

Problems we met

1. We found that there is an error about data after MFCC processing. This instruction cannot be executed.

```
splits = splitter.split(scaled_feature_vectors, classes_num)
```

Then we found the reason is that the dimensions of feature_vectors and classes_num are different. They are 4476 and 4477. Because there is an empty file and it cannot be processing with librosa.

```
[3] path='/content/Strings'

files = []
for root, dirnames, filenames in os.walk(path):
    for filename in fnmatch.filter(filenames, '*.mp3'):
        files.append(os.path.join(root, filename))

print("found %d audio files in %s"%(len(files),path))

found 4477 audio files in /content/Strings
```

```
[7] def get_features(y, sr=fs):
    S = librosa.feature.melspectrogram(y, sr=fs, n_mels=n_mels)
    mfcc = librosa.feature.mfcc(S=librosa.power_to_db(S), n_mfcc=n_mfcc)
    feature_vector = np.mean(mfcc, axis = 1)
    return feature_vector

feature_vectors = []
sound_paths = []
for i,f in enumerate(files):
    try:
        y, sr = librosa.load(f, sr=fs)
        y/=y.max() #Normalize
        if len(y) < 2:
            print("Error loading %s" % f)
            continue
        feat = get_features(y, sr)
        feature_vectors.append(feat)
        sound_paths.append(f)
    except Exception as e:
        print("Error loading %s. Error: %s" % (f,e))

print("Calculated %d feature vectors"%len(feature_vectors))

Error loading /content/Strings/viola/viola_D6_05_piano_arco-normal.mp3. Error:
Calculated 4476 feature vectors
```

 viola_D6_05_fortissimo_arco-normal.mp3	785 2
 viola_D6_05_pianissimo_arco-normal.mp3	166 5
 viola_D6_05_piano_arco-normal.mp3	
 viola_D6_15_forte_arco-normal.mp3	554 4
 viola_D6_025_forte_arco-normal.mp3	551 4

To fix this problem, we deleted this empty file. Then it worked.

```
[9] path='/content/Strings'

files = []
for root, dirnames, filenames in os.walk(path):
    for filename in fnmatch.filter(filenames, '*.mp3'):
        files.append(os.path.join(root, filename))

print("found %d audio files in %s"%(len(files),path))

found 4476 audio files in /content/Strings
```

2. In the model we used at first, the val_accuracy is only around 80%. So we tried to use models with more layers. Then we met another problem. The data we have cannot fit the model layers' dimensions.

```
from tensorflow.keras import layers
model = Sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=Xtr.shape[1:]))
model.add(layers.MaxPool2D(2,2))
model.add(layers.Conv2D(32, (3,3), activation='relu'))
model.add(layers.MaxPool2D(2,2))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(34, activation='relu'))
model.add(layers.Dense(NUM_LABELS))

model.summary()
```

```
ValueError                                Traceback (most recent call last)
<ipython-input-23-4daa084b1099> in <module>()
      1 from tensorflow.keras import layers
      2 model = Sequential()
----> 3 model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=Xtr.shape[1:]))
      4 model.add(layers.MaxPool2D(2,2))
      5 model.add(layers.Conv2D(32, (3,3), activation='relu'))

4 frames
/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/input_spec.py in assert_input_compatibility(input_spec, inputs, layer)
    194         ', found ndim=' + str(ndim) +
    195         '. Full shape received: ' +
--> 196         str(x.shape.as_list()))
    197     # Check dtype.
    198     if spec.dtype is not None:

ValueError: Input 0 of layer conv2d is incompatible with the layer: expected min_ndim=4, found ndim=2. Full shape received: [None, 13]
```