Music Genre Classification

- Balaji Patakula - 2021204008
- Phani Karnati - 2021204014
- Saurav Chhatani - 2020101113
- P A E Sai Raj - 2020101049

Group 42

Problem Statement

- **Style is not well-defined for music; the genre piece of music is highly related to its acoustic properties.**
- **Musical style encoding depends on extensive feature engineering and static definitions of components of style.**
- **Learning encodings directly from raw audio instead has significant applications in musical style transfer and audio processing**
- **Develop a deep learning algorithm for music genre encoding from raw audio samples without explicit feature engineering**

Prior knowledge & limitations

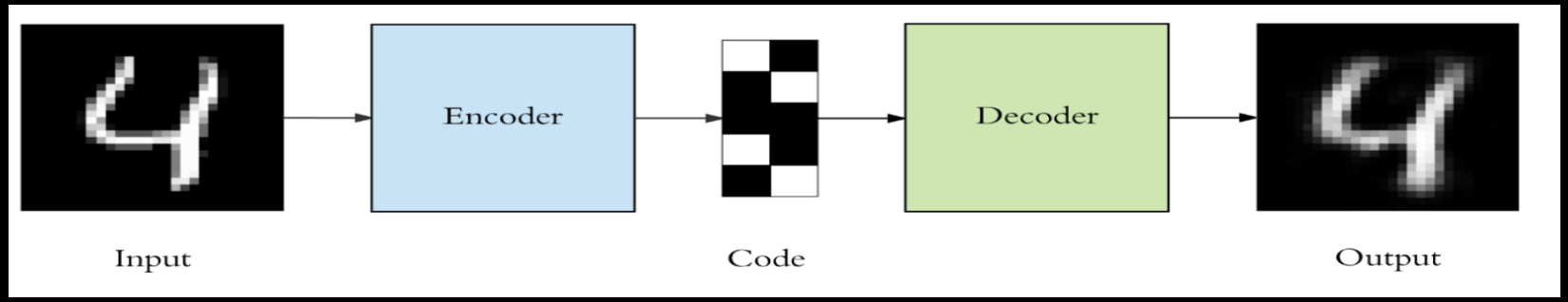
- **Extract time and frequency domain features from the audio files.**
    - **features are Spectrograms, Mel-spectrograms, MFCC, Spectral centroids, Energy, Spectral Roll-off, Spectral Flux, Spectral Entropy, Zero-crossing rate, and Pitch etc.,**
- **Use any of these features (or ensemble) and training dataset to develop classification models for music genre**
- **Feature engineering is quite cumbersome to achieve more accurate models**

Research Scope

- **Develop autoencoders to learn the latent space without the need for feature extraction.**
- **Optimize the latent vector dimension to enable loss less reconstruction**
- **Use latent vectors as feature set for developing classification models.**
- **Use latent vectors as feature set with fully connected deep neural network with soft max activation to generate the output layer and create the classification model**

Auto encoders

- **Dimensionality reduction technique for non-linear space, like PCA for linear space**
- **Compress the input into a lower-dimensional code and then reconstruct the output from this representation**
- **consists of 3 components: encoder, code and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code**



Soft max Activation

$$softmax(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

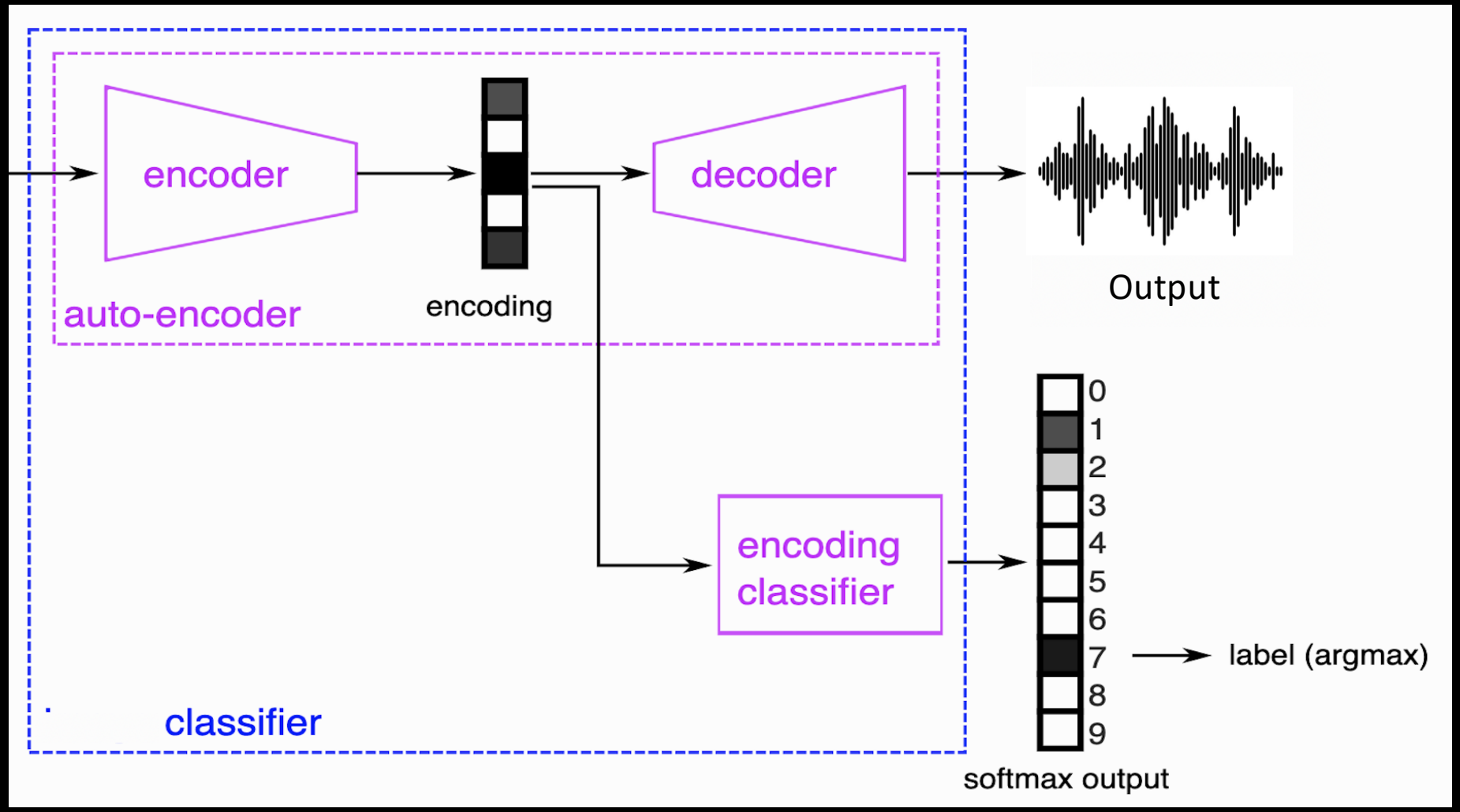
Here,

- **z** is the vector of raw outputs from the neural network
- The value of e ≈ 2.718
- The i-th entry in the softmax output vector softmax(**z**) can be thought of as the predicted probability of the test input belonging to class i.

- ***It is often used as the last activation function of a neural network to normalize the output of a network to a probability distribut***
- ***Soft max is an activation function that scales numbers/logits into probabilities. The output of a Soft max is a vector (say v) with probabilities of each possible outcome. The probabilities in vector v sums to one for all possible outcomes or classes.***

Network Diagram

Input



Model Architecture

Vanilla Autoencoder

Encoder: 3 hidden layers, learn $f(x) : \mathbb{R}^{500} \rightarrow \mathbb{R}^{64}$, where x is downsampled input

Decoder: 3 hidden layers, learn $g(x) : \mathbb{R}^{64} \rightarrow \mathbb{R}^{500}$, where x is encoder output

$\mathcal{L}_{reconstruction} = ||x - g(f(x))||_2^2$, where x is downsampled input

Two Layer Neural Network

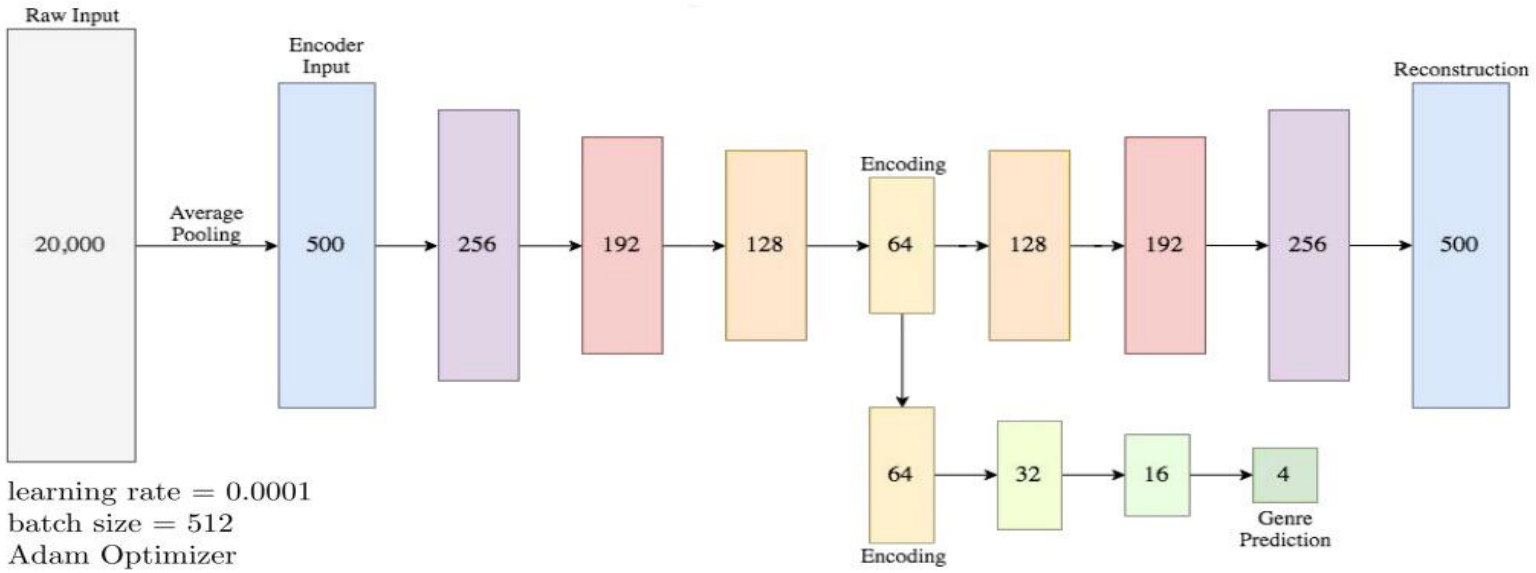
Hidden layer: 128-dim, tanh activation

$\mathcal{L}_{cross-entropy} = -\sum_{i=0}^{3} y_i \ log(\hat{y}_i)$

Deep Softmax Autoencoder (Final Architecture)

Simultaneously train a deep autoencoder and multi-class classifier using the 64-dim encoding as input to the classifier

$\mathcal{L} = \gamma||x - g(f(x))||_2^2 - (1-\gamma)\sum_{i=0}^{3} y_i \ log(\hat{y}_i)$, where reconstruction weight $\gamma = 0.9$



learning rate = 0.0001
batch size = 512
Adam Optimizer

Soft Max Auto Encoders Loss function

Deep Soft max Autoencoder

Simultaneously train a deep autoencoder and multi-class classifier using the 64-dim encoding as input to the classifier

$$L = \gamma ||x - g(f(x))||_2^2 - (1-\gamma) \sum_{i=0}^{3} y_i log(y_i^-)$$

Dataset

- **Music Analysis, Retrieval and Synthesis for Audio Signals (MARSYAS) GTZAN Dataset**
  - ❖ **10000 songs (30 seconds each) labelled as 10 different genre e.g. classical, jazz, metal, and pop etc.,**

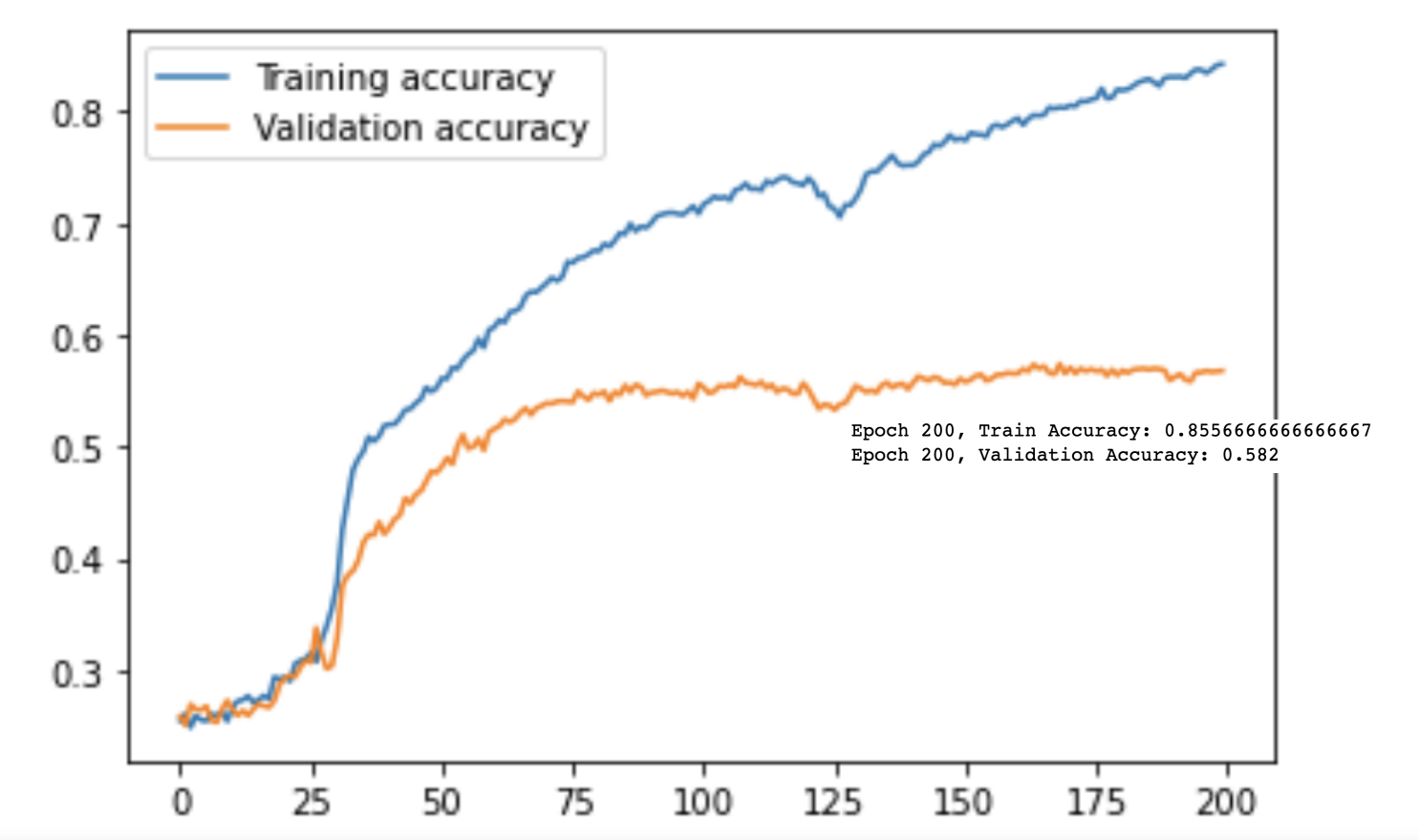
- **https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification**

Code Base

- Github Link
- Download the data set from **GTZAN data set (Music data set) be and copy into the project directory before running the jupyter notebooks**

Results

Model Training

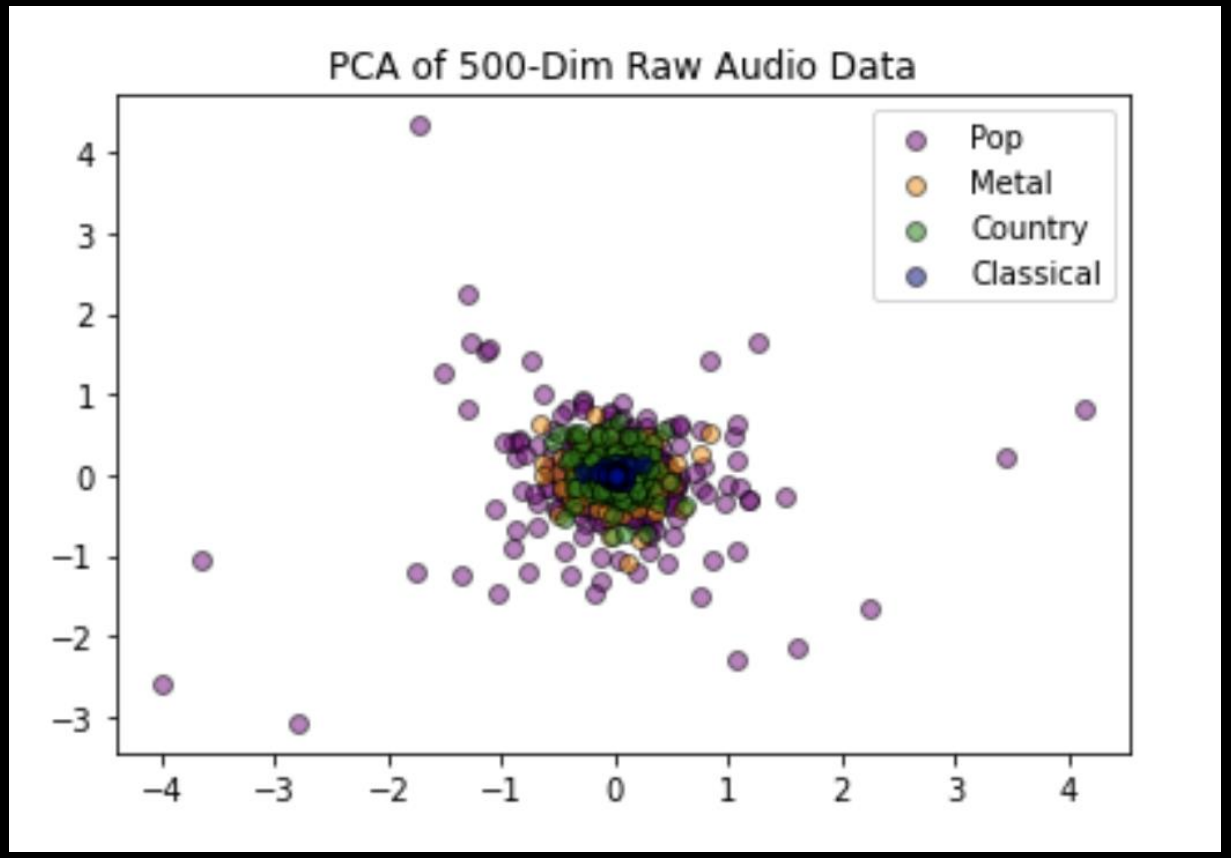


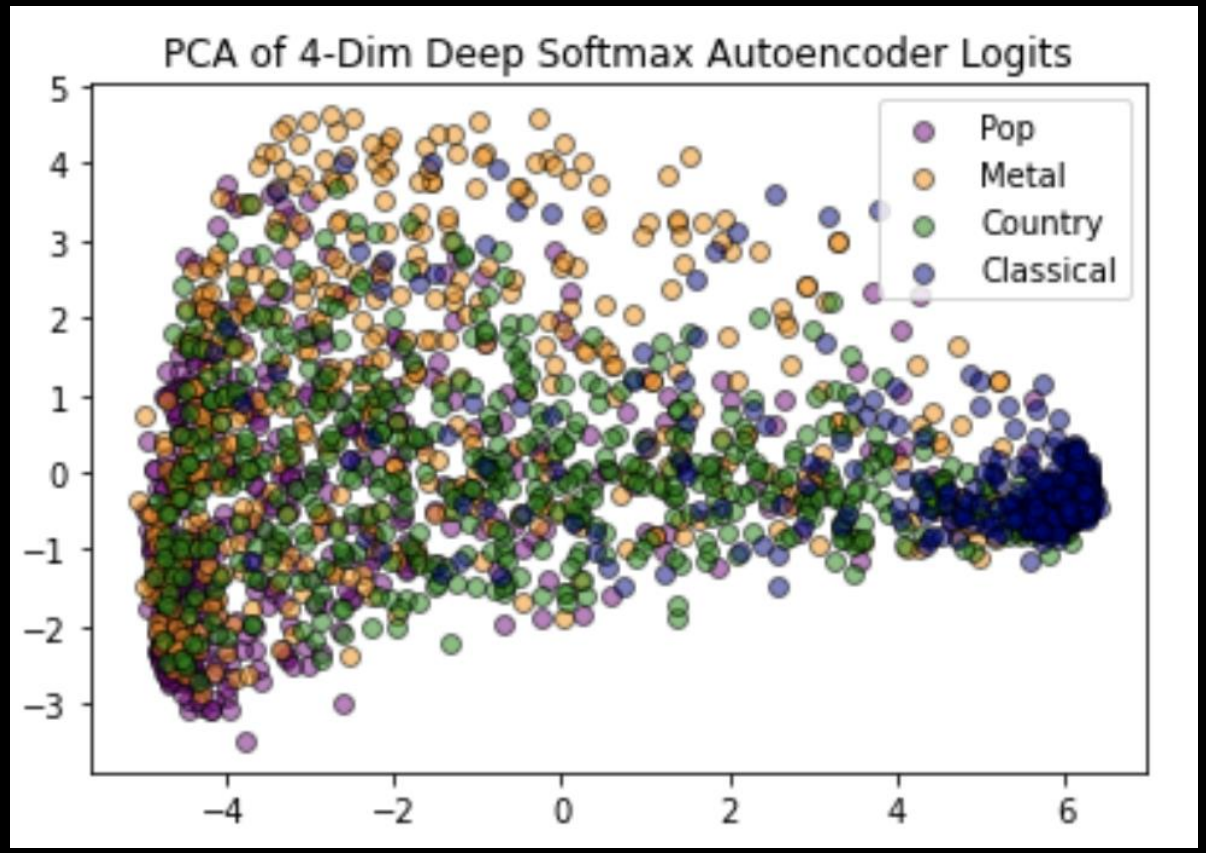

```
input_dim = 500
numexamples = 8000
num_classes = 4
alpha = 0.0001
num_epochs = 200
batch_size = 512
classificationweight = 0.1
```

Training Time: ~15 min

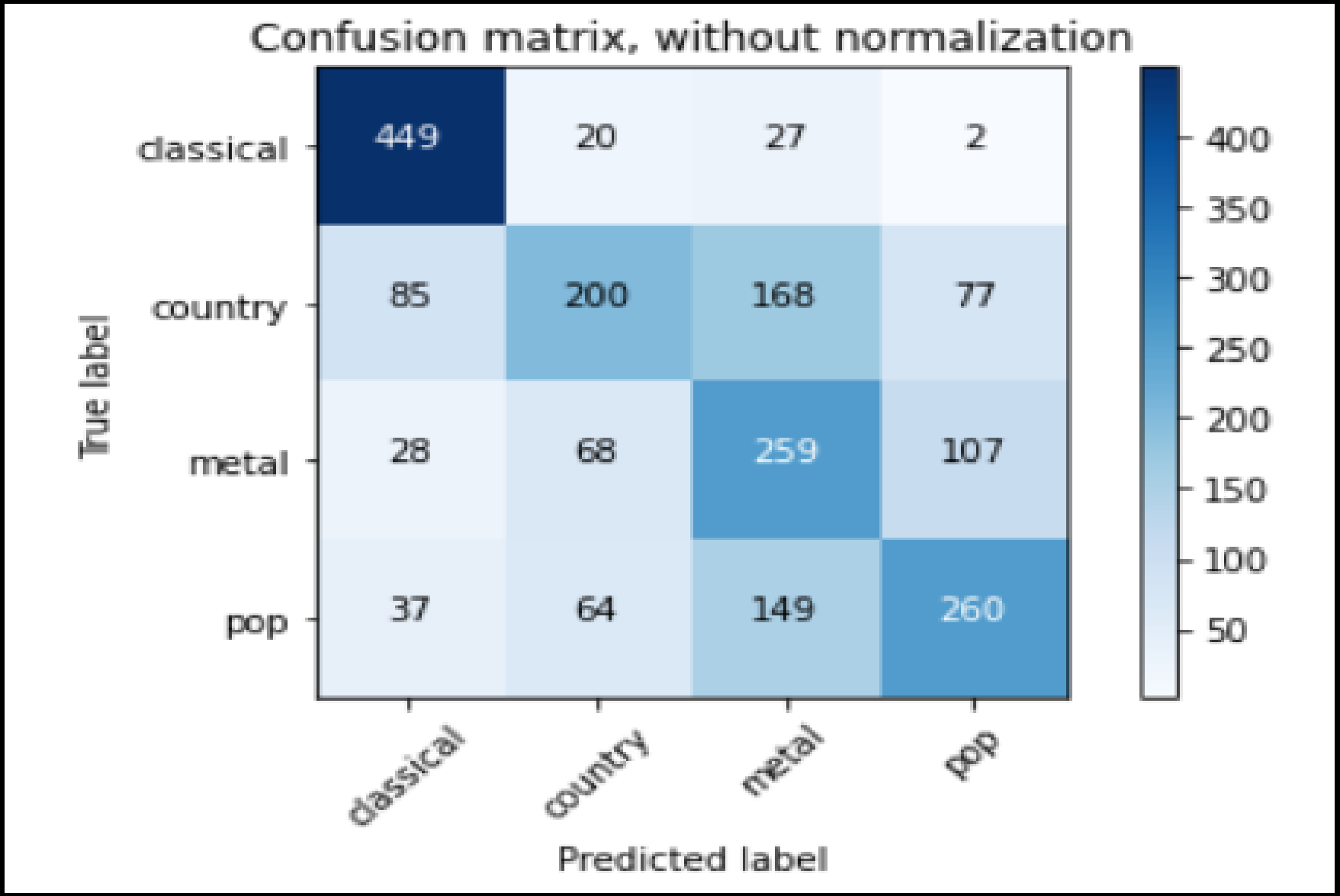
```
Epoch 200, Train Accuracy: 0.8556666666666667
Epoch 200, Validation Accuracy: 0.582
```

PCA Comparison





Confusion Matrix



Precision & Recall

	precision	recall	f1-score	support
classical	0.750	0.902	0.819	498
country	0.568	0.377	0.454	530
metal	0.430	0.561	0.486	462
pop	0.583	0.510	0.544	510
accuracy			0.584	2000
macro avg	0.583	0.587	0.576	2000
weighted avg	0.585	0.584	0.575	2000

Conclusion

- Music genres can be classified
- Model has the difficulty in classifying between metal and country
- Classical shows the highest precision, recall, and F1 score, likely due to its distinct style

References

- Music Analysis, Retrieval and Synthesis for Audio Signals (MARSYAS) GTZAN Dataset.
- H. Bahuleyan. Music Genre Classification using Machine Learning Techniques in arXiv, 2018.
- N. Mor et al. A Universal Music Translation Network in arXiv, 2018.
- I. Simon et al. Learning a Latent Space of Multitrack Measures in arXiv, 2018.
- S. Dai et al. Music Style Transfer: A Position Paper in arXiv, 2018.
- G. Tzanetakis et al. Musical Genre Classification of Audio Signals in IEEE, 2002.
- L. Gatys et al. A Neural Algorithm of Artistic Style in arXiv 2015.
- M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org, 2015.
- T. Li et al. "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network" in IMECS, 2010. R. Chen et al. "Isolating Sources of Disentanglement in VAEs" in arXiv, 2018.


THANK
YOU