学習モデル、Data Augmentation の有無、最適化アルゴリズムの違いによる音響シーン分類性能比較

伊藤 葵
Aoi Ito
法政大学情報科学部ディジタルメディア学科 B3 20K1105
aoi.ito.8q@stu.hosei.ac.jp

2023 年 1 月 26 日

1　はじめに

音響シーン・音響イベントの分類・検出に関する技術・研究分野 (Detection and Classification of Acoustic Scenes and Events) では、DCASE Community という組織が音響に関する問題に取り組む DCASE Challenge/Workshop を 2013 年から開催している [1]。この Challenge の一つに、音響シーン分類がある。

このレポートでは、DCASE2013 の Task1 にある音響シーン分類問題のデータセットを縮小したものに対し、異なる学習モデル (CNN, wavelet scattering を用いたアンサンブル学習、CNN とアンサンブル学習の Late Fusion) 間や Data Augmentation の有無、最適化アルゴリズムの違いという 3 つの観点から性能を比較する。

2　手法

この章では、今回の実験で用いたモデルの学習方法と Data Augmentation の方法を説明する。また、このレポートでは、`https://jp.mathworks.com/help/audio/ug/acoustic-scene-recognition-using-late-fusion.html` を参考とした。

2.1　CNN

CNN(Convolutional Neural Network) とは、ニューラルネットワークに畳み込みを追加したものである [2]。本レポートでは、図 1, 2 のネットワークで CNN を行う。最適化アルゴリズムは、sgdm (モーメンタム項付き確率的勾配降下法) と adam で比較する。

図 1　CNN のネットワーク 1

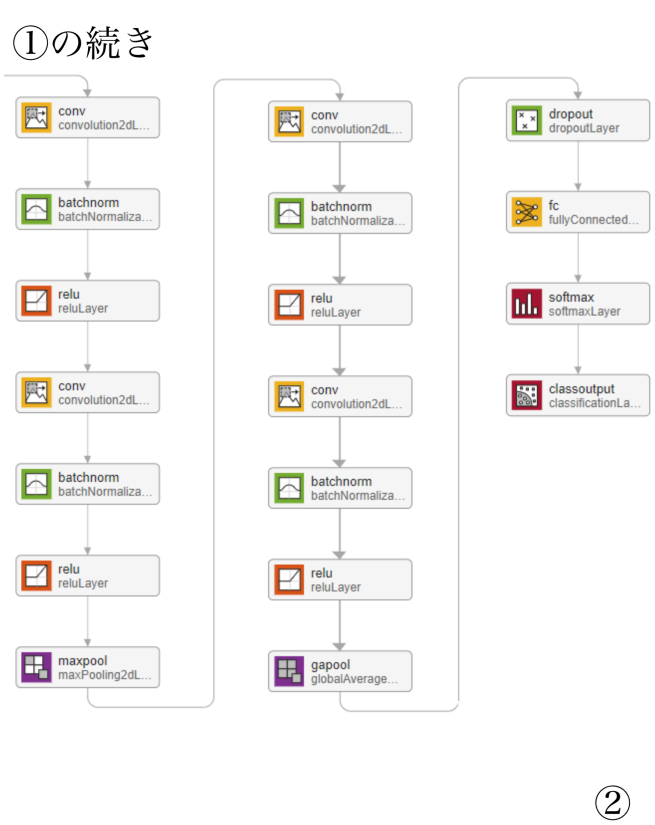

図 2　CNN のネットワーク 2

CNN の入力信号 (特徴量) は、対数をとったメルスペクトログラムとする。

2.2　wavelet scattering を用いたアンサンブル学習

wavelet 変換 とは、周波数解析の手法の一つである。

フーリエ解析では、ある入力信号を拡大縮小したサイン波、コサイン波の足し合わせで表現するが、時系列情報が損失する。これに対し、wavelet 変換は時間と周波数を同時に解析する手法である。基準となる小さな波 mother wavelet を様々な縮尺に引き伸ばし、周波数の物差しであるウェーブレットを多数用意する。これらのウェーブレットを、時間軸方向に平行移動させながら入力信号にあてがい，時間と周波数の情報を同時に得るのである [3]。[4] は wavelet scattering が音響シーンの表現に優れていることを示しているため、本レポートでは wavelet scattering を採用した。

アンサンブル学習とは、個々に別々の学習器として学習させたもののそれぞれの結果を融合させることで、未学習のデータに対しての予測能力向上を狙う学習である [5]。複数のモデルを学習し、最後に各モデルの予測結果を多数決原理で決めることで、各モデルの認識精度が低くても性能が向上する確率が高くなることがメリットである。

2.3　Late Fusion による出力の統合

Late Fusion モデルとは、複数のモデルの出力直後で出力結果を統合したものである。Early Fusion と異なり、統合後にニューラルネットワークの影響を受けない。本レポートでは、CNN の出力と wavelet scattering を用いたアンサンブル学習の出力を統合している。ブロック図で示すと、図 3 の通り [6]。

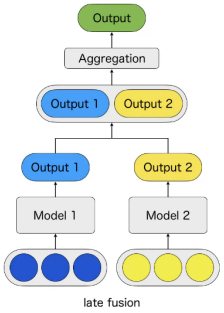

図 3　Late Fusion のブロック図

2.4　CNN 用の Data Augmentation

本レポートの実験では、学習用データセットとなる音声ファイル数が全部で 56 個と少ないため、学習用データ数の増加を目的に Data Augmentation を行う。本レポートでは、2 つの異なる音響シーンを 1:1 の割合で混ぜ合わせた。次式の x は音声ファイル、y は対応する正解ラベルであり、混合比は λ で調整可能である。今回の研究では、混合比 λ は 0.5 とした。実装では、異なるラベルを持つスペクトログラム同士を混合している。

$$\widetilde{x} = \lambda x_i + (1-\lambda)\, x_j \tag{1}$$

$$\widetilde{y} = \lambda y_i + (1-\lambda)\, y_j \tag{2}$$

2.5　実験に使用したデータ

実験に使用したデータは、DCASE2013 Task1 のデータセット (`https://archive.org/details/dcase2013_scene_classification`) を縮小し用いた。音響シーンの種類は、全部で 7 種類 (bus, busystreet, office, park, quietstreet, restaurant, supermarket) とする。各音響シーンは 10 個の wav ファイルが用意されており、音声の長さは 30 秒である。この内、各音響シーン No.01〜08 の音声ファイルを学習用データセット、No.09〜10 の音声ファイルを検証用データセットとした。

3　実験

3.1　実験条件

実行環境は、Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz、RAM 16.0 GB、使用 OS は Windows である。言語は MATLAB を使用した。

実験条件 (学習モデル、DataAugmentation の有無、最適化アルゴリズム) は表 1 の通りである。

3.2　実験方法

実験方法は以下の通りである。

1. 音響シーンデータセットのロード
2. 学習用データセット、検証用データセットへの分類
3. CNN の特徴量抽出
4. Data Augmentation
5. CNN の各層の定義、学習
6. CNN の評価
7. アンサンブル学習
8. アンサンブル学習の評価
9. Late Fusion の適用、評価

手順 1,2 のデータセットの内訳は、2.5 で述べたとおりであ

表 1　実験条件

Model	DataAugmentation	optimization
CNN		sgdm
Ensemble		sgdm
Late Fusion		sgdm
CNN	✓	sgdm
Ensemble	✓	sgdm
Late Fusion	✓	sgdm
CNN		adam
Ensemble		adam
Late Fusion		adam
CNN	✓	adam
Ensemble	✓	adam
Late Fusion	✓	adam

る。手順 3 では、音声の特徴量としてメルスペクトログラムを採用した。手順 6, 8, 9 の評価時には、 Average Accuracy だけでなく、混同行列を用いてどの音響シーンの分類を誤っているか可視化した。

3.3　実験結果

各モデルの正解率は表 2 の通り。

表 2　各モデルの正解率 (DA: Data Augmentation)

Model	sgdm	sgdm&DA	adam	adam&DA
CNN	57.14	64.29	71.43	50.00
Ensemble	78.57	78.57	85.71	78.57
Late Fusion	78.57	78.57	85.71	71.43

また、最適化アルゴリズム (sgdm, adam) における各モデル別の混同行列は図 4〜13 の通り。尚、アンサンブル学習は各最適化アルゴリズムによる学習方法の違いの影響はないため、DataAugmentation の有無にのみ触れている。

図 4　CNN の混同行列 (sgdm)

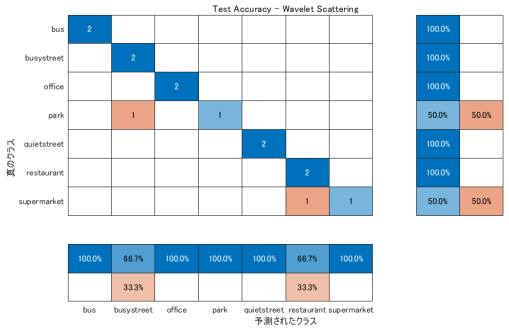

図 5　wavelet scattering を用いたアンサンブル学習の混同行列 (DataAugmentation なし)

図 6　Late Fusion の混同行列 (sgdm)

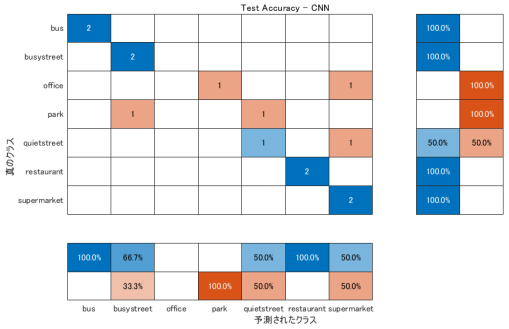

図 7　CNN の混同行列 (sgdm, DataAugmentation)

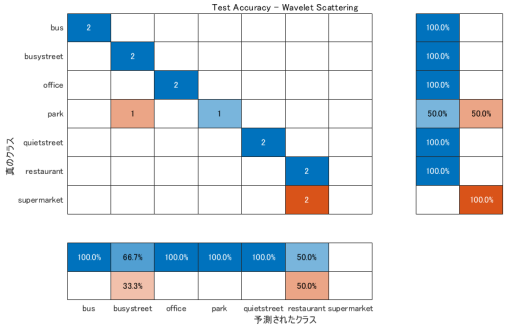

図 8　wavelet scattering を用いたアンサンブル学習の混同行列 (DataAugmentation あり)

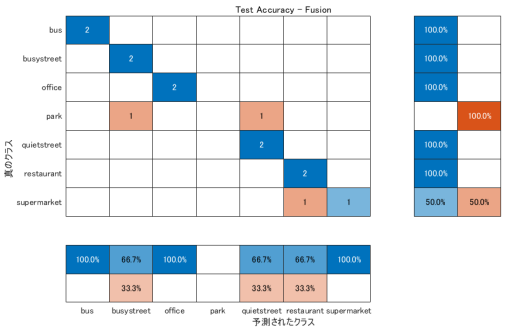

図 9　Late Fusion の混同行列 (sgdm, DataAugmentation)

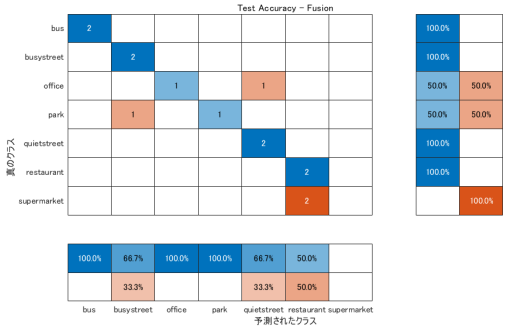

図 13　Late Fusion の混同行列 (adam, DataAugmentation)

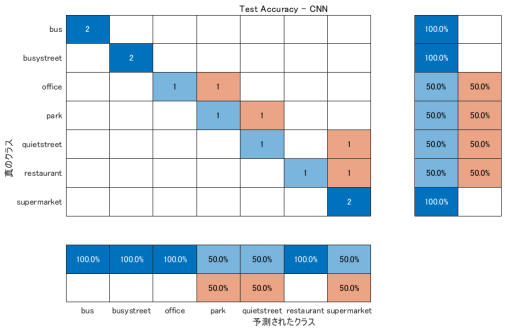

図 10　CNN の混同行列 (adam)

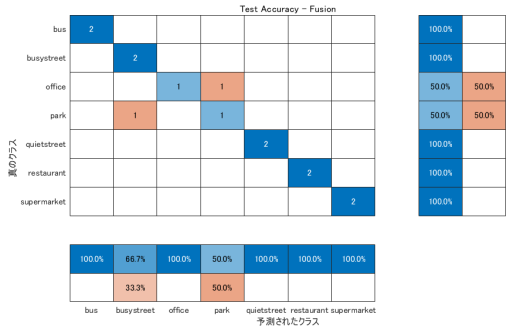

図 11　Late Fusion の混同行列 (adam)

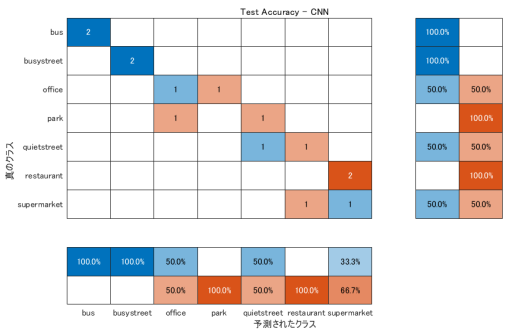

図 12　CNN の混同行列 (adam, DataAugmentation)

最も正解率が低かったモデルは、CNN のみのモデル (Data Augmentation 済、adam 使用) である。最も正解率が高かったモデルは、wavelet scattering を用いたアンサンブル学習のモデルとこのモデルに CNN (adam 使用) の結果を Late Fusion したモデルである。 CNN のみのモデルには、正解率が 0% となる音響シーンが存在する。

4　考察

まず、Data Augmentation の有無が認識精度に与える影響について考察する。最も正解率が低かった CNN のみのモデル (Data Augmentation 済、adam 使用) に着目する。一般に、Data Aumentation は、学習データ数の増加や過学習の抑制に効果がある。しかし、今回の実験では Data Augmentation をしても CNN のみ (最適化アルゴリズム: adam) の学習モデルと比較すると、21.43% 正解率が減少した。今回の実験の学習データセットの数は 56 と少なく、かつ人間が聞いても区別がしにくいほど似た音響シーンの組み合わせが多いデータセットであった。そのため、Data Augmentation を異なるラベルを持つスペクトログラム同士で行っても、似たようなデータがさらに増加されてしまったため、過学習の促進に繋がってしまった。反面、最適化アルゴリズムに sgdm を使用した際の実験では、CNN のみの学習モデルで 7.15% と正解率が高くなっている。

次に、学習モデルの違いの観点から考察する。CNN だけのモデルに対し、wavelet scattering を用いたアンサンブル学習の出力を CNN の出力に統合した Late Fusion モデルの方が認識性能が向上している。このことから、今回の実験のように、学習データ数が少ない場合は、特に複数のモデルを組み合わせた方が、単体のモデルによる音響シーン分類よりも、はるかに性能向上の効果が期待できると考える。例えば、学習データ数が少なく、似たデータが多かったために Data Augmentation が上手くいかなかったと考察した CNN のみのモデル (Data Augmentation 済、adam 使用) を見ても、Late Fusion をすることで、Data Augmentation をしなかった CNN のみ (最適化アルゴリズム: adam) の学習モデルと同等の性能となる。

最後に、混同行列に着目する。図 5,8 を比較すると、Data Augmentation をすることにより、supermarket の正解率が高くなった。これは、Data Augmentation による学習データ数増加が理由だと考察する。特に、supermarket は店内で鳴っている音の種類も多いため、他の音響シーンの音声が足しあわさることで、supermarket の未知のデータに含まれる音に柔軟に対応できるようになった。そして、図 4 〜 13 の混同行列

を通して、park の認識精度が低い。これに関して、図 4 と 6、図 7 と 9、図 10 と 11、図 12 と 13 を比較すると、複数の学習モデルの統合が認識精度が低い音響シーンの分類精度向上に効果があるといえる。さらに、図 4 と 10、図 6 と 11 といった最適化アルゴリズムが異なる CNN の混同行列を比較すると、adam を用いたモデルの方が park や office といった正解率が 0% と低かった音響シーンの認識精度が向上している。この結果から、音響シーン分類のタスクにおいて、最適化アルゴリズムは adam の方がふさわしい。ただし、adam は sgdm と比較すると汎化性能が劣る [7] といわれるため、AdaBound や AMSBound との比較が今後必要である。

5 おわりに

このレポートでは、DCASE2013 Task1 のデータセットを用いて、異なる学習モデル (CNN, wavelet scattering を用いたアンサンブル学習、CNN とアンサンブル学習の Late Fusion) 間や Data Augmentation の有無、最適化アルゴリズムの違いという 3 つの観点から音響シーン分類の性能を比較した。CNN だけでなく、アンサンブル学習の出力を Late Fusion にて統合し、最適化アルゴリズムには adam を採用することで、音響シーンの分類性能向上が期待できる結果となった。Data Augmentation は、元の学習データ数が少なく似たデータが多い場合は、どのようにデータを組み合わせて新たな学習データを作成するか、方法を検討する必要がある。課題として、CNN に対して、CNN 以外のモデル (LSTM 等) と wavelet scattering を用いたアンサンブル学習の Late Fusion の性能比較がある。

参考文献

[1] @diesekiefer. ”音環境理解のための技術コミュニティ DCASE の紹介”. Qiita.2022-12-15. https://qiita.com/diesekiefer/items/bd0c31c135ede8a9990e, (参照 2023-01-10).

[2] @icoxfog417. ”Convolutional Neural Network とは何なのか”. Qiita.2019-10-27. https://qiita.com/icoxfog417/items/5fd55fad152231d706c2, (参照 2023-01-10).

[3] 三谷 政昭 (2008). ”やり直しのための通信数学【オンデマンド版】”. CQ 出版社. (参照 2023-01-10).

[4] Vincent Lostanlen, Joakim Anden(2016). ”BINAURAL SCENE CLASSIFICATION WITH WAVELET SCATTERING”. Detection and Classification of Acoustic Scenes and Events 2016.2016-09-03. (参照 2023-01-10).

[5] ”アンサンブル学習（Ensemble learning）解説と実験”. S-Analysis. https://data-analysis-stats.jp/機械学習/アンサンブル学習（Ensemble learning）解説と実験, (参照 2023-01-10).

[6] ”ニューラルネットワークを用いた複数モーダルの統合について”. AGIRobots.2022-07-09. https://agirobots.com/multimodal-ai-fusion/, (参照 2023-01-10).

[7] @Phoeboooo. ”[最新論文] 新しい最適化手法誕生! AdaBound & AMSBound”. Qiita. 2019-03-04. https://qiita.com/Phoeboooo/items/f610affdcaaae0a28f34, (参照 2023-01-10).

付録

作成したソースコードを記載する。
最適化アルゴリズムは、ソースコード 333 行目の options にて ‘adam‘ もしくは ‘sgdm‘ を選択する。‘sgdm‘ を選択した場合は、‘’Momentum’,0.9,‘も追記する。

```
%% Acoustic Scene Recognition Using Late Fusion

%% Load Acoustic Scene Recognition Data Set
% To run the example, you must first download ... the data set [1]. The full
% data set is approximately 15.5 GB. Depending ... on your machine and internet
% connection, downloading the data can take ... about 4 hours.

downloadFolder = tempdir;
datasetFolder = ... fullfile(downloadFolder,'scenes_stereo');

%%
% Read in the development set metadata as a ... table. Name the table variables
% |FileName|, |AcousticScene|.

metadata_train = readtable("dataset.txt");
head(metadata_train)

metadata_test = readtable("dataset_test.txt");
head(metadata_test)

train_filePaths = metadata_train.FileName;
test_filePaths = metadata_test.FileName;

%%
% Create audio datastores for the train and ... test sets. Set the
% |Labels| property of the
% ... <docid:audio_ref#mw_6315b106-9a7b-4a11-a7c6-322c073e343a
% |audioDatastore|> to the acoustic scene. Call
% ... <docid:audio_ref#mw_27293e0c-5066-45c4-b34d-e2814c56921f
% |countEachLabel|> to verify an even ... distribution of labels in both the
% train and test sets.

adsTrain = audioDatastore(train_filePaths, ...
    'Labels',categorical(metadata_train.AcousticScene), ... ...
    'IncludeSubfolders',true);
display(countEachLabel(adsTrain))

adsTest = audioDatastore(test_filePaths, ...
    'Labels',categorical(metadata_test.AcousticScene), ... ...
    'IncludeSubfolders',true);
display(countEachLabel(adsTest))

%%
% You can reduce the data set used in this ... example to speed up the run time
% at the cost of performance. In general, ... reducing the data set is a good
% practice for development and debugging. Set ... |reduceDataset| to |true| to
```

```
% reduce the data set.
reduceDataset = false;
if reduceDataset
    adsTrain = splitEachLabel(adsTrain,20);
    adsTest = splitEachLabel(adsTest,10);
end

%%
% Call ...
    <docid:audio_ref#mw_4257f192-1344-4f89-8edb-414b0f9efc42 |read|> to
% get the data and sample rate of a file from ...
    the train set. Audio in the
% database has consistent sample rate and ...
    duration. Normalize the audio and
% listen to it. Display the corresponding label.

[data,adsInfo] = read(adsTrain);
data = data./max(data,[],'all');

fs = adsInfo.SampleRate;
sound(data,fs)

fprintf('Acoustic scene = ...
    %s\n',adsTrain.Labels(1))

%%
% Call ...
    <docid:audio_ref#mw_a427bc3f-f73e-448b-896a-65ddbb3a266a |reset|> to
% return the datastore to its initial condition.

reset(adsTrain)

%% Feature Extraction for CNN
% Each audio clip in the dataset consists of ...
    10 seconds of stereo
% (left-right) audio. The feature extraction ...
    pipeline and the CNN
% architecture in this example are based on ...
    [3]. Hyperparameters for the
% feature extraction, the CNN architecture, ...
    and the training options were
% modified from the original paper using a ...
    systematic hyperparameter
% optimization workflow.
%
% First, convert the audio to mid-side ...
    encoding. [3] suggests that mid-side
% encoded data provides better spatial ...
    information that the CNN can use to
% identify moving sources (such as a train ...
    moving across an acoustic
% scene).

dataMidSide = [sum(data,2),data(:,1)-data(:,2)];

%%
% Divide the signal into one-second segments ...
    with overlap. The final
% system uses a probability-weighted average ...
    on the one-second segments to
% predict the scene for each 10-second audio ...
    clip in the test set. Dividing
% the audio clips into one-second segments ...
    makes the network easier to
% train and helps prevent overfitting to ...
    specific acoustic events in the
% training set. The overlap helps to ensure ...
    all combinations of features
% relative to one another are captured by the ...
    training data. It also
% provides the system with additional data ...
    that can be mixed uniquely
% during augmentation.

segmentLength = 1;
segmentOverlap = 0.5;

[dataBufferedMid,¬] = ...
    buffer(dataMidSide(:,1),round(segmentLength*fs),round(segme
[dataBufferedSide,¬] = ...
    buffer(dataMidSide(:,2),round(segmentLength*fs),round(segme
dataBuffered = ...
    zeros(size(dataBufferedMid,1),size(dataBufferedMid,2)+size
dataBuffered(:,1:2:end) = dataBufferedMid;
dataBuffered(:,2:2:end) = dataBufferedSide;

%%
% Use ...
    <docid:audio_ref#mw_1bb316de-8018-4365-a351-73374473237f
% |melSpectrogram|> to transform the data into ...
    a compact frequency-domain
% representation. Define parameters for the ...
    mel spectrogram as suggested by
% [3].

windowLength = 2048;
samplesPerHop = 1024;
samplesOverlap = windowLength - samplesPerHop;
fftLength = 2*windowLength;
numBands = 128;

%%
% |melSpectrogram| operates along channels ...
    independently. To optimize
% processing time, call |melSpectrogram| with ...
    the entire buffered signal.

spec = melSpectrogram(dataBuffered,fs, ...
    'Window',hamming(windowLength,'periodic'), ...
    'OverlapLength',samplesOverlap, ...
    'FFTLength',fftLength, ...
    'NumBands',numBands);

%%
% Convert the mel spectrogram into the ...
    logarithmic scale.

spec = log10(spec+eps);

%%
% Reshape the array to dimensions (Number of ...
    bands)-by-(Number of
% hops)-by-(Number of channels)-by-(Number of ...
    segments). When you feed an
% image into a neural network, the first two ...
    dimensions are the height and
% width of the image, the third dimension is ...
    the channels, and the fourth
% dimension separates the individual images.

X = ...
    reshape(spec,size(spec,1),size(spec,2),size(data,2),[]);

%%
% Call |melSpectrogram| without output ...
    arguments to plot the mel
% spectrogram of the mid channel for the first ...
    six of the one-second
% increments.

for channel = 1:2:11
    figure
    melSpectrogram(dataBuffered(:,channel),fs, ...
        'Window',hamming(windowLength,'periodic'), ...
```

```
            ...
        'OverlapLength',samplesOverlap, ...
        'FFTLength',fftLength, ...
        'NumBands',numBands);
    title(sprintf('Segment %d',ceil(channel/2)))
end

%%
% The helper function ...
    |HelperSegmentedMelSpectrograms| performs ...
    the feature
% extraction steps outlined above.



%%
% To speed up processing, extract mel ...
    spectrograms of all audio files in
% the datastores using ...
    <docid:matlab_ref#bvaomuj-1 |tall|> ...
    arrays. Unlike
% in-memory arrays, tall arrays remain ...
    unevaluated until you request that
% the calculations be performed using the ...
    <docid:matlab_ref#bvaolov
% |gather|> function. This deferred evaluation ...
    enables you to work quickly
% with large data sets. When you eventually ...
    request the output using
% |gather|, MATLAB combines the queued ...
    calculations where possible and
% takes the minimum number of passes through ...
    the data. If you have Parallel
% Computing Toolbox(TM), you can use tall ...
    arrays in your local MATLAB
% session, or on a local parallel pool. You ...
    can also run tall array
% calculations on a cluster if you have ...
    MATLAB(R) Parallel Server(TM)
% installed.
%
% If you do not have Parallel Computing ...
    Toolbox(TM), the code in this
% example still runs.

train_set_tall = tall(adsTrain);
xTrain = ...
    cellfun(@(x)HelperSegmentedMelSpectrograms(x,fs, ...
    ...
    'SegmentLength',segmentLength, ...
    'SegmentOverlap',segmentOverlap, ...
    'WindowLength',windowLength, ...
    'HopLength',samplesPerHop, ...
    'NumBands',numBands, ...
    'FFTLength',fftLength), ...
    train_set_tall, ...
    'UniformOutput',false);
xTrain = gather(xTrain);
xTrain = cat(4,xTrain{:});

test_set_tall = tall(adsTest);
xTest = ...
    cellfun(@(x)HelperSegmentedMelSpectrograms(x,fs, ...
    ...
    'SegmentLength',segmentLength, ...
    'SegmentOverlap',segmentOverlap, ...
    'WindowLength',windowLength, ...
    'HopLength',samplesPerHop, ...
    'NumBands',numBands, ...
    'FFTLength',fftLength), ...
    test_set_tall, ...
    'UniformOutput',false);
xTest = gather(xTest);
xTest = cat(4,xTest{:});

%%
% Replicate the labels of the training set so ...
    that they are in one-to-one
% correspondence with the segments.

numSegmentsPer10seconds = size(dataBuffered,2)/2;
yTrain = ...
    repmat(adsTrain.Labels,1,numSegmentsPer10seconds)';
yTrain = yTrain(:);


%% Data Augmentation for CNN
% The DCASE 2017 dataset contains a relatively ...
    small number of acoustic
% recordings for the task, and the development ...
    set and evaluation set were
% recorded at different specific locations. As ...
    a result, it is easy to
% overfit to the data during training. One ...
    popular method to reduce
% overfitting is _mixup_. In mixup, you ...
    augment your dataset by mixing the
% features of two different classes. When you ...
    mix the features, you mix the
% labels in equal proportion. That is:
%
% $$\begin{array}{l}\tilde{\mathrm{x}} ...
    =\lambda {\mathrm{x}}_i +\left(1-\lambda
% \right){\mathrm{x}}_j \\\tilde{y} =\lambda ...
    y_i +\left(1-\lambda\right)y_j \end{array}$$
%
% Mixup was reformulated by [2] as labels ...
    drawn from a probability
% distribution instead of mixed labels. The ...
    implementation of mixup in this
% example is a simplified version of mixup: ...
    each spectrogram is mixed with
% a spectrogram of a different label with ...
    lambda set to 0.5. The
% original and mixed datasets are combined for ...
    training.

%{
xTrainExtra = xTrain;
yTrainExtra = yTrain;
lambda = 0.5;
for i = 1:size(xTrain,4)

    % Find all available spectrograms with ...
        different labels.
    availableSpectrograms = ...
        find(yTrain≠yTrain(i));

    % Randomly choose one of the available ...
        spectrograms with a different label.
    numAvailableSpectrograms = ...
        numel(availableSpectrograms);
    idx = randi([1,numAvailableSpectrograms]);

    % Mix.
    xTrainExtra(:,:,:,i) = ...
        lambda*xTrain(:,:,:,i) + ...
        (1-lambda)*xTrain(:,:,:,availableSpectrograms(idx));

    % Specify the label as randomly set by lambda.
    if rand > lambda
        yTrainExtra(i) = ...
            yTrain(availableSpectrograms(idx));
    end
end
xTrain = cat(4,xTrain,xTrainExtra);
```

```
yTrain = [yTrain;yTrainExtra];

%%
% Call |summary| to display the distribution of labels for the augmented
% training set.

summary(yTrain)
%}

%% Define and Train CNN
%
% Define the CNN architecture. This architecture is based on [1] and
% modified through trial and error. See
% ... <docid:nnet_ug#mw_25a9e4c2-614f-48b9-97e2-bbdd7aaf806f ...
List of Deep
% Learning Layers> to learn more about deep learning layers available in
% MATLAB(R).

imgSize = [size(xTrain,1),size(xTrain,2),size(xTrain,3)];
numF = 32;
layers = [ ...
    imageInputLayer(imgSize)

    batchNormalizationLayer

    convolution2dLayer(3,numF,'Padding','same')
    batchNormalizationLayer
    reluLayer
    convolution2dLayer(3,numF,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(3,'Stride',2,'Padding','same')

    convolution2dLayer(3,2*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer
    convolution2dLayer(3,2*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(3,'Stride',2,'Padding','same')

    convolution2dLayer(3,4*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer
    convolution2dLayer(3,4*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(3,'Stride',2,'Padding','same')

    convolution2dLayer(3,8*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer
    convolution2dLayer(3,8*numF,'Padding','same')
    batchNormalizationLayer
    reluLayer

    globalAveragePooling2dLayer

    dropoutLayer(0.5)

    fullyConnectedLayer(7)
    softmaxLayer
    classificationLayer];


%%
% Define <docid:nnet_ref#bu59f0q ... |traingOptions|> for the CNN. These
% options are based on [3] and modified ... through a systematic hyperparameter
% optimization workflow.

miniBatchSize = 128;
tuneme = 128;
lr = 0.05*miniBatchSize/tuneme;
options = trainingOptions('sgdm', ...
    'InitialLearnRate',lr, ...
    'MiniBatchSize',miniBatchSize, ...
    'L2Regularization',0.005, ...
    'MaxEpochs',8, ...
    'Momentum',0.9, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress', ...
    'Verbose',false, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',2, ...
    'LearnRateDropFactor',0.2);

%%
% Call <docid:nnet_ref#bu6sn4c |trainNetwork|> ... to train the network.

trainedNet = ...
    trainNetwork(xTrain,yTrain,layers,options);

%% Evaluate CNN
% Call ... <docid:nnet_ref#mw_0a51db93-cccf-4b2f-ae4c-6724cbf5ec46 ...
|predict|>
% to predict responses from the trained ... network using the held-out test
% set.

cnnResponsesPerSegment = ...
    predict(trainedNet,xTest);

%%
% Average the responses over each 10-second ... audio clip.

classes = trainedNet.Layers(end).Classes;
numFiles = numel(adsTest.Files);

counter = 1;
cnnResponses = zeros(numFiles,numel(classes));
for channel = 1:numFiles
    cnnResponses(channel,:) = ...
        sum(cnnResponsesPerSegment(counter:counter+numSegmentsP
    counter = counter + numSegmentsPer10seconds;
end

%%
% For each 10-second audio clip, choose the ... maximum of the predictions,
% then map it to the corresponding predicted ... location.

[~,classIdx] = max(cnnResponses,[],2);
cnnPredictedLabels = classes(classIdx);

%%
% Call ... <docid:nnet_ref#mw_b571feea-6af0-489f-b52c-a4b5141ac550
% |confusionchart|> to visualize the accuracy ... on the test set. Return the
% average accuracy to the Command Window.

figure('Units','normalized','Position',[0.2 ...
```

```
        0.2 0.5 0.5])
cm = ...
        confusionchart(adsTest.Labels,cnnPredictedLabels,'title',...
        Accuracy - CNN');
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';

fprintf('Average accuracy of CNN = ...
        %0.2f\n',mean(adsTest.Labels==cnnPredictedLabels)*100)


%% Feature Extraction for Ensemble Classifier
% Wavelet scattering has been shown in [4] to ...
        provide a good representation
% of acoustic scenes. Define a
% ...
        <docid:wavelet_ref#mw_a0dd2386-66f5-4c0b-b96d-7ae0ba327cd1 corresponding predicted location. Call
% |waveletScattering|> object. The invariance ...
        scale and quality factors
% were determined through trial and error.

sf = ...
        waveletScattering('SignalLength',size(data,1), ...
        ...
                          'SamplingFrequency',fs, ...
                          'InvarianceScale',0.75, ...
                          'QualityFactors',[4 1]);

%%
% Convert the audio signal to mono, and then call
% ...
        <docid:wavelet_ref#mw_edb9d918-ed3b-469f-8cb0-5e6cdd25e97
% |featureMatrix|> to return the scattering ...
        coefficients for the scattering
% decomposition framework, |sf|.

dataMono = mean(data,2);
scatteringCoeffients = ...
        featureMatrix(sf,dataMono,'Transform','log');

%%
% Average the scattering coefficients over the ...
        10-second audio clip.

featureVector = mean(scatteringCoeffients,2);
fprintf('Number of wavelet features per ...
        10-second clip = %d\n',numel(featureVector))

%%
% The helper function ...
        |HelperWaveletFeatureVector| performs the ...
        above steps.
% Use a <docid:matlab_ref#bvaomuj-1 |tall|> ...
        array with
% <docid:matlab_ref#bsz9tpz |cellfun|> and ...
        |HelperWaveletFeatureVector| to
% parallelize the feature extraction. Extract ...
        wavelet feature vectors for
% the train and test sets.

scatteringTrain = ...
        cellfun(@(x)HelperWaveletFeatureVector(x,sf),train_set_tall,'UniformOutput',false);
xTrain = gather(scatteringTrain);
xTrain = cell2mat(xTrain')';

scatteringTest = ...
        cellfun(@(x)HelperWaveletFeatureVector(x,sf),test_set_tall,'UniformOutput',false);
xTest = gather(scatteringTest);
xTest = cell2mat(xTest')';

%% Define and Train Ensemble Classifier
% Use |fitcensemble| to create a trained ...
        classification ensemble model
% (|ClassificationEnsemble|).

subspaceDimension = min(150,size(xTrain,2) - 1);
numLearningCycles = 30;
classificationEnsemble = ...
        fitcensemble(xTrain,adsTrain.Labels, ...
        'Method','Subspace', ...
        'NumLearningCycles',numLearningCycles, ...
        'Learners','discriminant', ...
        'NPredToSample',subspaceDimension, ...
        'ClassNames',removecats(unique(adsTrain.Labels)));

%% Evaluate Ensemble Classifier
% For each 10-second audio clip, call ...
        |predict| to return the labels and
% the weights, then map it to the ...
        <docid:nnet_ref#mw_b571feea-6af0-489f-b52c-a4b5141ac550 ...
        |confusionchart|>
% to visualize the accuracy on the test set. ...
        Print the average.

[waveletPredictedLabels,waveletResponses] = ...
        predict(classificationEnsemble,xTest);

figure('Units','normalized','Position',[0.2 ...
        0.2 0.5 0.5])
cm = ...
        confusionchart(adsTest.Labels,waveletPredictedLabels,'title
        Accuracy - Wavelet Scattering');
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';

fprintf('Average accuracy of classifier = ...
        %0.2f\n',mean(adsTest.Labels==waveletPredictedLabels)*100)


%% Apply Late Fusion
% For each 10-second clip, calling predict on ...
        the wavelet classifier and
% the CNN returns a vector indicating the ...
        relative confidence in their
% decision. Multiply the |waveletResponses| ...
        with the |cnnResponses| to
% create a late fusion system.

fused = waveletResponses .* cnnResponses;
[~,classIdx] = max(fused,[],2);

predictedLabels = classes(classIdx);

%% Evaluate Late Fusion
% Call |confusionchart| to visualize the fused ...
        classification accuracy.
% Print the average accuracy to the Command ...
        Window.

figure('Units','normalized','Position',[0.2 ...
        0.2 0.5 0.5])
cm = ...
        confusionchart(adsTest.Labels,predictedLabels,'title','Test
        Accuracy - Fusion');
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';

fprintf('Average accuracy of fused models = ...
        %0.2f\n',mean(adsTest.Labels==predictedLabels)*100)

function X = ...
        HelperSegmentedMelSpectrograms(x,fs,varargin)
% Copyright 2019 The MathWorks, Inc.
p = inputParser;
```

```
addParameter(p,'WindowLength',1024);
addParameter(p,'HopLength',512);
addParameter(p,'NumBands',128);
addParameter(p,'SegmentLength',1);
addParameter(p,'SegmentOverlap',0);
addParameter(p,'FFTLength',1024);
parse(p,varargin{:})
params = p.Results;

x = [sum(x,2),x(:,1)-x(:,2)];
x = x./max(max(x));

[xb_m,¬] = ...
    buffer(x(:,1),round(params.SegmentLength*fs),round(params.SegmentOverlap*fs),'nodelay');
[xb_s,¬] = ...
    buffer(x(:,2),round(params.SegmentLength*fs),round(params.SegmentOverlap*fs),'nodelay');
xb = ...
    zeros(size(xb_m,1),size(xb_m,2)+size(xb_s,2));
xb(:,1:2:end) = xb_m;
xb(:,2:2:end) = xb_s;

spec = melSpectrogram(xb,fs, ...
    'Window',hamming(params.WindowLength,'periodic'), ...
        ...
    'OverlapLength',params.WindowLength - ...
        params.HopLength, ...
    'FFTLength',params.FFTLength, ...
    'NumBands',params.NumBands, ...
    'FrequencyRange',[0,floor(fs/2)]);
spec = log10(spec+eps);

X = ...
    reshape(spec,size(spec,1),size(spec,2),size(x,2),[]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%HelperWaveletFeatureVector
function features = ...
    HelperWaveletFeatureVector(x,sf)
x = mean(x,2);
features = featureMatrix(sf,x,'Transform','log');
features = mean(features,2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
% Copyright 2019 The MathWorks, Inc.
```