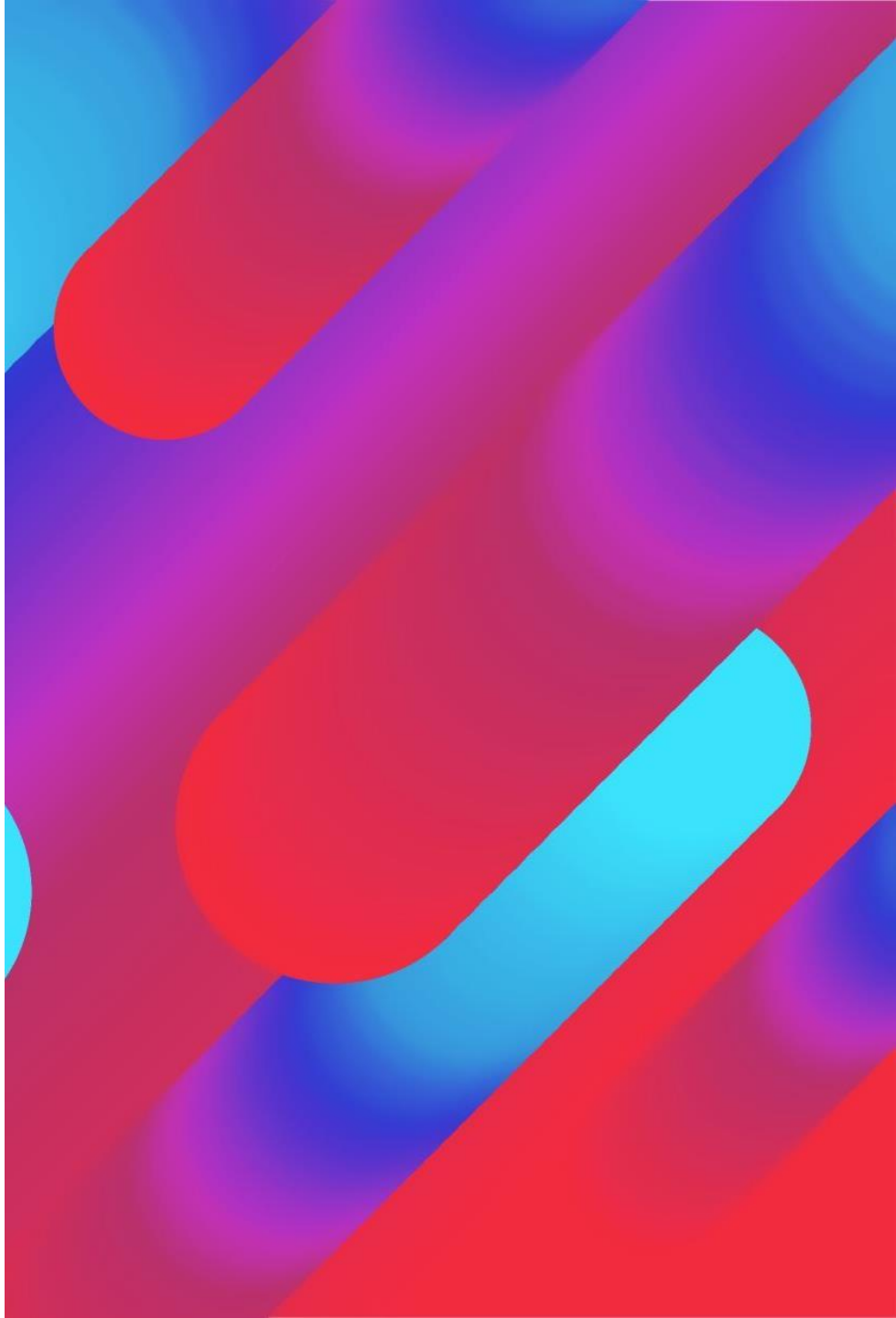
DLS Final Project

SRINIVAS YASHVANTH VALAVALA - SVALAVAL

SRI VENKATA SAI ANOOP BULUSU - SRBULUSU

PRANAV MAHAJAN - PMAHAJA



Abstract

We developed a CNN and RESNET to classify the genre of musical audio samples. The model was trained on a large dataset of musical audio samples, each labeled with a specific genre.

We used the models to classify the genres of a new set of musical audio samples. We then used the classification results to generate a custom playlist for the user, based on the distribution of genres they had been listening to.

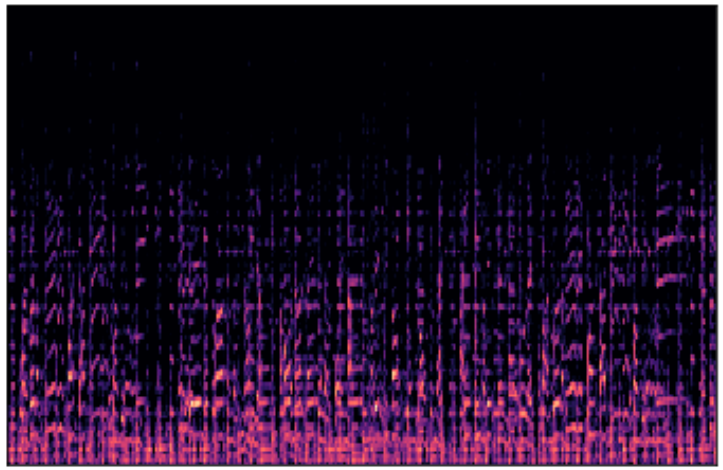
About the Dataset

For this project we have used the GTZAN dataset.

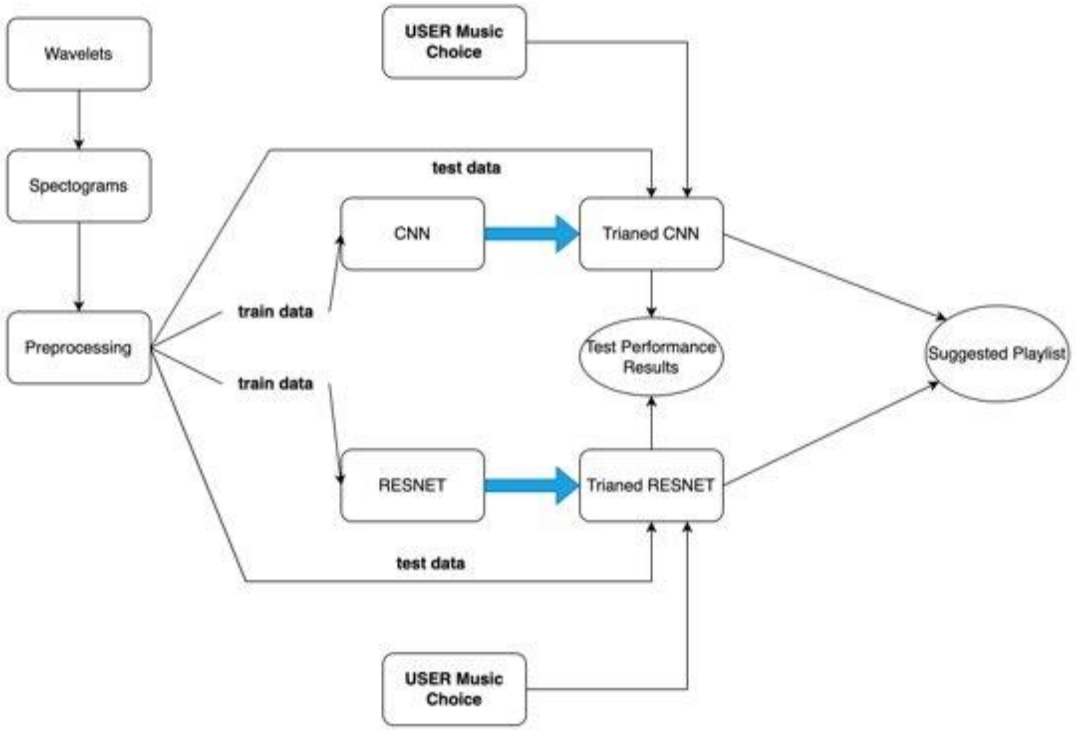
The dataset consists of audio samples of 10 different genres.

Each genre consists of 100 samples each, having length of 30 seconds.

We then converted all the audio samples into spectograms.



Approach


Wavelets
Spectograms
Preprocessing
USER Music Choice
test data
CNN
Trianed CNN
train data
train data
Test Performance Results
Suggested Playlist
RESNET
Trianed RESNET
test data
USER Music Choice

Networks We Used

CNN :

Convolutional neural networks (CNN) are a deep learning technique that is commonly used for image classification tasks and tasks that require pixel processing. In our case, we use it for the classification of music data by first converting them into an appropriate image format.

ResNet :

We also implemented ResNet18 model using transfer learning. ResNets solve the problem of the vanishing/exploding gradient by introducing Residual Blocks and can improve upon the accuracy of traditional CNNs.

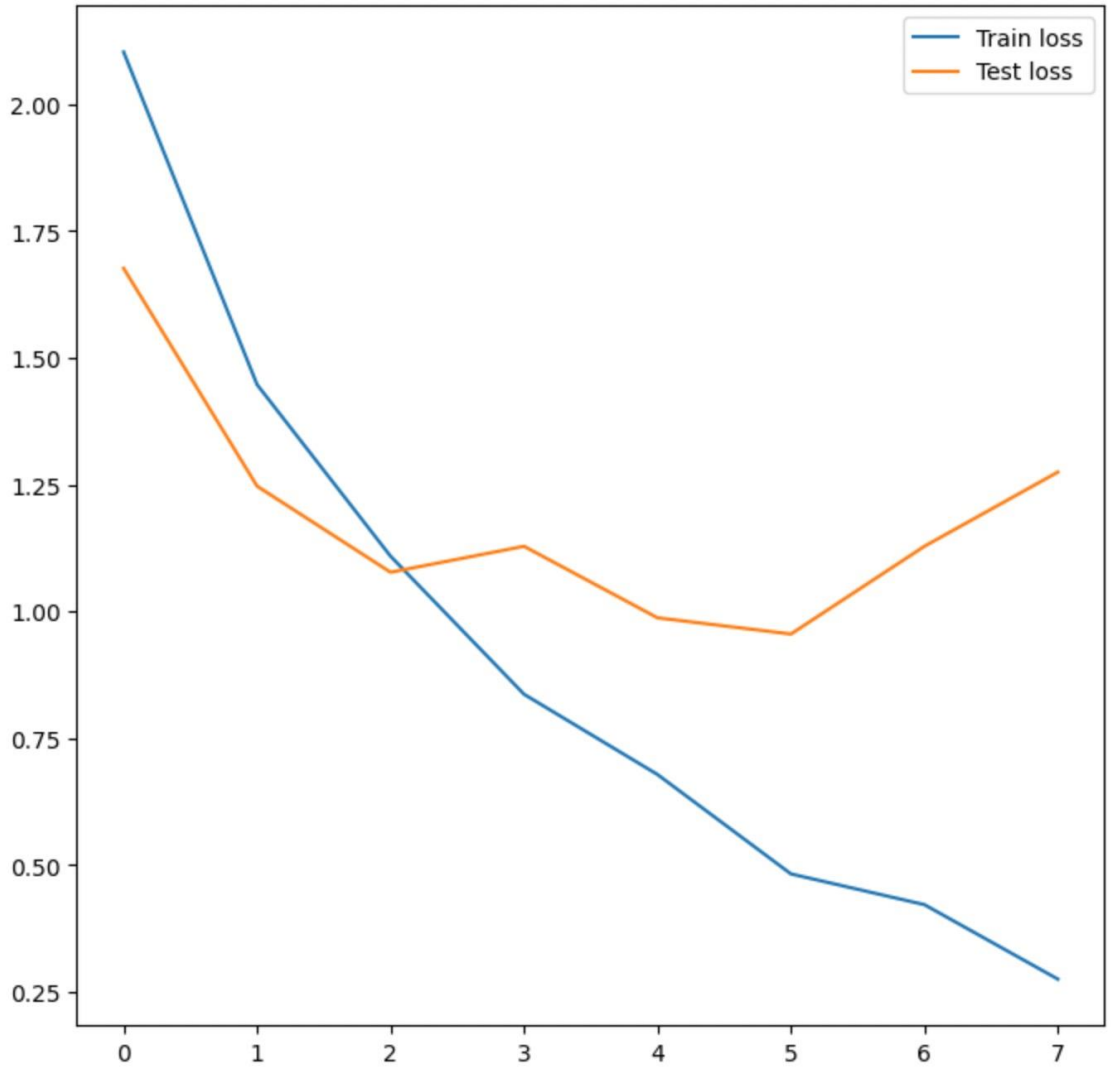
CNN

```
ConvNet(
  (conv1): Conv2d(1, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv3): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool3): MaxPool2d(kernel_size=4, stride=4, padding=0, dilation=1, ceil_mode=False)
  (conv4): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool4): MaxPool2d(kernel_size=4, stride=4, padding=0, dilation=1, ceil_mode=False)
  (dropout): Dropout(p=0.5, inplace=False)
  (fc1): Linear(in_features=77824, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=256, bias=True)
  (fc3): Linear(in_features=256, out_features=10, bias=True)
)
```

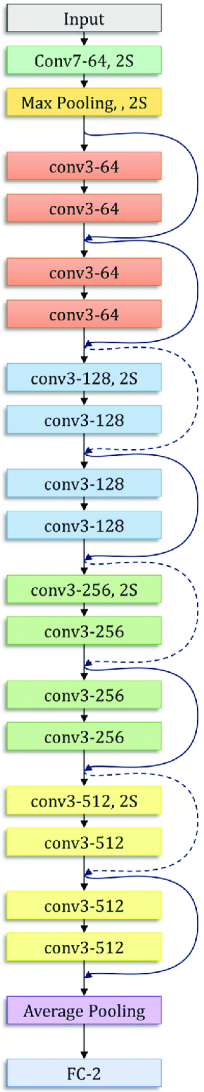
CNN Performance

We got a Model Test Accuracy of 66%.

But we can clearly observe that the model is overfitting.

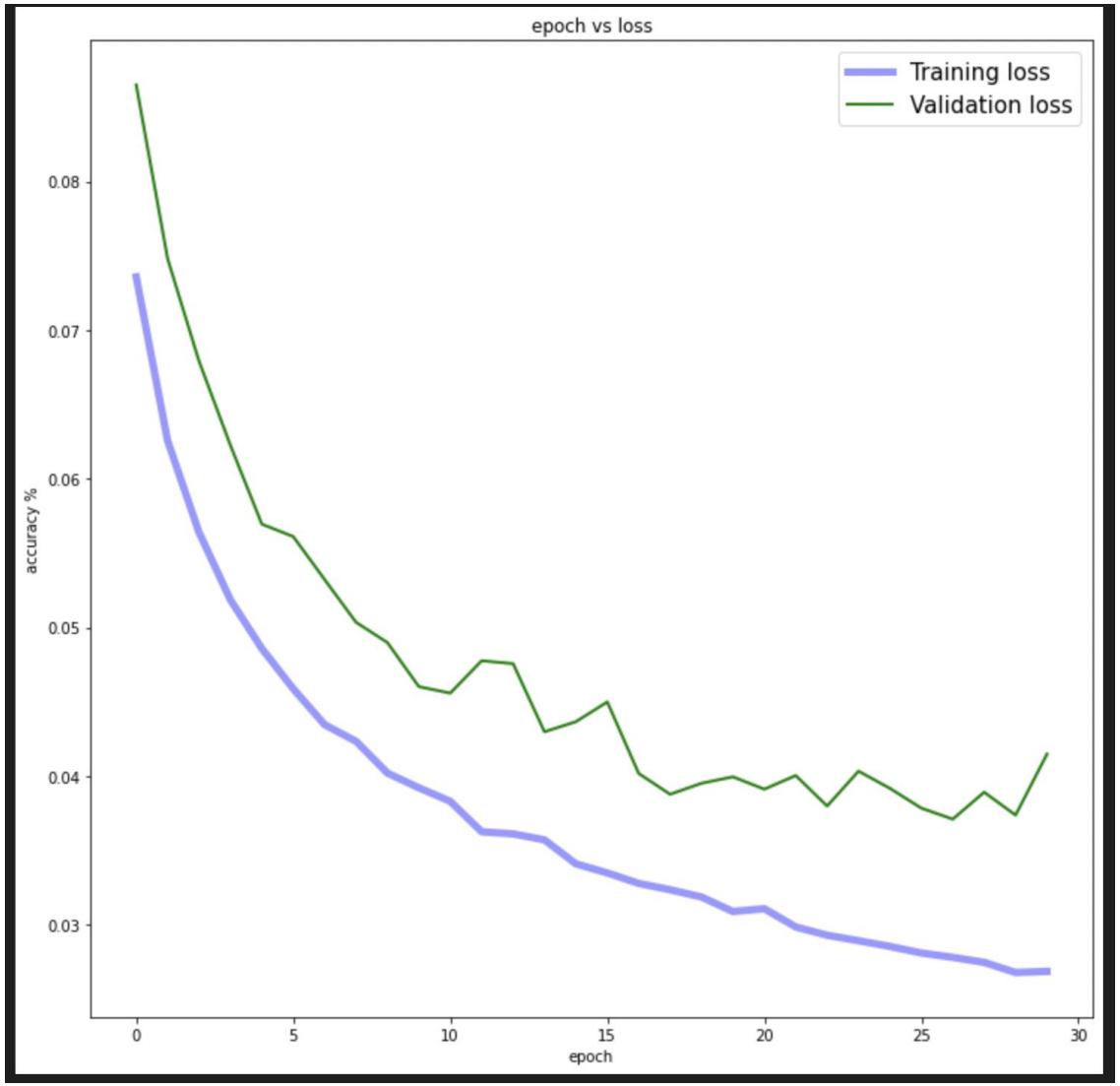


Resnet

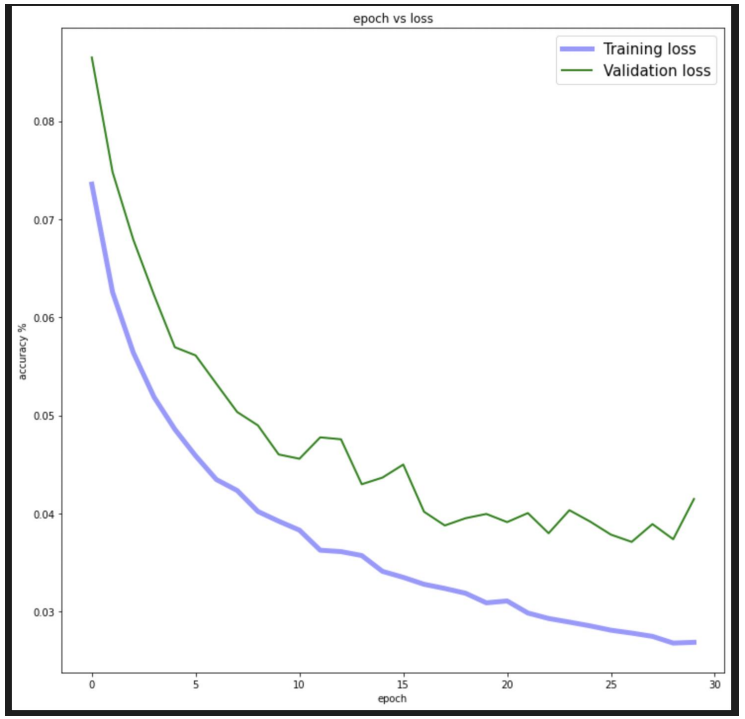

Input
Conv7-64, 2S
Max Pooling, , 2S
conv3-64
conv3-64
conv3-64
conv3-64
conv3-128, 2S
conv3-128
conv3-128
conv3-128
conv3-256, 2S
conv3-256
conv3-256
conv3-256
conv3-512, 2S
conv3-512
conv3-512
conv3-512
Average Pooling
FC-2

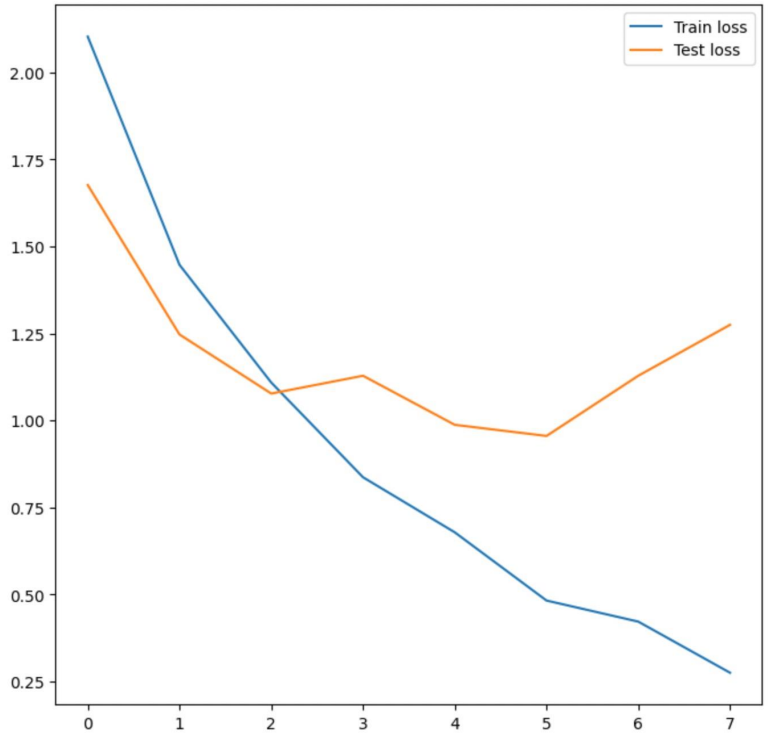
Resnet Performance

We achieved a test accuracy around 71%.



Performance Comparision





Results

Our use case is recommending users a playlist of music depending on the music they were already listening to, which doesn't require very high levels of accuracy to work. Instead, it only requires that similar music is recommended.

Also, some inaccuracy might be rather beneficial in our case to provide a wider variation of music to the user.

The two approaches used by us are a regular CNN and a pre-trained ResNet18 model. The CNN achieved an accuracy of about 60 percent while the pre-trained ResNet model was able to achieve a test accuracy of about 71 percent. Overall, both models learn the patterns in the audios.

Conclusion

Both of our models achieve decent accuracy for our use case. As discussed earlier this use case can benefit from being not fully accurate.

This thought process gives rise to one future implementation that is controlling a variable that helps us bring more variety to what is being recommended to the user. In other words, we are able to control how close to the original songs are the songs that are recommended.

Another future implementation would be using more data and a bigger data set involving a higher number of genres for the training.

Using data augmentation we can increase our data set by using the following transformations.

Thank You !