

# GROUP 25: Audio Feature Extraction And Classification for Urban Sound

Project Repository: <https://github.com/yuxuan3713/ECE-228-Project>

Chen, Cai  
cac005@ucsd.edu

Liu, Yuxuan  
yul067@ucsd.edu

Sun, Haoran  
hsun@ucsd.edu

Zhou, Moyan  
moz006@ucsd.edu

*Abstract*—Recent studies in urban planning and architectural design industries have demonstrated the importance of dynamic sound classification in the urban environment. In this paper, we further explore the characteristics of urban sound because the environmental sounds are unstructured and combined with noise. Six features are evaluated to recognize ten types of urban sounds. We trained three models: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Support Vector Machine (SVM) on different features of audio clips. Concerning the accuracy of classification, we investigated sample rate, window size, feature domains and all combinations of features. Currently, the best accuracy is 91.1% produced by RNN model with combined features.

## I. INTRODUCTION

The dynamic sound in the urban environment is still neglected aspects of city planning [1]. Most people are in general largely unaware of the importance of sounds for how we perceive the quality of a place and a good living environment. Urban sound classification could have much contribution in that it helps to unify independent areas related to sound and environment. The problem of classifying sound is that the feature of audio data is more complex than visual objects and how the feature is processed will have a huge impact on the result. Three models, CNN, RNN, and SVM, are trained on different features of audio clips (input), and output ten classes of audio with accuracy.

Automatic classification of urban sounds become increasingly useful in many applications, such as mobile devices aiming to provide the functionality of labeling audio or video by its audio data. Unlike images, audio data cannot be classified by the neural network trained for images and require a preprocessing for data pipelining. Also, it is unclear what feature is most robust and

discriminative that best represents audio data. We are aiming at exploring multiple features from audio – time domain features (RMSE), frequency domain features (FFT), perceptual features (MFCC), windowing features, etc – and apply both deep learning and standard statistical learning-based classification algorithms.

## II. RELATED WORK

Audio classification is one of the most popular research area nowadays. One classification problem that is similar to our project is environment sound recognition (ESR).[2] It has researched in, among others, temporal domain, frequency domain, and cepstral domain [3]. Sachin et al. [2] show that features extracted from frames (often using a Hanning or a Hamming window) can be used as an instance of training. However, a main drawback of the approach is that it is so hard to select an optimal framing-window length suited for all classes. Features based on psychoacoustic properties of sounds are more and more developed for ESR. Some researchers have investigated zero-crossing rate (ZCR), spectral flux, etc. of audio signals to discriminate different classes. But those features provide rough details about temporal and spectral properties which are not the state-of-the-art techniques [4], [5]. Auto-regressive based features, especially linear prediction coefficients (LPC), have been commonly used in speech processing applications. The Linear Prediction Cepstral Coefficient (LPCC) is an alternative representation of LPC and is also commonly used. However, LPC and LPCC reflect the source filter model of speech, so they are not useful for ESR [6], [7], [8].

A widely used feature is cepstral features such as MFCC [9], [10], [11], [12], [13] and  $\Delta MFCC$  and  $\Delta\Delta MFCC$ . MFCC works better on the neural network than the above features. Due to the limited amount of data in ESR, MFCC is often used by researchers as a benchmark to compare with other work. We use MFCC as one of our basic features and concatenate MFCC with our newly investigated features (they will be discussed in Section 3) to enhance the accuracy of our model. In the frequency domain, we also adopted Fourier transform as it describes the nature of the physical phenomenon forming the signal [14], [13], [15]. In the temporal domain, for decades, researchers use the wavelet transform to represent non-stationary signals. We analyzed the energy property of the wavelets to check its ability to classify the audio signals [16].

### III. DATASET AND FEATURES

#### A. Dataset

Our Dataset is called UrbanSound8K. It contains 8732 labeled environmental sound files in .wav format. These files are pre-sorted into 10 folders. Each file name indicates its ID. The dataset also has a CSV file linking each file ID, its corresponding folder, and its type of sound. There are 10 classes of sounds: Air conditioner, Dog bark, Engine idling, Children playing, Drilling, Gun shot, Jackhammer, Siren, and Street music. Each sound file is no longer than four seconds.

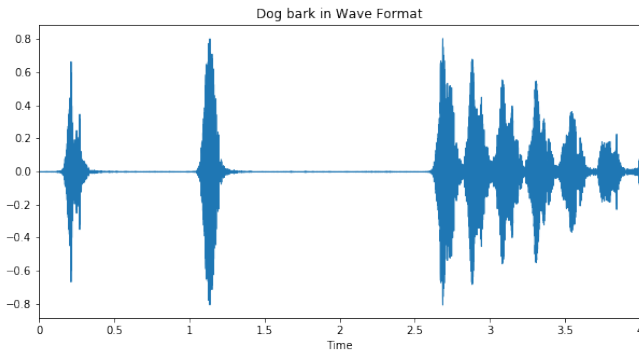


Fig. 1. Dog Bark: Wave Format

#### B. Features

The original feature size varies due to different sample rates of .wav files. Thus, we use padding

or duplication to obtain feature vectors of desirable length empirically and extract some combinations of the following features.

1) *Fast Fourier Transform (FFT)*: To transfer audio files into features, we use fast Fourier transform, specifically the short time Fourier transform (STFT), applies windows onto the signal and transferring the signal from the time domain to frequency domain in order to better characterize a sound. Windows need to be overlapped to prevent leakage. The feature length extracted is 2000.

2) *Root Mean Squared Energy (RMSE)*: The energy of a signal corresponds to the total magnitude of the signal (energy could also help to recognize a sound). In our case, the feature length is 100.

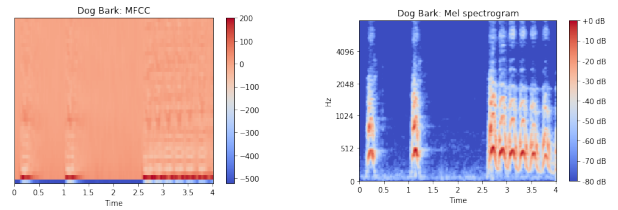


Fig. 2. The MFCC spectrogram

Fig. 3. Mel spectrogram

3) *Mel Frequency Cepstral Coefficient (MFCC)*: The Mel frequency resembles the human auditory system and is widely used in speech recognition [2]. Filtering power spectrum by using discrete cosine transform(DCT) can get the mel frequency cepstral coefficients. In our case, the MFCC feature length is 256.

4) *Chromagram*: Chromagram depicts twelve pitch-class profiles of a given audio as shown in figure 4 [17].

5) *Mel spectrogram*: Mel spectrogram is derived by mapping the spectrogram onto the mel-scale. The mel-scale is a pitch comparison which judged by the listeners. By applying the scale, the sound will be more suitable for human auditory system. The relation of f herz and m mels is shown below [18]:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (1)$$

6) *Spectral contrast*: Spectral contrast includes the spectral peak, the spectral valley, and their difference in each frequency sub-band features to enhance the performance [17].

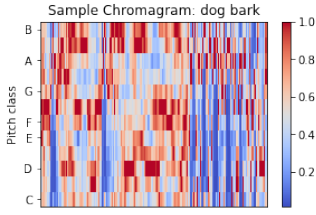


Fig. 4. The chromogram

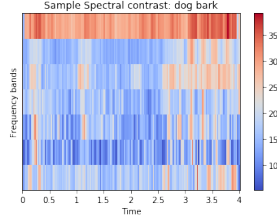


Fig. 5. The Spectral Contrast

#### IV. METHODS

To implement each method, we used the tools mentioned in table I.

Tool	Usage
Deep Learning Toolbox	Matlab package provides the framework of CNN, RNN, and SVM model and their related functions for tuning
Parallel Computing Toolbox	Matlab package used to do computation of eight thousand audio files in multiple cores
Audio Toolbox	Matlab package to extract the fast fourier transform feature, root mean squared energy, and Mel Frequency Cepstral Coefficient.
Librosa	Python package used to extract contrast, MFCC, and chromagram. It is also used to plot feature related spectrograms.
Matplotlib	Python 2D plotting package [9] to plot all the sample spectrograms.
Numpy, Scipy and Sklearn	Python packages used to normalizing the feature data and plot graphs with the normalized data.

TABLE I

MATLAB AND PYTHON TOOLS USED

##### A. CNN Architecture

Our CNN model is composed of four kinds of layers: convolution, ReLU, pooling and fully-connected as shown in figure 6. The convolution layers stacks 8, 16, and 32 feature maps, respectively, each with a  $3 \times 3$  filter. The convolution layer will capture some types of feature in the input. The ReLU layers apply the function  $f(x) = \max(0, x)$  to remove negative values and add some non-linearity. The operations of the first three kinds of layers are repeated three times. In the end, the prediction is made by fully-connected layer, which

has access to all parameters learned by previous layers.

##### B. RNN Architecture

A Long Short-Term Memory (LSTM) is a type of RNN that can be better used for time series data classification since it can optionally memorize or forget information based on its internal status. We use five LSTM layer to receive sequence or time series data, then the network also ends with a fully connected layer, a softmax layer and a classification output layer to predict the labels (see Figure 7). The diagram in Figure 8 illustrates the flow of a time series  $X$  with  $C$  features of length  $S$  through an LSTM layer. LSTM model is composed of a cell  $C$ , an input gate  $i$ , an output gate  $o$ , and a forget gate  $f$ .  $x_t$  is an input vector to LSTM,  $h_t$  is a hidden state vector,  $bs$  are bias terms, and  $\sigma(x)$  is an activation function [19]. The LSTM model has the following equations: .

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 C_t &= \sigma(f_t \circ C_{t-1} + i_t \circ \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)) \\
 h_t &= o_t \circ \tanh(C_t)
 \end{aligned} \tag{2}$$



Fig. 7. Architecture of a simple LSTM network for classification

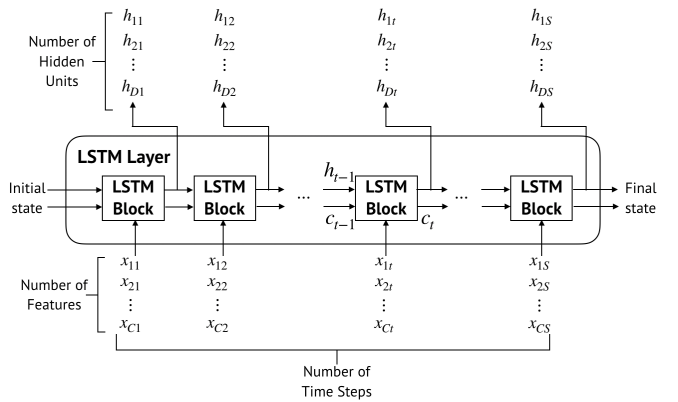


Fig. 8. Flow chart of LSTM layer

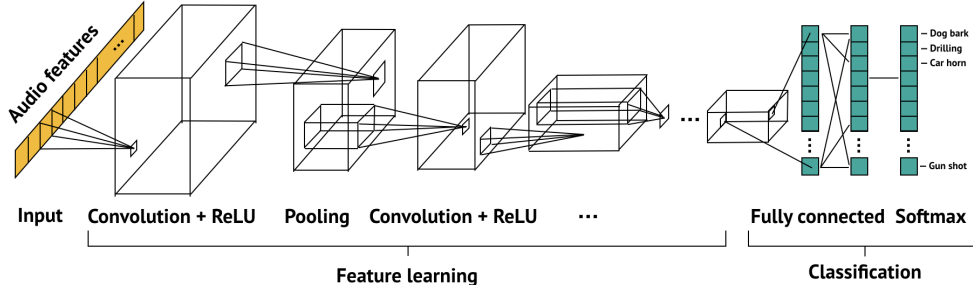


Fig. 6. Example of CNN network with some convolutional layers

### C. Kernel SVM Model

Support vector machine finds a hyperplane in an N-dimensional space which can maximize the margin among different data clusters. Since it is a supervised learning algorithm, we use the grid search to choose the type of the kernel, corresponding parameter  $C$ , and gamma in kernel function to optimize the result. We found radial basis function kernel (with  $C = 32$  and  $\gamma = 128$ ) trained on FFT with 2000 features gives the best result. The radial basis function is defined as  $K(x, x') = \exp(\gamma \|x - x'\|^2)$ , which can possibly map the original linearly inseparable feature space to a linearly separable one. Let  $\mathbf{w}$  be the weight vector of SVM and  $\phi(x)$  be some kernel function, then  $\mathbf{w}$  can be rewritten as the following form:

$$\mathbf{w} = \sum_{j=1}^n \alpha_j \mathbf{y}^{(j)} \phi(\mathbf{x}^{(j)}) \quad (3)$$

The kernel SVM formula in dual form is:

$$\begin{aligned} \max_{\alpha \in R^n} \sum_{i=1}^n \alpha_i - \sum_{i,j=1}^n \alpha_i \alpha_j \mathbf{y}^{(i)} \mathbf{y}^{(j)} \phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)}) \\ \text{s.t. } \sum_{i=1}^n \alpha_i \mathbf{y}^{(i)} = 0, 0 \leq \alpha_i \leq C \end{aligned} \quad (4)$$

The SVM implementation used in this project is provided by LIBSVM, which uses "one vs one" approach for multi-class classification problem. "One vs one" approach creates  $k(k+1)/2$  classifiers for  $k$  classes, and the prediction is determined by a majority vote among all classifiers.

## V. RESULTS AND DISCUSSION

### A. Experiments

For the CNN model, our hyper-parameters include learning rate and epoch size. Learning rate determine the step size the optimizer takes to find the local minimum; and epoch size determines how many times the training data are going to be feed into the model. A randomized test set is used to tune those hyper-parameters. Based on different feature used and the randomized test set, those hyper-parameters might vary.

For kernel SVM model, our hyper-parameters include  $C$  and  $\gamma$ .  $C$  determine how much the model wants to avoid misclassification. With a smaller  $C$ , the model will produce a hyper-plane with a large margin, and thus produces more misclassified samples; in contrast, a larger  $C$  will reduce the margin size and make more correct classifications but with a higher risk of over-fitting.  $\gamma$  in RBF kernel determines how far the influence of a single training example can reach, and it will determine the number of support vectors in the model. These hyper-parameters are tuned via grid search: checking all combinations between two hyper-parameter vectors and pick the parameter values that result in highest accuracy.

For RNN model, we compared *lstmLayer* and *bilstmLayer* options in LSTM model. The *lstmLayer* represents the unidirectional LSTM layer which only preserves information of the past. Using bidirectional will run the inputs in two ways, one from past to future and one from future to past, which can preserve information from both directions. By testing the two layer options, we find that *bilstmLayer* shows better results as they can understand context clearer.

## B. Results

Our evaluation metric is accuracy, which is the number of correctly classified samples divided by the total number of samples. For this project, we only care about what fraction of predictions is correct. Along with the experiments, we combine these six features with three models. We picked eleven relatively better results and listed in table II below. It shows that using combined features, MFCC, mel-spectrogram(Mel), cepstral contrast(Contrast), chromagram(Chroma), on RNN produces the best test accuracy.

Models Used	Features	Test Accuracy
CNN	FFT	85.5%
SVM	FFT	63.3%
CNN	RMSE	49.6%
RNN	MFCC	88.8%
RNN	MEL	74.3%
RNN	Contrast	40.2%
RNN	Chromagram	29.6%
RNN	MFCC + Mel	88.7%
RNN	MFCC + Contrast	89.0%
RNN	MFCC + Chromagram	89.2%
RNN	Combined	91.1%

TABLE II  
CLASSIFICATION ACCURACY OF DIFFERENT FEATURES ON DIFFERENT MODELS

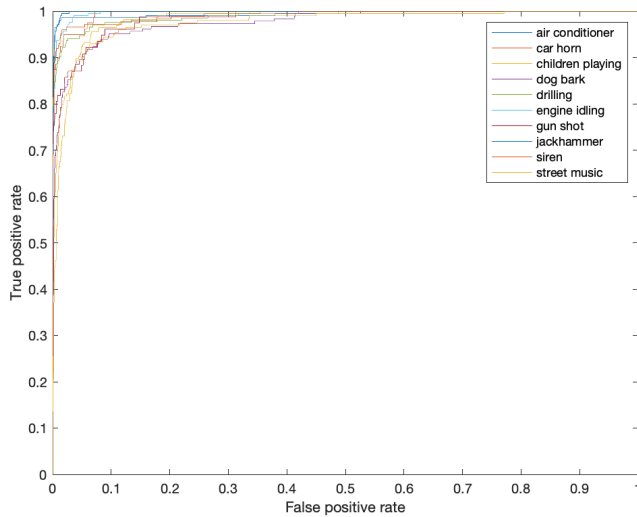


Fig. 9. AUC-ROC Curves for each Class Using Best Model on Test Set

Class	AUC	Class	AUC
air conditioner	0.9942	engine idling	0.9973
car horn	0.9954	gun shot	0.9847
children playing	0.9847	jackhammer	0.9990
dog bark	0.9766	siren	0.9918
drilling	0.9903	street music	0.9738

TABLE III  
AUCs FOR EACH CLASS USING BEST MODEL ON TEST SET

True Class	air_conditioner	car_horn	children_playing	dog_bark	drilling	engine_idling	gun_shot	jackhammer	siren	street_music	Accuracy	Percentage
air_conditioner	204			1						2	98.6%	1.4%
car_horn	1	70	3		1	2				2	88.6%	11.4%
children_playing	7	3	186	4	6	1	2			4	83.4%	16.6%
dog_bark	4	5	6	144		5	5	1	8	5	78.7%	21.3%
drilling	1	1	3	5	183				6		89.7%	10.3%
engine_idling	5		5	2		188					91.3%	8.7%
gun_shot	1		10	4			59			1	76.6%	23.4%
jackhammer		1			4			190			96.4%	3.6%
siren	2		4	3				2	160	2	92.5%	7.5%
street_music	17	1	15	5	5	2	1	2	3	146	74.1%	25.9%
	84.3%	86.4%	80.2%	85.7%	92.0%	94.9%	88.1%	94.5%	89.9%	81.1%		
	15.7%	13.6%	19.8%	14.3%	8.0%	5.1%	11.9%	5.5%	10.1%	18.9%		

Fig. 10. Confusion Matrix of Best Model on Test Set

## C. Discussion

Based on the result, MFCC, mel-spectrogram, spectral contrast, and chromagram features are not discriminative on CNN and not linearly separable on SVM. One reason that using the CNN has low accuracy is that CNN network does not preserve ordering of input data and not consider the time-domain correlation of the input features.

Combined features on RNN network gives the best classification among 11 results. One possible reason is that the combined feature covers more characteristics of an audio clip. For instance, although the chromagram itself gives a low classification accuracy, some types of sounds with distinguishable pitches can be better classified by using more general features along with the chromagram.

As for the RMSE feature, it captures the energy/magnitude of a sound clip; however, magnitude alone cannot be used to distinguished different kinds of sound, and noises in the clip can significantly lower the quality of this measurement.

From the confusion matrix of our best model, it is easy to see that the difficulties of classifying

different classes varies. For example, street music is the most confused class since the type of music varies too much, and they might have similar features to children playing in some aspects. In contrast, jackhammer is probably the least confused class since the sound fallen into this class are very distinguishable.

On the other hand, we found that in the Urbansound8k website, it provided pre-separated 10 folds for cross-validation propose. We did quite a lot research for the data set and considered carefully. It is easy to overlook that Urbansound8k's 10 folds put audios fractions of one long-term audio in the same fold. Under this circumstance, we thought dropping out one of these ten folds will reduce the information for training which leads us an improper model. Another reason why we don't choose to use these 10 folds for cross-validation is that we also thought ensuring randomness is important.

## VI. CONCLUSIONS

### A. Conclusion

For the project, we put six different features on three machine learning models respectively in order to produce a better result of classifying urban sound audio clips in 10 different classes. We examine that RNN network can produce better results for most of the features. The best classification accuracy is generated by using combined features (MFCC, Mel, Contrast, and Chroma) on RNN model based on the results. The reason that combined features on RNN is the best model is that the combined feature captures more general characteristics of a sound. Although some features themselves give low classification accuracy, the accuracy of classifying could increase by combining these features with the ones which produce better classification results.

### B. Future Work

Because of the limitation in time, there are still some ideas worth a closer look. Other than RNN and CNN, we aim to compare other neural networks, such as DNN. Applying filters such as the Gabor filter to input signals before extracting could also be a way to extract more representative feature data. Other datasets could also use to calculate the classification accuracy. By using the same

features and models on different datasets, such as the ESC-50 dataset which contains 50 classes, we can examine whether RNN and combined features always the best model for general urban sound audio files. Furthermore, since the training process shows that using CNN may lead to overfitting, we could use Principal Component Analysis (PCA) to reduce the input data's dimension.

## VII. CONTRIBUTIONS

*Cai Chen*: Did research on the audio classification related work and the data set, extracted cepstral features from the data and tuned parameters of CNN and LSTM models.

*Haoran Sun*: Implemented RMSE feature extraction and some MATLAB utilities for generating performance plots. Performed experiments using RMSE feature on some models. Visualized some basic features.

*Moyan Zhou*: Explored papers, found representative features, and researched the meaning behind each feature. Extracted features, created plots to visualize features, and organized in the feature plots notebook.

*Yuxuan Liu*: Built machine learning models and pipelined data into them. Developed and maintained code utilizing Matlab toolboxes. Created FFT / STFT feature extraction code. Ran various tests and validation on built ML model. Fine tuned ML model.

*Cai, Moyan, and Haoran* designed the poster and wrote the report.

## REFERENCES

- [1] T. Beatley, "Celebrating the natural soundscapes of cities," *The Nature of Cities*. Online post, 2013.
- [2] S. Chachada and C.-C. J. Kuo, "Environmental sound recognition: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.
- [3] D. Mitrović, M. Zeppezauer, and C. Breiteneder, "Features for content-based audio retrieval," in *Advances in computers*. Elsevier, 2010, vol. 78, pp. 71–150.
- [4] J. D. Deng, C. Simmermacher, and S. Cranefield, "A study on feature analysis for musical instrument classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 429–438, 2008.
- [5] F. Gouyon, F. Pachet, O. Delerue *et al.*, "On the use of zero-crossing rate for an application of classification of percussive sounds," in *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, 2000.
- [6] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnm," *Expert systems with applications*, vol. 36, no. 3, pp. 6069–6075, 2009.

- [7] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern recognition letters*, vol. 22, no. 5, pp. 533–544, 2001.
- [8] E. Tsau, S.-H. Kim, and C.-C. J. Kuo, "Environmental sound recognition with celp-based features," in *ISSCS 2011-International Symposium on Signals, Circuits and Systems*. IEEE, 2011, pp. 1–4.
- [9] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.
- [10] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [11] J. Andén and S. Mallat, "Multiscale scattering for audio classification," in *ISMIR*. Miami, FL, 2011, pp. 657–662.
- [12] V. Tiwari, "Mfcc and its applications in speaker recognition," *International journal on emerging technologies*, vol. 1, no. 1, pp. 19–22, 2010.
- [13] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.
- [14] S. K. Yadav, K. Tyagi, B. Shah, and P. K. Kalra, "Audio signature-based condition monitoring of internal combustion engine using fft and correlation approach," *IEEE Transactions on instrumentation and measurement*, vol. 60, no. 4, pp. 1217–1226, 2010.
- [15] F. Briggs, R. Raich, and X. Z. Fern, "Audio classification of bird species: A statistical manifold approach," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 51–60.
- [16] T. Zhang and C.-C. J. Kuo, "Content-based classification and retrieval of audio," in *Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, vol. 3461. International Society for Optics and Photonics, 1998, pp. 432–444.
- [17] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings. IEEE International Conference on Multimedia and Expo*, vol. 1. IEEE, 2002, pp. 113–116.
- [18] D. O'shaughnessy, *Speech communication: human and machine*. Universities press, 1987.
- [19] I. Part, "Cs 224d: Deep learning for nlp1," 2015.