

Machine Learning Project Report and Instructions

Ryan Do

My project is creating an algorithm that is able to detect what instrument is being played. The instruments I have decided to test were violin, sax, trumpet, piano, organ, flute, and clarinet. For the most part it works on simplistic songs with one main instrument being played. Some issues I have encountered are that organ is sometimes hard to differentiate between piano and organ since they share similar qualities. Occasionally the algorithm will be somewhat inaccurate and mess up the instruments if an instrument is being too loud. For example, sometimes a loud flute can be mistaken for a trumpet or a clarinet can be mistaken for a saxophone. Now I will show instructions on how to run the code with instructions and pictures. (a lot of installing extra packages torch torchvision torchaudio librosa scikit-learn matplotlib numpy).

- 1) The code was created in python jupyter notebook so it is best to use jupyter notebook for the algorithm.
- 2) Make sure the directories matchup with how the images display. Will be zipped like that as such too. The Final Project Machine Learning folder should contain the Testing folders, Training folders, and the notebook that runs the code Music Algorithm Final. In the Training folder there should also be a folder of the 7 instruments i used with the training datasets inside total 525 .wav files throughout the 7 folders.

Files	Running	Clusters
-------	---------	----------

Select items to perform actions on them.

Upload New ↻

0	/ Final Project Machine Learning	Name	Last Modified	File size
<input type="checkbox"/>	..		seconds ago	
<input type="checkbox"/>	Testing		12 minutes ago	
<input type="checkbox"/>	Training		23 minutes ago	
<input type="checkbox"/>	Music Alogirithm 2.0.ipynb	Running	an hour ago	12.1 kB
<input type="checkbox"/>	Music Alogirithm Final.ipynb	Running	a minute ago	12.4 kB
<input type="checkbox"/>	TestScript.ipynb	Running	18 hours ago	107 kB
<input type="checkbox"/>	trained_model.pth		9 minutes ago	23.5 MB

0	/ Final Project Machine Learning / Training	Name	Last Modified	File size
<input type="checkbox"/>	..		seconds ago	
<input type="checkbox"/>	clarinet		2 hours ago	
<input type="checkbox"/>	flute		2 hours ago	
<input type="checkbox"/>	organ		2 hours ago	
<input type="checkbox"/>	piano		2 hours ago	
<input type="checkbox"/>	sax		2 hours ago	
<input type="checkbox"/>	trumpet		2 hours ago	
<input type="checkbox"/>	violin		2 hours ago	

- 3) When running the code, in the 2nd code block there will be a `base_path` variable that you will need to change to wherever you save the Training folder. Created a check to see if the folder contains the Training and Testing folders

```
: instrument_to_label = {
    "violin": 0,
    "sax": 1,
    "trumpet": 2,
    "piano": 3,
    "organ": 4,
    "flute": 5,
    "clarinet": 6
}

# when running this make sure to change location of base path and have a Training and Testing Folder
base_path = os.path.dirname(os.path.abspath(r"C:\Users\ryan3\Final Project Machine Learning\Training")) #current path of folder
train_path = os.path.join(base_path, "Training")
test_path = os.path.join(base_path, "Testing")

# checking if the folder pathing is same as mine (will provide picture)
if not os.path.isdir(train_path) or not os.path.isdir(test_path):
    raise FileNotFoundError(f"Expected 'Training' and 'Testing' folders in {base_path}.")

X_train = []
y_train = []
```

- 4) In the very last code block there will be a `test_file` variable that just needs the file name of the test data you input. Make sure the `test_file` is in the Testing folder and it should be able to work. Also make sure it is in the .wav formatting.

```
In [64]: test_file = os.path.join(test_path, "[flu][cla]0377__3.wav") # only need audio file name here make sure it is .wav format
y, sr = librosa.load(test_file, sr=16000)
mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128, fmax=8000)
mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)

# normalizing and resizing the test data
mel_spec_db = (mel_spec_db - np.mean(mel_spec_db)) / np.std(mel_spec_db)
mel_spec_db = librosa.util.fix_length(mel_spec_db, size=max_time_steps, axis=1)
X_test = torch.tensor(mel_spec_db).unsqueeze(0).unsqueeze(0).float().to(device)

with torch.no_grad():
    prediction = model(X_test)
    predicted_class = torch.argmax(prediction, dim=1).item()

label_to_instrument = {v: k for k, v in instrument_to_label.items()}
predicted_instrument = label_to_instrument.get(predicted_class, "Unknown")
print(f"Predicted instrument: {predicted_instrument}")

Predicted instrument: flute
```

Hopefully everything should work as intended. The algorithm works best when the instrument is the main vocal point and is clear. Something I could have implemented in the future is adding a lot more instruments and also being able to detect all the instruments being played in the audio as right now it can only detect a limited number of instruments and one at a time.