```
import librosa
import numpy as np
import os
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, LSTM,
TimeDistributed
from tensorflow.keras.utils import to_categorical

# Feature extraction: MFCCs
def extract_features(file_path, max_pad_len=174):
    try:
        audio, sample_rate = librosa.load(file_path, res_type='kaiser_fast')
        mfccs = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
        pad_width = max_pad_len - mfccs.shape[1]
        mfccs = np.pad(mfccs, pad_width=((0, 0), (0, pad_width)), mode='constant')
        return mfccs
    except Exception as e:
        print("Error:", file_path, e)
        return None

# Example: load dataset
X, y = [], []
dataset_path = "RAVDESS"  # change to your dataset path
for emotion_folder in os.listdir(dataset_path):
    label = int(emotion_folder)  # you'll map labels properly
    for file in os.listdir(os.path.join(dataset_path, emotion_folder)):
        file_path = os.path.join(dataset_path, emotion_folder, file)
        features = extract_features(file_path)
        if features is not None:
            X.append(features)
            y.append(label)

X = np.array(X)
X = X[..., np.newaxis]  # Add channel dimension
y = to_categorical(y)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# CNN model
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(40,174,1)),
    MaxPooling2D((2,2)),
```

```
    Dropout(0.3),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D((2,2)),
    Dropout(0.3),

    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(y.shape[1], activation='softmax')  # number of emotion classes
])

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

# Train
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=30, batch_size=32)
```