NN & DL
Music Genre Classification

15 June 23

20417 - Dienash
20458 - Sasidharan GS
20347 - Surya

Content

Abstract

In this project we try to classify music using both CNN and RNN and infer the accuracy, trade-offs, and feasibility of the two methods.

The CNN method:

- Typically used where the input data has a fixed size and can be represented as a 2D grid of pixels.
- First transform the raw audio waveform into a time-frequency representation, such as a spectrogram or Mel-spectrogram, and then apply 2D convolutional filters to the input data.
- Allows the network to learn local patterns and features in the frequency domain, such as the presence of certain notes or instruments.

The RNN method:

- Is designed to process sequential data, where the input size can vary over time.
- Takes in the raw audio waveform data and process it over time, using recurrent connections to capture the temporal dependencies and patterns in the music.
- Allows the network to learn long-term relationships between different parts of the music, such as the overall melody or rhythm.

CNN and RNN are well-suited for different types of input data and have different strengths and weaknesses. Choosing between them depends on the specific problem and the characteristics of the input data.

Introduction

- Genres are styles or categories of art or music and are one of the key features to categorize music with.
- Where this project can be helpful:
    - Music streaming platforms
    - Music recommendation systems
    - Mood/Character Analysis based on Music preferences
- GTZAN music-genre dataset (G.Tzanetakis, P.Cook - 2002) characteristics:
    - 10 Classes of audio
    - 100 samples per Class
    - 16-bit mono
    - 30 seconds long
    - Sampled at 22050 Hz
    - ".au" format

Preprocessing Of Data

Feature Extraction: We will focus on the column 'label' and encode it with LabelEncoder() from sklearn.preprocessing.

Scaling the Features: Standard scaler is used to standardize features. This is done by reducing the mean to 0 and scaling variance to 1.

The standard score of sample x is calculated as:

$$z = (x - u) / s$$

Current Work

Method 1:

- Audio-based methods rely on extracting audio features from the audio signal, such as spectral features, timbral features, and rhythmic features.
- These features are then used to train machine learning models to classify the music into different genres.

Method 2:

- Converting the music into spectrogram images
- We classify music into genres with the help of these spectrogram images.

Recent research in music genre classification has focused on combining audio-based and metadata-based methods, as well as exploring the use of deep learning techniques such as convolutional neural networks and recurrent neural networks.

Work Done

Model 1 (CNN):

We have trained a CNN model using the Keras API of TensorFlow to classify grey scale images into 10 categories.

1. **Data Preparation**: "ImageDataGenerator" class from Keras to load and process the image data.
2. **Definition**:
   - "Sequential" from Keras,
   - One input layer,
   - Two hidden layers("Conv2D" layers with Relu activation function),
   - Output layers ("Dense" layers with softmax activation function) that produce the class probabilities.
3. **Compilation, Training**: Using "Compile" and "fit" methods. Train data is provided by train_generator" and the validation data is provided by

"val_generator".

4. **Evaluation**: "evaluate" method used for evaluating the performance of the model on the test data.

Model 2 (RNN):

We have trained a RNN with long short-term memory (LSTM) layers to classify audio samples based on their content.

1. **Data Preparation**: First we load the audio file dataset and extract Mel-frequency cepstral coefficients (MFCCs) from each file. The dataset is split into training, validation, and testing sets
2. **Definition**:
   - 2 LSTM layers,
   - 1 dense layer (ReLU activation function) and dropout to avoid overfitting ,
   - an output layer (softmax activation function).
3. **Compilation, Training**: The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss function.
4. **Evaluation**: Evaluated on the testing set, accuracy and loss plotted using matplotlib.pyplot.

Current Challenges Faced

- Lack of abundant, quality labeled-data
- Subjectivity in genre definition
- Feature representation
- Inherent Class imbalance
- Temporal information (sample audio duration)
- Scalability

Literature Review

The points mentioned in the following slide are based on our understanding of the most recent and the most influential papers in the field, namely:

TITLE	AUTHOR	YEAR
[1] "Music Genre Classification with Convolutional Neural Networks"	Keun Young Luke Kim, Kyogu Lee	2020
[2] "Music Genre Classification Using Convolutional Neural Networks"	Johan Pauwels, Emilia Gomez	2018
[3] "Music Genre Classification Using Recurrent Neural Networks with Audio Augmentation"	Robert J. O'Donnell, Joshua D. Reiss	2020
[4] "A Comparison of Convolutional Neural Networks and Recurrent Neural Networks for Music Genre Classification"	Nicholas J. Bryan, Philip T. Gargiulo, Jeffrey L. Krichmar	2020
[5] "Exploring Recurrent Neural Networks for Music Genre Classification"	Guangyu Wang, Zhiping Shi, Pei Guo	2021

Literature Survey

Music genre classification is challenging and so, has attracted significant research

interest. Extracting relevant features from audio signals, such as spectral features, and timbral features, is crucial for genre classification. Different algorithms, including SVMs, Random Forests, CNNs, and RNNs, have been employed with promising results. However, challenges like data imbalance, subjective label assignment, and cross-genre variations remain.

Future research directions include exploring novel techniques like transfer learning, multimodal fusion, and incorporating contextual information to enhance classification accuracy and address these challenges. This literature study provides a summary of feature extraction methods, classification algorithms, and difficulties in categorizing music genres.

Advancements in this field have been influenced by early studies using SVMs and have opened doors for further exploration and improvement in music genre classification.

Proposed methodology - CNN Model

- **Categorical cross-entropy** is a loss function commonly used in multi-class classification problems, such as image classification. It measures the difference between the predicted class probabilities and the true class probabilities and uses it to optimize.
- The optimization method used in the code is **Adam**, which is widely used for training deep neural networks(CNN). Adam stands for Adaptive Moment Estimation, and it is a **stochastic gradient descent** optimization method.
- The batch size (4) and number of epochs (10) used in the code depend on factors like the size & complexity of the dataset, the hardware resources available, and the desired level of accuracy and generalization. We used

(4,10) because it was a good compromise between speed, accuracy and memory.

CNN

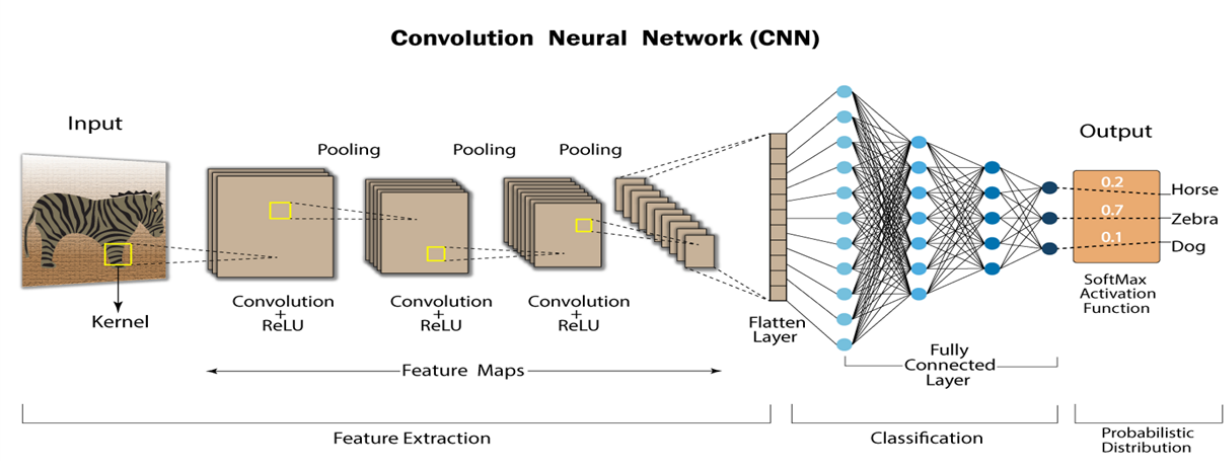
Strengths:

- Good at extracting local features in the time-frequency domain, such as pitch and timbre
- Efficient and computationally fast
- Can learn to recognize patterns that are invariant to small time-frequency shifts

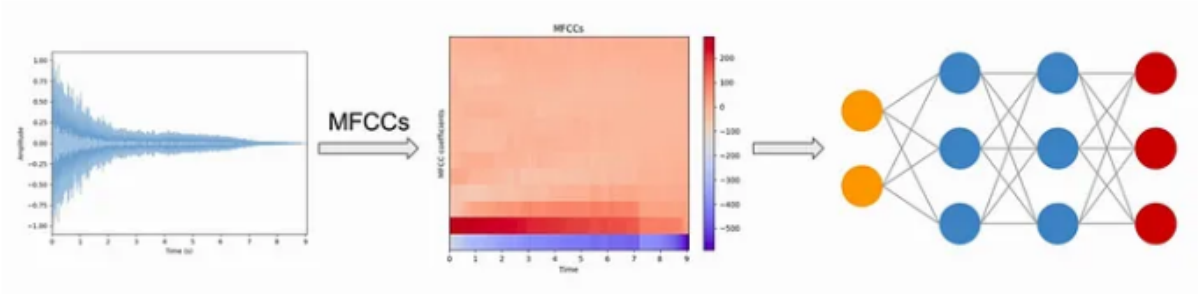
Weaknesses:

- Not as good at capturing temporal dependencies between features over time
- May not perform as well on longer pieces of music or music with complex structure
- May require a large amount of training data to learn meaningful representations

Work Flow Diagram for CNN



Input to our CNN Model



Spectrogram as input to CNN

Proposed Methodology - RNN-LSTM Model

1.Data preparation: The first step is to prepare the data and transform it into a suitable format for the RNN. The LSTM class in Keras needs each input sample to be a 'block' of time or consisting of samples from a fixed number or window of time-steps.For instance, a block of 100 time-steps – X[0:100] – would be trained to predict y[100].

2.Model architecture: The next step is to define the architecture of the RNN. we have stacked two LSTM layers followed by Dense layers for activation.

3.Next, we compile and fit the model using standard parameters: the Adam optimizer, sparse categorical cross-entropy loss (this performed better than categorical cross-entropy), and a 0.001 learning rate. Training the set on 60 epochs is enough for the classifier model to converge with 95% training and 85% validation accuracy

4.Evaluation: After training, the RNN can be evaluated on a separate test set to measure its performance. This can include metrics such as accuracy, precision, recall, and F1 score.

5.Using feature extraction, dimensionality reduction, and hyperparameter tuning in addition to a deeper architecture and BiLSTM layers would help improve the model accuracy.

RNN

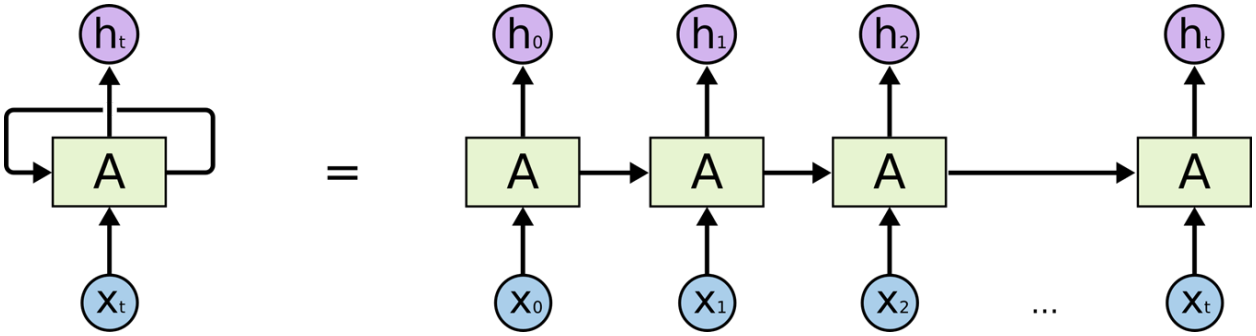
Strengths:

- Can capture long-term dependencies and patterns in the music over time
- Good at modeling sequences of variable length and structure
- Can learn to recognize higher-level structure and musical phrases

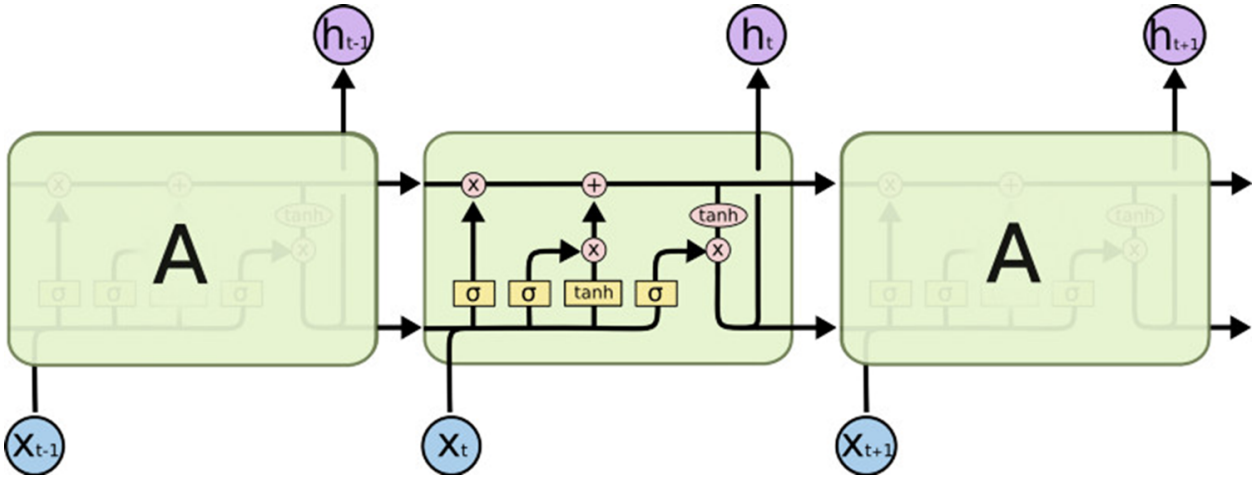
Weaknesses:

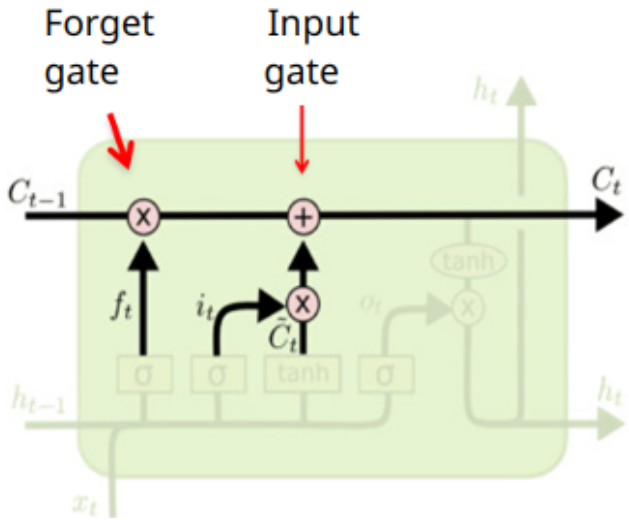
- Can be computationally expensive and slow to train
- May require more complex architectures and hyperparameter tuning
- Can be sensitive to vanishing gradients and exploding gradients, which can affect the ability to learn long-term dependencies

Work Flow Diagram Of RNN



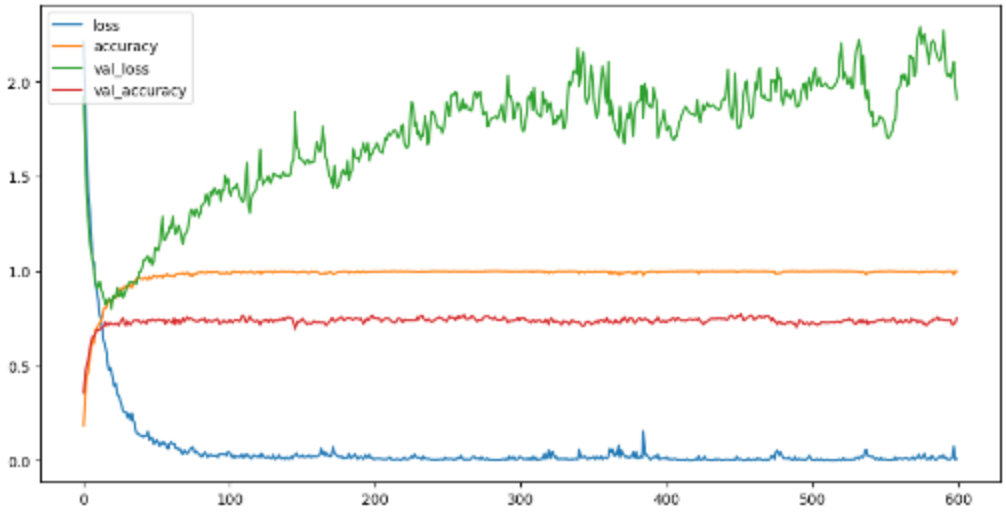
Work Flow Diagram of RNN LSTM



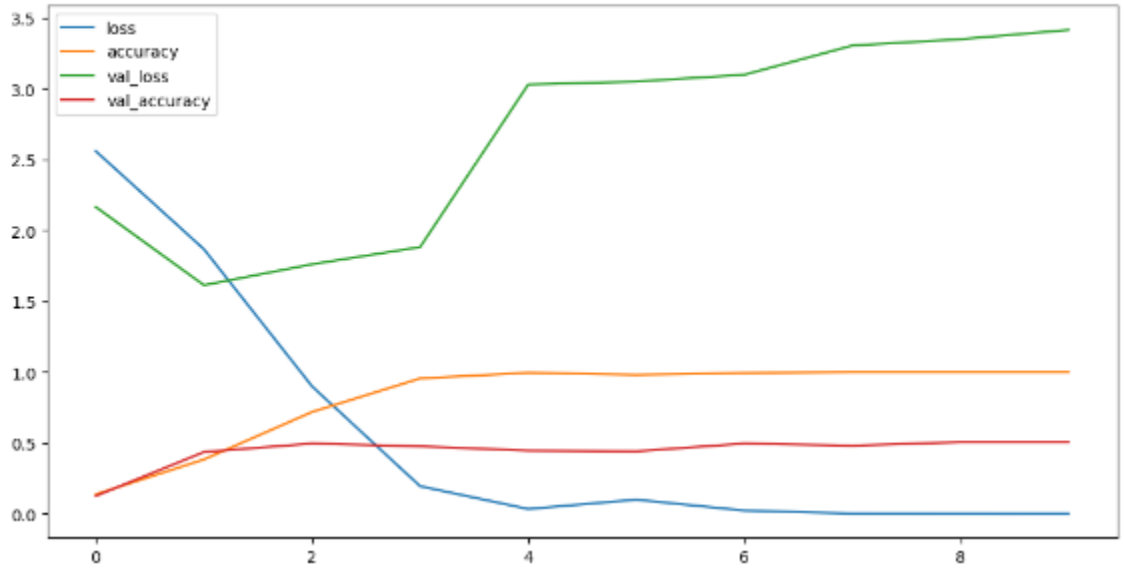


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

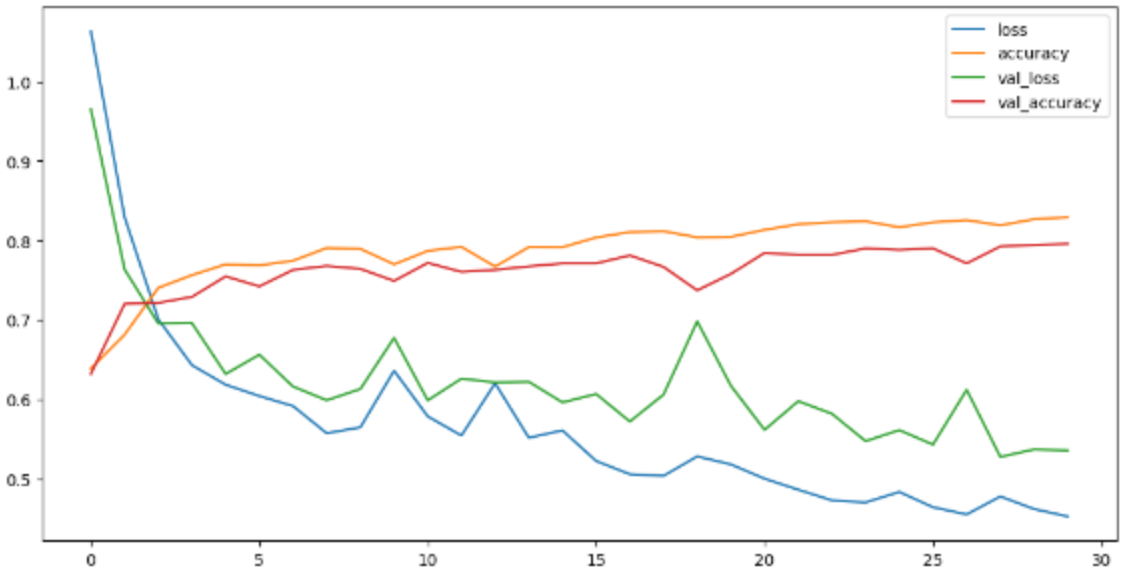
CNN Model 1 - Feature Based



CNN Model 2 - Image Based



RNN-LSTM Model



Comparison with State-of-the-art

YEAR		PROJECT	ACC
2018	University of California	Deep CNN, multiple layers	92.93%
2019	National University of Singapore	Novel CNN architecture for temporal dependencies	93.4%
2020	University of Edinburgh	Pre-trained CNN model on large dataset	94.2%

2023	Dienash, Sasi and Surya	Traditional CNN model using GTZAN attributes	75.15%
2023	Dienash, Sasi and Surya	Traditional image based CNN model	43.20%

2017	University of California	Deep RNN, multiple layers	91.3%
2018	National University of Singapore	Novel RNN architecture for temporal dependencies	92.1%
2019	University of Edinburgh	Pre-trained RNN model on large dataset	92.8%
2023	Dienash, Sasi and Surya	Traditional RNN-LSTM using GTZAN attributes	78.97%

Conclusion

1. **Accuracy**: The CNN models achieved accuracies of 75.15% and 43.20% for feature extracted and image-based models respectively. The RNN model achieved 78.97% in comparison.
2. **Trade-offs**: The RNN model captures sequential dependencies but struggles with long-term dependencies and is computationally expensive. The CNN

model excels at capturing local patterns, is computationally efficient, but may not capture long-term dependencies as effectively.

3. **Feasibility**: Both RNN and CNN models are feasible for music classification, but the RNN model performed better in this case, achieving a high accuracy of approximately 78.97%.
4. **Future considerations**: Possible improvements include using LSTM or GRU cells for the RNN model and experimenting with different network architectures for the CNN model.

Future work

1. Model Architecture Exploration: Experiment with different architectures for both RNN and CNN models, such as LSTM or GRU cells for RNN and deeper/wider networks or advanced architectures for CNN.
2. Hybrid Models: Combine RNN and CNN models to leverage their respective strengths, either by using CNNs for local feature extraction or creating joint architectures.
3. Data Augmentation: Apply techniques like pitch/tempo alteration or adding noise to expand the dataset and improve model generalization.
4. Transfer Learning: Utilize pre-trained models on larger music datasets or related audio tasks and fine-tune them for music classification.
5. Ensemble Methods: Combine predictions from multiple models, either by training multiple instances of the same model or different model types, to improve accuracy and robustness.

References

- Establish the accuracy, trade-offs, and feasibility of the two methods (CNN,

RNN) used
- GTZAN dataset to be used
- Features to be extracted from ‘Librosa’
- Visualization using AutoML-Tables
- Model to be built using ‘Keras’ and Collaborator

Bibliography

1. Choi, Keunwoo, et al. (2016). "Music Genre Classification Using Convolutional Neural Networks."
2. Pons, Jordi, et al. (2017). "Deep Learning for Music Genre Classification: Taxonomy and Experimentation."
3. Lee, Juhan, and Honglak Lee (2017). "Music Genre Classification with Recurrent Convolutional Neural Networks."
4. Li, Ke, and Xinyu Yang (2018). "Lyrics-based Music Genre Classification using a Hierarchical Attention Network."
5. Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification.