```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Flatten
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Dropout

import matplotlib.pyplot as plt
```

```
np.random.seed(42)
tf.random.set_seed(42)
```

```
data_dir = '/content/drive/MyDrive/DATASET/Brain_Tumor_Detection'
```

```
label_mapping = {'no': 0, 'yes': 1}
```

```
import os
images = []
labels = []
for folder in os.listdir(data_dir):
    folder_path = os.path.join(data_dir, folder)
    if os.path.isdir(folder_path):
        label = label_mapping.get(folder)
        if label is not None:
            for file_name in os.listdir(folder_path):
                if file_name.endswith(".jpg"):
                    file_path = os.path.join(folder_path, file_name)
                    image = load_img(file_path, color_mode="grayscale", target_size=(64, 64))
                    image_array = img_to_array(image)
                    images.append(image_array)
                    labels.append(label)
```

```
images = np.array(images)
labels = np.array(labels)
```

```
X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.2, random_state=42)
```

```
X_train = X_train / 255.0
X_test = X_test / 255.0
```

```
num_timesteps = X_train.shape[1]
height = X_train.shape[2]
width = X_train.shape[3]
input_dim = 1
```

```
X_train = X_train.reshape(-1, num_timesteps, height * width)
X_test = X_test.reshape(-1, num_timesteps, height * width)
```

```
model = Sequential()
model.add(LSTM(64, input_shape=(num_timesteps, height * width)))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an `input_shape`/`input_dim` argu
  super().__init__(**kwargs)
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
model.fit(X_train, y_train, epochs=150, batch_size=32, validation_data=(X_test, y_test))
```

```
Epoch 1/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 30ms/step - accuracy: 1.0000 - loss: 3.2098e-06 - val_accuracy: 0.9683 - val_loss: 0.3534
Epoch 2/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 8.7828e-06 - val_accuracy: 0.9683 - val_loss: 0.3615
Epoch 3/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 6.0396e-06 - val_accuracy: 0.9700 - val_loss: 0.3564
Epoch 4/150
```

```
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 48ms/step - accuracy: 1.0000 - loss: 2.5852e-06 - val_accuracy: 0.9700 - val_loss: 0.3575
Epoch 5/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 30ms/step - accuracy: 1.0000 - loss: 3.5603e-05 - val_accuracy: 0.9683 - val_loss: 0.3632
Epoch 6/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 3.7369e-06 - val_accuracy: 0.9683 - val_loss: 0.3677
Epoch 7/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 5.3972e-06 - val_accuracy: 0.9700 - val_loss: 0.3701
Epoch 8/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 3s 31ms/step - accuracy: 1.0000 - loss: 3.4366e-06 - val_accuracy: 0.9683 - val_loss: 0.3648
Epoch 9/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 51ms/step - accuracy: 1.0000 - loss: 1.2046e-05 - val_accuracy: 0.9683 - val_loss: 0.3750
Epoch 10/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 30ms/step - accuracy: 1.0000 - loss: 7.3774e-06 - val_accuracy: 0.9733 - val_loss: 0.3349
Epoch 11/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 5.8996e-06 - val_accuracy: 0.9717 - val_loss: 0.3409
Epoch 12/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 2.6302e-06 - val_accuracy: 0.9700 - val_loss: 0.3460
Epoch 13/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 1.4542e-06 - val_accuracy: 0.9700 - val_loss: 0.3489
Epoch 14/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 29ms/step - accuracy: 1.0000 - loss: 2.1031e-06 - val_accuracy: 0.9667 - val_loss: 0.3539
Epoch 15/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 48ms/step - accuracy: 1.0000 - loss: 2.9673e-06 - val_accuracy: 0.9667 - val_loss: 0.3570
Epoch 16/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 29ms/step - accuracy: 1.0000 - loss: 4.5282e-06 - val_accuracy: 0.9667 - val_loss: 0.3613
Epoch 17/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 30ms/step - accuracy: 1.0000 - loss: 3.9110e-06 - val_accuracy: 0.9667 - val_loss: 0.3730
Epoch 18/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 5.4550e-06 - val_accuracy: 0.9667 - val_loss: 0.3707
Epoch 19/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 3s 29ms/step - accuracy: 1.0000 - loss: 2.3962e-06 - val_accuracy: 0.9683 - val_loss: 0.3660
Epoch 20/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 47ms/step - accuracy: 1.0000 - loss: 3.3487e-06 - val_accuracy: 0.9683 - val_loss: 0.3695
Epoch 21/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 3s 41ms/step - accuracy: 1.0000 - loss: 2.9774e-06 - val_accuracy: 0.9667 - val_loss: 0.3834
Epoch 22/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 1.3362e-06 - val_accuracy: 0.9667 - val_loss: 0.3912
Epoch 23/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 3s 29ms/step - accuracy: 1.0000 - loss: 2.1004e-06 - val_accuracy: 0.9650 - val_loss: 0.3985
Epoch 24/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 2.4720e-06 - val_accuracy: 0.9633 - val_loss: 0.4099
Epoch 25/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 2.1069e-06 - val_accuracy: 0.9633 - val_loss: 0.4134
Epoch 26/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 4s 47ms/step - accuracy: 1.0000 - loss: 1.5061e-06 - val_accuracy: 0.9633 - val_loss: 0.4132
Epoch 27/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 3s 38ms/step - accuracy: 1.0000 - loss: 4.1643e-06 - val_accuracy: 0.9633 - val_loss: 0.4239
Epoch 28/150
75/75 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - accuracy: 1.0000 - loss: 1.3833e-06 - val_accuracy: 0.9633 - val_loss: 0.4234
Epoch 29/150
```

```
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9728 - loss: 0.2170
Test Loss: 0.25713786482810974
Test Accuracy: 0.9666666388511658
```

```
def predict_single_image(image_path):
    image = load_img(image_path, color_mode="grayscale", target_size=(64, 64))
    image_array = img_to_array(image) / 255.0
    image_array = image_array.reshape(1, num_timesteps, height * width)
    prediction = model.predict(image_array)
    return prediction[0][0] > 0.5
```

```
test_directory_path = '/content/drive/MyDrive/DATASET/Brain_Tumor_Detection/pred'

def analyze_images_in_directory(directory_path):
    image_files = [f for f in os.listdir(directory_path) if f.endswith(".jpg")]
    if not image_files:
        print("No images found in the directory.")
    for file_name in image_files:
        file_path = os.path.join(directory_path, file_name)
        try:
            prediction = predict_single_image(file_path)
            print(f"Image: {file_name}, Tumor present: {'Yes' if prediction else 'No'}")
        except Exception as e:
            print(f"Error processing image {file_name}: {e}")
            analyze_images_in_directory(test_directory_path)
```

```
analyze_images_in_directory(test_directory_path)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 288ms/step
Image: pred18.jpg, Tumor present: No
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Image: pred2.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Image: pred24.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
Image: pred22.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
Image: pred13.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
Image: pred20.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Image: pred16.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Image: pred17.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Image: pred15.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Image: pred1.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Image: pred0.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Image: pred23.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Image: pred25.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred12.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Image: pred3.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred28.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Image: pred19.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred11.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
Image: pred14.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred10.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred21.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Image: pred27.jpg, Tumor present: Yes
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
Image: pred26.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Image: pred29.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Image: pred46.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Image: pred41.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Image: pred51.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Image: pred50.jpg, Tumor present: No
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Image: pred33.jpg, Tumor present: No
```