ABSTRACT

In this paper, we propose to explore several deep machine learning methods for Urban sound classification (USC) tasks. Our network architecture extracts high-level feature representations from spectrogram-like features (MFCC and Mel spectrogram). Furthermore, we test traditional machine learning methods and compare their performance with deep learning models. Experiments are conducted on UrbanSound8K. Our experimental results demonstrate that deep learning module (ResNet) has achieved the best performance.

Index Terms—Sound Classification, Machine Learning, MFCC, Mel spectrogram, Urbansound8k

1. INTRODUCTION

Over the past five years, developments in artificial intelligence have moved into the medium of sound, whether it be in generating new forms of music (with varying degrees of success), or identifying specific instruments from a video. Sound recognition is a front and center topic in today’s pattern recognition theories, which covers a rich variety of fields. Some of sound recognition topics have made remarkable research progress, such as automatic speech recognition (ASR) [1] [2] and music information retrieval (MIR) [3] [4]. Urban sound classification (USC) is another branch of sound recognition and is widely applied in surveillance [5], home automation [6], scene analysis [7] and machine hearing [8]. However, unlike speech and music, sound events are more diverse with a wide range of frequencies and often less well defined, which makes Urban Sound Classification tasks more difficult than ASR and MIR. Hence, USC still faces critical design issues in performance and accuracy improvement.

Traditional machine learning methods such as random forest, are usually associated with audio-based machine learning projects, but convolutional neural networks also work well on sound classification. In this paper, we will use neural networks, together with some helpful audio analysis methods, and compare the performance of traditional machine learning method and convolutional neural network. In this paper, we will use some neural network made in pytorch, together with some helpful audio analysis libraries, which can distinguish between 10 different sounds with high accuracy.

2. RELATED WORK

Sound classification has always been an important research topic and traditional machine learning algorithms including KNN, SVM and random forest are the main classification methods. However, they performed poorly in large sound classification datasets including Urbansound8k dataset [9] and ESC dataset [10] proposed in recent years. With the emerge of deep learning in computer vision area, using convolutional neural networks (CNN) has proved to be a promising direction in several researches [11] [12]. It also adopted several useful feature extraction techniques including MFCC and Mel spectrogram.

CNN has become the most popular method in image classification task since its first success in large scale image recognition tasks [13]. CNN has evolved through several generation since then. Important milestones also includes VGGNet [14] and GoogLeNet [15]. ResNet [16] is the latest and most successful CNN architecture so far. Before ResNet, deep networks usually suffer from severe gradient vanishing problem. Kaiming He introduced a residual connection that establish shortcuts between each CNN block to help the gradient to propagate to deeper layers. Because of its stable performance in spatial feature extraction, variations of ResNet are also commonly used in tasks other than image classification. Sound recognition problem can also be considered as a variation of sequence understanding. LSTM [17] and GRU [18] module are most frequent temporal context extractor in sequence problem.

3. DATASET AND FEATURES

3.1. Urbansound8k

We used Urbansound8k, a standard benchmark for sound classification, to evaluate out methods. It contains 8732 sounds excerpts of various lengths. The excerpts are sampled from the 10 classes including air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. Visualization of random chosen examples are shown in figure 1. A detailed statistics of Urbansound8k is shown in table 1. We have to use 10-fold cross validation to fully evaluate our methods following the origin split of the dataset. The 10 folds are not as difficult...
and there are correlations between some folds. Sound clips in different fold might be sampled from a same sound source. Thus, using different data splits will produce false results.

![Fig. 1. Audio samples](image)

<table>
<thead>
<tr>
<th>categories</th>
<th>ac</th>
<th>ch</th>
<th>cp</th>
<th>db</th>
<th>dr</th>
</tr>
</thead>
<tbody>
<tr>
<td>samples</td>
<td>1000</td>
<td>429</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>categories</td>
<td>ei</td>
<td>gs</td>
<td>jh</td>
<td>si</td>
<td>sm</td>
</tr>
<tr>
<td>samples</td>
<td>1000</td>
<td>374</td>
<td>1000</td>
<td>929</td>
<td>1000</td>
</tr>
</tbody>
</table>


### 3.2. Feature Extraction

In audio classification, we always transform the audio data from time field into the frequency field for speech processing and then analysis. Here we apply two features which are mostly used and helpful – Mel-Spectrogram and Mel-Frequency Cepstral Coefficient (MFCC). We use the Python library “librosa” to extract them.

#### 3.2.1. Mel-Spectrogram

The Mel-Spectrogram is a Spectrogram with the Mel Scale as its y axis. Usually, the Mel Spectrogram is the result of these four steps: [19] 1. Separate to windows: Sample the input with windows of size $n_{fft}$. And to sample the next window, make hops of size $hop\_length$ each time. 2. Compute FFT (Fast Fourier Transform) for each window to transform from time domain to frequency domain. 3. Generate a Mel scale: Take the entire frequency spectrum, and separate it into $n_{mels}$ evenly spaced frequencies. This Mel Scale is constructed such that sounds of equal distance from each other on the Mel Scale, also “sound” to humans as they are equal in distance from one another. 4. Generate Spectrogram: For each window, decompose the magnitude of the signal into its components, corresponding to the frequencies in the mel scale. Mel-Spectrograms plots of 10 urban sound classes are shown in figure 2.

![Fig. 2. Mel-Spectrograms plots of 10 urban sound classes](image)

#### 3.2.2. Mel-Frequency Cepstral Coefficient (MFCC)

Mel-Frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an Mel-Frequency Cepstrum (MFC). And the MFC is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. MFCC mimics how human’s hearing system works. [20] Usually, MFCC is the result of these five steps: [21] 1. Take the Fourier transform of (a windowed excerpt of) a signal. 2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows. 3. Take the logs of the powers at each of the mel frequencies. 4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal. 5. The MFCCs are the amplitudes of the resulting spectrum. MFCCs plots of 10 urban sound classes are shown in figure 3.

![Fig. 3. MFCC plots of 10 urban sound classes](image)

### 4. METHODS

#### 4.1. CNN Architecture

VGG is a Convolutional Neural Network architecture invented by Visual Geometry Group (Oxford University). [14] There are some VGGn models in which n is the layer number. Compared the results and because of the time limit, we choose VGG11 finally. In figure 4, there shows the architecture of VGG11.

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts
to jump over some layers. Typical ResNet models are implemented with double- or triple-layer skips that contain nonlinearities (ReLU) and batch normalization in between. Models with several parallel skips are referred to as DenseNet. For single skips, the layers may be indexed either as $l - 2$ to $l$ or as $l$ to $l + 2$. Given a weight matrix $W_{l-1,l}$, for connection weights from layer $l - 1$ to $l$, the forward propagation through the activation function would be as follow:

$$a^l := g(W_{l-1,l}a^{l-1} + b^l + W^{-2,l}a^{l-2})$$

Where $a^l$ is the activation (outputs) of neurons in layer $l$, $g$ is the activation function for layer $l$, $W_{l-1,l}$ is the weight matrix for neurons between layer $l - 1$ and $l$, $Z^l = W_{l-1,l}a^{l-1} + b^l$. The backward propagation is shown as: For normal path:

$$\Delta w_{l-1,l} = -\eta \frac{\partial E^l}{\partial w_{l-1,l}} = -\eta \alpha (l-1) \delta^l$$ (2)

For skip path:

$$\Delta w_{l-2,l} = -\eta \frac{\partial E^l}{\partial w_{l-2,l}} = -\eta \alpha (l-2) \delta^l$$ (3)

Skipping effectively simplifies the network, using fewer layers in the initial training stages. This method speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The configuration of ResNet with different depth is shown in figure 5.

4.2. RNN

Both mel-spectrogram and MFCC can be seen as a sequence of consecutive features. We tried to use an LSTM cell in order to capture not only spatial but also temporal patterns. A 2-layer LSTM aims at aggregating the feature overtime to the hidden state of the final LSTM cell. The state is considered as the feature and feed to a fully connected classifier.

4.3. Traditional Methods

Although there are some reasons that deep learning methods behave better. For example, traditional machine learning methods experience underfit in high dimensional data and dimension reduction may lose detailed information. But we do try some traditional method to verify its inability. We then use K-Nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), Random Forest (RF) and XGBoost. We use the Python library "sklearn" [22] to apply them. The results are shown in part V.

5. EXPERIMENTS

5.1. Preprocessing

We use a 22.05kHz sampling rate for UrbanSound8K datasets. Different preprocessing is applied depending on desired input feature for deep learning models. We used two spectrogram-like representations, Mel spectrogram and Mel-frequency cepstral coefficients (MFCC) as described in section 3.2. When using RGB image as input, we apply random crop for data augmentation and normalization for projecting them to preferable input distribution.

5.2. Training settings

All deep learning models are trained with batch size of 32. We used a learning rate decay schedule with an initial learning rate of 0.001. Then the learning rate will decrease with a factor $\gamma = 0.1$ every 20 epoch of UrbanSound8K. The models are trained for 100 epochs for UrbanSound8K. We use cross entropy as the loss function, which is typically used for multi-classification task. In the testing stage, feature extraction and normalization remains the same. Random crop will not be applied to testing data. Model outputs are projected to probability space by Softmax function. The classification performance of the methods is evaluated by the 10-fold cross validation.

5.3. Results

As presented in table 3, we used the data split of fold 3 to briefly evaluate traditional machine learning methods including SVM, KNN, Random Forest and XGBoost. Based on our testing result, in general the performance is ranked as XGBoost > Random Forest > KNN > SVM. MFCC is the better feature for KNN, Random Forest and XGBoost, while Mel spectrogram is most suitable for SVM. MFCC feature $(n_{mfcc} = 80)$ with XGBoost achieves the best testing accuracy, 51.03%, out of all possible combinations of features and methods. In comparison of this, the third column of table 2 presents the performance of ResNet18 using the same data.

We did a comprehensive evaluation on the performance of CNN models. Figure 6 and 7 shows the difference be-
We also tested the performance of RNN, DNN model but they both had unsatisfying performance of around 50% and 35% respectively. To explore the underlying cause and fact of our results we investigate in to number of parameters in these 6 deep learning models that we tested. As presented in table 6, ResNet18 has $5 \times$ more parameters than LSTM model while LSTM model has $10 \times$ more parameters than DNN. Thus the poor performance of DNN and LSTM model is due to the lack of representational abilities. In other words, the models experience underfit when training such a large dataset. VGG11 has $10 \times$ more parameters than ResNet18, but as mentioned before, network degrading problem and gradient vanishing problem make the parameters less efficient compared with ResNet. ResNet34 and ResNet50 both have $2 \times$ parameters than ResNet18. But these deep models have stronger representational abilities we can observe overfitting in the training.

![Fig. 6. Accuracy and loss in fold 3](image1)

![Fig. 7. Accuracy and loss in fold 10](image2)

ResNet50 has more filter but similar total number of parameters, thus, it outperformed ResNet34 in this task.

![Table 6. Parameters](image3)
7. REFERENCES


A. INDIVIDUAL CONTRIBUTION

Wei Wang: For theory foundation, I took part in every discussion like theme choosing and so on. For code part, I developed lower dimensional features extraction, all the traditional methods and vgg11 method. Also, I participated in some experiments of Resnet34 and Resnet50. As for writing, I helped with PPT presentation and final report about what I did (about 1/3 of all work).

Tianyu Zhao: In theory part, I researched into feature extraction methods especially MFCC. I contribute to the development of dataset API and CNN, RNN training framework. For experiments, I tuned the hyper-parameter, evaluate the performance of ResNet18 and etc. I also participate in presentation and final report writing. (about 1/3 of all work).

Hangquan Zhao: I participate in every project selection, discussion and design. In the coding part, I respond for the ResNet18, 34, 50 part. I construct the training and testing network and make data collection of the result. Also, I make some contribution to the VGG and feature extraction part, where I extract the Mel Spectrum feature and output the feature map into the specific directory. In the writing part, I take part in all the process of PPT presentation, code reviewing and the final report(nearly 1/3 of all work).
B. REPLY TO REVIEW

Critical review from team 12:

Question: You should cite all references you used images from.
Response: We have deleted some unnecessary images and cited references for others.

Question: Instead of only top-1 accuracy, you can try more metrics like top-3 accuracy.
Response: Since we just have 10 classes, although top-3 accuracy might be larger, but top-1 accuracy is more accurate and meaningful for comparison.

Question: You should explain MFCC and MEL since I think not everyone is familiar with signal processing.
Response: They are now explained on the “feature extraction” part of the report.

Question: You should provide more detailed observations instead of just going through which models perform better, and list them down.

Question: e.g. Why do you think resnet 50 performs worse than resnet 18 and 34?
Response: According to our further experiments, ResNet50 outperformed ResNet34 but still can’t exceed the accuracy achieved by ResNet18. Possible reasons are summarized in the last paragraph of section 5.3.

Question: Is 10-fold cross validation necessary? Maybe 4-fold or 3-fold is enough?
Response: In the dataset website “https://urbansounddataset.weebly.com/urbansound8k.html”, it says “Use the predefined 10 folds and perform 10-fold (not 5-fold) cross validation”. The reason is that our results will NOT be comparable to previous results in the literature, meaning any claims to an improvement on previous research will be invalid.

Question: References should also include the papers of xgboost, knn, and svm, random forest, related works you presented, mel-frequency spectrogram and MFCC.
Response: We have deleted some unnecessary introductions for the traditional methods and cited references for the left part.

Critical review from team 24:

Question: Some of the slides seem to be crowded. I suggest you could either delete some images that are duplicated in context, or put some content on a new slide.
Response: When doing the presentation, we followed the requirements strictly on canvas. Because of the page limitation and lots of contents. That makes some of the slides crowded. Sorry for your inconvenience when reading.

Question: In the “Why deep learning” slide, I see the need to use state-of-the-art vision model like VGG. Could you please explain why such models could outperform the traditional methods even if the overfitting still exists?
Response: In this slide, we explained that it is because traditional methods can’t do well on higher dimensional data like images and reduce the dimension will cause data information loss. And from the experiments, there are always overfitting problems which we must admit. For example, the accuracy is 99% on the training dataset and 75% on the test dataset. But compared with the all the experiments with dataset “UrbanSound8K” on the Internet, our results are among the best ones which means overfitting problem is inevitable. Maybe you can find some results online reached 99% accuracy also on the test dataset. But after we read the original papers, we found either they used the wrong methods or faked the results.

Question: I can see the project is kind of results-oriented. But I think it's better if you could explain the difference between the results of different combinations. Why Mel-Spectrum and ResNet is much better than the others you tried?
Response: The explanations are in the results part of the report.

Question: On a personal note, I would have the “Results/Observations” between “Traditional Methods” and “Deep Learning Methods” to be first of this parts. Because it seems that you first try to show deep learning is much better than traditional ways and then tried to find the best one among the deep learning methods.

Response: Yes, we followed strictly the requirements on canvas so the contents are ordered also following the requirements. Sorry for the inconvenience when reading.

Critical review from team 42:

Question: For the background part, did not clearly explain what you have done. I expected a brief-intro like stuff rather than what the outcome could be used for.

Response: The improvement background part is shown in the report.

Question: It is unclear about what this project actually did, stated the superiority of deep leaning before things started while seems still used some traditional methods later. Concludes after comparison maybe comes better.

Response: Yes, it’s better that concludes after comparison. But when presentation, we followed strictly the requirements on canvas. It let us to talk “Why Deep Learning Methods” first. Sorry for the inconvenience when reading.

Question: Images quoted lack of consistency, some with label while some are not.

Response: This has been fixed in the report.

Question: For the relate works part, explanation is too brief, did not explain the figures shown on the slides.

Response: Related works are well explained then in the report.

Question: For the feature extraction part, there are some raw codes for dimensionality reduction, which seems unnecessary.

Response: Those codes have been removed in the report.

Question: For the details on used models, the frames of CNN models are too small to read.

Response: This has been fixed in the report.