet for weakly supervised AED: (a) the architecture of DSNet, where the dense layer marked with * is replaced by a recurrent layer in DSNet-RNN; (b) the schema of the dense block. C denotes a concatenation operation; and (c) the schema of the SE block.

3. Method

3.1. DenseNet

In a standard CNN, the output of the \( l \)th layer \( x_l \) is calculated by applying a non-linear transformation to the output of the previous layer \( x_{l-1} \)

\[
x_l = L_l(x_{l-1})
\]

where \( L_l \) is a convolution followed by a non-linearity activation function. Conventional CNNs consist of a stack of convolutional layers. However, deeper CNNs are more difficult to train due to vanishing gradients. In ResNet [5], residual blocks are used to train deeply structured neural networks. A residual block sums the identity mapping of the input to the output of the layer. The output \( x_l \) of a residual block is given by

\[
x_l = H_l(x_{l-1}) + x_{l-1}
\]

where \( H_l \) is a non-linear transformation, which usually consists of a single layer or a stack of multiple layers. The identity mapping acts as a skip connection from a lower layer to the upper layer, which enables input features to be reused and the gradient to flow directly from the upper layer to the lower layer.

DenseNet [6] is built from stacks of dense blocks. To improve the information flow between layers, DenseNet uses skip connections from any layer to all subsequent layers in each dense block. Each layer in dense blocks can be expressed as

\[
x_l = L_l([x_{l-1}, \ldots, x_0])
\]

where \([ \ ]\) represents the concatenation of feature maps. DenseNet may look similar to ResNet, which introduces skip connections, however this small modification makes a noticeable difference between the two networks. DenseNet is more efficient than ResNet in parameter usage. Thanks to short connections to all feature maps in the architecture, information from previously computed feature maps can be reused easily.
We use four convolutional layers and a single bottleneck layer (a layer that contains few channels compared to the previous layer) in each dense block. To improve computational efficiency, the bottleneck layer compresses all feature maps in the dense block into a reduced number of feature maps using $1 \times 1$ convolution. For all convolutional layers in the model, each side of the inputs is zero-padded by one pixel to keep the feature map size fixed and batch normalization is applied before a rectified linear unit (ReLU) for better training performance. We use more feature maps on the upper dense blocks to compensate for feature map size reduction in each max pooling layer.

3.2. Squeeze-and-Excitation

We use SE blocks [7] to consider interdependencies between channels. In the SE block, global spatial information is squeezed into a channel descriptor using global average pooling. A channel descriptor $z \in \mathbb{R}^C$ is extracted by averaging the input feature map $U \in \mathbb{R}^{H \times W \times C}$ through its spatial dimensions $H \times W$. To utilize the information aggregated in the squeeze operation, a simple gating mechanism is employed. The channel descriptor $z$ is transformed into a set of channel weights $s \in \mathbb{R}^C$, which is given by

$$s = \sigma(W_2 \delta(W_1 z))$$

(4)

where $\sigma$ and $\delta$, respectively, refer to sigmoid and rectified linear functions. To reduce model complexity and aid generalization, a bottleneck is formed by $W_1 \in \mathbb{R}^{C \times r \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$. We set the dimensionality-reduction ratio $r$ to 4 in our system. The final output of the SE block is obtained by scaling $U$ with channel weights $s$ for each channel. In this manner, channels possessing more important information can be emphasized.

3.3. Global Pooling for Aggregation

The proposed DSNet aims to predict both segment-level and clip-level labels. To train DSNet with only weak labels (clip-level labels), we need to aggregate segment-level predictions to form clip-level predictions. A common approach would be taking an average over all segment predictions corresponding to a clip-level prediction. In this approach, all segments of the clip have the same influence on the clip-level prediction. However, clips with a positive label can also contain negative segments, which disturb the training process. In the multiple instance learning framework, a global max-pooling is applied to aggregate segment-level predictions into a clip-level prediction. In the max pooling approach, the clip-level prediction focuses on the most positive segment in the clip and disturbance from negative segments can be reduced. However, with global max-pooling aggregation, only the most positive segment in each clip is active in training during backpropagation and other segments are ignored.

To take advantage of both methods, we apply the LogSumExp (LSE) function, which is a smooth approximation of the max function. The LSE function is given as

$$y_k = \frac{1}{\alpha} \log \left( \frac{1}{T} \sum_{t=1}^{T} \exp(\alpha \cdot s_{t,k}) \right)$$

(5)

where $y_k$ is a clip-level prediction for class $k$, $s_{t,k}$ is the segment label of the $t$th segment for class $k$ and $T$ is the number of segments in a clip. In Equation (5), $\alpha$ is a hyperparameter (a parameter whose value is set before training) to control the sharpness of the function. As $\alpha$ increases, the function approaches the max function and, as $\alpha$ decreases, the function approaches the average function. With the LSE pooling, we can use all the segments of the clip during training and also focus on positive segments in positive clips. We set the parameter $\alpha = 0.5$ in our system. To train our model with only weak labels, we apply the mean square error as the cost function, which is given by

$$C_{cl} = \frac{1}{N} \sum_{n=1}^{N} ||L_n - Y_n||^2$$

(6)
where \( L_n \) denotes the true label for the \( n \)th clip, \( Y_n \) is the clip-level prediction for the \( n \)th clip, and \( N \) is the number of clips in the training data.

### 3.4. Structured Prediction for Accurate Event Localization

#### 3.4.1. RNN-Based Structured Prediction

Segment-level prediction can be performed using DSNet as described above. However, segment-level predictions may not be robust since DSNet does not make good use of long-term contextual information. Better segment-level prediction results can be obtained by considering long-term contextual information and incorporating prior knowledge into our model. To consider long-term dependency between segment predictions, an RNN is applied at the top of DSNet. We refer to DSNet with RNN as DSNet-RNN. The structure of DSNet-RNN is almost the same with DSNet except that the dense layer marked with \(*\) in Figure 1 is replaced by a single layer RNN with bi-directional gated recurrent units (Bi-GRUs) \([35]\). However, in weakly supervised learning, there is a lack of accurate label information on each segment. The incorrect information on each segment may affect other segments through the RNN. To mitigate this problem, it is desirable to train our model by applying some prior knowledge that audio events are generally continuous. To utilize this prior knowledge, we define a prediction smoothness cost \( C_{ps} \) as

\[
C_{ps} = \sum_{i,j=1}^{N} \mu(y_i, y_j)s(i, j),
\]

\[
\mu(y_i, y_j) = ||y_i - y_j||,
\]

\[
s(i, j) = \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_{ps}^2}\right),
\]

where \( y_i \) is the segment prediction of the \( i \)th segment and \( p_i \) denotes its normalized temporal position. The prediction smoothness cost \( C_{ps} \) encourages segment predictions to be continuous over time by penalizing nearby segments with different predictions. The cost function for training the DSNet-RNN is given by

\[
C = C_{cl} + \lambda C_{ps}
\]

where \( \lambda \) is a compromising parameter.

#### 3.4.2. CRF Post-Processing

As the proposed model can produce segment-level predictions, we can determine the border of audio events through post-processing. A common approach is to smooth the segment-level predictions and threshold them for boundary decision. However, since this approach does not take dependency between the segments into account, it is not easy to determine the borders of audio events precisely. To address this issue, we apply CRF for post-processing the segment-level predictions. To reflect the full relationship among segments, we incorporate the fully connected CRF model proposed in \([34]\) into our system.

In the conventional approach, segment-level predictions \( y_i \) are smoothed and thresholded for segment-level classification. The threshold value \( th_v \) is determined to have the best F1 score on the validation set. In the CRF post-processing approach, label assignment probability of each class for the \( i \)th segment \( P(i) \) is calculated as

\[
P(i) = \text{sigmoid}(y_i - th_v).
\]

The energy function for the fully connected CRF is given as

\[
E(x) = \sum_i \theta_i + \sum_{ij} \theta_{ij},
\]
\[ \theta_i = -\log(P(i)), \]

\[ \theta_{ij} = \mu(i,j) \ast \left[ w_{\text{mel}} \ast \exp\left( -\frac{||m_i - m_j||^2}{2\sigma_{\text{mel}}^2} \right) \right. \]

\[ \left. + w_{\text{pos}} \ast \exp\left( -\frac{||p_i - p_j||^2}{2\sigma_{\text{pos}}^2} \right) \right], \]

where \( \theta_i \) represents the unary potential at the \( i \)th segment and \( \theta_{ij} \) is the pairwise potential between the \( i \)th and \( j \)th segments. In the pairwise potential, \( \mu(i,j) = 1 \) if the \( i \)th and \( j \)th segments have different label assignments, and zero otherwise. \( p_i \) denotes the temporal position and \( m_i \) is the log mel spectrum of the \( i \)th segment. The hyperparameters \( w_{\text{mel}}, \sigma_{\text{mel}}, w_{\text{pos}}, \) and \( \sigma_{\text{pos}} \) control the Gaussian kernels. The pairwise potential penalizes segments with similar log mel spectra and positions having different labels. This model can efficiently infer the probabilities using mean field approximation and efficient message passing through high-dimensional filtering [34].

### 4. Experiments

#### 4.1. Dataset

The DCASE 2017 task 4 Dataset [30] was published for the task of “Large-scale weakly supervised sound event detection for smart cars” in the DCASE 2017 challenge. The dataset employs a subset of AudioSet by Google [26]. The DCASE 2017 task 4 Dataset consists of 17 audio events divided into two categories: “Warning” and “Vehicle”. The dataset contains audio classes for self-driving cars, smart cities and related areas. The dataset contains 51,172 clips of training set, 488 clips of validation set and 1103 clips of evaluation set. Every clip is less than 10 s long. Each clip may correspond to more than one audio event and possibly has overlapping audio events. The dataset is obtained by collecting real-life recordings that contain noise and unknown class signals. The training set has weak labels denoting the presence of a given audio event in the clip and no timestamps are provided. For the validation and evaluation sets, strong labels with timestamps are provided for the purpose of performance evaluation.

#### 4.2. Metrics

In our work, both clip-level and segment-level evaluation metrics were used. The default segment length used in this work was 100 ms, which is shorter than the segment length used in the DCASE challenge, 1 s. This was because our system aims to detect audio events accurately in time via structured prediction. Since the dataset for evaluation has multi-label annotations, we used the metrics proposed in [36]. F1 score with precision (P) and recall (R) was calculated as the primary evaluation metric for both clip-level and segment-level evaluation. For segment-level evaluation, segment-based error rate (ER) was also measured. A detailed explanation of both evaluation metrics is described in [36].

#### 4.3. Feature Extraction

As inputs to the neural networks, we used log mel band features. We extracted 128 mel bands from 0 Hz to 22,050 Hz. We applied a window size of 1100 samples with a shift of 365 samples for frame segmentation to produce 800 frames in a 10-s clip. The logarithms of the mel band energies were calculated and each log mel energy was normalized by subtracting its mean and dividing by its standard deviation computed over the training set. As a result, a 800 \( \times \) 128 normalized log mel spectrogram image was extracted for each 10-s clip.

#### 4.4. DSNet and DSNet-RNN Structures

The specific configuration of the proposed model is described in Table 1. The extracted normalized log mel spectrogram image was used for input to the neural networks. A convolution layer was used to produce feature maps for dense blocks. These networks consisted of four dense blocks each...
with four convolution layers and one bottleneck layer. The convolution layers consisted of three consecutive operations: 3 × 3 convolution, batch normalization and ReLU. We used 1 × 1 convolution layer to reduce channels. An SE block and a max pooling layer were placed after each dense block. For segment-level prediction, two dense layers were applied in the DSNet, and Bi-GRU and a dense layer were applied in the DSNet-RNN. Finally, the segment-level predictions were aggregated through the global pooling layer for clip-level prediction. We set σ ps = 0.1 in Equation (7) and λ = 0.01 in Equation (8) to train the DSNet-RNN. The parameter size of the DSNet is 0.32M, which is similar to that of the baseline CNN. The DSNet-RNN has more parameters than the others due to the Bi-GRUs used for structured prediction.

Table 1. DSNet and DSNet-RNN architectures.

| Layers               | Output Size       | DSNet             | DSNet-RNN          |
|----------------------|-------------------|-------------------|--------------------|
| Convolution          | 800 × 128 × 32    | [3 × 3, 32 conv]  |                    |
| Dense block          | 800 × 128 × 32    | [3 × 3, 16 conv] × 4 | [1 × 1, 16 conv] × 4 |
| SE block             | 800 × 128 × 32    | bottleneck size 8 |                    |
| Max-pooling          | 800 × 64 × 32     | 1 × 2 max pool    |                    |
| Dense block          | 800 × 64 × 48     | [3 × 3, 16 conv] × 4 | [1 × 1, 48 conv] × 4 |
| SE block             | 800 × 64 × 48     | bottleneck size 12|                    |
| Max-pooling          | 400 × 32 × 48     | 2 × 2 max pool    |                    |
| Dense block          | 400 × 32 × 64     | [3 × 3, 16 conv] × 4 | [1 × 1, 64 conv] × 4 |
| SE block             | 400 × 32 × 64     | bottleneck size 16|                    |
| Max-pooling          | 200 × 16 × 64     | 2 × 2 max pool    |                    |
| Dense block          | 200 × 16 × 64     | [3 × 3, 16 conv] × 4 | [1 × 1, 64 conv] × 4 |
| SE block             | 200 × 16 × 64     | bottleneck size 16|                    |
| Max-pooling          | 100 × 8 × 64      | 2 × 2 max pool    |                    |
| Reshape              | 100 × 512         | 100 × 8 × 64 to 100 × 512 |                |
| Segment-level prediction | 100 × 17      | 256 dense(ReLU)     | 128 Bi-GRUs   |
|                      |                   | 17 dense(sigmoid) | 17 dense(sigmoid) |
| Clip-level prediction | 100 × 17        | global LSE pooling |                    |
| Parameters           | -                 | 0.32 M           | 0.69 M           |

4.5. Baseline CNN Structure

To verify the performance of the proposed method, we compared the proposed method with a baseline model. In the DCASE 2017 challenge, several CNN-based models were proposed and showed good performance in weakly supervised AED [31–33]. We chose a CNN baseline model similar to the models proposed in the DCASE 2017 Challenge. The specific configuration of the baseline model is described in Table 2. The audio feature for the baseline was the same as that of the proposed model, a 800 × 128 normalized log mel spectrogram image. The baseline model consisted of four stacks of two convolution layers and a max pooling layer. The last max pooling layer was connected to two dense layers to produce segment-level predictions, and the segment-level predictions were aggregated in the global pooling layer.
Table 2. Baseline CNN architecture.

| Layers       | Output Size       | CNN                        |
|--------------|-------------------|----------------------------|
| Convolution  | $800 \times 128 \times 32$ | $[3 \times 3, 32 \text{ conv}] \times 2$ |
| Max-pooling  | $800 \times 64 \times 32$ | $1 \times 2 \text{ max pool}$ |
| Convolution  | $800 \times 64 \times 32$ | $[3 \times 3, 32 \text{ conv}] \times 2$ |
| Max-pooling  | $400 \times 32 \times 32$ | $2 \times 2 \text{ max pool}$ |
| Convolution  | $400 \times 32 \times 64$ | $[3 \times 3, 64 \text{ conv}] \times 2$ |
| Max-pooling  | $200 \times 16 \times 64$ | $2 \times 2 \text{ max pool}$ |
| Convolution  | $200 \times 16 \times 64$ | $[3 \times 3, 64 \text{ conv}] \times 2$ |
| Max-pooling  | $100 \times 8 \times 64$ | $2 \times 2 \text{ max pool}$ |
| Reshape      | $100 \times 512$ | $100 \times 8 \times 64$ to $100 \times 512$ |
| Segment-level prediction | $100 \times 17$ | 256 dense(ReLU) |
|              |                   | 17 dense(sigmoid) |
| Clip-level prediction | 17 | global LSE pooling |
| Parameters   | -                 | 0.29 M                    |

4.6. Training and Evaluation

The neural network models were implemented using TensorFlow [37]. We set the hyperparameters such that they provided the highest segmental F1 score on the validation set. All networks were trained with Adam (an algorithm for first-order gradient-based optimization of stochastic objective functions) [38]. A dropout [39] rate of 0.1 was applied to the output of the SE blocks and the dense layer with ReLU. We used mini-batches (a subset of data used for updating parameters during one iteration) of 10 clips and a learning rate of 0.0001. We used the validation set to earlystop (stop training based on the validation error to avoid overfitting) the training based on the segmental F1 score. To deal with the unbalance between classes on the training set, we applied undersampling to the classes with more than 1000 clips. The networks were trained on NVIDIA Tesla M40 GPUs.

For evaluation, the optimal thresholds were selected to have the best performance on the validation set. The segment-level predictions were smoothed with a Hanning window of length 41 before thresholding. We set the CRF parameters in Equation (12) to $w_{\text{mel}} = 1$, $c_{\text{mel}} = 1$, $w_{\text{pos}} = 1$ and $\sigma_{\text{pos}} = 25$, which showed the best segment-level F1 score on the validation set. To perform multi-labeled classification, CRF post-processing was performed separately for each class. We employed 10 mean field iterations in the test phase.

5. Results and Discussion

5.1. Audio Tagging

Table 3 presents the clip-level tagging results on the DCASE 2017 task 4 evaluation set and parameter sizes of each model. The results show that the DSNet had an absolute improvement of 0.0347 over the baseline CNN in terms of F1 score. The performance of the DSNet indicates that DenseNet and SENet are suitable not only for image processing but also for audio processing. The DSNet-RNN showed almost the same performance as the DSNet in clip-level metrics, which means the structured prediction has little effect on the clip-level performance.

The class-wise F1 score results for the CNN, DSNet and DSNet-RNN models are presented in Table 4. While there was some variation across classes, the DSNet and DSNet-RNN showed better performance than CNN on most classes. The performance of the DSNet was considerably better compared to the baseline CNN for the “air horn, truck horn”, “police car”, “skateboard” and “motorcycle” classes and the DSNet-RNN showed better performance than the baseline CNN in the
“air horn, truck horn”, “police car”, “screaming” and “motorcycle” classes. The best performing class for all models was “civil defense siren”, which consists of long and high volume sounds, and the worst performing class was “car passing by”, which consists of short and low volume sounds.

Table 3. Clip-level results on the DCASE 2017 task 4 evaluation set.

| Model       | F1    | P     | R     |
|-------------|-------|-------|-------|
| CNN         | 0.5506| 0.5667| 0.5353|
| DSNet       | 0.5853| 0.5822| 0.5883|
| DSNet-RNN   | 0.5839| 0.5504| 0.6281|

Table 4. Class-wise clip-level F1 score results.

| Class                      | CNN     | DSNet   | DSNet-RNN |
|----------------------------|---------|---------|-----------|
| Train horn                 | 0.5273  | 0.4615  | 0.5102    |
| Air horn, truck horn       | 0.4000  | 0.5455  | 0.5783    |
| Car alarm                  | 0.4267  | 0.4500  | 0.3836    |
| Reversing beeps            | 0.3373  | 0.3765  | 0.4186    |
| Ambulance                  | 0.5556  | 0.4681  | 0.4854    |
| Police car                 | 0.4906  | 0.5778  | 0.6525    |
| Fire engine, fire truck    | 0.5606  | 0.6055  | 0.5586    |
| Civil defense siren        | 0.7704  | 0.8160  | 0.8189    |
| Screaming                  | 0.6833  | 0.7059  | 0.8333    |
| Bicycle                    | 0.4675  | 0.4615  | 0.3294    |
| Skateboard                 | 0.5946  | 0.7627  | 0.6372    |
| Car                        | 0.6266  | 0.6759  | 0.6411    |
| Car passing by             | 0.2727  | 0.2931  | 0.2468    |
| Bus                        | 0.4238  | 0.4000  | 0.2637    |
| Truck                      | 0.4455  | 0.4541  | 0.4505    |
| Motorcycle                 | 0.5465  | 0.6324  | 0.7009    |
| Train                      | 0.7209  | 0.7883  | 0.7759    |

5.2. Event Detection with Localization

Table 5 presents the segment-level results on the DCASE 2017 task 4 evaluation set. Both the DSNet and DSNet-RNN outperformed the baseline CNN model in F1 score by 0.0148 and 0.0367, respectively. Similar to the clip-level results, the DSNet performed better than conventional CNN by using DenseNet and SENet. Especially, the DSNet-RNN showed the best performance in segment-level results. This indicates that each segment-level prediction benefits from considering contextual information in the neural network.

Table 5. Segment-level results on the DCASE 2017 task 4 evaluation set.

| Model       | F1    | P     | R     | ER  |
|-------------|-------|-------|-------|-----|
| CNN         | 0.4987| 0.4598| 0.5447| 0.7568|
| DSNet       | 0.5135| 0.4746| 0.5593| 0.7039|
| DSNet-RNN   | 0.5354| 0.5074| 0.5667| 0.6213|

The weight $\lambda$ introduced in Equation (8) is a hyperparameter, which allows us to control the dependency of the cost function on structured prediction. The effect of the weight $\lambda$ on the DSNet-RNN is presented in Table 6. The result shows that, when $\lambda = 0$, the model did not show significant performance improvement over the DSNet. This means that the flow of uncertain information in the RNN may hinder the training of the model in weakly supervised learning. The overall results show that the performance of the model could be improved by restricting uncertain information flow with appropriate constraints based on prior information. The model had the best performance when $\lambda = 0.01$, which was used in training the DSNet-RNN.
Table 6. Segment-level results for the DSNet-RNN at different $\lambda$.

| $\lambda$ | F1   | ER   |
|-----------|------|------|
| 0         | 0.5168 | 0.6564 |
| 0.005     | 0.5184 | 0.7048 |
| 0.01      | 0.5354 | 0.6213 |
| 0.02      | 0.5281 | 0.6867 |
| 0.05      | 0.5039 | 0.8109 |

Table 7 presents the influence of CRF post-processing on the segment-level performances. All models show performance improvement through CRF post-processing. In the DSNet-RNN, the performance improvement was relatively low. This indicates that the DSNet-RNN already reflected contextual information and hence the additional benefit from CRF post-processing is relatively small. The results of the DSNet-RNN with and without CRF post-processing are visualized in Figure 2. We could correct isolated inaccurate predictions and improve the predictions particularly in the boundaries of the events by employing CRF post-processing.

![Figure 2](image)

Figure 2. The results of the DSNet-RNN before and after CRF: (a) log mel spectrograms of audio events; (b) segment-level ground truth labels; (c) predicted segment-level labels before CRF; and (d) predicted segment-level labels after CRF.

Table 7. Effect of CRF on segment-level performance.

| Model      | Before CRF | After CRF |
|------------|------------|-----------|
|            | F1         | ER        | F1    | ER    |
| CNN        | 0.4987     | 0.7568    | 0.5195 | 0.6680 |
| DSNet      | 0.5135     | 0.7039    | 0.5265 | 0.6849 |
| DSNet-RNN  | 0.5354     | 0.6213    | 0.5432 | 0.6131 |

5.3. Comparison with the DCASE 2017 Task 4 Results

For comparison, the results of our models and the top results from the DCASE 2017 task 4 are presented in Table 8. In the DCASE 2017 task 4, Xu et al. [33] and Lee et al. [31] showed the best performance in audio tagging (clip-level) and sound event detection (segment-level), respectively.
Xu et al. [33] used the learnable gated activation function and Lee et al. [31] used a multi-scale input framework. Both also used the fusion or ensemble of models for the best performance. For a fair comparison, we compared the segment-level results of our proposed model in 1 s time resolution. Our models showed better performance in both clip-level and segment-level results, even without the fusion or ensemble of models. The proposed models outperformed Xu et al. [33] in clip-level F1 score. In the segment-level metrics, the DSNet-RNN achieved a similar performance as Lee et al. [31] in F1 score and showed a better performance in ER.

Table 8. Comparison with the DCASE 2017 results on evaluation set.

| Model                | F1<sub>tag</sub> | F1<sub>1s</sub> | ER<sub>1s</sub> |
|----------------------|-----------------|-----------------|-----------------|
| DSNet                | 0.585           | 0.530           | 0.689           |
| DSNet + CRF          | 0.585           | 0.542           | 0.644           |
| DSNet-RNN            | 0.584           | 0.550           | 0.606           |
| DSNet-RNN + CRF      | 0.584           | 0.557           | 0.570           |
| Xu et al. [33]       | 0.556           | 0.518           | 0.730           |
| Lee et al. [31]      | 0.526           | 0.555           | 0.660           |

6. Conclusions

In this paper, we propose DSNet, which is a combination of DenseNet and SENet, for weakly supervised AED. DSNet allows better information and gradient flow through direct connections between any two layers in dense blocks and adaptively recalibrates channel-wise feature responses using SE blocks. Moreover, we propose a structured prediction framework and adopted it to DSNet. DSNet-RNN utilizes contextual information while minimizing the propagation of uncertainty and CRF post-processing helps to refine segment-level predictions. Experiments showed that DSNet with structured prediction achieved state-of-the-art results in the DCASE 2017 task 4 dataset.

Author Contributions: Conceptualization, I.C. and N.S.K.; Funding acquisition, N.S.K.; Investigation, I.C.; Methodology, I.C.; Software, I.C. and S.H.B.; Writing—original draft, I.C.; and Writing—review and editing, N.S.K.

Acknowledgments: This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, A.; Raj, B. Audio event detection using weakly labeled data. In Proceedings of the ACM on Multimedia Conference, Amsterdam, The Netherlands, 15–19 October 2016; pp. 1038–1047.
2. Kumar, A.; Raj, B. Audio event and scene recognition: A unified approach using strongly and weakly labeled data. arXiv 2016, arXiv:1611.04871.
3. Su, T.W.; Liu, J.Y.; Yang, Y.H. Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 791–795.
4. McFee, B.; Salamon, J.; Bello, J.P. Adaptive pooling operators for weakly labeled sound event detection. arXiv 2018, arXiv:1804.10070.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
6. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
7. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
8. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]

9. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015; pp. 1529–1537.

10. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, Williamsport, MA, USA, 28 June–1 July 2001; pp. 282–289.

11. Zhou, X.; Zhuang, X.; Liu, M.; Tang, H.; Hasegawa-Johnson, M.; Huang, T. HMM-based acoustic event detection with AdaBoost feature selection. In *Multimodal Technologies for Perception of Humans*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 345–353.

12. Mesaros, A.; Heittola, T.; Eronen, A.; Virtanen, T. Acoustic event detection in real life recordings. In *Proceedings of the 18th European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, 23–27 August 2010; pp. 1267–1271.

13. Temko, A.; Nadeu, C. Classification of acoustic events using SVM-based clustering schemes. *Pattern Recognit.* **2006**, *39*, 682–694. [CrossRef]

14. Temko, A.; Nadeu, C. Acoustic event detection in meeting-room environments. *Pattern Recognit. Lett.* **2009**, *30*, 1281–1288. [CrossRef]

15. Portelo, J.; Bugalho, M.; Trancoso, I.; Neto, J.; Abad, A.; Serralheiro, A. Non-speech audio event detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, 19–24 April 2009; pp. 1973–1976.

16. Chin, M.L.; Burred, J.J. Audio event detection based on layered symbolic sequence representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 25–30 March 2012; pp. 1953–1956.

17. Gemmeke, J.F.; Vaeugen, L.; Karsmakers, P.; Vanrumste, B.; hamme, H.V. An exemplar-based NMF approach to audio event detection. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 20–23 October 2013; pp. 1–4.

18. Mesaros, A.; Heittola, T.; Dikmen, O.; Virtanen, T. Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 9–24 April 2015; pp. 151–155.

19. Pancoast, S.; Akbacak, M. Bag-of-audio-words approach for multimedia event classification. In *Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association*, Portland, OR, USA, 9–13 September 2012; pp. 2105–2108.

20. Lu, X.; Tsao, Y.; Matsuda, S.; Hori, C. Sparse representation based on a bag of spectral exemplars for acoustic event detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 4–9 May 2014; pp. 6255–6259.

21. Cakir, E.; Heittola, T.; Huttunen, H.; Virtanen, T. Polyphonic sound event detection using multi label deep neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 12–17 July 2015, pp. 1–7.

22. Zhang, H.; McLoughlin, I.; Song, Y. Robust sound event recognition using convolutional neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 19–24 April 2015; pp. 559–563.

23. Phan, H.; Hertel, L.; Maass, M.; Mertins, A. Robust audio event recognition with 1-max pooling convolutional neural networks. *arXiv 2016*, arXiv:1604.06338.

24. Cakir, E.; Parascandolo, G.; Heittola, T.; Huttunen, H.; Virtanen, T.; Cakir, E.; Parascandolo, G.; Heittola, T.; Huttunen, H.; Virtanen, T. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP)* **2017**, *25*, 1291–1303. [CrossRef]

25. Lee, J.; Kim, T.; Park, J.; Nam, J. Raw waveform-based audio classification using sample-level CNN architectures. *arXiv 2017*, arXiv:1712.00866.
26. Gemmeke, J.F.; Ellis, D.P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R.C.; Plakal, M.; Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 776–780.
27. Maron, O.; Lozano-Pérez, T. A framework for multiple-instance learning. In Advances in Neural Information Processing Systems; MIT Press: Cambridge, MA, USA, 1998; pp. 570–576.
28. Kumar, A.; Raj, B. Deep CNN framework for audio event recognition using weakly labeled web data. arXiv 2017, arXiv:1707.02530.
29. Kumar, A.; Khadkevich, M.; Fügen, C. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 326–330.
30. Mesaros, A.; Heittola, T.; Diment, A.; Elizalde, B.; Shah, A.; Vincent, E.; Raj, B.; Virtanen, T. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In Proceedings of the DCASE2017 Workshop, Munich, Germany, 16–17 November 2017.
31. Lee, D.; Lee, S.; Han, Y.; Lee, K. Ensemble of Convolutional Neural Networks for Weakly-Supervised Sound Event Detection Using Multiple Scale Input; Technical Report; DCASE2017 Challenge: Tampere, Finland, 2017.
32. Lee, J.; Park, J.; Nam, J. Combining Multi-Scale Features Using Sample-Level Deep Convolutional Neural Networks for Weakly Supervised Sound Event Detection; Technical Report; DCASE2017 Challenge: Tampere, Finland, 2017.
33. Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M.D. Surrey-CVSSP System for DCASE2017 Challenge Task4; Technical Report; DCASE2017 Challenge: Tampere, Finland, 2017.
34. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In Advances in Neural Information Processing Systems; MIT Press: Cambridge, MA, USA, 2011; pp. 109–117.
35. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv 2014, arXiv:1406.1078.
36. Mesaros, A.; Heittola, T.; Virtanen, T. Metrics for polyphonic sound event detection. Appl. Sci. 2016, 6, 162.
37. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv 2016, arXiv:1603.04467.
38. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.
39. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 2014, 15, 1929–1958.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).