Efficient Music Genre Classification with Convolutional Neural Networks

Yaoning Wang
1023041120
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—Over recent years, the complexity of music production has gradually decreased, and more and more people are creating music and uploading it to streaming media. The huge amount of music streaming has led people to spend a lot of time searching for specific music. Therefore, the technique of quickly classifying musical genres is of great importance in today's society. In this paper, music genre classification with deep neural networks is explored. A Convolutional Neural Network (CNN) was trained to identify 10 different music genres. Using the GTZAN dataset, 1000 files were split into train data and test data. In the pre-processing, we use Librosa to convert the original audio files into their corresponding Mel spectra. The converted Mel spectrum is then fed into the proposed CNN model for training. Our results show that CNN models are much more efficient than traditional machine learning models. Traditional machine learning baselines correctly classified 65% of the samples on average. This CNN achieved 85% accuracy on test sets.**

***Index Terms*—Music genre classification, Convolutional Neural Networks, Deep Learning, Mel-spectrum**

I. Introduction

Following the emergence of the Internet, a significant number of individuals began uploading music initially stored on vinyl records and compact discs to online streaming platforms [1]. With the proliferation of expansive digital music libraries and the surge in popularity of music streaming services, locating specific genres or particular pieces of music has become a challenging endeavor. Consequently, the need for tools capable of accurately identifying and classifying music has become a pertinent concern for both newcomers and certain musicians. Presently, the majority of music within contemporary streaming platforms is presented with only the title and author information, lacking specific tags. This absence of detailed metadata complicates the task of recognizing hidden tags within songs and classifying them based on genre.

There are two steps are required in music genre classification [2]. The first step is to extract the audio features of the input music, and the second step is to construct a classifier through these features. In this study, we use the Mel spectrum to simulate human perception. Convolutional neural networks have proven their powerful classification capabilities. We try to use convolutional neural networks to construct a classifier and verify its effectiveness by comparing it with traditional machine learning methods.

A. Machine Learning and Neural Networks

Machine learning is a multidisciplinary field that focuses on developing algorithms and models that enable computers to learn patterns, relationships, and representations from data without explicit programming [3]. It encompasses various paradigms, including supervised learning, unsupervised learning, and reinforcement learning, each addressing specific learning scenarios. Supervised learning utilizes a fully labeled data set to build a mathematical model whereas unsupervised learning attempts to extract useful features from an unlabeled data set without any specific target in mind. Correspondingly, Reinforcement learning takes a different approach through the use of a feedback mechanism [4]. In reinforcement learning, the learning is accomplished through a process of being rewarded for correct actions or predictions.

Neural networks consist of layers of interconnected nodes, or neurons, where each connection is associated with a weight that is adjusted during the training process. This adjustment is guided by a process known as backpropagation, where the model iteratively updates its parameters to minimize a predefined loss function. A convolutional neural network (CNN) is a type of neural network that is intended to process multidimensional vectors such as images [5]. A CNN can be used for both binary classification and multiclassification tasks where these classifiers differ only in the number of output classes. A CNN consists of one or more convolutional layers and a fully connected layer at the top, as well as associated weights and pooling layers. This structure enables convolutional neural networks to exploit the two-dimensional structure of the input data. Convolutional neural networks can give better results in image and speech recognition compared to other deep-learning structures.

The convolutional layer is a pivotal component within CNN, serving the critical function of detecting local patterns and extracting hierarchical features from input data [6]. It employs learnable filters, or kernels, to systematically slide across the input, performing convolution operations that capture features like edges or textures. By sharing parameters across the entire input, the layer allows the network to learn spatial hierarchies efficiently. The convolutional operation involves element-wise multiplication of the filter with overlapping regions of the input, producing a feature map. Stride controls the movement

of the filter, while padding mitigates size reduction. Non-linear activation functions, often ReLU, introduce non-linearity, and multiple channels are utilized for colored images. These layers are integral in building deep representations, recognizing complex patterns by combining simpler features.Figure 1 shows the convolutional layer.

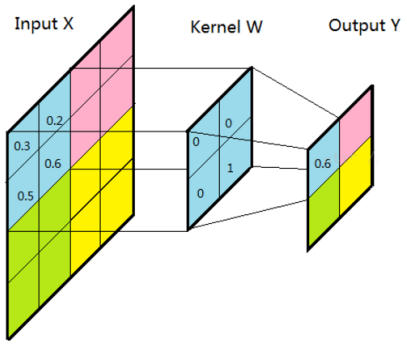

Fig. 1. Visualization of a typical convolutional layer.

The pooling layer is a crucial element in CNN, contributing to the network's ability to downsample and reduce the spatial dimensions of the feature maps obtained from the convolutional layers. Its primary purpose is to retain important information while decreasing the computational complexity of the model. Commonly used pooling techniques include max pooling, which selects the maximum value from a local region, and average pooling, which computes the average [7]. Pooling is typically applied after convolutional layers and introduces a form of translation invariance, making the network more robust to variations in spatial orientation. By reducing the resolution of the feature maps, pooling helps control overfitting and improves computational efficiency. Integrating pooling layers into CNN architectures enhances the network's ability to capture essential features, contributing to its success in tasks such as image recognition and object detection.Figure 2 shows the max pooling layer.

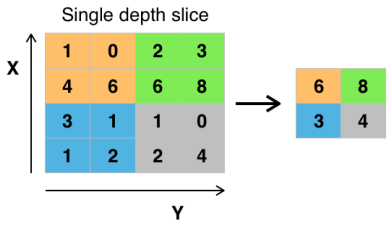

Fig. 2. Visualization of a typical pooling layer.

The fully connected layer is a crucial component in neural network architectures, including CNN. Also known as dense layers, these layers connect every neuron to every neuron in the previous and subsequent layers, forming a fully connected network. In the context of CNNs, fully connected layers are typically employed after the convolutional and pooling layers to perform high-level reasoning and global pattern recognition. The neurons in these layers have learnable weights, and during training, these weights are adjusted to minimize the loss function through backpropagation [8]. Fully connected layers are essential for capturing complex relationships and interactions within the learned features, enabling the network to make predictions based on the global context of the input. While convolutional layers excel at capturing local patterns, fully connected layers contribute to the network's ability to understand the broader context and relationships among features. The integration of fully connected layers into CNN architectures enhances the model's capacity to make accurate predictions in tasks such as image classification and object recognition.

B. The Mel Spectrogram

A signal is a variation in a certain quantity over time. For audio, the quantity that varies is air pressure. How do we capture this information digitally? We can take samples of the air pressure over time. The rate at which we sample the data can vary, but is most commonly 44.1kHz, or 44,100 samples per second. What we have captured is a waveform for the signal, and this can be interpreted, modified, and analyzed with computer software [9].

An audio signal is comprised of several single-frequency sound waves. When taking samples of the signal over time, we only capture the resulting amplitudes. The Fourier transform is a mathematical formula that allows us to decompose a signal into it's individual frequencies and the frequency's amplitude. In other words, it converts the signal from the time domain into the frequency domain. The result is called a spectrum. This is possible because every signal can be decomposed into a set of sine and cosine waves that add up to the original signal [10]. This is a remarkable theorem known as Fourier's theorem. The fast Fourier transform (FFT) is an algorithm that can efficiently compute the Fourier transform. It is widely used in signal processing.Figure 3 shows the waveplot and power spectrum.

he fast Fourier transform is a powerful tool that allows us to analyze the frequency content of a signal, but what if our signal's frequency content varies over time? Such is the case with most audio signals such as music and speech. These signals are known as non periodic signals. We need a way to represent the spectrum of these signals as they vary over time. We can calculate multiple spectra by performing an FFT on multiple windowed segments of the signal [11]. This is called a Short-Time Fourier Transform. The FFT is computed on overlapping windowed segments of the signal, and we get what is called the spectrogram.

Studies have shown that humans do not perceive frequencies on a linear scale. We are better at detecting differences in lower frequencies than higher frequencies. For example, we can easily tell the difference between 500 and 1000 Hz, but we will hardly be able to tell a difference between 10,000 and 10,500 Hz, even though the distance between the two pairs are the same.In 1937, Stevens, Volkmann, and Newmann proposed a unit of pitch such that equal distances in pitch sounded

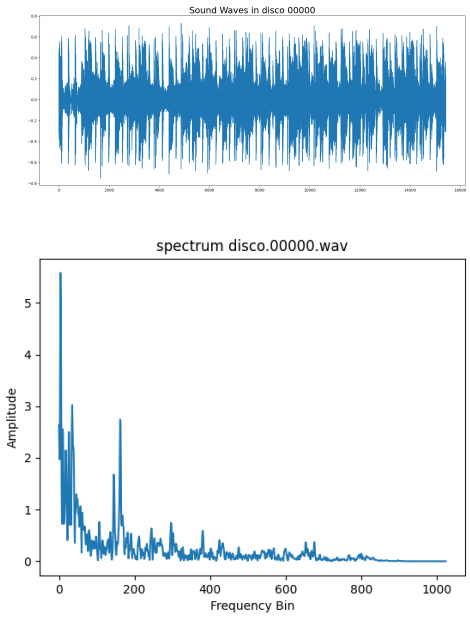

Fig. 3. Waveplot and power spectrum.

equally distant to the listener. This is called the mel scale [12]. We perform a mathematical operation on frequencies to convert them to the mel scale. A mel spectrogram is a spectrogram where the frequencies are converted to the mel scale.

II. Implementation Tools

A. Tensorflow

TensorFlow is an open-source machine learning framework developed by the Google Brain team. It is widely used for building and training various machine learning models, including deep learning models. TensorFlow is designed to facilitate the development and deployment of machine learning models, particularly deep neural networks. And is an open-source project, allowing researchers and developers to access, modify, and contribute to the source code. TensorFlow provides a flexible platform for building and deploying machine learning models across various domains [13]. It supports both CPU and GPU computations, enabling the scaling of models to handle large datasets and complex architectures. TensorFlow has a large and active community, and it offers a rich ecosystem of tools and libraries.

B. Sklearn

Scikit-learn, often abbreviated as sklearn, is a popular open-source machine-learning library for the Python programming language. It provides simple and efficient tools for data analysis and modeling, including various machine-learning algorithms. Scikit-learn is designed to be a user-friendly and accessible machine-learning library, making it easy for both beginners and experts to implement machine-learning algorithms. It has the following advantages, Consistency: A well-designed, consistent API that makes it easy to switch between different algorithms. Extensibility: Easily extendable with new algorithms implemented in Python. Community Support: Large and active community that contributes to the library's development. Scikit-learn is an excellent tool for practitioners and researchers alike, providing a rich set of functionalities for various machine-learning tasks [14]. Its simplicity and consistency make it a go-to library for implementing and experimenting with machine-learning algorithms in Python.

III. Music Classification

A. Datasets

We use GTZAN as our dataset, which was created by Tzanetakis and Cook [15]. The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. We split the dataset into 700 songs in the training set and 300 songs in the test set.table 1 shows the dataset genre distribution.

TABLE I
Dataset Genre Distribution

Genre	Number of Records
blues	100
classical	100
country	100
disco	100
Hip-hop	100
jazz	100
metal	100
pop	100
reggae	100
rock	100
TOTAL	**1000**

B. Preprocessing

This study uses Mel spectrum as input. Aaron et al. [16]have used MFCC to preprocess songs. In this experiment, 30 seconds of audio is sent to the preprocessing stage and converted into their respective Mel spectrums, quantize into audio signal at 660,000 sampling rate per second, perform fast Fourier transform on 1,024 frames, Hop size is set to 256. The preprocessing of this experiment is carried out through Librosa. Its working principle is to perform pre-emphasis, framing, and windowing functions on the original audio, and map the amplitude and frequency of each frame after the fast Fourier transform to the Mel scale. And then merge them according to the fast Fourier frame and the number of Hop size, and finally perform cepstrum analysis to obtain the MFCC.Figure 4 shows the calculation of the mel spectrogram of the data set.Before constructing the model, a few steps have to be taken:

- Values of the mel spectrograms should be scaled so that they are between 0 and 1 for computational efficiency.
- The data is currently 1000 rows of mel spectrograms that are 128 x 660. We need to reshape this to be 1000 rows of 128 x 660 x 1 to represent that there is a single color

channel. If our image had three color channels, RGB, we would need this additional dimension to be 3.

- Target values have to be one-hot-encoded in order to be fed into a neural network.It is important that we complete these steps after creating our holdout set to prevent data leakage.

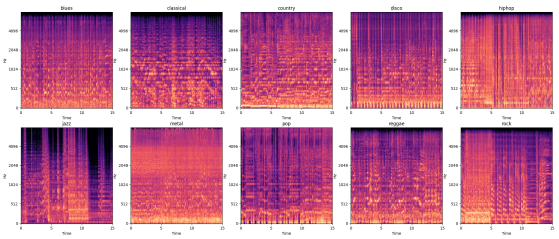

Fig. 4. The mel spectrogram of the data set.

C. Architecture for Convolutional Neural Network

The purpose of using dropout is to prevent overfitting from happening. The overfitting means that the neural network is matches a specific dataset too closely or precisely when learning features. This leads to the very low generalization and recognition accuracy of the neural network. Dropout is currently a technique used in deep learning to reduce overfitting. The dropout means that the neural network randomly disconnects neurons during learning, that is, these disconnected neurons will not participate in the training process during the current training. After iteration random sampling, a subnetwork is constructed from the original neural network, and the structure of the sub-network is also different from the original network structure. Figure 4 is a schematic diagram of dropout.

We add an activation function after each convolutional layer. The activation function we use is Rectified Linear Unit (ReLU). If it is negative than setting ReLU to 0, and output the value if it is positive. ReLU can solve the problem of gradient explosion, and has the characteristics of fast calculation speed and fast convergence speed. Figure 5 and (2) respectively show the function graph of ReLU and the mathematical formula of ReLU.

The CNN model architecture used in this research is composed of 4 convolutional layers. The preprocessed spectrogram is used as input and sent to the CNN model. We set the convolutional kernel as (3, 3), the stride as (1, 1), the activation function as ReLu, the max pooling layer as (2, 2), and the dropout as 0.2. The output of the first convolutional layer is 128, which is compressed to 64 after the max pooling layer and sent to the second layer. The output of the second convolutional layer is 64, which is compressed to 32 after the max pooling layer and sent to the third layer. The output of the third convolutional layer is 32, which is compressed to 16 through the maximum pooling layer and sent to the fourth layer. The output of the fourth layer of convolutional layer is16, and it is compressed to 8 after the maximum pooling layer and sent to the fifth layer. The output of the fifth layer is 8, and it is compressed to 4 after the maximum pooling layer and sent to fully connected layer. After classification by fully connected layer, the result is obtained and the majority vote is taken to obtain the accuracy. Figure 5 shows the proposed CNN model architecture.

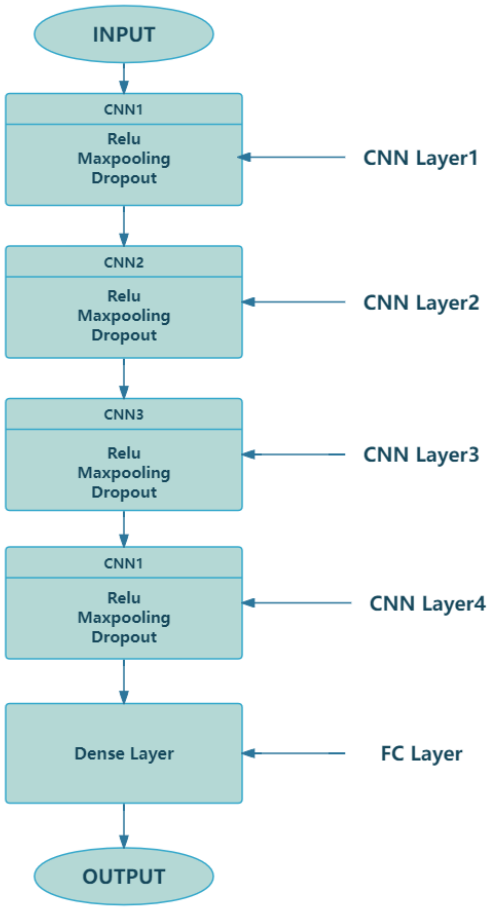

Fig. 5. Proposed CNN model architecture.

IV. EXPERIMENT

A. Experiment Environment

First of all, this research splits the GTZAN dataset into 70% for training set and 30% for test set, and convert all the audio in the dataset into their respective MFCC and send them to the proposed CNN model for training. Librosa is a tool for audio signal processing, we use it for audio conversion in preprocessing to obtain the spectrogram we need. The experiment was performed under a GPU server with Windows11based on NVIDIA GTX 1660Ti, and it is equipped with 16GB of memory for training actions. The total number of iteration performed in the experiment is 2,180, batch size is set to 32, epochs is set to 100.

ADAM is an optimizer for controlling the learning rate, it can iteratively update neural network weights and optimize the

objective function based on training data. Therefore, we also used ADAM into our architecture.

AUC-ROC is a coordinate schema analysis tool, and it usually used as a scoring standard for audio classification, as shown in (1). Among them, true positive (TP) means something is detected, and it exist indeed. False positive (FP) means something is detected, but it does not exist actually. True negative (TN) means something is not detected, and it also does not exist in fact. False negative (FN) means that something is not detected, but it exist actually.

$$ROC = \frac{TP/(TP+FN)}{FP/(FP+TN)} \tag{1}$$

AUC-ROC is mostly applied to unbalanced datasets. The GTZAN used in this experiment is a balanced dataset, so we change to adopt majority voting as our scoring indicator, as shown in (2). Among them, T indicates that there are Tclassifiers, and N indicates that there are N categories. If the prediction results of T classifiers for category j are greater than half of the total voting results, the prediction is category j, otherwise the prediction is refuse.

$$H(x) = \begin{cases} C_j, \sum_{i=1}^{T} h_i j(x) > \frac{1}{2} \sum_{k=1}^{N} \sum_{i}^{T} h_i k(x) \\ \text{Refuse ; Other} \end{cases} \tag{2}$$

B. Experimental Results

The architecture proposed in this study has an accuracy rate of 75% based on the confusion matrix, and the accuracy rate of majority voting is 85.30%.Table II shows the results for our method compares with traditional machine learning method. Figure 6 respectively show the accuracy curve and loss curve of the model. Figure 7 shows the confusion matrix of the CNN model. In the confusion matrix,The model also had a tough time distinguishing between reggae and hiphop. Half of the misclassifications for reggae were hiphop and vise versa. Again, this makes sense since reggae music heavily influenced hiphop music and share similar characteristics.The model misclassified several genres as rock, particularly blues and country. It's no wonder though because there are so many sub-genres of rock music that branch into other genres. Blues rock is very popular as well as southern rock which has country influences.The model hardly ever predicted blues, and only correctly classified 35% of blues songs, but a majority of the misclassifications were jazz and rock. This makes a lot of sense! Jazz and blues are very similar styles of music, and rock music was heavily influenced by, and really came out of, blues music.

V. CONCLUSION

The categorization of music genres plays a pivotal role in assisting users, especially newcomers and specific musicians, in discovering music tailored to their preferences. Beginners, unfamiliar with diverse musical styles, often struggle to efficiently locate specific genres within streaming platforms, leading to time-consuming searches. Similarly, for musicians seeking music within a desired genre, prolonged listening sessions can result in auditory fatigue and hinder accurate genre judgment. Consequently, a dedicated music genre classification tool proves to be a time-saving solution for these individuals. Our proposed convolutional neural network demonstrates an impressive accuracy of 85.30% in music genre classification, paving the way for future advancements in this domain. To enhance the model's precision further, we plan to integrate streaming media and web crawlers, fortifying our CNN architecture for a more comprehensive approach. This endeavor aims to streamline the music discovery process, reducing the time and effort for both music novices and specific musicians, ultimately boosting efficiency.

TABLE II
COMPARISON OF RESULTS AMONG DIFFERENT METHODS

Method	Accuracy
Our Method	85.30%
KNN	68.70%
Decission trees	62.73%
Random Forest	65.70%
Support Vector Machine	61.28%
Logistic Regression	68.31%

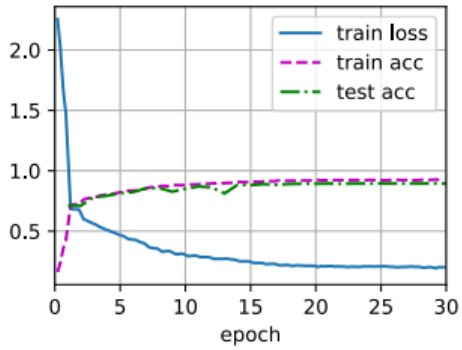

Fig. 6. The accuracy curve and loss curve of the model.

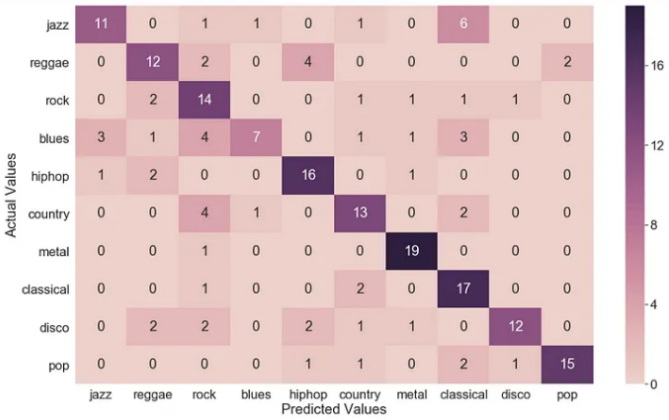

Fig. 7. Confusion matrix of the CNN model .

REFERENCES

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE transactions on acoustics, speech, and signal processing, vol. 28, no. 4, pp. 357-366, 1980.

[2] 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.

[3] E. Zheng, M. Moh, and T.-S. Moh, "Music genre classification: A ngram based musicological approach," in 2017 IEEE 7th International Advance Computing Conference (IACC), 2017, pp. 671-677: IEEE.

[4] A. Elbir, H. B. Çam, M. E. Iyican, B. Öztürk, and N. Aydin, "Music Genre Classification and Recommendation by Using Machine Learning Techniques," in 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), 2018, pp. 1-5: IEEE.

[5] R. Rajan and H. A. Murthy, "Music genre classification by fusion of modified group delay and melodic features," in 2017 Twenty-third National Conference on Communications (NCC), 2017, pp. 1-6: IEEE.

[6] T. Kobayashi, A. Kubota, and Y. Suzuki, "Audio feature extraction based on sub-band signal correlations for music genre classification," in 2018 IEEE International Symposium on Multimedia (ISM), 2018, pp. 180-181: IEEE.

[7] M. Roopaei, P. Rad, and M. Jamshidi, "Deep learning control for complex and large scale cloud systems," Intelligent Automation & Soft Computing, vol. 23, no. 3, pp. 389-391, 2017.

[8] T. Li, A. B. Chan, and A. H. Chun, "Automatic musical pattern feature extraction using convolutional neural network," Genre, vol. 10, p. 1x1, 2010.

[9] T. Nakashika, C. Garcia, and T. Takiguchi, "Local-feature-map integration using convolutional neural networks for music genreclassification," in Thirteenth Annual Conference of the International Speech Communication Association, 2012.

[10] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2014, pp. 6959-6963: IEEE.

[11] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep contentbased music recommendation,"in Advances in neural information processing systems, 2013, pp. 2643-2651.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning.(Report)," Nature, vol. 521, no. 7553, p. 436, May 2015, 2015.

[14] S. Gollapudi, Practial Machine Learning. Birmingham, U.K.: Packt, 2016.

[15] T. Mikolov et al., "Recurrent neural network based language model," in Proc. INTERSPEECH, vol. 2, 2010, p. 4.

[16] Y. M. G. Costa et al., "Music genre recognition using spectrograms," in Proc. 18th Int. Conf. Syst., Signals Image Process., 2011, pp. 1–4.