Spectrogram
CPSC 393
Final Project

Rob Farmer: 2293819
Charles Filce: 2287651
Dan Haub: 2315346

Welcome

ROBERT "ROB" FARMER

CHARLES "CHARLIE" FILCE

DAN "The Man" HAUB

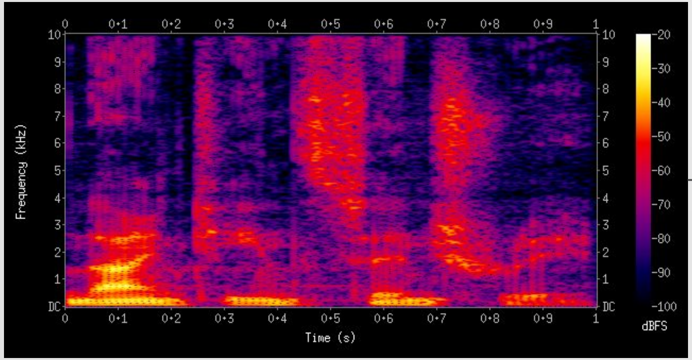
The Problem

Music streaming platforms currently rely on content creators to classify their songs as a specific genre/style and there are currently no protections against the mislabeling of songs.

This has led to an abuse of the system, leading to misclassification of tracks and dissatisfied customers.

The Solution

By feeding the samples through our proprietary Convolutional Neural Network, our team aims to automatically and correctly classify music tracks by their genre/mood.



Our CNN

ROCK
POP
REGGAE
ELECTRO SEA-SHANTY*
PUNK

*Genres may not be accurate to model

The Data

Our team utilized Free Music Archive (FMA) data acquired through Kaggle. The data was adequately sanitized and sorted, packaged in a CSV.

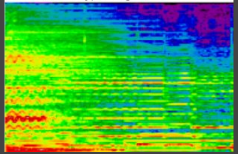
Spectrogram data was created using Short-term Fourier Transformations of 20 second audio samples.

Data was then processed by converting the spectrogram data into a 2d numpy array.

The System

Spectrogram

```
Label                       International
FileName    19252.fused.full.npy.gz
Name: 1436, dtype: object
```



Spectrogram sorted from 0hz-4khz on y-axis, 20s time sample on x-axis

Data Modelling

```
<class 'http.client.HTTPResponse'>
<class '_io.BufferedReader'>
array([[-80.        , -80.        , -80.        , ..., -80.        ,
        -80.        , -71.5089389 ],
       [-80.        , -80.        , -80.        , ..., -80.        ,
        -80.        , -71.45252049],
       [-80.        , -80.        , -80.        , ..., -80.        ,
        -80.        , -71.36522916],
       ...,
       [-80.        , -80.        , -80.        , ..., -13.03568181,
         -8.08395914,  -5.72313642],
       [-80.        , -80.        , -80.        , ...,  -9.43161413,
         -9.95325236, -14.14820794],
       [-80.        , -80.        , -80.        , ..., -11.6952153 ,
        -13.24206394, -15.70028749]])
```

Spectrogram was read into the CoLab as a numpy 2d array

Training

```
[ ] testData = np.array(testDataList)
    testData = testData.reshape(testData.shape + (1,))
    testLabs = np.array(testLabsList)

[ ] del testDataList
    del testLabsList

[ ] print(trainData.shape)
```

Model was trained using a split of training and test data written into CSVs

Max Pooling & Convolution

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 126, 598, 32)      320
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 63, 299, 32)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 61, 297, 64)       18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 30, 148, 64)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 146, 128)      73856
_________________________________________________________________
```

Parameter Evaluation

```
Total params: 67,081,236
Trainable params: 67,081,236
Non-trainable params: 0
```

Parameters were readjusted after determining that 67 million was too many

Parameter Evaluation cont.

```
Total params: 2,236,164
Trainable params: 2,236,164
Non-trainable params: 0
```

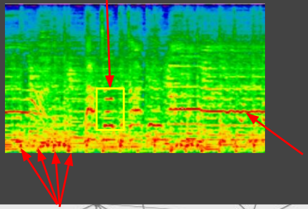
Significantly fewer parameters gave heavier weights to the "neurons"

CNN Evaluation

```
Epoch 1/15
384/384 [==============================] - 414s 1s/step
Epoch 2/15
384/384 [==============================] - 409s 1s/step
Epoch 3/15
236/384 [=================>............] - ETA: 2:29 -
```

CNN uses Google CoLab TPUs to analyze the efficacy of our training

Conclusion



Does this look like a rock song to you?

About The Model

We struggled with overfitting; we remedied this by utilizing dropout in our model and reducing the number of parameters in order to reduce redundancy.

Data did not have metadata, had to be obtained separately.

The Struggles

Not last on the Kaggle leaderboard (#13!)

Greater accuracy than simply guessing randomly

Very accurately guesses the Old-Time genre

The Success(es)

About The Model

```
=================================================================
==
conv2d (Conv2D)               (None, 125, 253, 16)      272
_________________________________________________________________
__
max_pooling2d (MaxPooling2D)  (None, 62, 126, 16)       0
_________________________________________________________________
__
conv2d_1 (Conv2D)             (None, 59, 123, 32)       8224
_________________________________________________________________
__
max_pooling2d_1 (MaxPooling2D) (None, 29, 61, 32)       0
_________________________________________________________________
__
conv2d_2 (Conv2D)             (None, 27, 59, 64)        18496
_________________________________________________________________
__
max_pooling2d_2 (MaxPooling2D) (None, 13, 29, 64)       0
_________________________________________________________________
__
conv2d_3 (Conv2D)             (None, 11, 27, 128)       73856
_________________________________________________________________
__
max_pooling2d_3 (MaxPooling2D) (None, 5, 13, 128)       0
_________________________________________________________________
__
dropout (Dropout)             (None, 5, 13, 128)        0
_________________________________________________________________
__
flatten (Flatten)             (None, 8320)              0
```

Total params: 2,256,164

Results of The Model

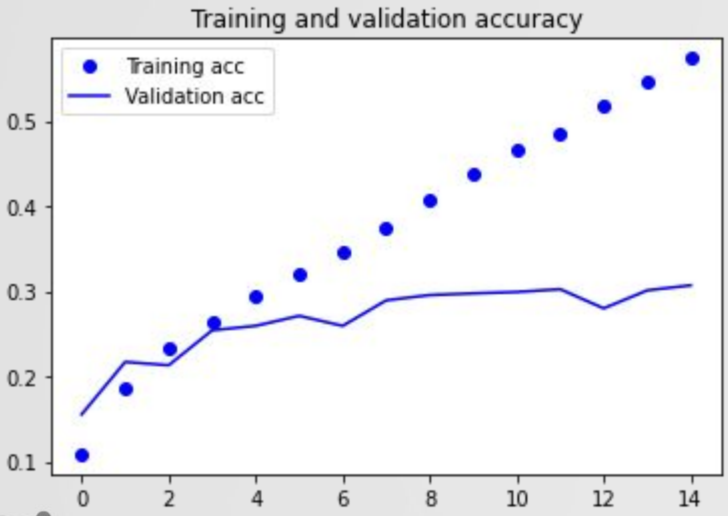
.5841 — Accuracy

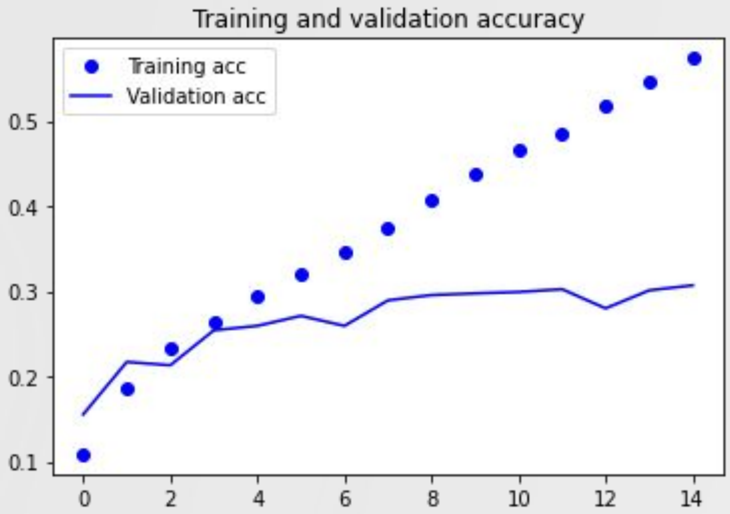
1.362 — Loss

.3069 — Validation Accuracy

2.572 — Validation Loss

Results of The Model

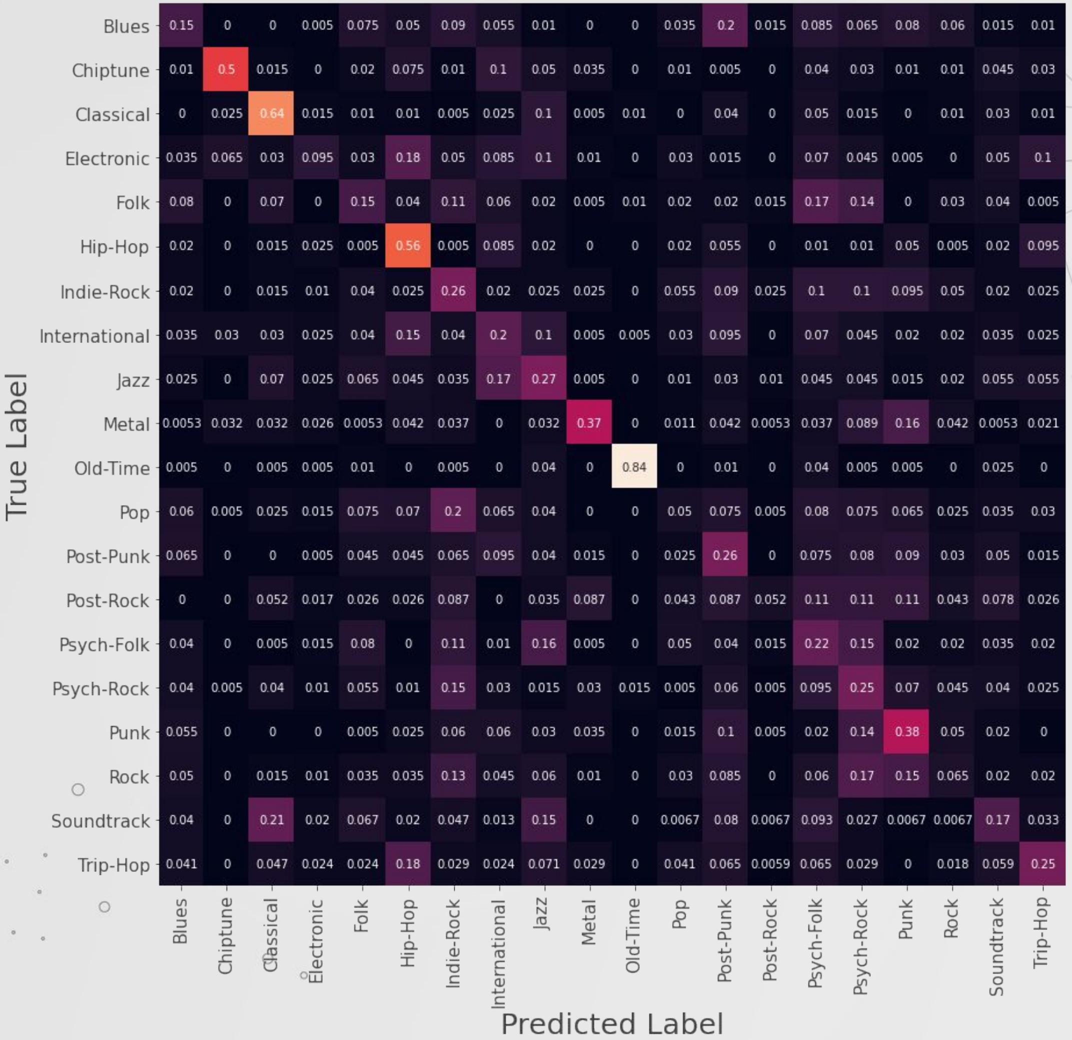




Results of The Model

2.602 — Test Loss

.2944 — Test Accuracy


True Label
Predicted Label
Blues 0.15 0 0 0.005 0.075 0.05 0.09 0.055 0.01 0 0 0.035 0.2 0.015 0.085 0.065 0.08 0.06 0.015 0.01
Chiptune 0.01 0.5 0.015 0 0.02 0.075 0.01 0.1 0.05 0.035 0 0.01 0.005 0 0.04 0.03 0.01 0.01 0.045 0.03
Classical 0 0.025 0.64 0.015 0.01 0.01 0.005 0.025 0.1 0.005 0.01 0 0.04 0 0.05 0.015 0 0.01 0.03 0.01
Electronic 0.035 0.065 0.03 0.095 0.03 0.18 0.05 0.085 0.1 0.01 0 0.03 0.015 0 0.07 0.045 0.005 0 0.05 0.1
Folk 0.08 0 0.07 0 0.15 0.04 0.11 0.06 0.02 0.005 0.01 0.02 0.02 0.015 0.17 0.14 0 0.03 0.04 0.005
Hip-Hop 0.02 0 0.015 0.025 0.005 0.56 0.005 0.085 0.02 0 0 0.02 0.055 0 0.01 0.01 0.05 0.005 0.02 0.095
Indie-Rock 0.02 0 0.015 0.01 0.04 0.025 0.26 0.02 0.025 0.025 0 0.055 0.09 0.025 0.1 0.1 0.095 0.05 0.02 0.025
International 0.035 0.03 0.03 0.025 0.04 0.15 0.04 0.2 0.1 0.005 0.005 0.03 0.095 0 0.07 0.045 0.02 0.02 0.035 0.025
Jazz 0.025 0 0.07 0.025 0.065 0.045 0.035 0.17 0.27 0.005 0 0.01 0.03 0.01 0.045 0.045 0.015 0.02 0.055 0.055
Metal 0.0053 0.032 0.032 0.026 0.0053 0.042 0.037 0 0.032 0.37 0 0.011 0.042 0.0053 0.037 0.089 0.16 0.042 0.0053 0.021
Old-Time 0.005 0 0.005 0.005 0.01 0 0.005 0 0.04 0 0.84 0 0.01 0 0.04 0.005 0.005 0 0.025 0
Pop 0.06 0.005 0.025 0.015 0.075 0.07 0.2 0.065 0.04 0 0 0.05 0.075 0.005 0.08 0.075 0.065 0.025 0.035 0.03
Post-Punk 0.065 0 0 0.005 0.045 0.045 0.065 0.095 0.04 0.015 0 0.025 0.26 0 0.075 0.08 0.09 0.03 0.05 0.015
Post-Rock 0 0 0.052 0.017 0.026 0.026 0.087 0 0.035 0.087 0 0.043 0.087 0.052 0.11 0.11 0.11 0.043 0.078 0.026
Psych-Folk 0.04 0 0.005 0.015 0.08 0 0.11 0.01 0.16 0.005 0 0.05 0.04 0.015 0.22 0.15 0.02 0.02 0.035 0.02
Psych-Rock 0.04 0.005 0.04 0.01 0.055 0.01 0.15 0.03 0.015 0.03 0.015 0.005 0.06 0.005 0.095 0.25 0.07 0.045 0.04 0.025
Punk 0.055 0 0 0 0.005 0.025 0.06 0.06 0.03 0.035 0 0.015 0.1 0.005 0.02 0.14 0.38 0.05 0.02 0
Rock 0.05 0 0.015 0.01 0.035 0.035 0.13 0.045 0.06 0.01 0 0.03 0.085 0 0.06 0.17 0.15 0.065 0.02 0.02
Soundtrack 0.04 0 0.21 0.02 0.067 0.02 0.047 0.013 0.15 0 0 0.0067 0.08 0.0067 0.093 0.027 0.0067 0.0067 0.17 0.033
Trip-Hop 0.041 0 0.047 0.024 0.024 0.18 0.029 0.024 0.071 0.029 0 0.041 0.065 0.0059 0.065 0.029 0 0.018 0.059 0.25
Blues Chiptune Classical Electronic Folk Hip-Hop Indie-Rock International Jazz Metal Old-Time Pop Post-Punk Post-Rock Psych-Folk Psych-Rock Punk Rock Soundtrack Trip-Hop

Future Improvements

With ImageNet and pre-trained neural networks, our model could be further refined to produce data with higher accuracy in its scoring.

This implementation of transfer learning would allow the reading of spectrograms as if they were images read by our CNN.

With additional time and resources our team would aim to score higher on the classification leaderboard, though we have achieved 13th place.

Sources

Data:
https://www.kaggle.com/c/multitask-music-classification/data?select=data.tar.gz

References:
https://towardsdatascience.com/understanding-input-and-output-shapes-in-convolution-network-keras-f143923d56ca

Documentation for various python packages

Course Textbook: Deep Learning with Python

Spectrogram