# FOR THE BETTER, CRNN IN MUSIC GENRE RECOGNITION PROBLEM

ESRAT MARIA, BYUNGUK MIN, SEUNGHYEON OH, JAMIL SAFDAR

# WHAT IS GENRE CLASSIFICATION?

Classical = 89 %
Jazz = 7 %

Country = 85 %
Blues = 11 %


Disco = 85 %
Classical = 10 %


Rock = 85 %
Metal = 9 %

# OUR APPROACH

**How to make a model that can predict music's genre by merely listening to it?**

- Mel-Spectogram: We can not train a model with raw audio signal data. We should convert it to image data.
- CNN: We use CNN to train visualized signal's information.
- RNN: Lastly, we use a layer of RNN (LSTM, long short time memory) so that the model can detect dependencies across play time of the music.

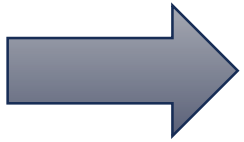
# FRAMEWORK

- Audio library processor:  Librosa
- Deep learning framework:  Keras
- Music datasets:
  - GTZAN dataset

# DATASET AND PRE-PROCESSING

- GTZAN Music Genre Dataset
  - Collection of 1000 songs
  - 10 genres

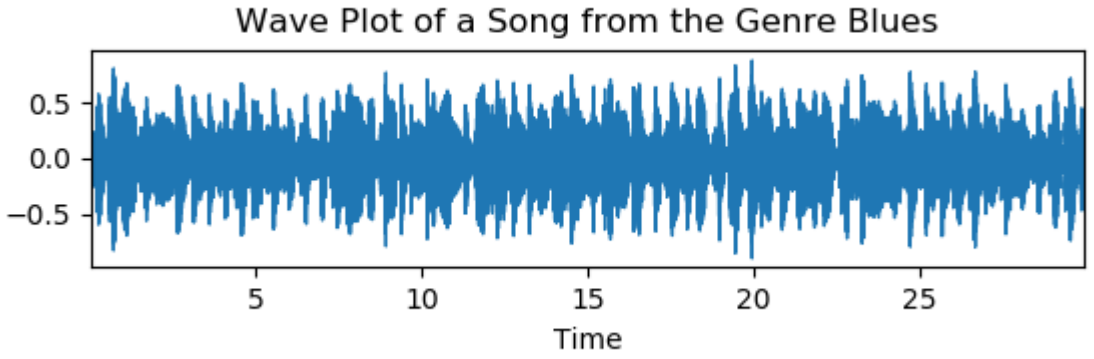
| Class label | Class description | Mean duration (s) | Number of samples |
|---|---|---|---|
| 1 | Blues | 30 | 100 |
| 2 | Classical | 30 | 100 |
| 3 | Country | 30 | 100 |
| 4 | Disco | 30 | 100 |
| 5 | Hip Hop | 30 | 100 |
| 6 | Jazz | 30 | 100 |
| 7 | Metal | 30 | 100 |
| 8 | Pop | 30 | 100 |
| 9 | Reggae | 30 | 100 |
| 10 | Rock | 30 | 100 |



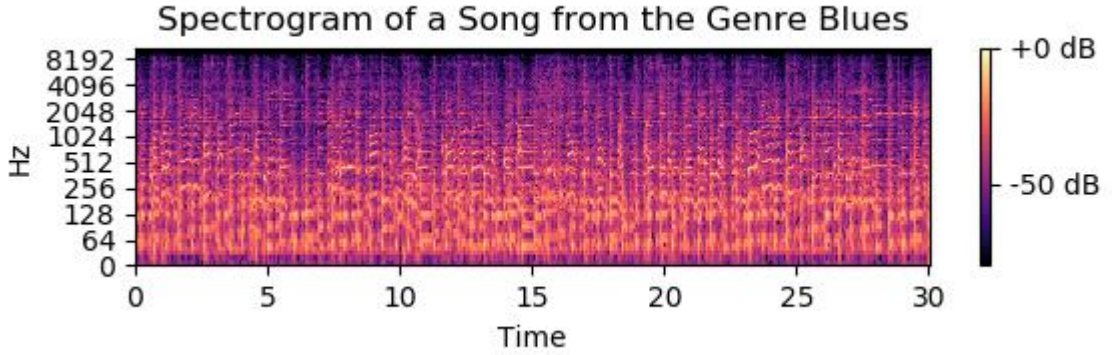
| Feature | Feature Type |
|---|---|
| OMSC | Long-term |
| Low-Energy | Long-term |
| MSCM | Long-term |
| MSFM | Long-term |
| OSC | Short-term |
| MFCC | Short-term |
| Spectral Centroid | Short-term |
| Spectral Rolloff | Short-term |
| Spectral Flux | Short-term |
| Zero Crossings | Short-term |

# SPECTROGRAMS

```
filename = 'C:/Users/Esrat Maria/Desktop/genres/blues/blues.00000.wav'
y, sr = librosa.load(filename)
plt.figure()
plt.subplot(3, 1, 1)
# trim silent edges
whale_song, _ = librosa.effects.trim(y)
librosa.display.waveplot(whale_song, sr=sr)
plt.title("Waveplot of a Song from the Genre Blues")
plt.show()
```

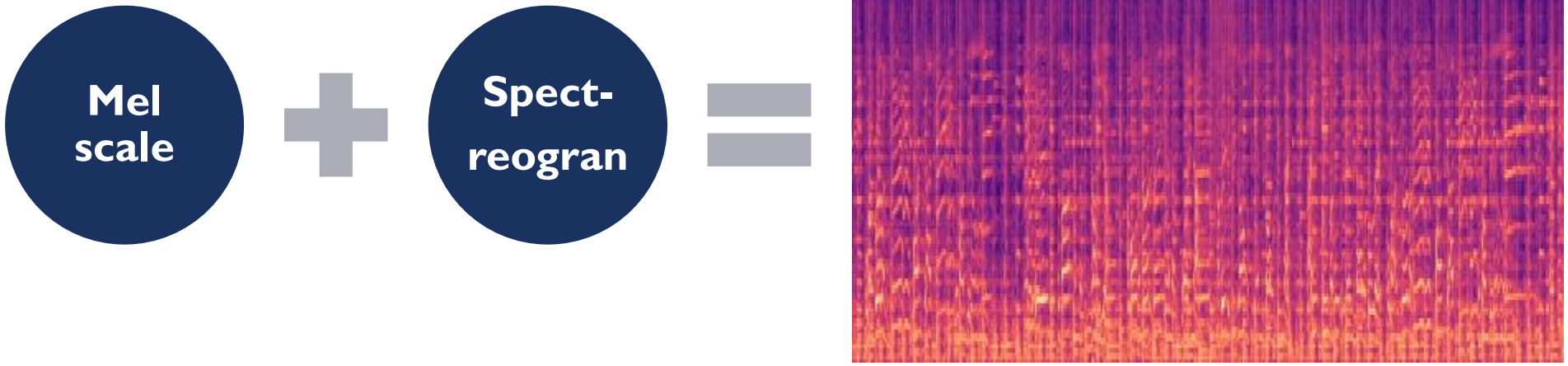



```
hop_length = 512
D = np.abs(librosa.stft(whale_song, n_fft=2048,
                        hop_length=hop_length))
DB = librosa.amplitude_to_db(D, ref=np.max)
librosa.display.specshow(DB, sr=sr, hop_length=hop_length,
                         x_axis='time', y_axis='log')
plt.colorbar(format='%+2.0f dB')
plt.title("Spectrogram of a Song from the Genre Blues")
plt.show()
```



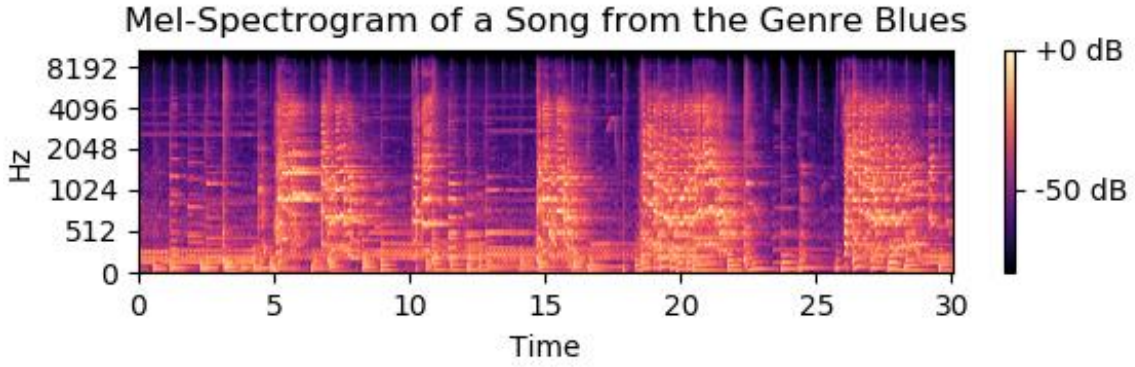
# MEL-SPECTROGRAMS

A spectrogram is a visual representation of the spectrum of frequencies in a sound or other signal as they vary with time or some other variable.



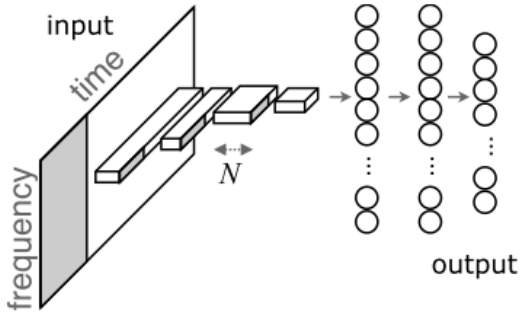
# MEL-SPECTROGRAM

```
S = librosa.feature.melspectrogram(whale_song, sr=sr, n_fft=2048,
                                   hop_length=hop_length,
                                   n_mels=128)
S_DB = librosa.power_to_db(S, ref=np.max)
librosa.display.specshow(S_DB, sr=sr, hop_length=hop_length,
                         x_axis='time', y_axis='mel')
```

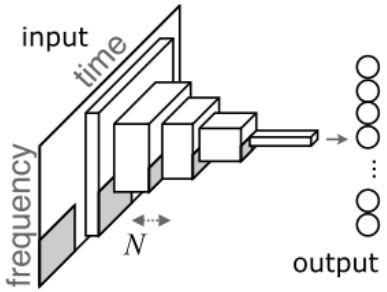


# CNN

- **1-Dimensional Convolution & 2D kernels ? <u>No, 2D Convs & 2D Kernels.</u>**



(b) k2c1



(c) k2c2

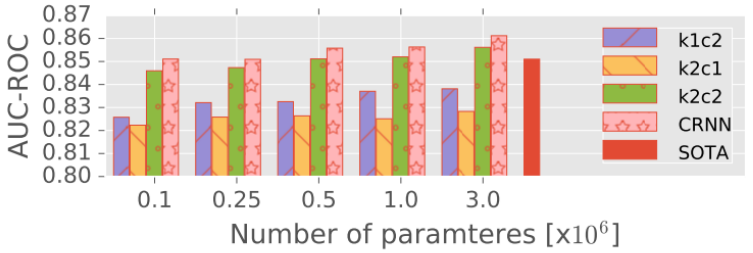
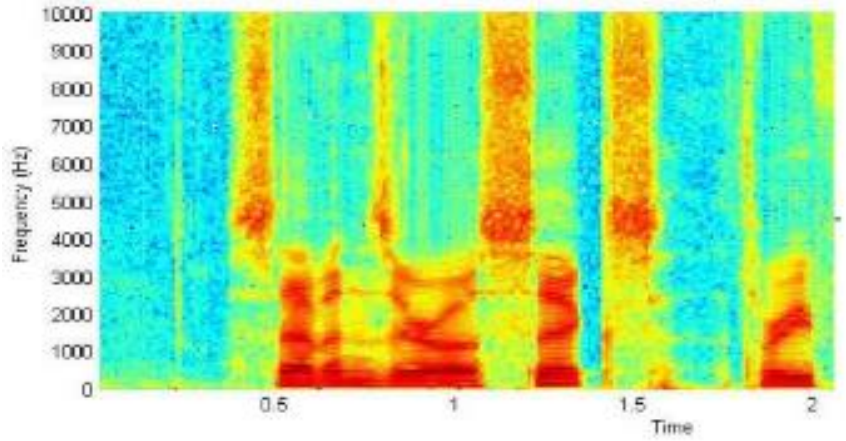


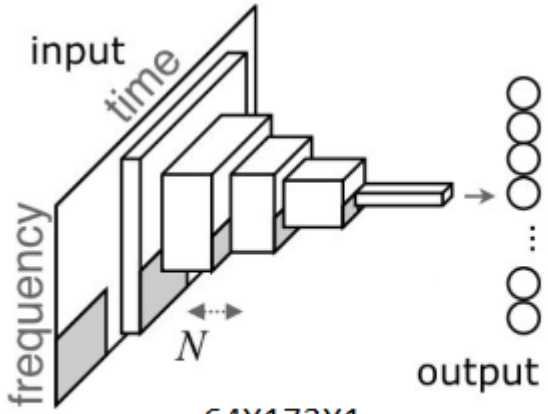

**Fig. 2**: AUCs for the three structures with $\{0.1, 0.25, 0.5, 1.0, 3.0\} \times 10^6$ parameters. The AUC of SOTA is .851 [2].

# CNN-GRU FOR MUSIC CLASSIFICATION

Song Audio File




Mel-Spectrogram



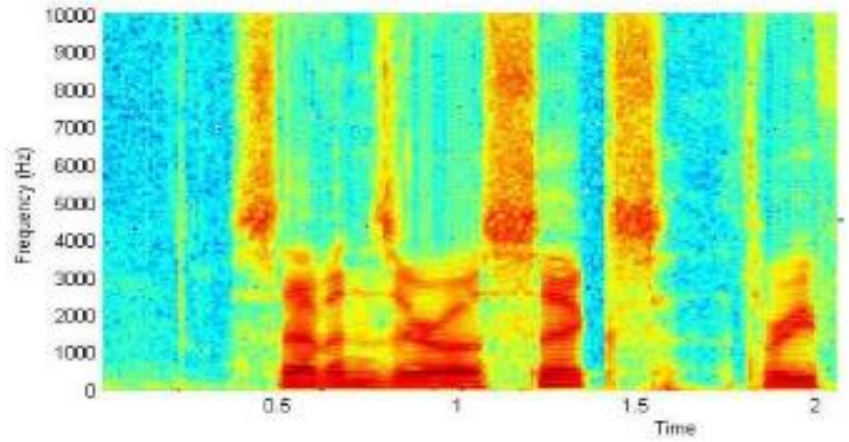

64X173X1

Feature extraction & Classification

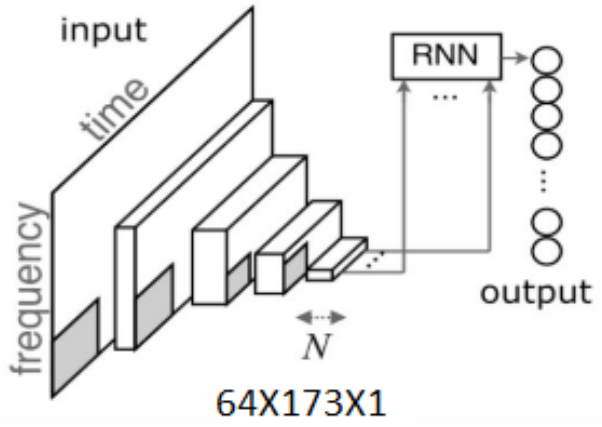
```
model.add(Conv2D(16, kernel_size=(3, 3),
                 data_format="channels_last", input_shape=(64, 173, 1)))
model.add(TimeDistributed(Activation("relu")))
model.add(Conv2D(16, kernel_size=(3, 3)))
model.add(TimeDistributed(Activation("relu")))
model.add(MaxPooling2D(pool_size=2))
model.add(TimeDistributed(Flatten()))
model.add(TimeDistributed(Dense(32)))
model.add(GRU(64))
model.add(Dense(10))
model.add(Activation("softmax"))
```

# CRNN FOR MUSIC CLASSIFICATION

Song Audio File



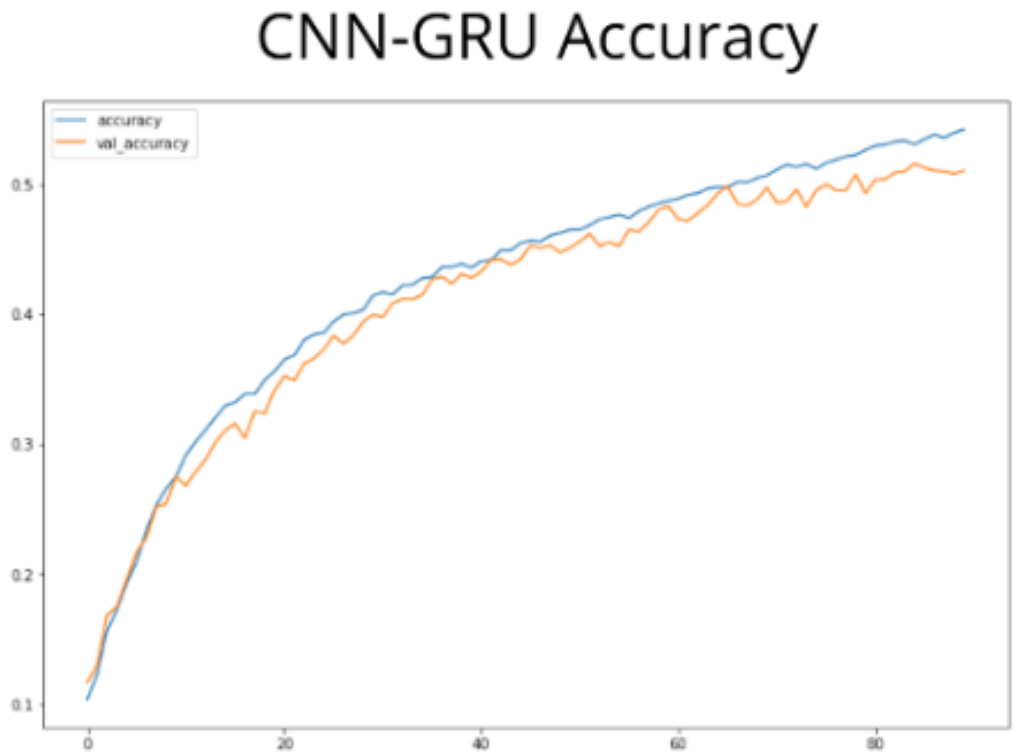

Mel-Spectrogram



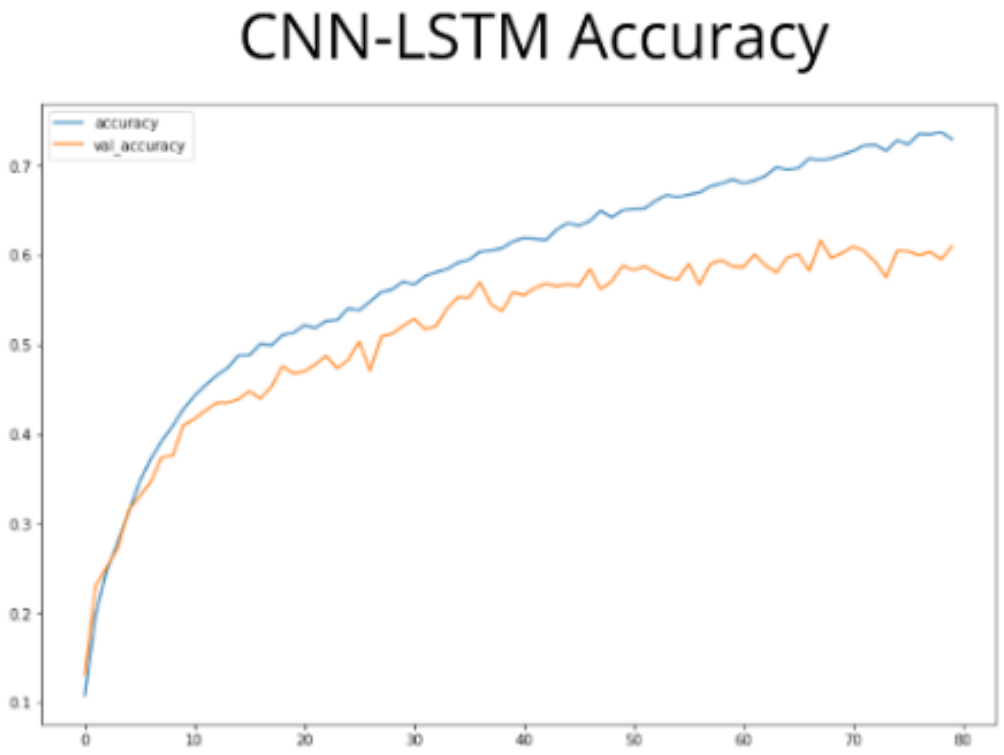

Feature extraction & Classification

```
model.add(Conv2D(20, (5, 5), input_shape=(64, 173, 1),
                 activation="relu", strides=1, padding="valid"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(50, (5, 5), use_bias=50))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(20, activation="relu"))
model.add(Lambda(lambda x: backend.expand_dims(model.output, axis=-1)))
model.add(LSTM(512, activation="relu", return_sequences=False))
model.add(Dense(10, activation = "softmax"))
model.summary()
```

# EXPERIMENT RESULT (CNN-GRU VS CNN-LSTM)




Test accuracy: 50.301%




Test accuracy: 61.007%

# IMPROVEMENTS

- Over fitting problem in the classifier layers.
- Our is not complex enough.
- Final performance on GTZAN around 61% of accuracy.
- Lack of data (initially started with small number).

# REFERENCES


- Recommending music on Spotify with deep learning https://benanne.github.io/2014/08/05/spotify-cnns.html
- K. Choi, G. Fazekas, K. Cho, and M. Sandler, "A tutorial on deep learning for music information retrieval," arXiv preprint arXiv:1709.04396, 2017.
- Music Genre Recognition by Deep Sound http://deepsound.io/music_genre_recognition.html
- Using CNN and RNN for genre recognition by Medium https://towardsdatascience.com/using-cnns-and-rnns-for-music-genre-recognition-2435fb2ed6af
- K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in Proc. Int. Conf. Acoust, Speech, Signal Process., 2017
- " K. Choi, G. Fazekas, and M. Sandler. Explaining deep convolutional neural networks on music classification" - https://arxiv.org/pdf/1607.02444.pdf
- GTZAN dataset - http://marsyas.info/downloads/datasets.html
- Librosa on github - https://github.com/librosa/librosa

# THANK YOU

**QUESTIONS ?**