Recompose Classics

Predict completion of Chopin pieces through use of various DNN models

Hannah Do **Ethan Cobb** **Bethold Owusu** **CSCI 1910**

Data and Goal

Previously

Old Data: 48 Midi files from famous composer Chopin

Featuring Engineering

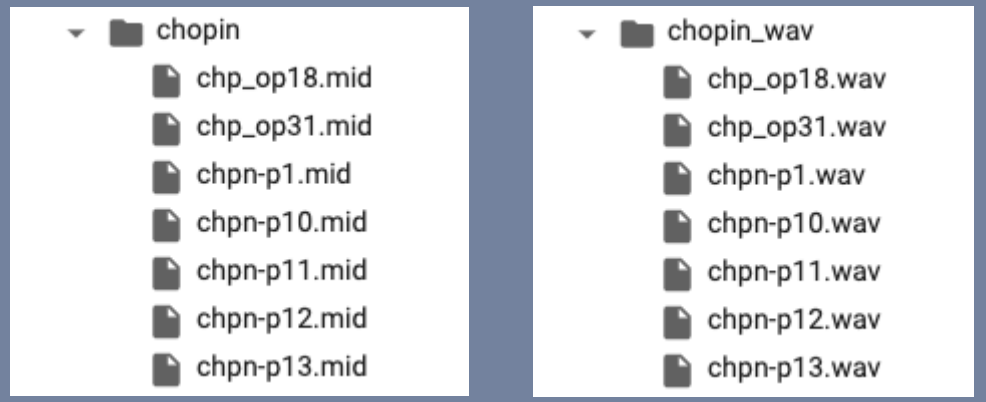
2862 wave audio file (15 seconds each)

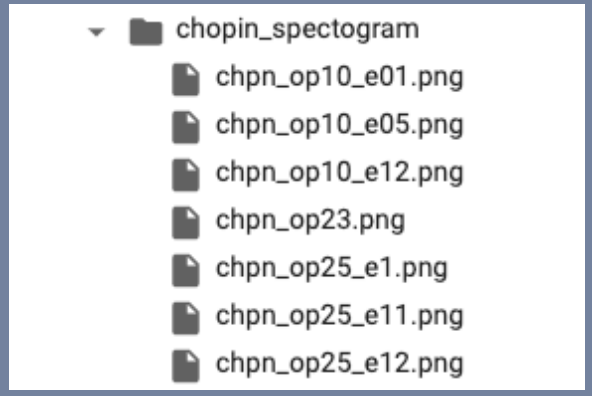
Goal

Predicting and Regenerating Music sequence within an interval using DNN architectures

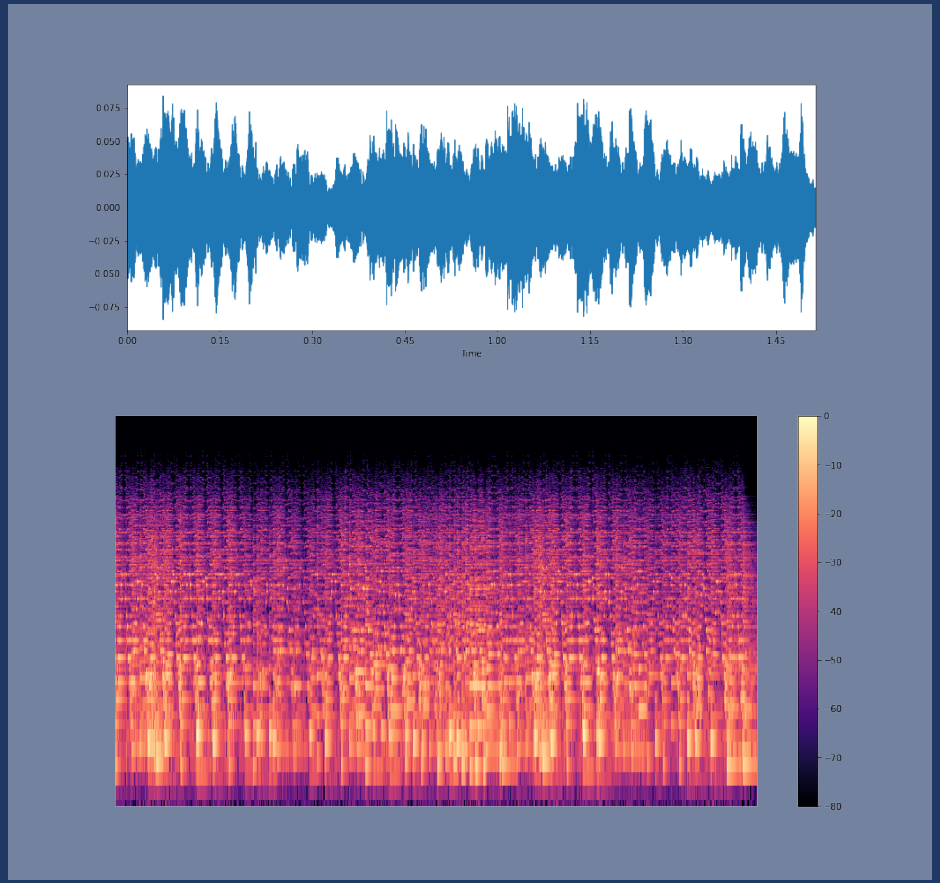
APPROACH

1. Conversion from 'midi' to 'wav' files

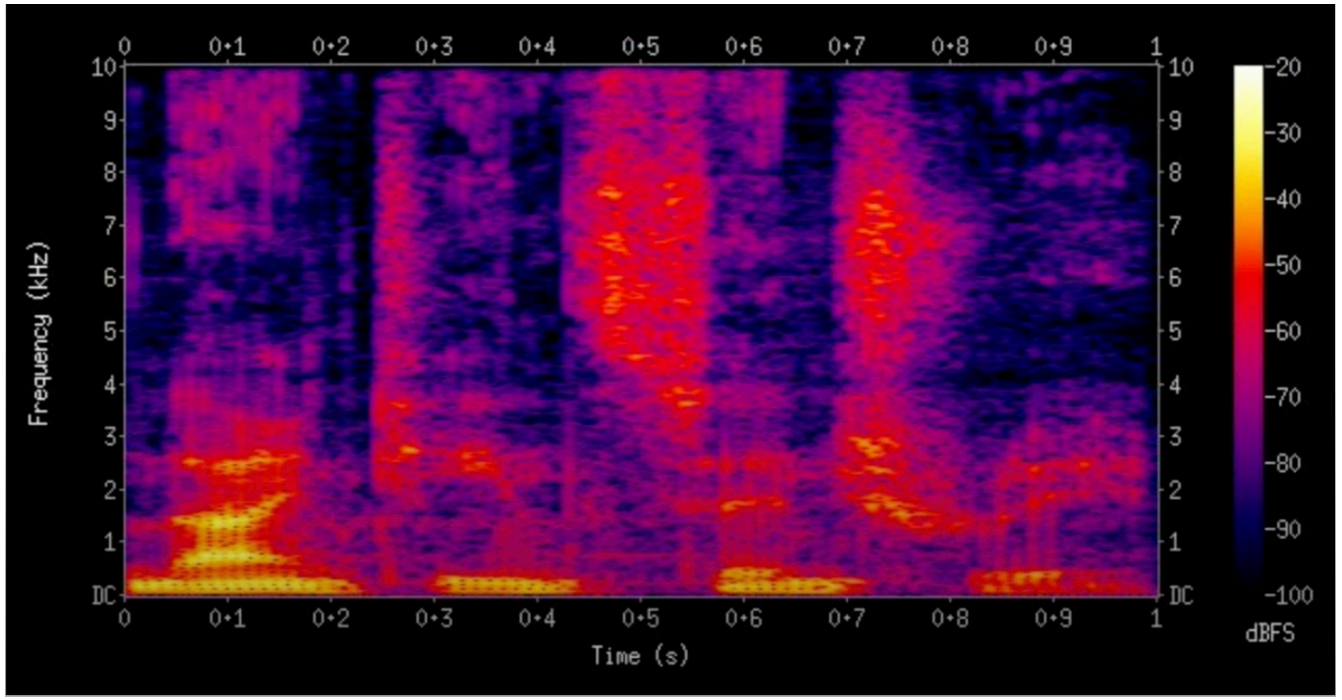




2. Transformed '**wav**' files to 'audio waveform' and to 'spectrogram'

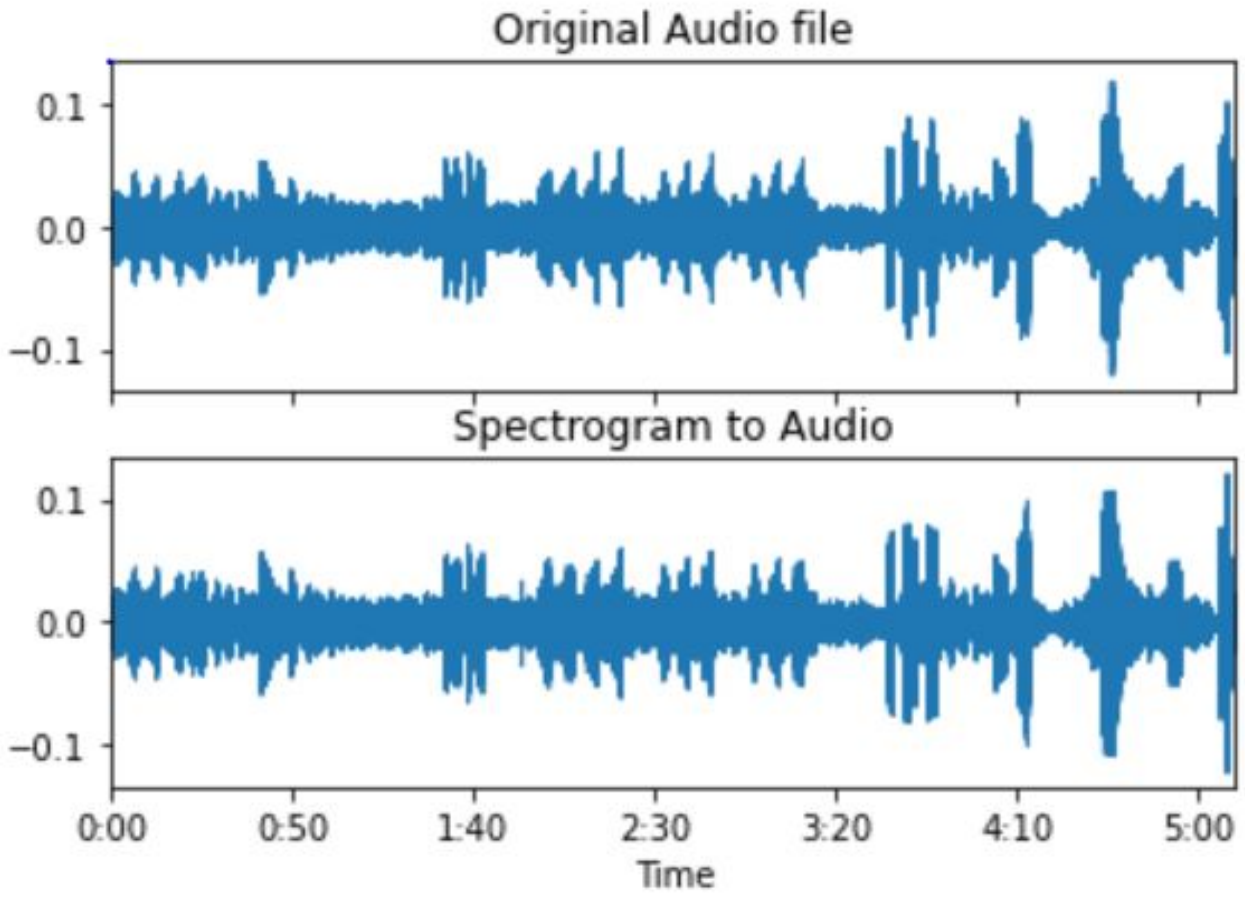


SPECTROGRAM



- Describes contributions of frequency bins across time
- "Time/frequency" representation

Spectrogram to Audio



- Easy Conversion using Librosa library

- Converted Audio file from spectrogram sounded the same as original audio file

Data Pre-processing - Hannah

- Segmentation to 15 second wav files with 5 seconds of overlap
  - Removed files less than 15 seconds (outliers)
- Train, Test, Input, Output datasets divided as wav files using FluidSynth
  - Before conversion to numpy array to preserve sound quality
- Wav files converted to spectrogram with Fourier transform, and compressed to melspectrograms
- Spectrogram arrays stacked as (1275, 128, 646) - train, (158, 128, 646) - test dataset
- Data Reshaping – flattened to 1D arrays for Linear Regression and Random Forest, expanded to 4D array for CNN – train: (1275, 128, 646, 1), test: (158, 128, 646, 1)

Baseline Models

1. Linear Regression

MAE of 9.34
MSE of 167

```
from sklearn.linear_model import LinearRegression

clf = LinearRegression()

clf.fit(x_train_1d, y_train_1d)
y_pred = clf.predict(x_test_1d)
```

```
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

print("MAE:",mean_absolute_error(y_test_1d, y_pred))
print("MSE:",mean_squared_error(y_test_1d, y_pred))
```

```
MAE: 9.343149
MSE: 167.1632
```

Baseline Models

2. Random Forest

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=2, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=0, verbose=0, warm_start=False)
```

```
regr.score(X_spec_, y_spec_)
```

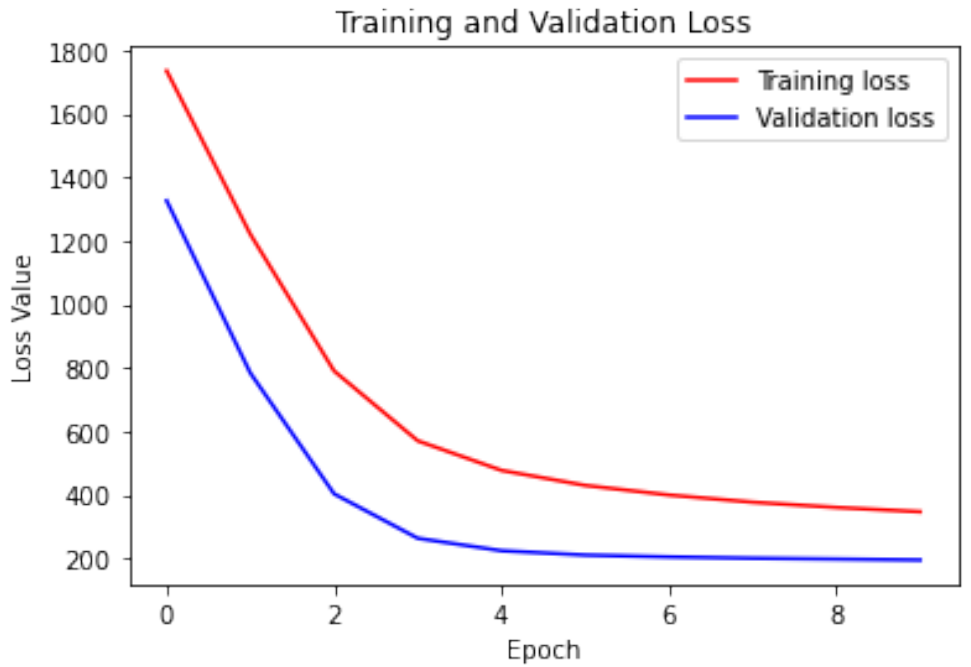
```
0.3967978758216605
```

CNN Models

1. CNN Model One

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 128, 646, 1)	10
max_pooling2d_1 (MaxPooling2	(None, 64, 323, 1)	0
dropout_1 (Dropout)	(None, 64, 323, 1)	0
conv2d_transpose_1 (Conv2DTr	(None, 128, 646, 1)	17
leaky_re_lu_1 (LeakyReLU)	(None, 128, 646, 1)	0
dense_1 (Dense)	(None, 128, 646, 1)	2

Total params: 29
Trainable params: 29
Non-trainable params: 0

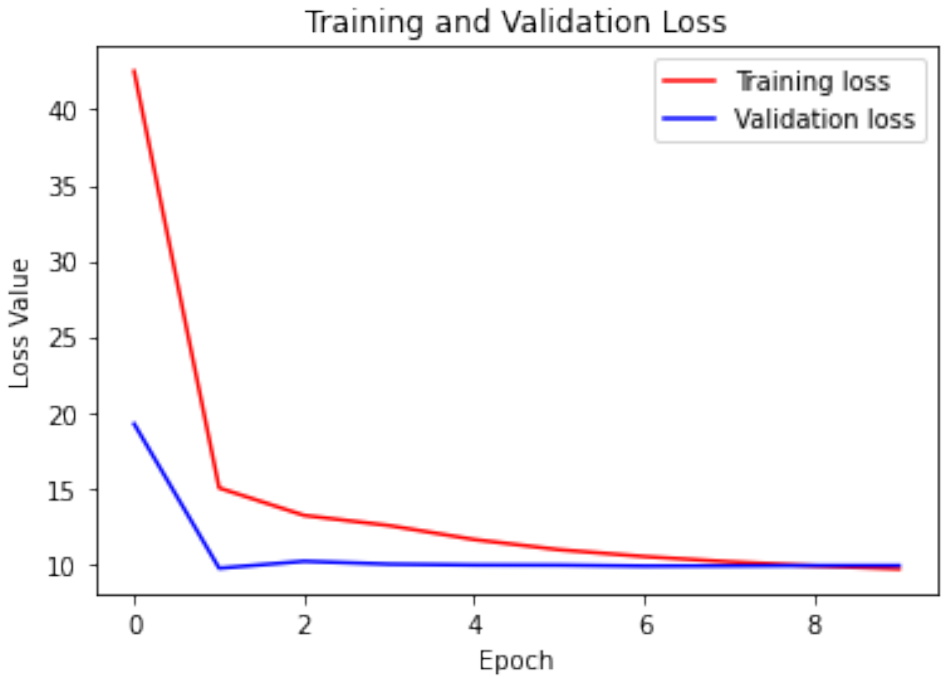


CNN Models

2. CNN Model Two

Layer (type)	Output Shape	Param #
conv2d_62 (Conv2D)	(None, 128, 646, 1)	10
conv2d_63 (Conv2D)	(None, 128, 646, 5)	50
conv2d_transpose_41 (Conv2DT	(None, 256, 1292, 2)	162
leaky_re_lu_45 (LeakyReLU)	(None, 256, 1292, 2)	0
conv2d_64 (Conv2D)	(None, 256, 1292, 5)	15
max_pooling2d_38 (MaxPooling	(None, 256, 1292, 5)	0
conv2d_65 (Conv2D)	(None, 256, 1292, 5)	230
max_pooling2d_39 (MaxPooling	(None, 128, 646, 5)	0
conv2d_transpose_42 (Conv2DT	(None, 256, 1292, 2)	162
leaky_re_lu_46 (LeakyReLU)	(None, 256, 1292, 2)	0
conv2d_66 (Conv2D)	(None, 128, 646, 5)	165
leaky_re_lu_47 (LeakyReLU)	(None, 128, 646, 5)	0
dropout_16 (Dropout)	(None, 128, 646, 5)	0
dense_16 (Dense)	(None, 128, 646, 1)	6

Total params: 800
Trainable params: 800
Non-trainable params: 0

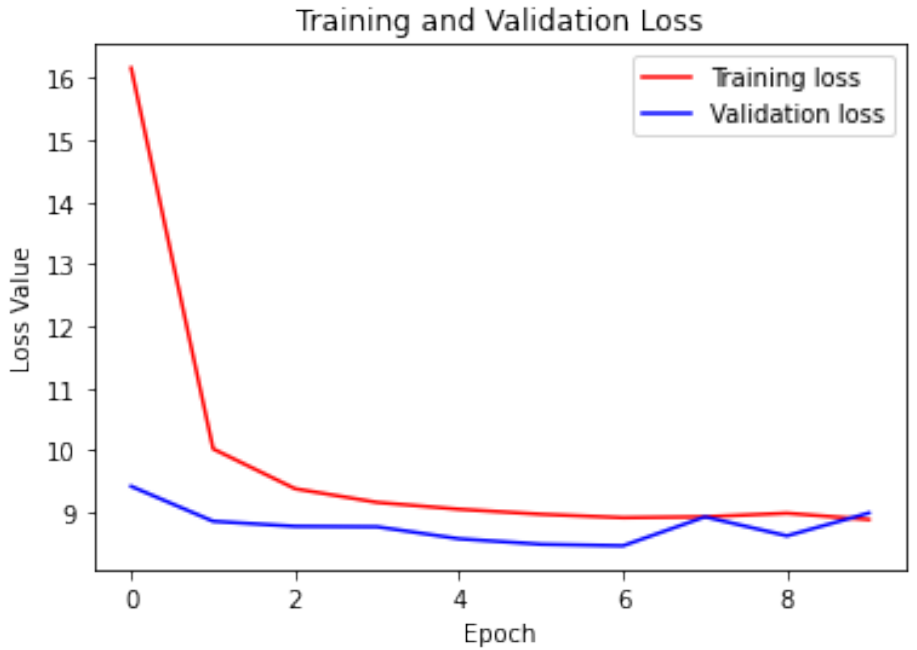


CNN Models

3. CNN Model Three

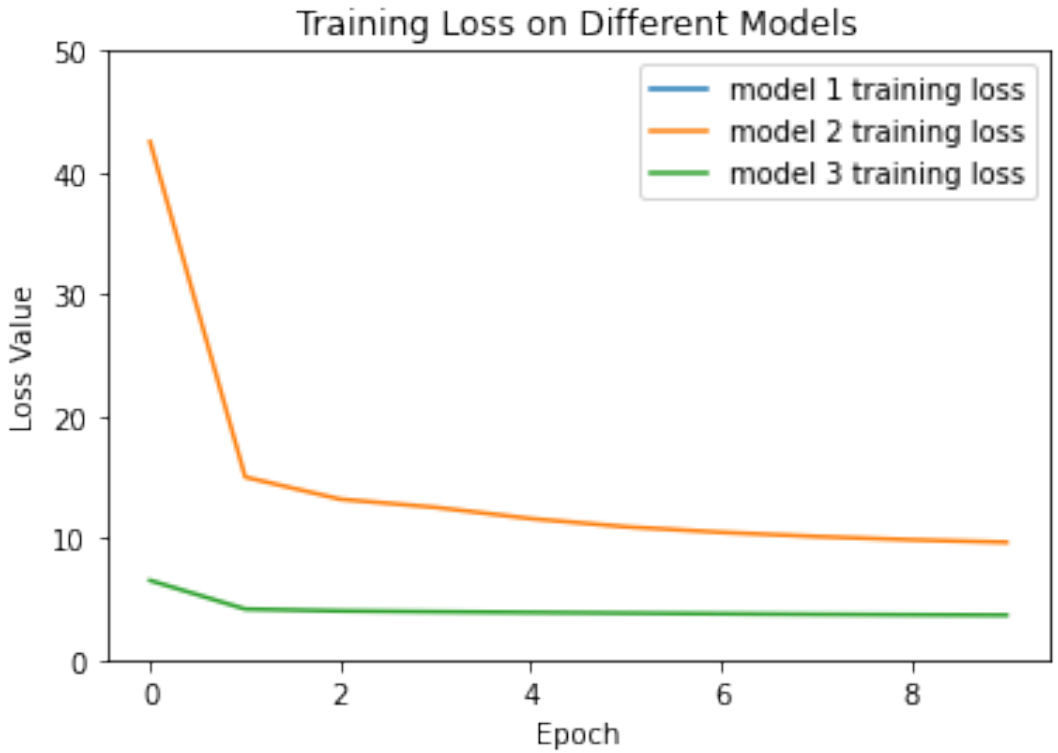
Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 128, 646, 25)	250
conv2d_19 (Conv2D)	(None, 128, 646, 40)	9040
max_pooling2d_8 (MaxPooling2	(None, 64, 323, 40)	0
conv2d_transpose_8 (Conv2DTr	(None, 128, 646, 20)	12820
leaky_re_lu_11 (LeakyReLU)	(None, 128, 646, 20)	0
conv2d_20 (Conv2D)	(None, 128, 646, 20)	3620
max_pooling2d_9 (MaxPooling2	(None, 64, 323, 20)	0
conv2d_transpose_9 (Conv2DTr	(None, 128, 646, 40)	12840
leaky_re_lu_12 (LeakyReLU)	(None, 128, 646, 40)	0
dropout_4 (Dropout)	(None, 128, 646, 40)	0
dense_4 (Dense)	(None, 128, 646, 1)	41

Total params: 38,611
Trainable params: 38,611
Non-trainable params: 0



CNN Model Comparison

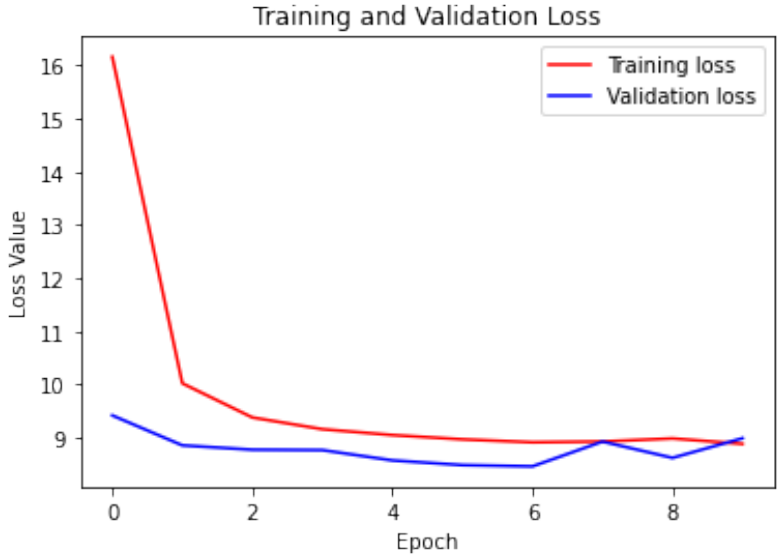
Performance of different CNN models

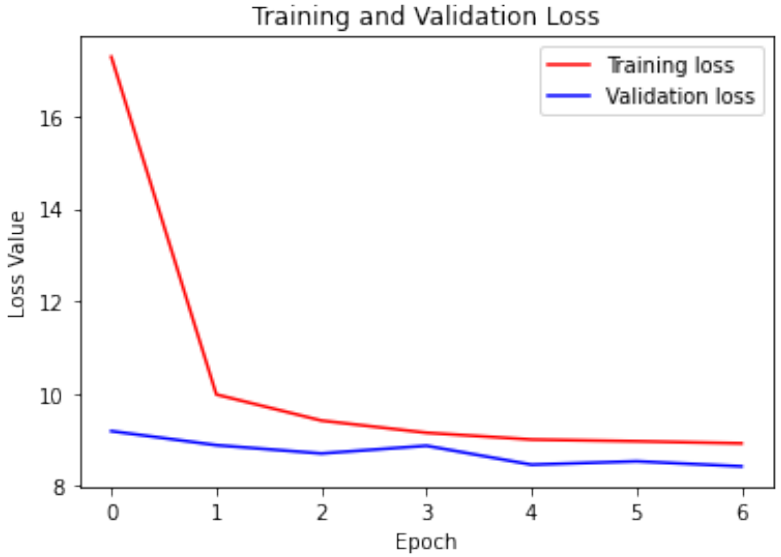


CNN Model Three

Early stop at epoch 7:
Validation MAE of 8.42, MSE of 146

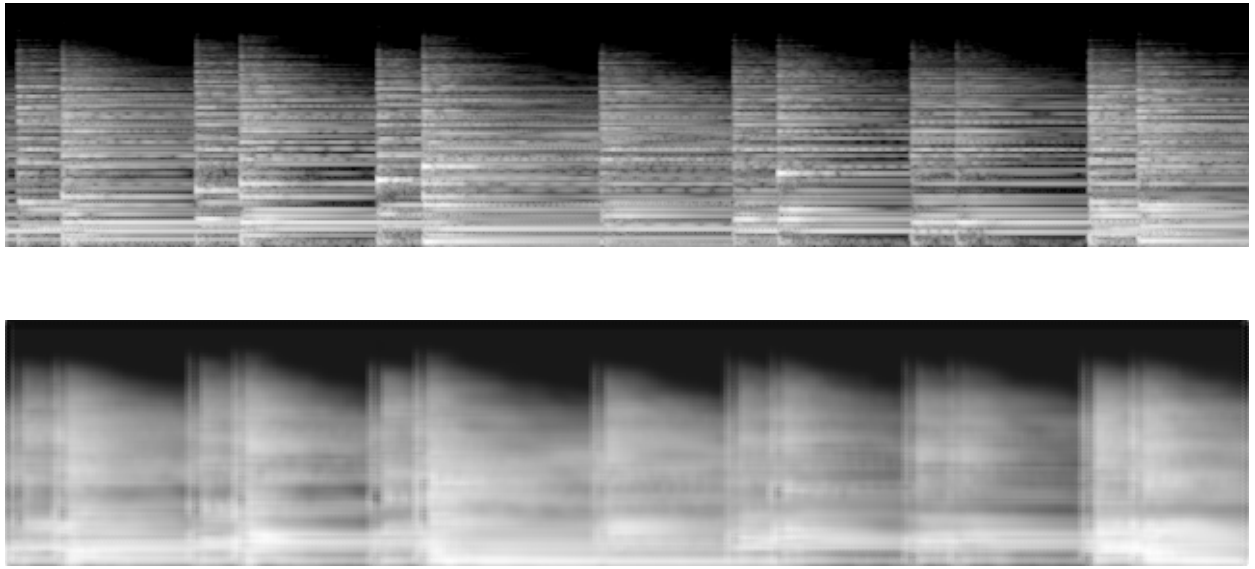
```
Epoch 1/7
40/40 [==============================] - 194s 5s/step - loss: 1139.8225 - mae: 26.2944 - val_loss: 172.5020 - val_mae: 9.1900
Epoch 2/7
40/40 [==============================] - 194s 5s/step - loss: 175.0071 - mae: 10.1181 - val_loss: 154.2593 - val_mae: 8.8856
Epoch 3/7
40/40 [==============================] - 194s 5s/step - loss: 152.9021 - mae: 9.5242 - val_loss: 149.8727 - val_mae: 8.7059
Epoch 4/7
40/40 [==============================] - 195s 5s/step - loss: 147.3443 - mae: 9.2866 - val_loss: 151.8076 - val_mae: 8.8730
Epoch 5/7
40/40 [==============================] - 195s 5s/step - loss: 136.5246 - mae: 8.9639 - val_loss: 146.4015 - val_mae: 8.4636
Epoch 6/7
40/40 [==============================] - 195s 5s/step - loss: 134.4252 - mae: 8.8763 - val_loss: 146.4263 - val_mae: 8.5363
Epoch 7/7
40/40 [==============================] - 195s 5s/step - loss: 135.7883 - mae: 8.9157 - val_loss: 146.1080 - val_mae: 8.4267
```



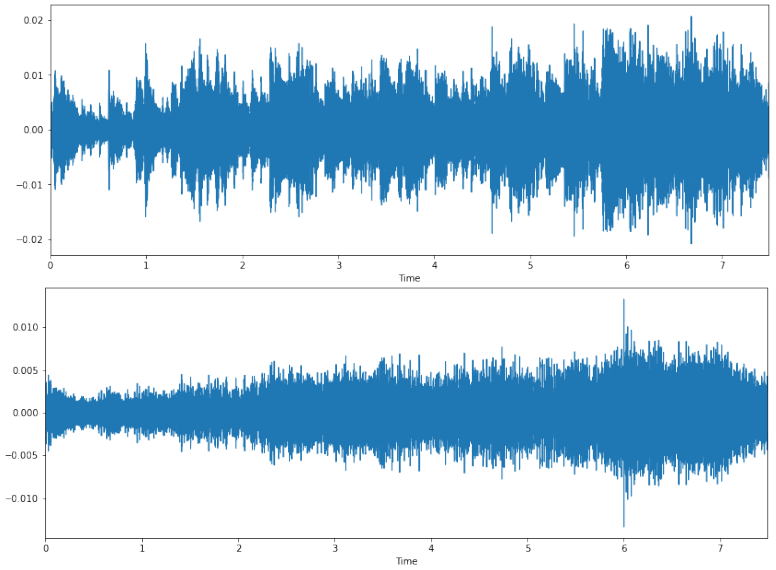


CNN Model Evaluation

Spectograms of targeted (top) and predicted (bottom) spectograms



Waveplots of targeted (top) and predicted (bottom) spectograms

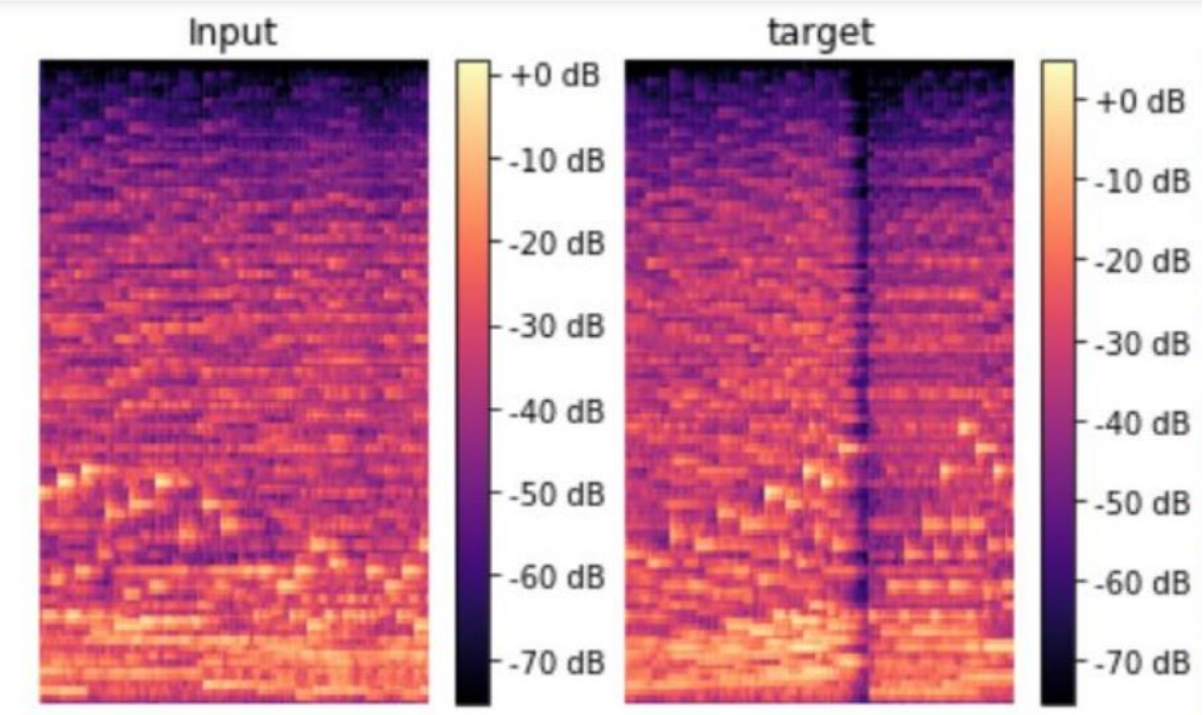


Bethold: Pickled png files to numpy/ Normalization

Train/ Target Data Set: (2, 1288, 256, 128) - 5 Seconds

Test Data Set: (2, 144, 256, 128) - 5 Seconds

Sample train and Target ->



Auto Encoders

Model 1(fewer params), Model 2(More params)

Model_1 (4 hidden layers)

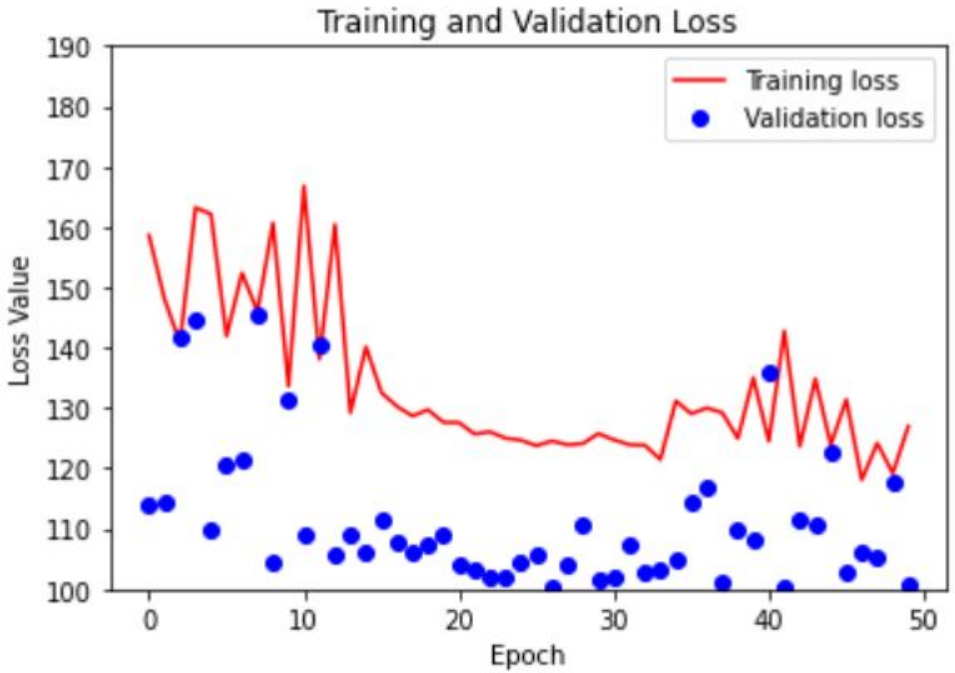
```
Model: "autoencoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
img (InputLayer)             [(None, 440, 128, 1)]     0
_________________________________________________________________
encoder (Functional)         (None, 16)                18672
_________________________________________________________________
decoder (Functional)         (None, 440, 128, 1)       31233
=================================================================
Total params: 49,905
Trainable params: 49,905
Non-trainable params: 0
_________________________________________________________________
```

Model_2 (5 hidden layers)

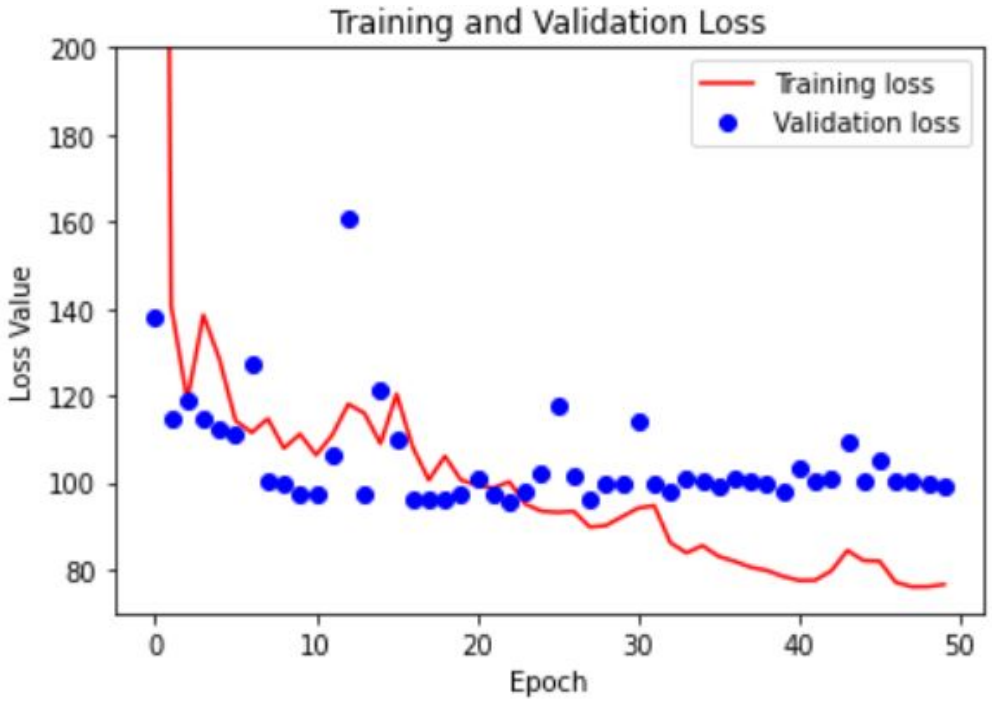
```
Model: "autoencoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
img (InputLayer)             [(None, 300, 128, 1)]     0
_________________________________________________________________
encoder (Functional)         (None, 256)               535424
_________________________________________________________________
decoder (Functional)         (None, 300, 128, 1)       15336321
=================================================================
Total params: 15,871,745
Trainable params: 15,871,745
Non-trainable params: 0
_________________________________________________________________
```

MSE LOSSES

Model_1 Results

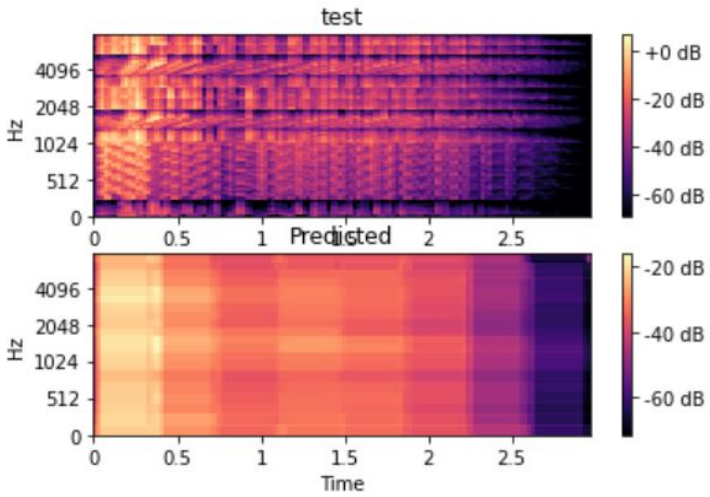


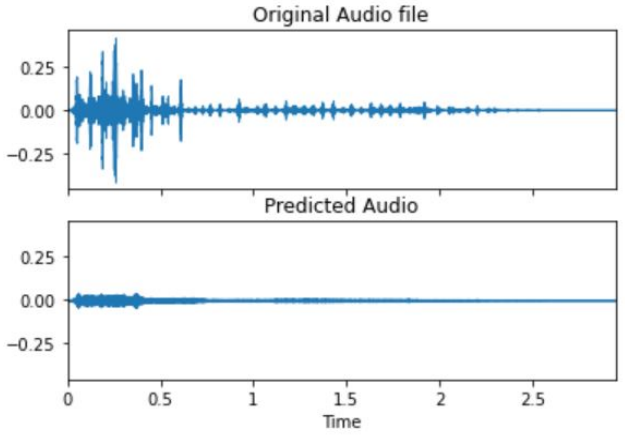
Model_ 2 Results



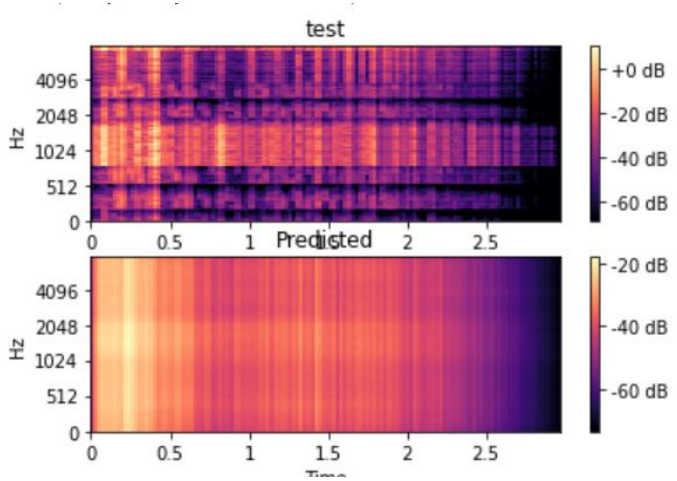
AUTOENCODER SPECTROGRAM AND AUDIO RESULTS

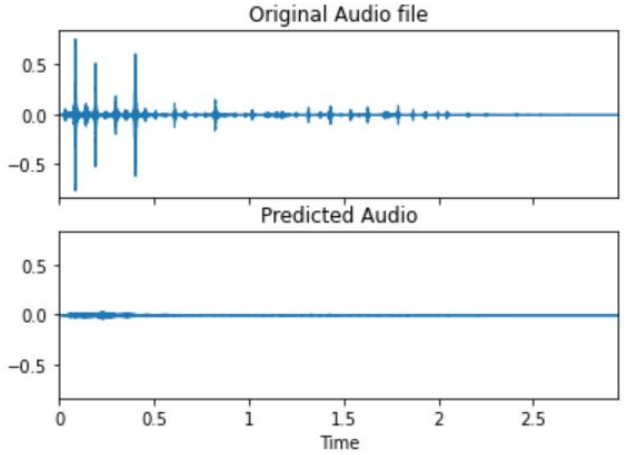
Model 1 Spectrogram and Audio





Model 2 Spectrogram and Audio





DCGAN ARCHITECTURE

Generator Architecture

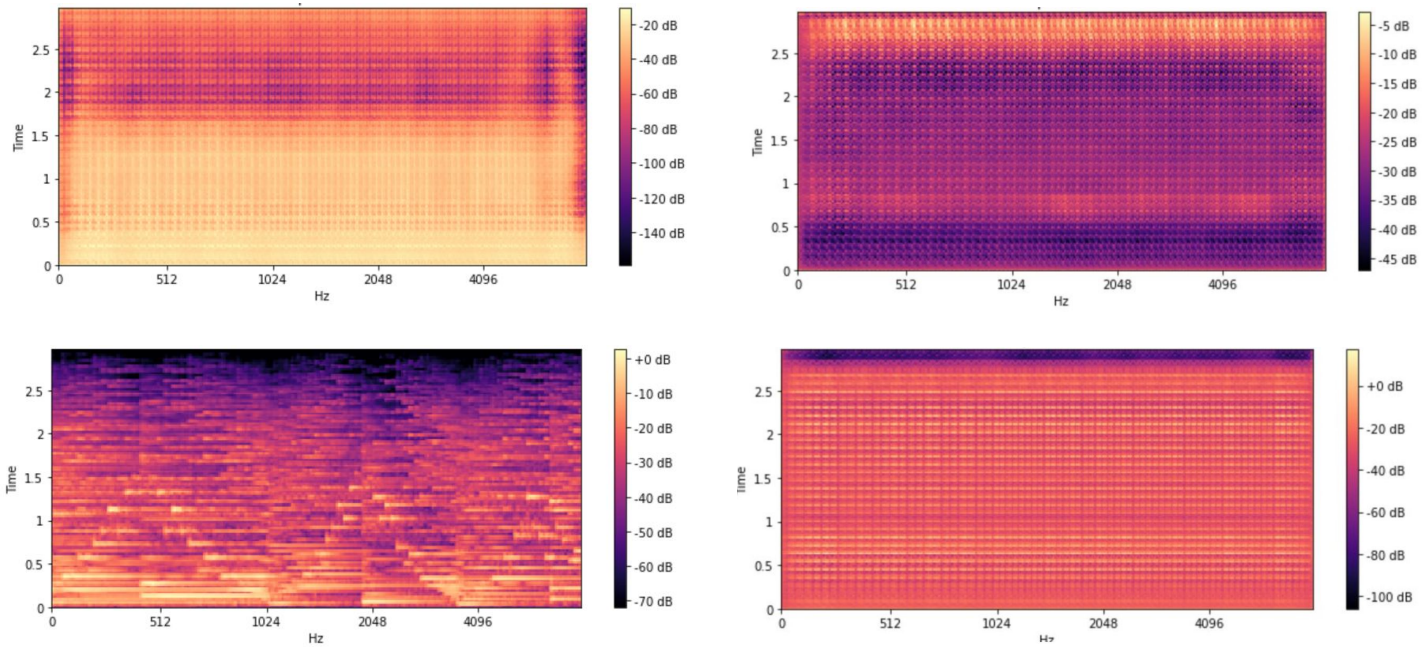
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 55296)             5584896
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 55296)             0
_________________________________________________________________
reshape (Reshape)            (None, 18, 12, 256)       0
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 37, 14, 256)       590080
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 37, 14, 256)       0
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 75, 16, 128)       295040
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 75, 16, 128)       0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 75, 32, 128)       147584
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 75, 32, 128)       0


_________________________________________________________________
conv2d_transpose_3 (Conv2DTr (None, 150, 64, 64)       73792
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 150, 64, 64)       0
_________________________________________________________________
conv2d_transpose_4 (Conv2DTr (None, 300, 128, 32)      18464
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 300, 128, 32)      0
_________________________________________________________________
conv2d_transpose_5 (Conv2DTr (None, 300, 128, 1)       289
=================================================================
Total params: 6,710,145
Trainable params: 6,710,145
Non-trainable params: 0
```

Discriminator Architecture

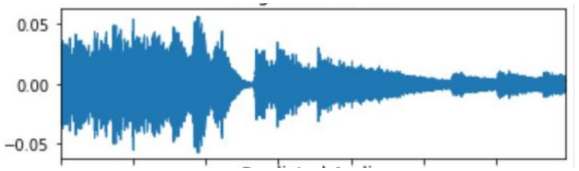
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 150, 64, 64)       640
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 150, 64, 64)       0
_________________________________________________________________
dropout (Dropout)            (None, 150, 64, 64)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 75, 32, 128)       73856
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 75, 32, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 75, 32, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 307200)            0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 307201
=================================================================
Total params: 381,697
Trainable params: 381,697
Non-trainable params: 0
```

DCGAN RESULTS (generator- 6 hidden layers/ Discriminator-2 hidden layers)

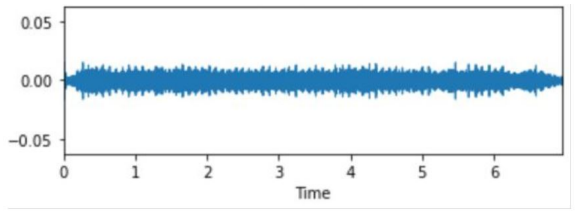
Sample test (bottom left) and Generated Spectrograms



Sample audio



Generated Audio



PIX2PIX MODEL ARCHITECTURE

Generator Architecture

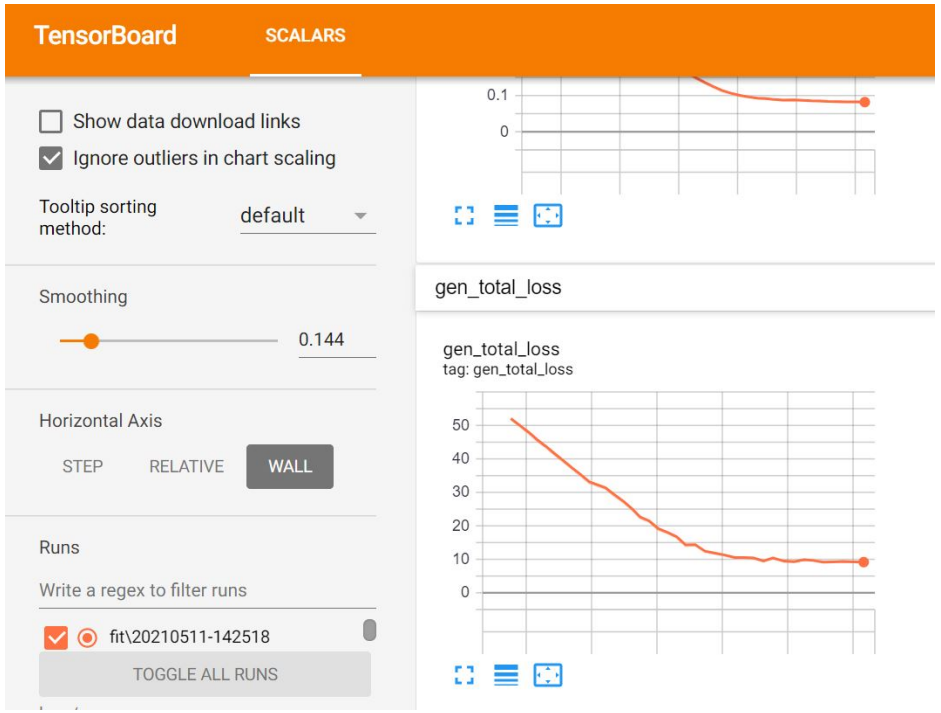
```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 256, 128, 3) 0
__________________________________________________________________________________________________
sequential_8 (Sequential)       (None, 128, 64, 32)  1536        input_2[0][0]
__________________________________________________________________________________________________
sequential_9 (Sequential)       (None, 64, 32, 64)   33024       sequential_8[0][0]
__________________________________________________________________________________________________
sequential_10 (Sequential)      (None, 32, 16, 128)  131584      sequential_9[0][0]
__________________________________________________________________________________________________
sequential_11 (Sequential)      (None, 16, 8, 128)   262656      sequential_10[0][0]
__________________________________________________________________________________________________
sequential_12 (Sequential)      (None, 32, 16, 128)  262656      sequential_11[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 32, 16, 256)  0           sequential_12[0][0]
                                                                 sequential_10[0][0]
__________________________________________________________________________________________________
sequential_13 (Sequential)      (None, 64, 32, 128)  524800      concatenate_3[0][0]
__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 64, 32, 192)  0           sequential_13[0][0]
                                                                 sequential_9[0][0]


__________________________________________________________________________________________________
sequential_14 (Sequential)      (None, 128, 64, 64)  196864      concatenate_4[0][0]
__________________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 128, 64, 96)  0           sequential_14[0][0]
                                                                 sequential_8[0][0]
__________________________________________________________________________________________________
conv2d_transpose_9 (Conv2DTrans (None, 256, 128, 1)  1537        concatenate_5[0][0]
==================================================================================================
Total params: 1,414,657
Trainable params: 1,413,377
Non-trainable params: 1,280
```

Discriminator Architecture

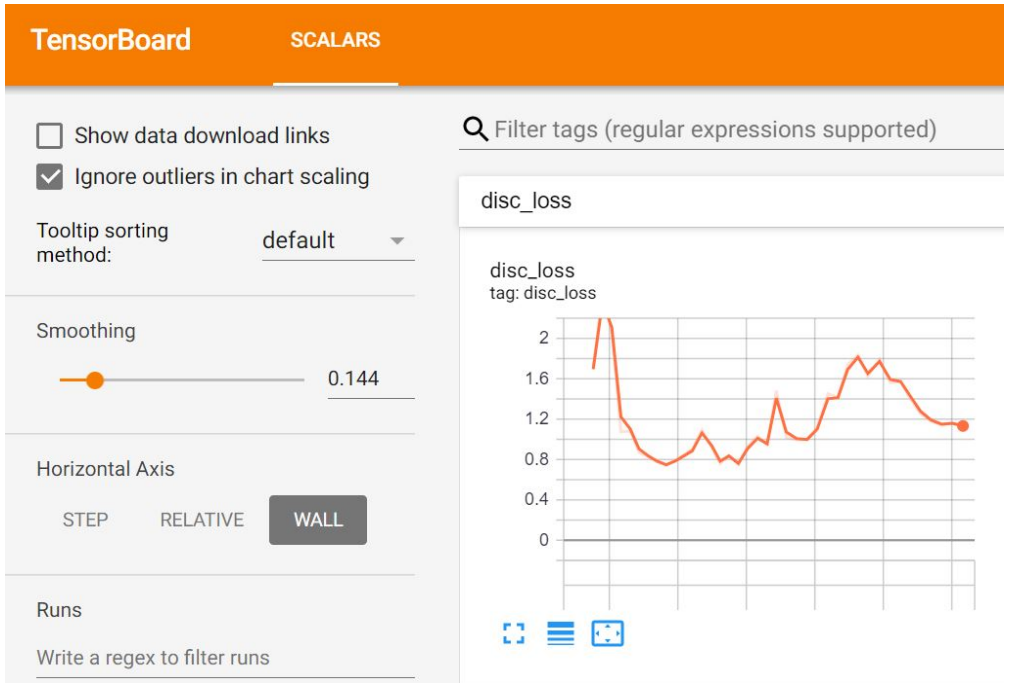
```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_image (InputLayer)        [(None, 256, 128, 3) 0
__________________________________________________________________________________________________
target_image (InputLayer)       [(None, 256, 128, 3) 0
__________________________________________________________________________________________________
concatenate_6 (Concatenate)     (None, 256, 128, 6)  0           input_image[0][0]
                                                                 target_image[0][0]
__________________________________________________________________________________________________
sequential_16 (Sequential)      (None, 128, 64, 64)  6144        concatenate_6[0][0]
__________________________________________________________________________________________________
sequential_17 (Sequential)      (None, 64, 32, 128)  131584      sequential_16[0][0]
__________________________________________________________________________________________________
sequential_18 (Sequential)      (None, 32, 16, 256)  525312      sequential_17[0][0]
__________________________________________________________________________________________________
zero_padding2d (ZeroPadding2D)  (None, 34, 18, 256)  0           sequential_18[0][0]
__________________________________________________________________________________________________
conv2d_11 (Conv2D)              (None, 31, 15, 512)  2097152     zero_padding2d[0][0]
__________________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 31, 15, 512)  2048        conv2d_11[0][0]


__________________________________________________________________________________________________
leaky_re_lu_11 (LeakyReLU)      (None, 31, 15, 512)  0           batch_normalization_16[0][0]
__________________________________________________________________________________________________
zero_padding2d_1 (ZeroPadding2D (None, 33, 17, 512)  0           leaky_re_lu_11[0][0]
__________________________________________________________________________________________________
conv2d_12 (Conv2D)              (None, 30, 14, 1)    8193        zero_padding2d_1[0][0]
==================================================================================================
Total params: 2,770,433
Trainable params: 2,768,641
Non-trainable params: 1,792
```

PIX2PIX SCALAR LOSS WITH TENSORBOARD

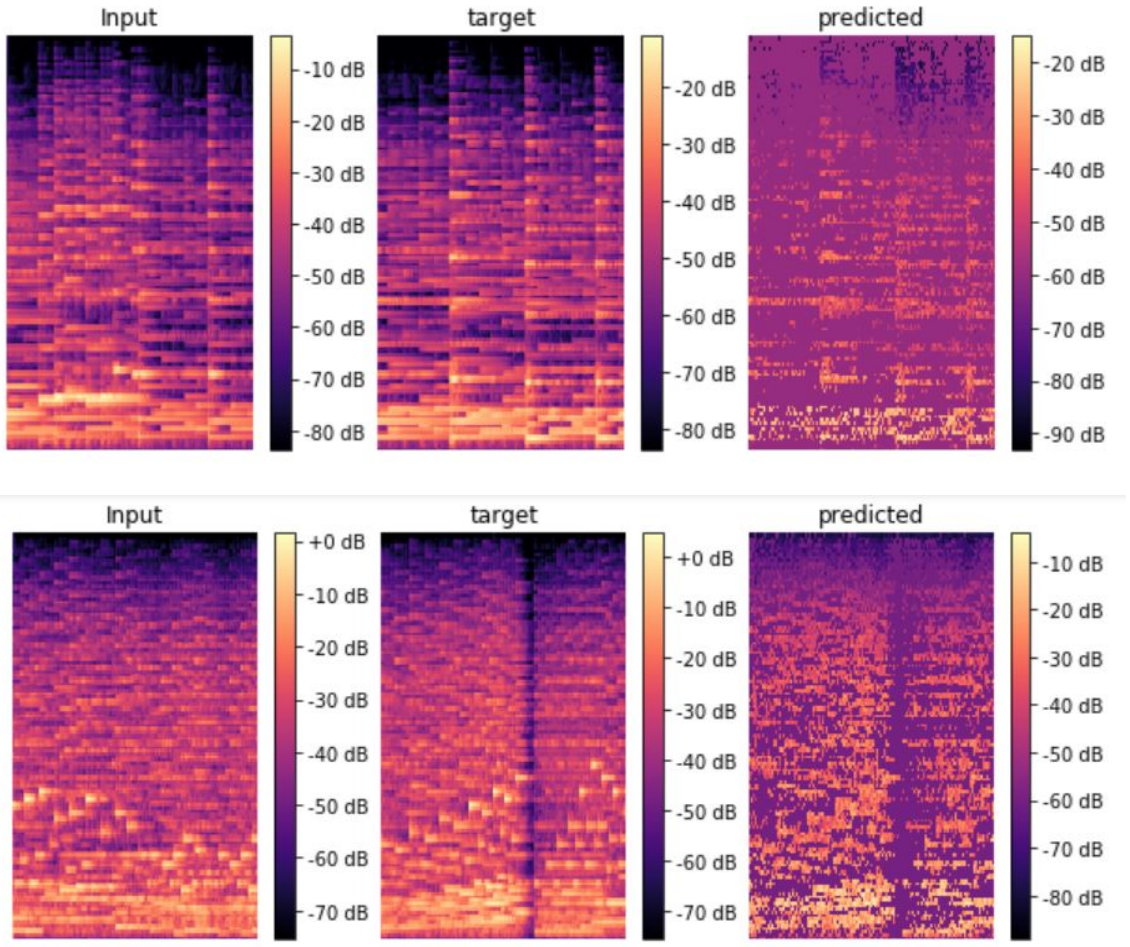
Generator Loss (Binary Cross Entropy with lambda constant

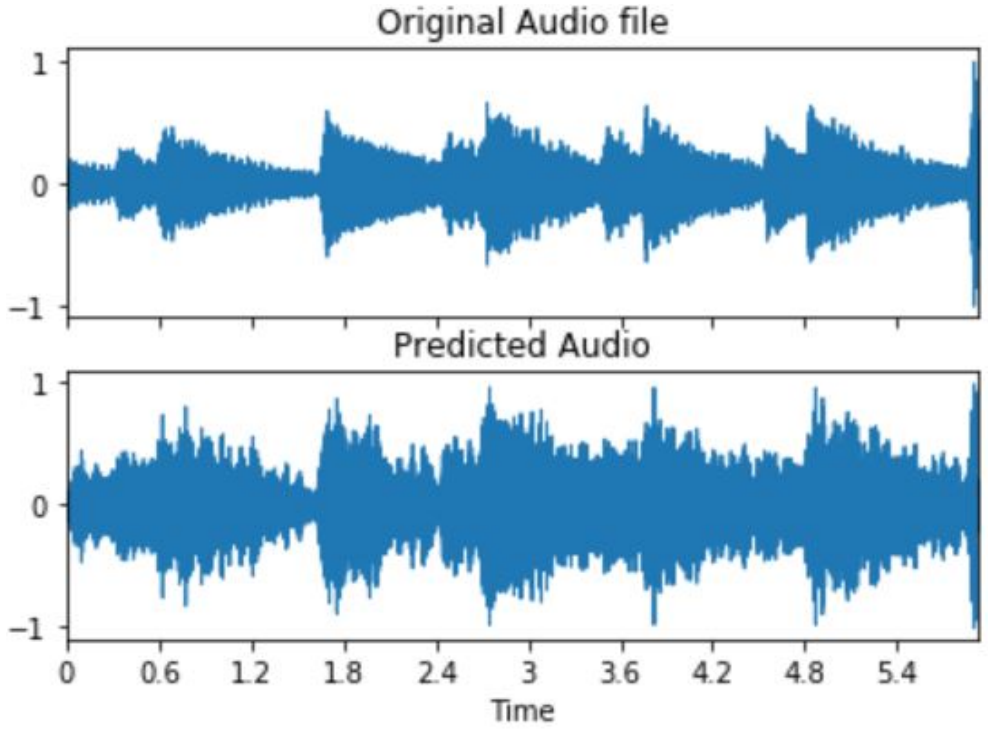


Discriminator Loss (Binary Cross Entropy)

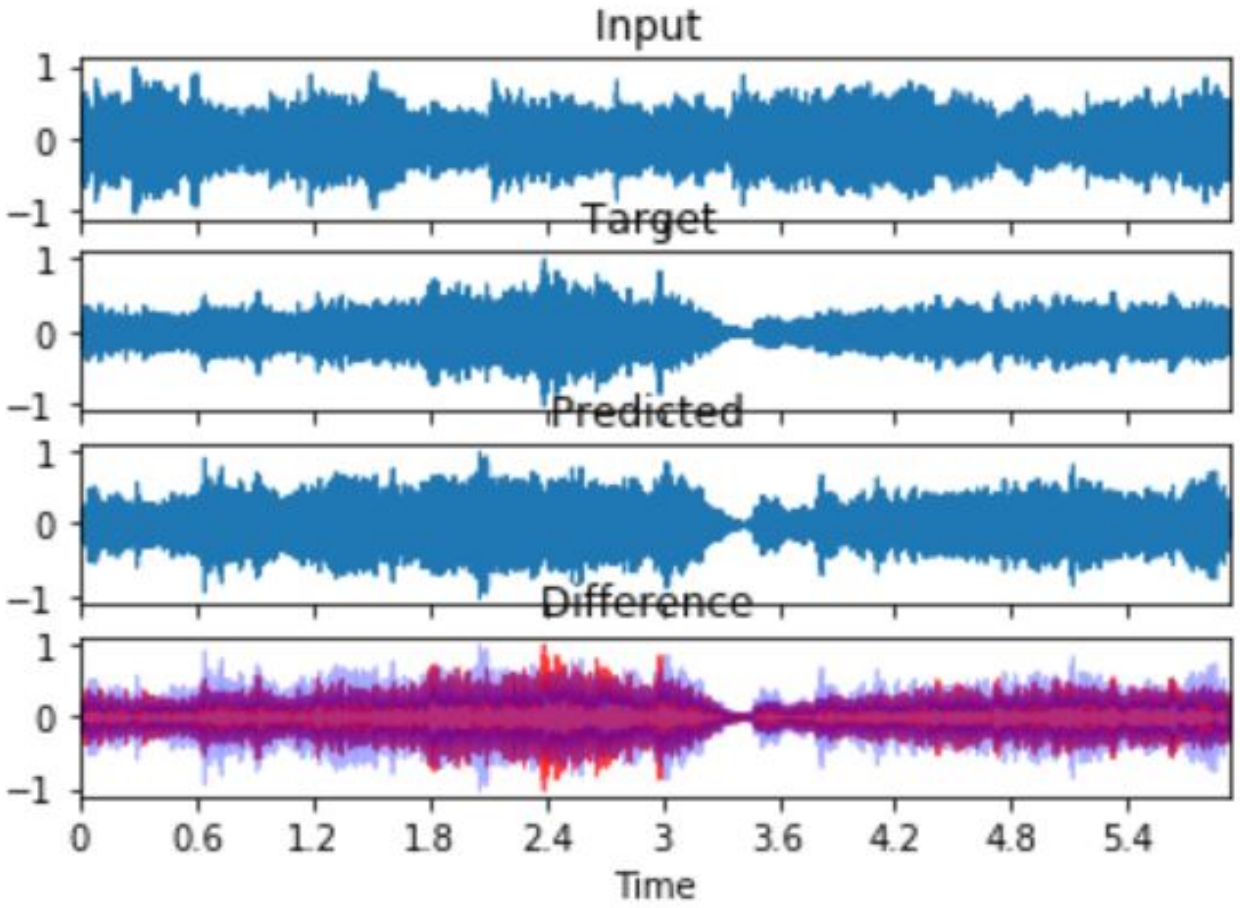


PIX2PIX SPECTROGRAMS AND AUDIO





PIX2PIX AUDIO DEMO



	Sample 1	Sample 2
Input	Input 1	Input 2
Target	Target 1	Target 2
Predicted	Predicted 1	Predicted 2

STANDBY AND ENJOY

CONCLUSION AND FUTURE WORK

- CNN Models
  - The final model preserves the rhythm of a sequence, however the keys being played are difficult to hear

- Auto Encoders Models
  - These models generalizes the pattern of the song but did not preserve the rhythm and keys being played

- DCGAN Model
  - Could not be used to regenerate music piece because of the uniqueness in each spectrogram

- PIX2PIX Model
  - A pixel to pixel image transformation did preserve the rhythm and keys being played in a song
  - Allowed the model to successfully learn and predict target

- Future Work
  - Try different hyperparameters such as learning rate of optimizers and lambda constant of the PIX2PIX model to get even closer to the target sound
-

Accomplished So Far

Bethold

1. Converted Midi files into wav files
2. Converted Spectrograms to images
3. Converted Spectrogram back to wav files
4. Picked spectrogram to numpy array
4. Pre-processed raw data for training
5. Implemented baseline Random Forest
6. Implemented different autoencoders; VAE, and DCGAN.
7. Successfully predicted and regenerated target audio using PIX2PIX model

Hannah

1. Converted Midi files into wav files
2. Converted Spectrograms to image s
3. Segmented wav files into short 15 second files with 5 seconds of overlap, amplifying the dataset to over 2,000 files
4. Performed Baseline model of Linear Regression
5. Implemented three types of CNN model with different regularizations – basic, intermediate, complex
6. Predicted and regenerated audio using Librosa and Pydub

Ethan

1. Converted Midi files into wav files
2. Created preprocessing pipeline to compute log spectrograms and max/min values of each sample, stored in google drive directory.
3. Normalized spectrograms and stored min/max values for all log spectrograms
4. Implemented Variational Autoencoder with 5 convolutional layers and generated 5 audio samples compared against original audio samples.

Contributions