Trying to:

Convert Google's YAMNet to a .mlmodel file using coremltools

First attempt:

1. Downloaded yamnet model from Kaggle in TensorFlow 2 format (from here)
2. Called coremltools.convert on the model
3. Then call mlmodel.save with the desired directory

Example code:

```
import coremltools as ct

# Path to the directory containing saved_model.pb, assets/, variables/
saved_model_dir = "./saved_models/yamnet"

# Tell coremltools it's a TensorFlow SavedModel
mlmodel = ct.convert(
  saved_model_dir,
  source="tensorflow",
)

# Save out the .mlmodel file
mlmodel.save("./saved_models/YAMNet.mlmodel")
```

This process did not work and ended up with NotImplementedError for operation ComplexAbs, which is likely due to YAMNet having custom operations that are not known by coremltools.

Second attempt:

1. Try to extract just the frame component from the "frames only" model of YAMNet
2. Extract the embeddings layer of the model and convert to a keras model
3. Attempt to convert that new keras model

Code Example:

```
#    Convert *only* that embeddings-only model (or your combined model)…
#    giving Core ML an explicit, concrete input shape:
mlmodel = ct.convert(
    embeddings_model,                       # or model_to_convert
    convert_to="mlprogram",                 # iOS 15+ targets
    minimum_deployment_target=ct.target.iOS15,
    inputs=[ ct.TensorType(
        name=embeddings_model.input_names[0],        # e.g. "input_1"
```

```
        shape=(1, params.patch_frames, params.mel_bands)
    )]
)
mlmodel.save("YAMNet_EmbeddingsOnly.mlmodel")
```

Still gets the same error as first attempt, seems like the unsupported operation is deeper in model than previously thought.

Third Attempt:

1. Use keras2onnx python library (pip install keras2onnx)
2. Convert keras model from previous attempt to onnx then attempt a coremltools.convert on the onnx model

Example code:

```
import tensorflow as tf
import keras2onnx

# Load the full YAMNet model from TF Hub
yamnet_model_handle = 'https://tfhub.dev/google/yamnet/1'

yamnet_model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(15600,), dtype=tf.float32),
    tf.keras.layers.Reshape((15600, 1)),
    tf.keras.layers.Lambda(lambda x: tf.squeeze(x, axis=-1)),  # sometimes
required
    tf.keras.layers.Lambda(lambda x: yamnet(x))  # wrap YAMNet from hub
])



onnx_model = keras2onnx.convert_keras(yamnet_model, yamnet_model.name)
keras2onnx.save_model(onnx_model, "yamnet.onnx")
```

Converting to Onnx still gives a similar unsupported ops error.

Conclusion:

YAMNet's complex nature makes it difficult to convert to .mlmodel format without extensive knowledge of its structure, due to this YAMNet is not a good candidate for this app. Apple's Sound Analysis or a custom trained CreateML model are still possible candidates. Other

candidates could include using PyTorch or TorchAudio models and attempt to convert to .mlmodel format.