University of Toronto

APS360 Final Report:
J.A.A.Ms Classifier: Music Genre Recognition Model

Jacklyn Becker
Addis Semagn
Ayesha Khan
Mariem Ahmed

Date of Submission: April 9th, 2021

Word Count: 2228 (excluding titles, figures, references), Penalty: 0%

Introduction

With over 50 million songs on streaming platforms such as Spotify, classification is an important functionality for improving music management, filtering, and recommendations. Music genre classification can be subjective to humans, which can achieve an average accuracy of 70% [1]. Through analyzing qualitative characteristics of audio, including frequency and wavelength, music can quantitatively be classified into categories such as pop, hip hop, jazz, etc. We want to build a machine learning model that not only classifies audio files into their music genre but does so at an accuracy comparable to or better than humans.

Illustration

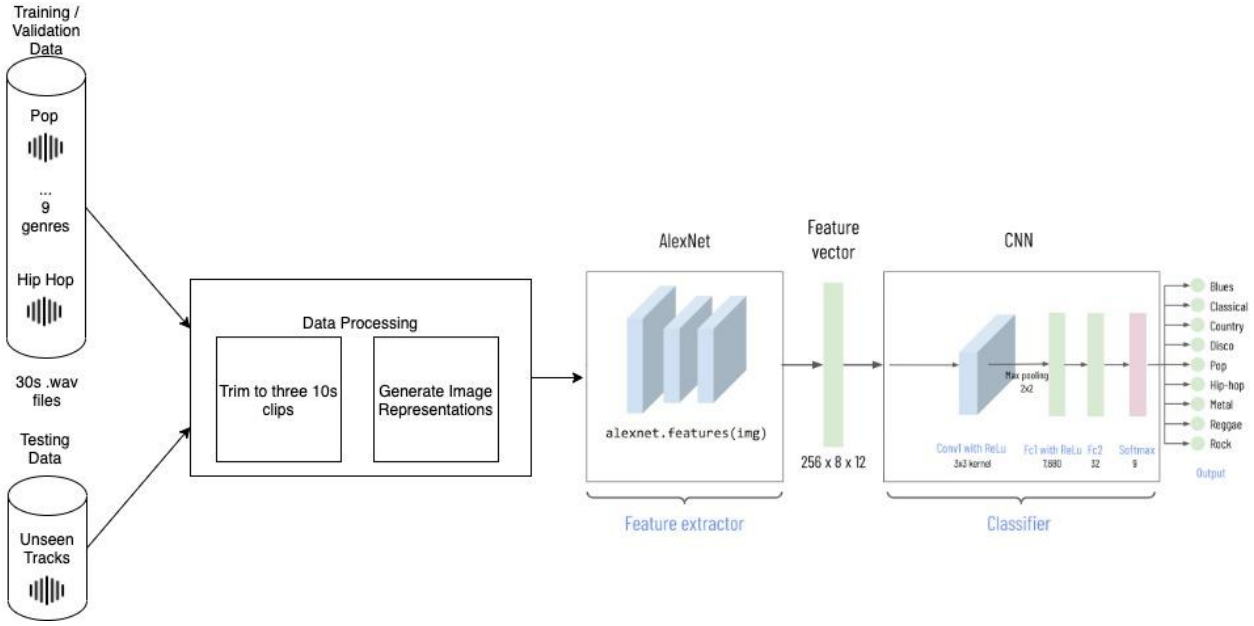


Figure 1: Visual representation of model architecture

Background & Related Work

Machine learning has come far in genre classification. The most popular dataset used in research is GTZAN [2], classified with the manual approach of using features such as rhythm and melody. However, this dataset has been found to have defects, including repetitions and mislabeling [3]. Though research has disproved that all models derived using this dataset are flawed [3], it highlights a key weakness in human-based genre classification: it's subjective and inconsistent.

The ambiguity in classifying genres both motivated and proved to be a suitable use for AI music genre recognition systems. One such system was built in 2002 by Tzanetakis and Cook who achieved 61% accuracy using a mixture of Gaussian models and the k-nearest-neighbours. They transformed the audio into mel-frequency cepstral coefficients (a more compact representation of audio) and analyzed its timbre, rhythm, and pitch features [1]. This laid the foundational work that, over the past two decades, led to

accuracies as high as 90%. One such work was that of Yang Y et al [4]., using parallel convolutional networks to analyze and merge temporal and timbre patterns [5].

These works are the building blocks of models used in popular audio streaming platforms and continue to inspire teams, including ours, to continue to push the boundaries of accuracy.

Data Processing

We primarily used the GTZAN dataset, containing 900 30-second audio files, each labelled as one of nine genres (blues, classical, country, disco, pop, hip-hop, metal, reggae or rock). Our second dataset contained 15 songs, which we manually downloaded directly from YouTube and trimmed to a random 30-sec subsection. These 30 second audio files were 44kHz/s, providing us with over 1.32 million data points per song. Since this was too much data to efficiently train our model, we followed the following steps to process our data:

1. Split each 30-sec audio file into 3 10-second segments, i.e. {0s, 10s}, {10s, 20s}, {20s, 30s}.
2. Convert them into image format. We used three representations: Mel spectrogram, spectral centroid and pitch class visualization. These images were generated using the Python Librosa library [Figure 2].

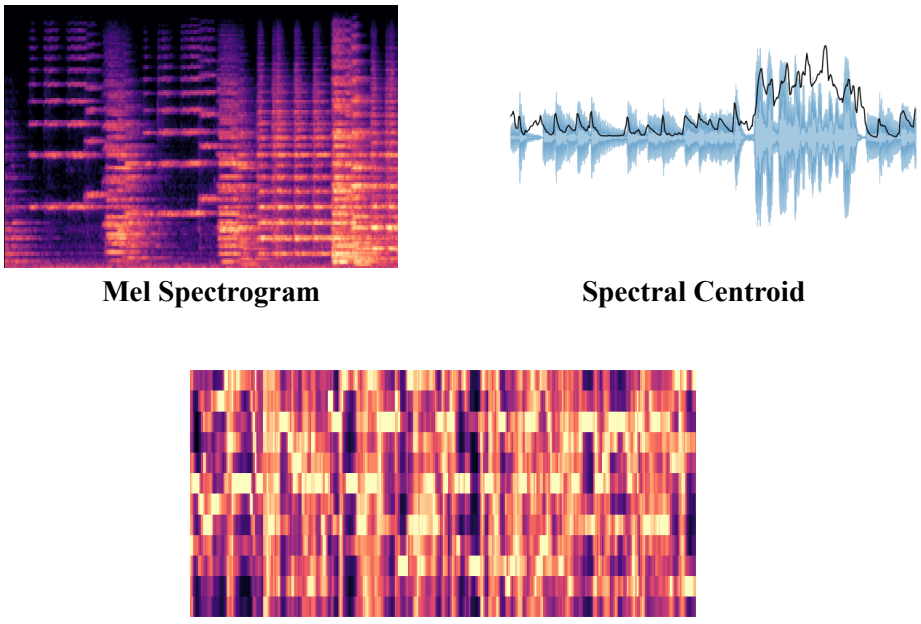


Figure 2: Visual representations of audio files

3. Resize images to 288 x 432 (which may change the aspect ratio): this aids in training the model, by allowing for batching, and reducing the computational time needed for forward passes.

These images were then shuffled and split into 60% training, 20% validation, and 20% test sets.
The distribution of our total dataset is summarized below. [Table 1].

	Training	Validation	Testing	
Audio clips per genre	180 GTZAN	60 GTZAN	60 GTZAN	15 YouTube
Total audio clips	1620	540	675	
Total images (x3 image types)	4860	1620	2025	

Table 1: Summary of data samples in each data set

Architecture

We experimented with two different models: convolutional neural network (CNN) and convolutional neural network using AlexNet for transfer learning (AlexCNN).

CNN

Audio can be processed as a 2D image, which is a great use case for CNN models. By converting our audio input into spectrograms, the input will have the 2D structure of frequency over time, which can be consumed as an image for pattern-finding. Our model consists of two convolution layers with ReLu activation and max pooling applied after each layer. Followed by 2 fully connected layers [Figure 3]. The final classification will use the combined outputs which should increase the pattern-recognition accuracy [5].

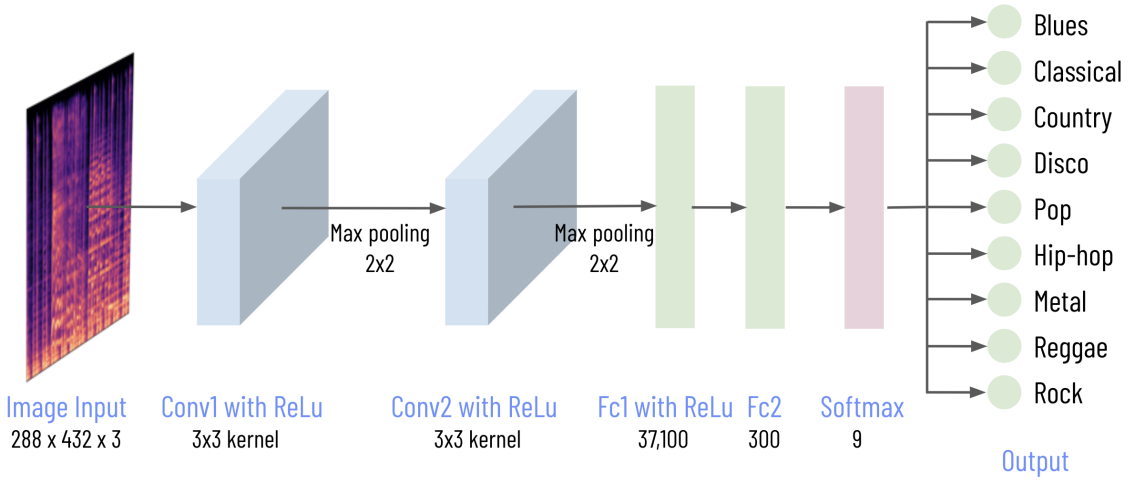


Figure 3: CNN Architecture

AlexCNN

Our second model is a CNN that uses transfer learning by using the pre-trained AlexNet model for feature extraction. Spectrograms are sent into alexnet.features and those features are saved and sent into a convolution layer, which goes through ReLu and max pooling. It's then sent to two fully connected layers and finally, softmax is applied to predict the genre [Figure 4].

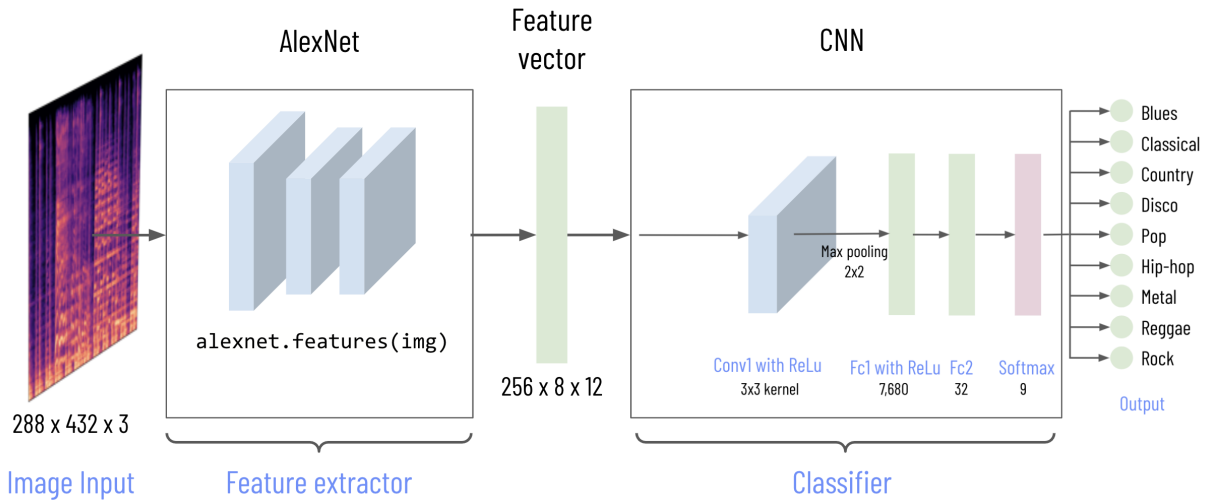


Figure 4: AlexCNN Architecture

Loss and Activation Functions

As this is a multi-class classification problem, the optimal functions are a categorical cross-entropy loss function and a softmax activation function [6].

Baseline Model

As a baseline model, we chose a simple 2-layer ANN with 300 neurons in the hidden layer [Figure 5]. The model flattens the mel-spectrogram images into a 1D vector and outputs a prediction score for each of the 9 genres. Relu activation functions are applied to each layer and the training code uses Adam optimization algorithm. Since this is a classification problem, the cross entropy loss function is used. The model predicts the genre with the highest prediction score and requires very minimal tuning as the only hyperparameters are batch size, learning rate, number of epochs, and number of neurons in the hidden layer.

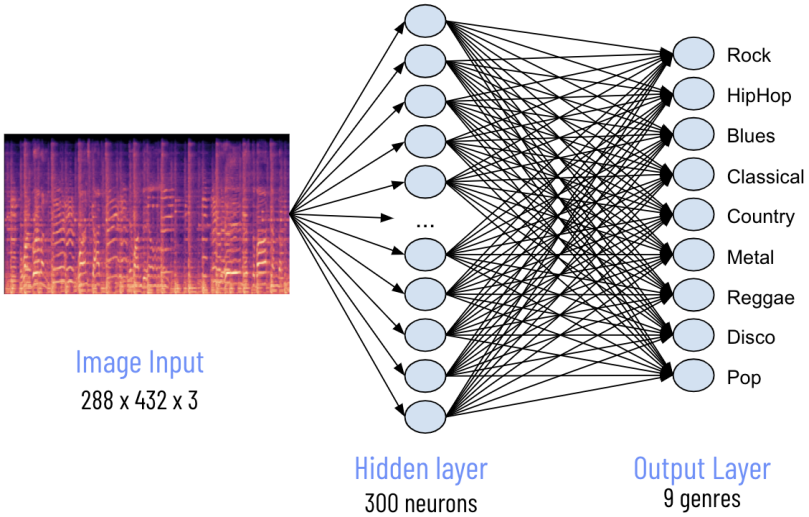
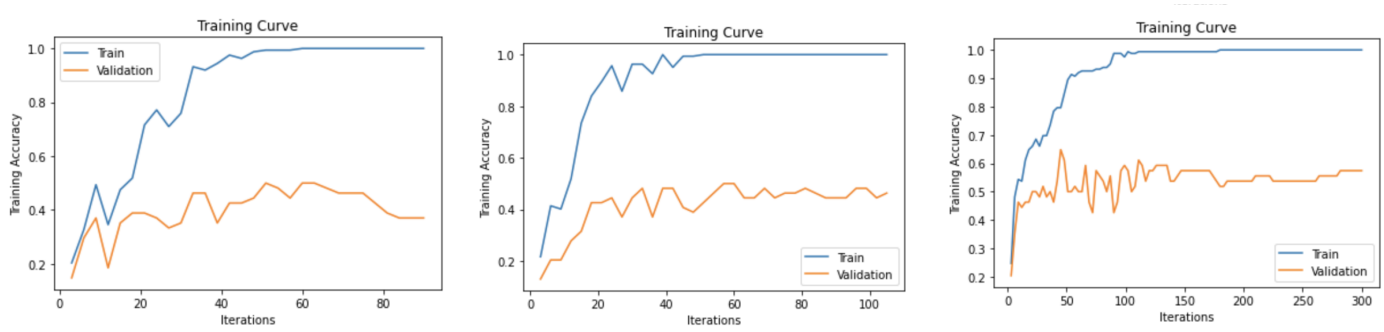


Figure 5: Baseline (ANN) Model

Quantitative Results

To ensure the models were working correctly, we overfit them on a small dataset. All the models were able to memorize the small dataset and achieve 100% training accuracy [Figure 6].



Baseline **Simple CNN** **AlexNet CNN**

Figure 6: Overfit Curves

Training, validation, and test accuracies for all three models are outlined in Table 2. Improving from our baseline model, we achieved a validation accuracy of 68.15% with our simple CNN model [Figure 7]. To further improve this and achieve an accuracy closer to or higher than humans, we added our second model: CNN with AlexNet features. The incorporation of transfer learning allowed us to finally exceed human accuracy [Figure 8].

Model	Training Accuracy	Validation Accuracy	Test Accuracy
Baseline - ANN	0.9907	0.6278	0.6130
Simple CNN	0.9988	0.6815	0.6796
Human	0.7000		
AlexNet CNN	0.9660	0.7981	0.7944

Table 2: Model accuracies

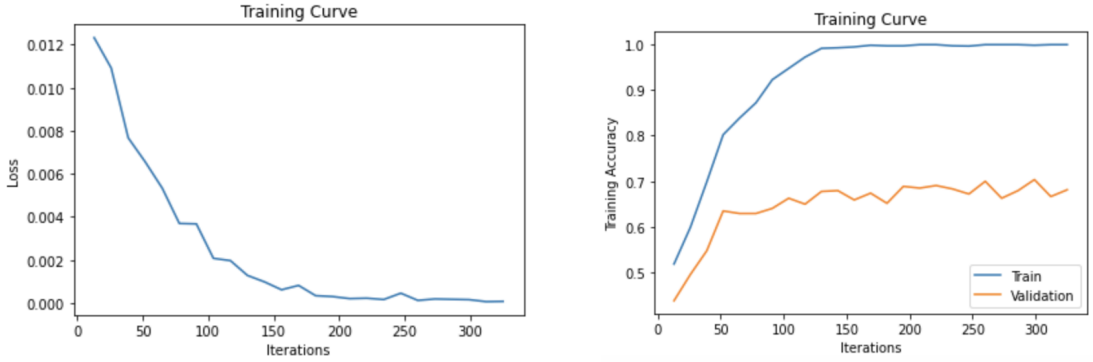


Figure 7: CNN training curves

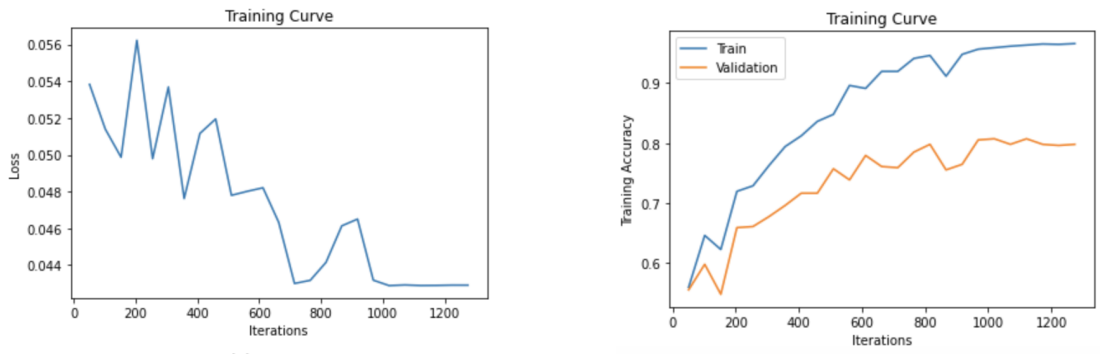


Figure 8: CNN with AlexNet features training curves

To further ensure our model achieved the best validation accuracy possible, we attempted to train and tune it on all three image representations: mel-spectrogram, spectral centroid, and pitch-class. Mel-spectrograms achieved the highest validation accuracy and also had the smallest gap between validation and training curves [Figure 9]. Therefore, we decided to use this image representation to train our final model.

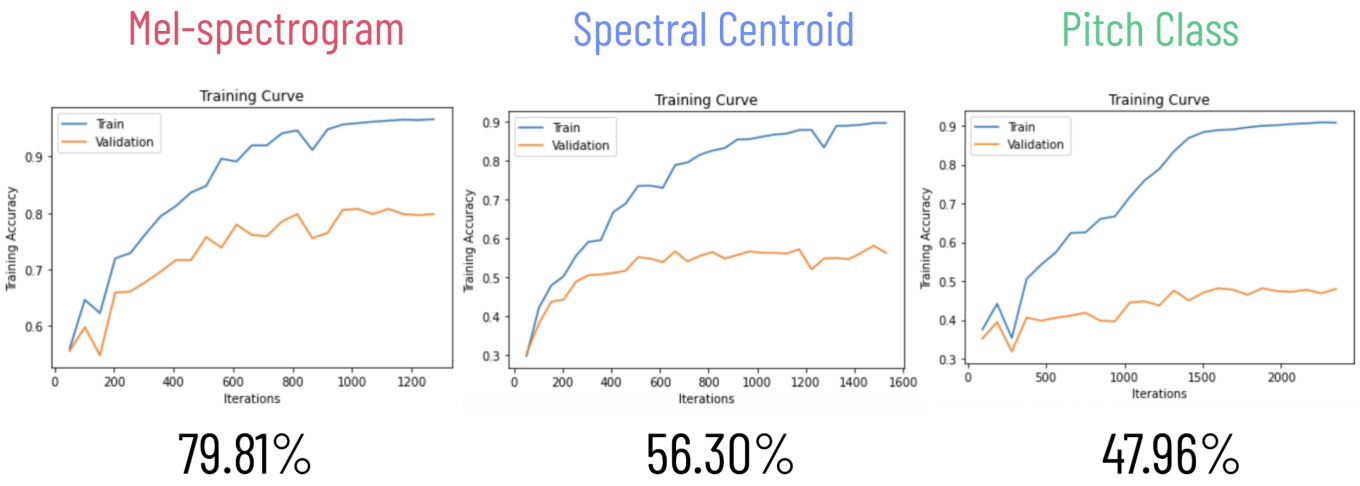


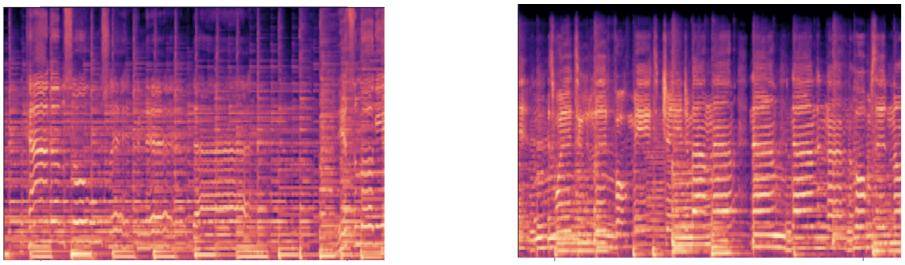
Figure 9: Comparing validation accuracy across image types

Qualitative Results

Looking at the confusion matrix for our test data [Table 3], we can see that our model confused many songs with rock and it specifically had a hard time distinguishing between country and rock (shown in blue). If we look at the spectrograms for country and rock [Figure 10], we can see that they have similar features so it makes sense that the model would have a harder time distinguishing between these two genres**.** Whereas classical and blues have very distinct spectrograms and therefore can be classified at a higher accuracy [Figure 11].

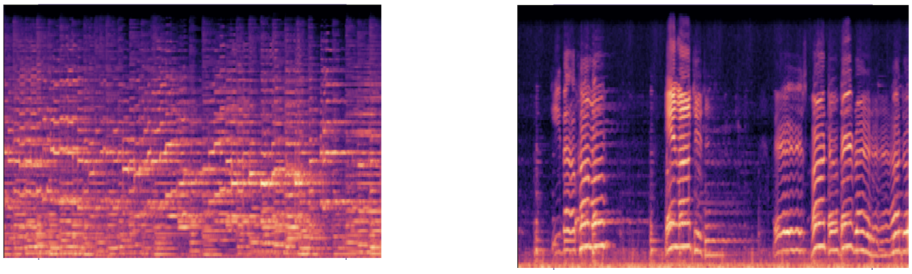
Actual Label	Predicted Label									
		Blues	Classical	Country	Disco	HipHop	Metal	Pop	Reggae	Rock	Recall
	Blues	55	0	1	0	0	0	0	0	4	0.917
	Classical	0	54	2	0	0	0	0	2	2	0.9
	Country	2	0	43	2	0	0	1	2	10	0.717
	Disco	0	1	2	41	0	0	2	8	6	0.683
	HipHop	0	0	1	0	48	1	3	4	3	0.8
	Metal	0	1	0	0	1	54	0	0	4	0.9
	Pop	0	0	1	4	5	0	45	3	2	0.75
	Reggae	0	0	0	3	7	1	0	47	2	0.783
	Rock	2	1	8	1	0	3	1	2	42	0.7
	Precision	0.932	0.947	0.741	0.804	0.787	0.915	0.865	0.691	0.56	

Table 3: GTZAN test data confusion matrix



Country **Rock**

Figure 10: Country vs. Rock spectrograms



Classical **Blues**

Figure 11: Classical vs. Blues spectrograms

Evaluate model on new data

New data samples were collected by downloading songs directly from Youtube and selecting a 30-sec subsection of the audio file. The songs selected were ensured to be more recent than the GTZAN dataset (i.e. newer than 2000) to guarantee the audio data did not already exist in our dataset [2]. Our model was trained and validated using only the GTZAN dataset so all of our manually collected data was completely new.

The new data collection process also involved ensuring that the audio segments selected did not contain non-musical data such as interludes, dialogues or audio gaps. The data only contained actual music audio to properly test our model's ability to classify genres.

A total of 15 new songs were obtained (5 per genre). Each was trimmed into three 10-sec clips to get 15 samples per genre and a total of 135 manually obtained samples. Since our model had achieved the best validation accuracy using the mel-spectrogram representation, we decided to process each of our 10-sec audio clips into mel-spectrograms to be fed into our model [Figure 12].

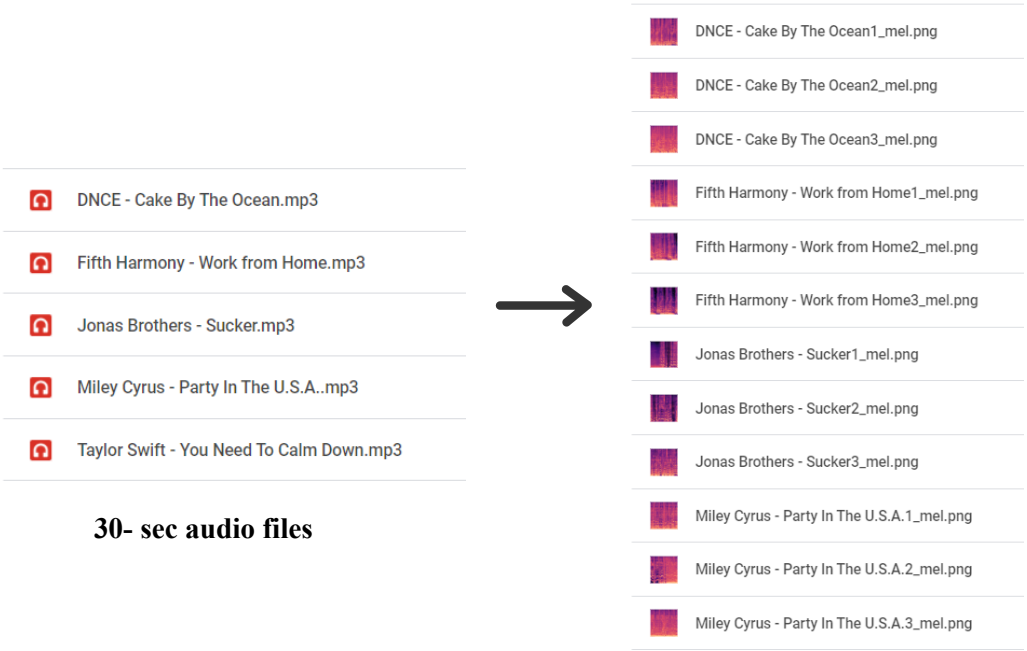


Figure 12: Manually obtained 30-sec song audio files processed into 3 mel-spectrograms

We tested our final model using these new spectrograms and achieved a final test accuracy of 48.8%. Our model's output for this data as well as possible reasons for why a low accuracy was achieved is discussed below.

Discussion

Our model achieved an accuracy of 79.4% on the GTZAN test set but only an accuracy of 48.8% on our new data test set. There are many reasons we believe this is justified.

The new data set contains more recent songs which do not easily fit into a single genre category in comparison to the older songs found in the GTZAN dataset. In the confusion matrix below [Table 4], we see that our model had difficulty differentiating between pop and hip hop songs (shown in blue) as well as rock and metal songs (shown in red). These genres are generally very similar and even more so in modern music as fusion genres such as 'pop rap' (hip hop and pop) have become increasingly popular. [7]

Our model also had difficulty identifying blues songs (shown in yellow). We believe this is most likely because purely 'blues' songs are not commonly found today. Most of the songs found for the blues category on Youtube were a fusion of one or more genres.

Pop and Reggae songs with an upbeat tempo were commonly mistaken for disco (shown in green). The genre of disco is ambiguous as it has evolved over time and now generally represents upbeat dance songs. Since disco music was most popular in the 1970s, we had difficulty finding purely disco songs to add to our new test set.

		Predicted Label									
Actual Label		Blues	Classical	Country	Disco	HipHop	Metal	Pop	Reggae	Rock	Recall
	Blues	0	0	3	0	0	2	0	4	6	0
	Classical	1	14	0	0	0	0	0	0	0	0.933
	Country	0	0	9	0	1	0	0	3	2	0.6
	Disco	0	0	1	9	0	0	1	3	2	0.6
	HipHop	0	0	0	0	4	0	8	2	1	0.267
	Metal	0	3	0	0	0	12	0	2	0	0.8
	Pop	0	0	0	3	3	0	6	0	1	0.4
	Reggae	0	0	0	4	0	0	2	9	0	0.6
	Rock	0	0	3	0	0	7	2	0	3	0.2
	Precision	0	0.824	0.563	0.563	0.5	0.571	0.316	0.391	0.2	

Table 4: New test data confusion matrix

With a higher accuracy on the GTZAN dataset, our model clearly performs well on older songs since it was trained on this dataset. With more ambiguity in genres today, our model was not able to perform as

well on modern music. The test accuracy on new data could have been improved by incorporating newer music into our training and validation datasets or narrowing down our classification problem to genres that are commonly found both today as well as in previous decades.

Ethical Considerations

Collecting our own data requires careful consideration to abide by copyright laws [8]. Downloading music without proper permissions is not only illegal but an unethical use of an artist's work without compensation. It's also important to consider the diversity of our data. Popular datasets, including GTZAN, contain mostly US-popular music [9]. This could lead to our model being a poor classifier of non-American or even non-English music. Music genres can also be underrepresented. For example, songs with explicit content are typically excluded from popular datasets as it's extra data cleaning to add censorship. This could result in our model having an exclusion bias against genres that are more heavily censored. These data biases can result in a skewed model. An unethical application of this could be within a recommendation system of a streaming platform. If our model is weak at classifying i.e. non-American or non-hip hop music, it could heavily favour certain music over others for promotion.

Project Difficulty / Quality

Genre classification can be ambiguous and complex as a song can share features of multiple genres, falling on a spectrum as opposed to a clear class. This resulted in the following difficulties:

Difficulty	Solution / Result
There exists a large number of genres.	We selected 9 major genres, large enough to represent a wide range of music and produce an effective classifier while small enough for manageable training.
Our classifier is limited by the training dataset, GTZAN, which had songs that clearly fit into one genre. Thus, our new test data had to reasonably be within the capabilities of this model.	We intentionally looked for songs that had one clearly dominant genre throughout the song, though this was challenging to find among modern music. We noticed our model was weaker with songs that alternated between a fusion of genres.
Genre definitions are not concretely defined.	We developed a model that relied on visual information for learning patterns. We trained it with three types of music representation to find the most suitable, resulting in a 31.85% improvement in accuracy.
Our goal was to beat human accuracy of 70%.	We experimented with multiple different models, ultimately achieving this goal by leveraging transfer learning with AlexNet.

Table 5: Project difficulty and how we overcame it

Overall, our final test accuracy on the GTZAN dataset of 79.4% is on par with prior art in genre classification using the same dataset. Within a collection of models curated by researchers at the Aristotle University of Thessaloniki, the highest accuracy achieved by such a neural network was 81.4%, a 2% difference [Figure 13].

Method	Features	GTZAN	ISMIR	Homburg	1517-Artists	Unique
LRSMs	Fusion cmc	*87.00* (2.62)	*82.99*	*62.40* (3.65)	*54.91* (2.54)	72.90 (1.26)
	Fusion cm	86.80 (2.85)	82.30	62.29 (4.04)	54.74 (2.68)	72.84 (1.11)
	Cortical	85.50 (2.79)	81.62	61.71 (4.02)	54.43 (2.58)	72.35 (1.05)
	MFCCs	50.60 (5.35)	59.08	43.26 (2.30)	23.45 (1.96)	54.60 (1.87)
	Chroma	17.6 (4.03)	43.90	26.93 (1.39)	9.77 (1.13)	24.71 (1.87)
SRC	Fusion cmc	84.40 (2.27)	82.85	59.64 (3.24)	53.08 (2.83)	72.61 (1.18)
	Fusion cm	84.40 (2.71)	80.50	58.10 (4.15)	50.78 (2.41)	71.97 (1.85)
	Cortical	84.10 (3.04)	79.97	57.52 (3.98)	50.72 (2.61)	67.48 (1.14)
	MFCCs	63.60 (5.01)	70.50	38.10 (2.74)	30.12 (1.87)	56.59 (1.06)
	Chroma	36.80 (5.67)	47.73	26.61 (2.58)	17.01 (1.31)	31.20 (2.94)
SVMs	Fusion cmc	86.80 (2.82)	*82.99*	62.61 (3.22)	53.30 (3.19)	*75.15* (1.48)
	Fusion cm	86.40 (2.98)	73.93	61.07 (3.32)	53.08 (3.38)	73.54 (1.87)
	Cortical	86.00 (2.83)	73.79	60.92 (2.83)	53.71 (3.18)	68.89 (2.22)
	MFCCs	54.90 (3.14)	52.67	43.95 (2.05)	26.16 (2.96)	53.22 (1.06)
	Chroma	16.90 (4.02)	48.42	34.99 (1.96)	12.16 (2.27)	39.87 (2.67)
NN	Fusion cmc	81.40 (3.20)	78.64	50.26 (4.21)	44.87 (2.21)	64.68 (2.31)
	Fusion cm	81.10 (3.31)	79.02	50.21 (3.48)	44.90 (2.43)	64.68 (2.31)
	Cortical	80.70 (3.26)	79.69	49.78 (2.98)	44.84 (2.55)	64.43 (2.57)
	MFCCs	57.60 (5.05)	67.76	29.79 (3.13)	26.57 (1.84)	48.82 (2.17)
	Chroma	34.10 (4.67)	42.24	23.64 (1.93)	14.40 (1.80)	25.32 (2.96)

Figure 13: Music classification accuracies using different methods and datasets [10]

The complexity of our project could be increased by exploring the following:

1. Augmenting the audio samples by adding a slight noise to increase the dataset. The reason we did not implement this is due to storage issues.
2. Implementing more types of models.
    a. We could expand our CNN to implement parallel CNNs, each detecting a pattern in a different feature (i.e. frequencies and rhythm) and combining the output predictions for the final classification [5].
    b. Though we did attempt to implement an RNN but were limited by computational power, an RNN would work well given that audio represents sequential data. By feeding the RNN multiple short slices from the audio and classifying each slice, we could use an average of the results to inform the genre of the entire audio sequence. A weakness of the standard RNN is that earlier slices have increasingly less influence on the result; to resolve, we could implement a second RNN by adding a Long Short-Term Memory layer to more effectively find the long-term structure of the song [11].

Though we surpassed our goal for J.A.A.Ms, with more time and resources, our model could be greatly improved and contribute to increasing the efficiency of music retrieval systems.

Google Colab links

Main code notebook:
https://colab.research.google.com/drive/1vTZnbCkwQZ0npFHaTUUfG8PxtBpT-0lW?usp=sharing

New dataset processing and code for demonstration:
https://colab.research.google.com/drive/1lqqLbOQJadUwVPF0v60xrB2XIUu2YK14?usp=sharing

References

[1] D. A. Huang, A. A. Serafini, and E. J. Pugh, "Music Genre Classification," *standford.edu*, 2018. [Online]. Available: http://cs229.stanford.edu/proj2018/report/21.pdf. [Accessed: 09-Feb-2021].

[2] A. Olteanu, "GTZAN Dataset - Music Genre Classification," Kaggle, 24-Mar-2020. [Online]. Available: https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification. [Accessed: 10-Feb-2021].

[3] B. L. Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," 10-Jun-2013. [Online]. Available: https://arxiv.org/pdf/1306.1461.pdf. [Accessed: 10-Feb-2013].

[4] Y. Xu and W. Zhou, "A deep music genres classification model based on CNN with Squeeze & Excitation Block," *IEEE Xplore*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9306374. [Accessed: 09-Feb-2021].

[5] T. Lidy and A. Schindler, "Parallel Convolutional Neural Networks for Music Genre and Mood Classification," 01-Aug-2016. [Accessed: 10-Feb-2021].

[6] S. Ronaghan, "Deep Learning: Which Loss and Activation Functions should I use?," *Medium*, 01-Aug-2019. [Online]. Available: https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8. [Accessed: 12-Feb-2021].

[7] "Pop rap," *Wikipedia*, 04-Apr-2021. [Online]. Available: https://en.wikipedia.org/wiki/Pop_rap. [Accessed: 08-Apr-2021].

[8] "Music Copyrights 101 - Protect and Copyright Your Music," *United States*, 20-Nov-2019. [Online]. Available: https://www.tunecore.com/guides/copyrights-101. [Accessed: 12-Feb-2021].

[9] A. Holzapfel and M. Coeckelbergh, "Ethical Dimensions of Music Information Retrieval Technology," Sep-2018. [Online]. Available: https://www.researchgate.net/publication/327807581_Ethical_Dimensions_of_Music_Information_Retrieval_Technology. [Accessed: 09-Feb-2021].

[10] C. Kotropoulos, “Music classification by low-rank semantic mappings,” *Research Gate*, Dec-2013. [Online]. Available: https://www.researchgate.net/figure/Music-genre-classification-accuracies-for-the-GTZAN-ISMIR-Homburg-1517-Artists-and_tbl1_257879201. [Accessed: 08-Apr-2021].

[11] K. Choi and M. Sandler, “Convolutional Recurrent Neural Networks for Music Classification,” 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7952585. [Accessed: 09-Feb-2021].