Wavelet Scattering Transform for Multiclass Support Vector Machines in Audio Devices Classification System

Cheng Siong Chin  
Faculty of Science, Agriculture, and Engineering, Newcastle University Singapore, Singapore 599493  
cheng.chin@ncl.ac.uk

Jianhua Zhang  
School of Information and Control Engineering, Qingdao University of Technology, Shandong, 266525, China  
jianhuazhang@qut.edu.cn

Abstract—This paper presents an acoustic device classification system using multiclass support vector machines (MCSVM) on wavelet features. Using the MCSVM classifier that includes different binary SVMs, device-wise classification accuracy is achieved with shorter computational time than a convolutional neural network (CNN). Other types of kernels for the MCSVM classifier are compared with the k-nearest neighbors (kNN) algorithm, multiclass Naive Bayes, and Decision Tree. The experiment results demonstrate that the classification accuracy of MCSVM is better than Naive Bayes, kNN, and Decision Tree when tested on Detection and Classification of Acoustic Scenes and Events for 2020 (DCASE2020) dataset. The device-wise classification accuracy for the proposed MCSVM classifier exhibits approximately 15.6% better than the baseline (via CNN) results in DCASE2020-Task1A. Hence, it has good potential in robotic and drone systems for acoustic device detection.

Keywords—acoustic device classification, multiclass support vector machines, convolutional neural network, device-wise classification.

I. INTRODUCTION

An acoustic scenes classification system (ACS) [1] classifies sounds obtained from various environments and devices. Methods to extract auditory information have shown practical applications in healthcare [2], surveillance robotic systems [3], and other applications. The best-performing ACS performed in challenges such as Detection and Classification of Acoustic Scenes and Events (DCASE) Challenges was the convolutional neural network (CNN) and a fusion of multiple different models. CNN [4] is an artificial neural network that is commonly used to analyze images. It has demonstrated exemplary performance in other challenges such as Computer Vision and Image Processing [5].

In this paper, a few standard methods that performed moderately well in the DCASE challenges are discussed. For example, CNN combined with different techniques such as Recurrent neural networks [6], support vector machines [7], Gaussian mixture models [8], and multilayer perceptron [9-10] were used. Convolutional Recurrent network architectures were applied to the datasets. In contrast, others such as Bi-Long Short Term Memory networks (Bi-LSTM) [11] and LSTM[12] were also demonstrated quite well in the DCASE challenges. However, it is known that ensemble CNN is quite computational exhaustive. As a result, MLP and SVM[13] were used as ensemble classifiers[14]. Mel-scale representations generally perform well for feature extraction before acoustic scenes classification using CNN. Mel-frequency cepstral coefficients (MFCCs) [15] and other signal representations such as spectrogram and Constant Q Transform (CQT) [16-18] could preserve the relative positions of harmonics and its exponential frequency resolution was used to replace the MFCCs. In addition, wavelet time scattering [19-20] is insensitive to deformation. Hence, it has fewer impacts on the conceptual contents of a signal and generate an accurate classifier for acoustic scenes with background noise. However, the audio-based context recognition of the sound devices and MCSVM classifier for automatic classification has not been studied widely.

To address the issue, we propose a wavelet scattering with parametric log transformation using the Gabor wavelet. To enhance unknown devices for recording the acoustic scenes, an averaged data augmentation technique (refers to increase the amount of data by including different sets of existing data) mixed and averaged groups of datasets to improve the generalization. The feature representations from scattering transform provide inputs to MCSVM classifiers to improve the acoustic device classification.

The paper is organized as follows. In Section II, the proposed methodology used in feature extraction and classification algorithms is discussed. The TAU2020 Mobile dataset is addressed in Section III, and Section IV concludes the paper.

II. PROPOSED METHOD

The averaged data augmentation on the wavelet scattering-based feature extraction, and the MCSVM classifier framework for training different features are discussed. Before that, the TAU Urban Acoustic Scene 2020 Mobile dataset [21] is discussed. The development dataset consists of a total of 23040 samples. The dataset includes different audio gathered from...
simulated (S1 to S6) and real devices or sources (A, B, and C). Much of the data were obtained from device A using a binaural microphone. Other devices such as iPhone SE (Device C) and Samsung Galaxy S7 (Device B) are used. The simulated devices, S1 to S6 consists of randomly selected segments from the simultaneous recordings. The metadata of split consisting of 13,962 samples used for training (Device A, B, C, S1 to S3) and 2,967 samples for evaluation (Device A, B, C, S1 to S6), respectively. Note the dataset from S4, S5 and S6 are not provided for the training. Thus, making the evaluation more difficult as the dataset is not seen during the training stage.

A. Averaged Data Augmentation

The scatter transform [19-20] was used to compute modulation spectrum coefficients of multiple orders through cascades of wavelet convolutions and modulus operators prior to the data augmentation. The common types of data augmentation techniques on training datasets such as pitch shifting and Mixup [22], temporal cropping [23], block mixing [24], HPSS [25], and adding Gaussian noise [26] were used to improve generalization. Instead of directly using the Mixup to mix two inputs and labels together, the inputs from different sources or groups are combined and averaged [20] to form unseen sources for the training. The features and labels are varied, as shown.

\[ \tilde{x}_{i,j} = \gamma x_{i,j} + (1-\gamma) y_{i,j} \]
\[ \tilde{y}_{i,j} = \gamma y_{i,j} + (1-\gamma) y_{i,j} \]
where \( x_{i,j} \) and \( y_{i,j} \) are the seen source and corresponding labels, respectively. \( \tilde{x}_{i,j} \) and \( \tilde{y}_{i,j} \) refers to the unseen devices, namely: S4, S5 and S6 with index \( j = 4, 5, 6 \), respectively. The value of the mixed coefficient \( \gamma \) is set to 0.5.

The allocations of the features for the unseen source namely: S4, S5 and S6 are grouped and averaged as follows.

\[ x_{i,4} = (\tilde{x}_{i,4} + \tilde{x}_{i,5} + \tilde{x}_{i,6}) / 3 \]
\[ x_{i,5} = (\tilde{x}_{i,4} + \tilde{x}_{i,5} + \tilde{x}_{i,6}) / 2 \]
\[ x_{i,6} = (\tilde{x}_{i,5} + \tilde{x}_{i,6}) / 2 \]
and their labels are mixed accordingly to (3) to (5) with labels named as S4, S5, and S6, respectively. The training dataset will now consist of both the seen sources \( x_i \) and unseen sources \( \tilde{x}_i \).

B. Multiclass Support Vector Machines Using Linear Kernel

The multiclass support vector machines was used. A brief overview of SVM [27] is given. Let inputs \( x_{i=1...m} \) where \( m \) samples can belong to either class II or I. The corresponding labels are \( y_i = -1 \) for class II and \( y_i = 1 \) for class I, respectively. SVM uses an optimal separating hyperplane for linearly separable data.

\[ f(x) = w^T x + b = 0 \]
where \( w \) is a normal vector called weight that controls the direction of the hyperplane and \( b \) is a scalar named bias to maintain the position of the hyperplane.

The hyperplane divides the training samples into two classes with a maximal margin between the classes. The input samples are mapped into space via \( \phi \)-function where \( f(x) \) is trained in this space. The equation of the linear hyperplane can be written as follows.

\[ f(x) = w^T \phi(x) + b = 0 \]
and has a decision rule

\[ d(x) = \text{sign}(w^T \phi(x) + b) \]

By considering the noise with slack variables \( \beta \) and regularized terms \( \mu \sum_{i=1}^{m} \beta_i \), the optimal hyperplane can be determined through the following optimization where \( w, b, \) and \( \beta \) are selected such that

\[ \min \frac{1}{2} \|w\|^2 + \mu \sum_{i=1}^{m} \beta_i \]

For all \( (x_i, y_i), i = 1, ..., m \):

\[ s.t. \quad y_i (w^T \phi(x_i) + b) \geq 1 - \beta_i \]

where \( \mu \) is a regularization term. With a small \( \mu \), it emphasizes the margin while ignoring the outliers in the dataset. On the other hand, large \( \mu \) tends to overfit the training data.

Most problems have more than two classes; the need for multiclass SVM classification is therefore required. One approach decomposes the data into several binary problems [28] as the binary classifiers are simple to perform. In addition, SVM is designed for the binary problem. Hence, the two common decompositions are one-vs-all [29] and one-vs-one [30]. The one-vs-all method constructs a similar number of classes \( n \) of SVMs. The one-vs-one has \( n(n-1)/2 \) decision functions for all different class pairs for reducing the computation time.

The input space is mapped into a high dimensional space using kernel trick [31] to improve the linear separability.

Let \( \phi(x_i) = \phi(x_i) \) as kernel functions for training data. The solution to the above problems can be determined as follows.

To find \( \alpha_1, ..., \alpha_m \) such that

\[ q(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \]

where \( \alpha_i \) is Lagrange multiplier linked with each inequality constraints \( y_i (w^T \phi(x_i) + b) \geq 1 \) is maximized under the constraints with respect to \( \alpha_i \)

\[ \sum_{i=1}^{m} \alpha_i y_i = 0, \quad h \geq \alpha_i \geq 0, \quad \forall \alpha_i \]

The classification function becomes

\[ f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \]

where the common types of kernel functions \( K(x_i, x_j) \) can be linear kernel and sigmoid kernel. Here, the linear kernel function is used.

\[ K(x_i, x_j) = x_i^T x_j \]
The solution of the dual problem in (11) could be determined via the common Iterative Single Data Algorithm [32] where optimal non-zero Lagrange multipliers $\alpha^*$, $w$ and $b$ can be determined. The final prediction is achieved by the majority voting to enhance the classification rate. Suppose that $Y = \{y_1, y_2, \ldots, y_n\}$ is the labels for a given test sample $x_i$ in the classification function. The majority voting method can be written as follows.

$$\text{Label} = \arg\max_{i=1,\ldots,n} f(x_i)$$  \hspace{1cm} (15)$$

where $n$ is the number of classes and $f(x_i)$ is the classification function in (15).

Equation (17) implies that $x_i$ is assigned to the class with maximum votes. It returns a classification error indicated by 'NoUniqueMode' that incurs an additional column in the confusion matrix with no unique mode.

In summary, the proposed method is depicted in Fig. 1. The input audio signals will undergo scattering transform to extract features for sound sources followed by the averaged data augmentation. The main parameters for the wavelet time scattering are the number of wavelet filter banks, duration of the time invariance, and the number of wavelets per octave. The first and second filter bank has four wavelets (or the resolution of four) and one wavelet per octave. The invariant scale is 0.75s, which corresponds to slightly more than 220500 samples for the sampling rate of 44100 Hz. The feature matrix of the training data is $55848 \times 290$. It is obtained from the number of training examples of 13962 multiplied by the number of scattering windows per example of four. On the other hand, the scattering feature matrix for the test data is $11868 \times 290$. There are 2967 test examples and four windows per example.

Subsequently, an MCSVM classifier with a linear kernel function is used to classify the acoustic devices. The outputs of the classifier are obtained via majority voting. The evaluation set is used to produce the final prediction of the sound coming from different types of devices such as A, B, C, S1 to S6.

### III. RESULTS AND DISCUSSION

TAU Urban Acoustic Scene 2020 Mobile dataset is used for the experiments. In this experiment, we seek to classify the audio signal from measurement devices A, B, C, and S1 to S6. The input training dataset consists of 13,962 samples. Another 2,967 samples are used for testing. The evaluation datasets have the same number of samples as the testing datasets. The challenge is the dataset from S4, S5, and S6 are not available for training. Hence, the proposed method aims to classify the complete sources (A, B, C, and S1 to S6) of the audio signals. The Intel® Core i7-9750H CPU, 2.6GHz, 6 Cores, and Geforce RTX 2060 were used to run the experiment.

#### A. Data Augmentation Comparisons for Device-Wise

The proposed averaged data augmentation method used in MCSVM classifiers compared to the Mixup is tabulated in Table I. The proposed method increases the diversity of data used for the training, as seen in the classification in device S4 to S6. In Mixup, the dataset is mixed by randomly combining the features of two different classes. The results show that the device S4 to S6 does not perform well in the classification. The confusion chart with the averaged data augmentation method for MCSVM using linear kernel can be seen in Fig. 2. The classification error indicated by 'NoUniqueMode' remains small as compared to other device-wise classification errors.

<table>
<thead>
<tr>
<th>Device</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mixup</td>
</tr>
<tr>
<td>A</td>
<td>98.4</td>
</tr>
<tr>
<td>B</td>
<td>95.0</td>
</tr>
<tr>
<td>C</td>
<td>99.7</td>
</tr>
<tr>
<td>S1</td>
<td>51.2</td>
</tr>
<tr>
<td>S2</td>
<td>94.5</td>
</tr>
<tr>
<td>S3</td>
<td>37.2</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>52.9</td>
</tr>
</tbody>
</table>

TABLE I. DEVICE-WISE COMPARISON OF MIXUP AND AVERAGED DATA AUGMENTATION FOR MCSVM USING LINEAR KERNEL
B. Comparison with Different Kernel Functions

In this section, different types of kernel functions such as polynomial, linear, radial basis function (Gaussian), and sigmoid are compared using the evaluation dataset with device class labels. The results are tabulated in Table II. With higher-order polynomial function and using Gaussian distribution on the inputs, the device-wise classification accuracy does not improve. As seen in Table II, the linear kernel performs better than polynomial, Gaussian, and sigmoid functions. In addition, the one-vs-one of constructing MCSVM produces better classification accuracy than the one-vs-all. With a smaller number of the training data required for one-vs-one, the computation effort is more efficient than one-vs-all.

![Figure 2. Device-Wise Confusion chart for MCSVM using Linear Kernel](image)

TABLE II. DEVICE-WISE COMPARISON BETWEEN DIFFERENT MULTICLASS SVM

<table>
<thead>
<tr>
<th>SVM Kernel Functions</th>
<th>Device-wise Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One-vs-One</td>
</tr>
<tr>
<td>2\textsuperscript{nd}-order polynomial</td>
<td>69.6</td>
</tr>
<tr>
<td>3\textsuperscript{rd}-order polynomial</td>
<td>69.5</td>
</tr>
<tr>
<td>Linear</td>
<td>69.8</td>
</tr>
<tr>
<td>Radial basis function (Gaussian)</td>
<td>65.6</td>
</tr>
<tr>
<td>Sigmoid function</td>
<td>19.6</td>
</tr>
</tbody>
</table>

C. Comparisons with Other Types of Classifiers

The multiclass SVM with linear kernel function using a one-vs-one approach is chosen for its highest classification accuracy. Compared to using a decision tree, the classification accuracy, Naive Bayes classifier, and k-mean nearest neighbors (k-NN) are tabulated in Table III. The development data was partitioned into $k$-fold, where $k=4$ was used in both DCASE 2016 and 2017 challenges. The out-of-sample misclassification rate by using 10-fold cross-validation is computed for comparisons. The final classification error for the 10-fold cross-validation can be seen in Fig. 3.

The multiclass SVM with linear kernel function using a one-vs-one approach has the highest classification accuracy of 69.7%. Comparing with the decision tree and Naive Bayes classifier, it gives a lower classification accuracy. Furthermore, the k-NN does not perform better than the proposed method. The proposed multiclass SVM with linear kernel function remains the best method, among other ways in Table III.

Similarly, the final classification error for the 10-fold cross-validation where multiclass SVM with linear kernel function has the lowest value. The misclassification rate of 0.56% can be observed in Fig. 3. In contrast, the multiclass Naive Bayes classifier has not performed well. It is due to the variables in the dataset are dependent on one another.

![Figure 3. Out-of-Sample Misclassification Rate via 10-Fold Cross-Validation for MCSVM Classifier using Linear Kernel Function](image)

In general, the precision is higher than 63.4% for the device-wise classification except for S4 to S6, as seen in Fig. 2. The false-negative rate is generally low, even though S4 to S6 devices remain higher. It implies that there are some devices that are still not classified correctly and are often classified wrongly. However, the average device-wise classification is around 69.7%.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Classification Accuracy (%)</th>
<th>Final Misclassification Rate via 10-Fold Cross Validation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiclass SVM with linear kernel function (One-vs-One)</td>
<td>69.7</td>
<td>0.56</td>
</tr>
<tr>
<td>Multiclass SVM with Decision Tree (One-vs-One)</td>
<td>40.7</td>
<td>21.7</td>
</tr>
<tr>
<td>Multiclass Naive Bayes Classifier</td>
<td>38.9</td>
<td>47.4</td>
</tr>
<tr>
<td>k-mean nearest neighbors (k-NN)</td>
<td>52.8</td>
<td>18.6</td>
</tr>
</tbody>
</table>

738
D. Comparisons with other Methods in DCASE2020

As compared with the baseline results obtained from the DCASE2020-Task1A challenge, the device-wise classification accuracy using CNN exceeds 54.1%, as seen in Table IV. In general, the S4 to S6 are lower than the baseline results. However, the classification accuracy for the proposed multiclass SVM with linear kernel function shows 69.7% or 15.6% better than the baseline result. Unsurprisingly, most of the methods used CNNs, and pre-trained networks such as ResNet and DenseNet with log-Mel energies generate a higher classification accuracy. Instead of a CNN model, the CNN ensemble improved accuracy despite the higher computational effort required.

The winner for the DCASE2020 challenge used the Snapshot Ensemble Deep Learning Neural Network with feature extraction via log-Mel energies. It has a top classification accuracy of 76.5%. The proposed device-wise classification accuracy falls short of 6.8%. However, it remains higher than the baseline system value of 54.1%, and the accuracy remains in the top quartile of the total 179 entries from 47 teams globally. Besides, the computational time and parameters used are lesser than CNN.

Table IV. Comparison of Device-wise and Scene-wise Baseline Results in DCASE2020 Challenge- Task 1A with Proposed MCSVM Using Linear Kernel Function

<table>
<thead>
<tr>
<th>Device</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline (using CNN)</td>
</tr>
<tr>
<td>A</td>
<td>70.6</td>
</tr>
<tr>
<td>B</td>
<td>60.6</td>
</tr>
<tr>
<td>C</td>
<td>62.6</td>
</tr>
<tr>
<td>S1</td>
<td>55.0</td>
</tr>
<tr>
<td>S2</td>
<td>53.3</td>
</tr>
<tr>
<td>S3</td>
<td>51.7</td>
</tr>
<tr>
<td>S4</td>
<td>48.2</td>
</tr>
<tr>
<td>S5</td>
<td>45.2</td>
</tr>
<tr>
<td>S6</td>
<td>39.6</td>
</tr>
<tr>
<td>Average</td>
<td>54.1</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

The wavelet scattering feature extraction for a multiclass support vector machine (MCSVM) with one-vs-one approach using linear kernel function for audio devices) the classification was proposed. The wavelet scattering with parametric log transformation using the Gabor wavelet was used. It was followed by using the averaged data augmentation to mix and average groups of datasets to improve the generalization for unknown devices used.

The classification performance of the different classifiers was compared. The results for device-wise classification showed a 15.6% improvement compared with the baseline results obtained in the DCASE2020-Task1A challenge. However, the proposed max-fusion method falls short of 6.8% compared to the Snapshot ensembles (that fused multiple Deep Neural Network models) that won the DCASE2020 Challenges-Task 1A. Nevertheless, the proposed method remains at the top of 179 entries from 47 teams globally. The proposed MCSVM is not demanding in terms of memory or computational, as fewer parameters were used than convolutional neural networks (CNNs).

ACKNOWLEDGMENT

This work was supported by the author's University.

REFERENCES


