ATTENTION-BASED CNN WITH GENERALIZED LABEL TREE EMBEDDING FOR AUDIO SCENE CLASSIFICATION

Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, and Alfred Mertins

University of Lübeck, Institute for Signal Processing, Lübeck, Germany
{phan,koch,katzberg,maass,mazur,mertins}@isip.uni-luebeck.de

ABSTRACT
This report presents our audio scene classification systems submitted for Task 1 (“acoustic scene classification”) of DCASE 2017 challenge [1]. The systems rely on combinations of generalized label tree embedding representation, convolutional neural networks (CNNs), and attention mechanism. Our experimental results on the development data of the challenge show that our proposed system significantly outperform the challenge’s baseline, improving the average classification accuracy from 74.8% of the baseline to 83.8%. However, we achieve significantly lower accuracies on the evaluation data, underperforming the DCASE baseline, due to overfitting caused by cross-validation errors in our submission systems.

Index Terms— audio scene classification, CNN, attention, generalized label tree embeddings

1. INTRODUCTION
Label tree embedding (LTE) [2] has been shown to be efficient in transforming and reducing complex audio scenes into semantic representations to expose their useful patterns. As a result, these representations facilitates training deep networks for recognition, such as template matching with CNNs [3, 4] and sequence modeling with RNNs [5], with good performances reported for audio scene classification (ASC) [3, 4, 5].

In this work, we introduce generalized label tree embedding (GLTE) which is an improved and generalized version of LTE. The idea behind GLTE is to identify ambiguous categories and direct them in both directions during label tree construction rather than forcing them to be split too early at a split node as in the case of LTE. As a result, we expect to obtain a better representation, i.e., GLTE, compared to LTE [2]. We investigate coupling CNNs with the GLTE representation for ASC. First, a CNN similar to those used in [6, 3, 4] for template learning and matching will be explored. We further study to integrate the attention mechanism [7, 8, 9] to this CNN to produce an attentive CNN.

In general, an audio scene contains different kinds of foreground events mixed with background noise. The foreground events are usually informative for recognizing a scene [10, 11]. It is, therefore, reasonable to somehow weight different parts of a scene differently in a classification model in hope that the weights will be adapted to focus stronger on those informative parts. Ideally, these weights should be automatically learned by the model. This can be accomplished with an attention layer [7, 8, 9]. Attention mechanism has been commonly used with RNNs [7, 8, 9], however, we will show that it can be integrated with the proposed CNN, thank to the CNN’s over-time convolution making this possible.

The overall pipeline of the proposed system is illustrated in Figure 1.

2. GENERALIZED LABEL TREE EMBEDDING
2.1. Learning to construct a generalized label tree
Given a training set $\mathcal{S} = \{(x_n, c_n)\}_{n=1}^N$ where $x \in \mathbb{R}^M$ denotes a low-level feature vector of size $M$ and $c \in \{1, \ldots, C\}$ denotes a class label with $C$ is the number of categories. For convenience, let us denote the label set as $\mathcal{L} = \{1, \ldots, C\}$. Our goal is to use $\mathcal{S}$ to recursively construct a label tree to encode the hierarchy of the class labels [2, 3].

Similar to the LTE algorithm proposed in [2, 3], the construction algorithm starts at the root node which is associated with the entire set $\mathcal{L}$. Without loss of generality, let us consider a current split node with a label subset $\ell \subset \mathcal{L}$ and the corresponding sample subset $\mathcal{S}^\ell \subset \mathcal{S}$. We then aim at splitting $\ell$ into two subsets $\ell^L$ and $\ell^R$ that satisfy $\ell^L \neq \emptyset$, $\ell^R \neq \emptyset$, $\ell^L \cap \ell^R = \emptyset$, and $\ell^L \cup \ell^R = \ell$. Among all possible partitions, we want to seek for the optimal one such that $\ell^L$ and $\ell^R$ can be separated with as few errors as possible using a binary classifier. To accomplish this, two-fold cross validation is performed on $\mathcal{S}^\ell$. For each fold, we decompose $\mathcal{S}^\ell$ into two halves: $\mathcal{S}_{\text{train}}^\ell$ and $\mathcal{S}_{\text{eval}}^\ell$. The former is used to train a multi-class classifier $\mathcal{M}^\ell$ which is then evaluated on $\mathcal{S}_{\text{eval}}^\ell$ to obtain the confusion matrix $\mathbf{A} \in \{0, 1\}^{|\ell| \times |\ell|}$, where $|\cdot|$ represents cardinality of a set. Each element $\mathbf{A}_{ij}$ of $\mathbf{A}$ is computed by

$$\mathbf{A}_{ij} = \frac{1}{|\mathcal{S}_{\text{eval},i}^\ell|} \sum_{x \in \mathcal{S}_{\text{eval},i}^\ell} P(j|x, \mathcal{M}^\ell),$$

where $\mathcal{S}_{\text{eval},i}^\ell \subset \mathcal{S}_{\text{eval}}^\ell$ denotes the set of samples with label $i$. $P(j|x, \mathcal{M}^\ell)$ denotes the posterior probability that $\mathcal{M}^\ell$ predicts $x$ as class $j$. Hence, $\mathbf{A}_{ij}$ with $i \neq j$ indicates how likely a sample of class $i$ is wrongly predicted to belong to class $j$ by the classifier. The overall confusion matrix $\mathbf{A}$ is computed as the average of the confusion matrices obtained by two-fold cross validation. It is further symmetrized as:

$$\mathbf{A} = (\mathbf{A} + \mathbf{A}^T)/2.$$

The optimal partition $\{\ell^L, \ell^R\}$ is then derived to minimize:

$$E(\ell) = \sum_{i,j \in \ell^L} \mathbf{A}_{ij} + \sum_{m,n \in \ell^R} \mathbf{A}_{mn}.$$

By this, the ambiguous categories tend to be grouped into the same subset and, hence, produce two meta-classes $\{\ell^L, \ell^R\}$ that are easy to separate from one another. Spectral clustering [12] is applied on $\mathbf{A}$ to solve a relaxed version of the optimization problem in (3).

In the LTE algorithm in [2, 3], $\ell^L$ and $\ell^R$ would be immediately directed to the left and right child nodes, respectively. However,
Figure 1: The overall pipeline of the proposed audio scene classification systems.

Figure 2: A part of the constructed generalized label tree. The labels in red indicate the ambiguous categories which, therefore, are directed in both left and right child nodes.

3. CLASSIFICATION MODELS FOR ASC

3.1. GLTE feature extraction

GLTE representation can be extracted for audio scenes as similarly done for LTE one in [3, 5]. An audio snippet was first decomposed into segments of length 250 ms with 50% overlap. In case of DCASE 2017 challenge, a 10-second snippet yields $T = 78$ segments each of which is represented by a set of low-level features. The per-segment low-level feature vectors were then employed to construct a label tree with the algorithm in Section 2.1. Afterwards, an unseen segment-wise feature vector of the test data was finally transformed into a GLTE feature vector using the embedding in (6).

In [3, 5], we made use of three low-level feature sets: (1) 64 Gammatone cepstral coefficients extracted in the frequency range of 50 Hz to 22050 Hz, (2) 20 MFCC coefficients, their delta and acceleration coefficients, (3) 20 log Mel-scale filter bank coefficients, their first and second derivatives, zero-crossing rate, short-time energy, four subband energies, spectral centroid, and spectral bandwidth. For low-level feature extraction, a 250-ms segment was further decomposed into frames with a length of 50 ms and 50% overlap. The feature extraction was performed on the frame level. In turn, the feature vector for the entire 250-ms segment was calculated by averaging the per-frame feature vectors.

Moreover, for each low-level feature set, we extracted two GLTE features corresponding to the presence/absence of background noise as they can complement each other [3, 4]. The background noise was subtracted using the minimum statistics estimation and subtraction method [13] when necessary. As a result,
six GLTE feature images were obtained for a scene snippet. Finally, they were concatenated to form a single GLTE feature image $S \in \mathbb{R}^{F \times T}$, where $F$ is the size of a segment-wise concatenated GLTE feature vector. In particular, to extract GLTE features for training examples, 10-fold cross validation was conducted.

### 3.2. From CNN to attentive CNN

#### 3.2.1. CNN

We employed the CNN proposed in [6] for template learning and matching. The network architecture is illustrated in Figure 3. The CNN is able to learn templates that are useful for the classification task [4, 3] thanks to its over-time convolution and 1-max pooling. These templates are then convolved with an input GLTE image to extract features for classification [4, 3]. Specifically, over-time convolution between a filter $w \in \mathbb{R}^{F \times w}$ of width $w$ and a GLTE image $S \in \mathbb{R}^{F \times T}$ resulted in a feature map $o = (o_1, \ldots, o_{T-w+1}) \in \mathbb{R}^{T-w+1}$:

$$a_i = \text{ReLU}(a_i), \quad (9)$$
$$a_i = (S \ast w)_i = \sum_{k,l} (S[i : i + w - 1] \odot w)_{k,l}. \quad (10)$$

Here, $\ast$ and $\odot$ denote the convolution operation and element-wise multiplication, respectively. Rectified Linear Units (ReLU) activation [14] is used in (9). We then perform 1-max pooling [15, 16] over time to obtain a most dominant feature which corresponds to the maximum matching score of the template (i.e. the convolution kernel) $w$ and the input GLTE image $S$ [4, 3].

Furthermore, since this CNN supports filters with different widths $w$, using $Q \times R$ filters (i.e. $Q$ sets with different widths $w$ and $R$ filters for each set [4, 3]) will produce a vector of $Q \times R$ features. The resulted feature vector is finally presented to a softmax layer for classification. The network is trained to minimize the cross-entropy loss over the training examples:

$$E(\theta) = -\frac{1}{N} \sum_{n=1}^{N} y_n \log (\hat{y}_n(\theta)) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (11)$$

where $\theta$ denotes the network parameters and $\lambda$ is used to compromise the error term and the $\ell_2$-norm regularization term.

#### 3.2.2. Attentive CNN (ACNN)

With the CNN in Figure 3, using $R$ filters of width $w$ will result in a feature map $O = (O_1, \ldots, O_{T-w+1}) \in \mathbb{R}^{R \times (T-w+1)}$ (i.e. a matrix). However, instead of using the 1-max pooling over time as in Section 3.2, we learn an attention weight $\alpha_i$ for a vector $O_i$ at the time $i$, where $1 \leq i \leq T-w+1$, using an attention layer:

$$\alpha_i = \frac{\exp(f(O_i^T))}{\sum_{j=1}^{T-w+1} \exp(f(O_j^T))}, \quad (12)$$

In (12), $f(O^T)$ denotes the scoring function of the attention layer which is given by

$$f(O^T) = O^T W, \quad (13)$$

where $W$ is trainable weight matrix of the attention layer. The attentive output feature vector is given by:

$$O_{\text{att}}^T = \sum_{i=1}^{T-w+1} \alpha_i O_i^T. \quad (14)$$

$O_{\text{att}}$ is finally presented to a softmax layer for classification as in the case of CNN.

### 3.3. Data augmentation with GLTE feature

Data augmentation is important to improve generalization of a network [17, 18]. For low-level features, this can be done via systematically synthesizing or transforming existing data. We show that, data augmentation on the high-level GLTE feature can be accomplished by extracting GLTE features for the training data with different data split when cross-validation. We performed cross-validation both with and without taking into account the locations of the audio scenes. All extracted GLTE features extracted were included then into the training set. We experimental saw significant improvements in the classification performance with this data augmentation.

### 3.4. Calibration with support vector machine (SVM)

As in [3, 5], for all employed networks, after training, we calibrated the final classifier by employing a linear SVM in replacement for the softmax layer for classification. The output vectors of a network extracted from the training examples were used to train the linear SVM classifier which was subsequently employed to classify those output vectors extracted from the test examples.

### 4. EXPERIMENTS

#### 4.1. DCASE 2017 development data

The “acoustic scene classification” task of the challenge targets 15 categories (cf. Table 1). Each category consists of 312 audio snippets of 10 seconds. The data is split into 4-fold cross validation (cf. [19] for more details).
The classifiers in the tree construction and GLTE extraction algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-forest classification algorithms in Section 2 were trained with random-fore
7. REFERENCES


