```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases
#https://www.kaggle.com/datasets/emmarex/plantdisease
```

```
# !mkdir -p ~/.kaggle
# !cp kaggle.json ~/.kaggle/
```

```
# !kaggle datasets download -d vbookshelf/rice-leaf-diseases
# # !kaggle datasets download -d emmarex/plantdisease
```

```
import zipfile
zip_ref = zipfile.ZipFile('/content/drive/MyDrive/Crop-Disease/Plant_leaf_diseases_dataset_without_augmentation.zip', 'r')
zip_ref.extractall('/content')
zip_ref.close()
```

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
import os
import pandas as pd
```

```
# Define the image directory and parameters
image_dir = '/content/Plant_leave_diseases_dataset_without_augmentation'
image_size = (64, 64)
batch_size = 32
```

```
# Load the image paths and labels
image_paths = []
labels = []
for class_name in os.listdir(image_dir):
    class_path = os.path.join(image_dir, class_name)
    for img_path in os.listdir(class_path):
        img_path = os.path.join(class_path, img_path)
        image_paths.append(img_path)
        labels.append(class_name)

# Create a dataframe to store the image paths and labels
data = pd.DataFrame({'image_path': image_paths, 'label': labels})
```

```
data.head()
```

	image_path	lab
0	/content/Plant_leave_diseases_dataset_without_...	Tomato__Tomato_Yellow_Leaf_Curl_Viru
1	/content/Plant_leave_diseases_dataset_without_...	Tomato__Tomato_Yellow_Leaf_Curl_Viru
2	/content/Plant_leave_diseases_dataset_without_...	Tomato__Tomato_Yellow_Leaf_Curl_Viru
3	/content/Plant_leave_diseases_dataset_without_...	Tomato__Tomato_Yellow_Leaf_Curl_Viru
4	/content/Plant_leave_diseases_dataset_without_...	Tomato__Tomato_Yellow_Leaf_Curl_Viru

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
generator = ImageDataGenerator(validation_split=0.2)
train = generator.flow_from_dataframe(
    data,
    x_col='image_path',
    y_col='label',
    color_mode="grayscale",
    target_size=(64,64))
```

```
Found 55448 validated image filenames belonging to 39 classes.
```

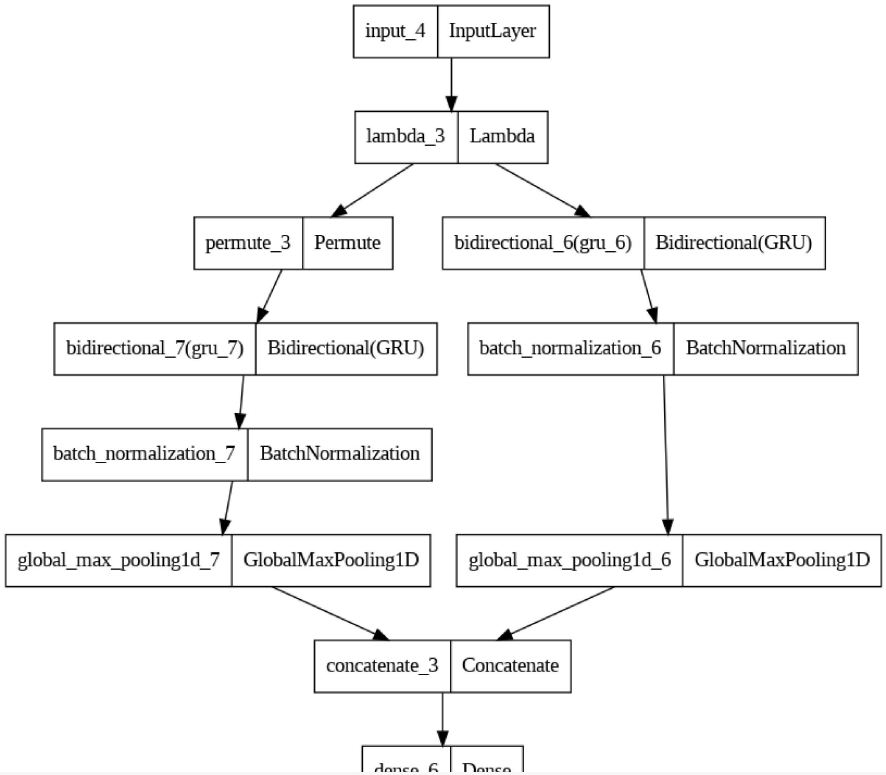
```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input,Bidirectional,LSTM,Lambda, GRU
from tensorflow.keras.layers import Permute,GlobalMaxPool1D,Concatenate, Dense, BatchNormalization, Dropout, GlobalAveragePooling1D
from tensorflow.keras.utils import plot_model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

```
input_ = Input(shape=(64,64,1))
lambda_ = Lambda(lambda x: tf.squeeze(x,axis=3))(input_)

lstm_ = Bidirectional(GRU(8,activation='relu',return_sequences=True))(lambda_)

permute_=Permute((2,1),input_shape=(64,64))(lambda_)
lstm_2 = Bidirectional(GRU(8,activation='relu',return_sequences=True))(permute_)


lstm_ = BatchNormalization()(lstm_)
maxpool1 = GlobalMaxPool1D()(lstm_)


lstm_2 = BatchNormalization()(lstm_2)
maxpool2 = GlobalMaxPool1D()(lstm_2)


concat_ = Concatenate(axis=1)([maxpool1,maxpool2])
dense_1 = Dense(20,activation='relu')(concat_)
output_ = Dense(39,activation='softmax')(dense_1)

model = Model(input_,output_)
```

```
model.summary()
```

```
Model: "model_3"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 input_4 (InputLayer)        [(None, 64, 64, 1)]       0         []

 lambda_3 (Lambda)           (None, 64, 64)            0         ['input_4[0][0]']

 permute_3 (Permute)         (None, 64, 64)            0         ['lambda_3[0][0]']

 bidirectional_6 (Bidirectional  (None, 64, 16)        3552      ['lambda_3[0][0]']
 )

 bidirectional_7 (Bidirectional  (None, 64, 16)        3552      ['permute_3[0][0]']
 )

 batch_normalization_6 (BatchNo  (None, 64, 16)        64        ['bidirectional_6[0][0]']
 rmalization)

 batch_normalization_7 (BatchNo  (None, 64, 16)        64        ['bidirectional_7[0][0]']
 rmalization)

 global_max_pooling1d_6 (Global  (None, 16)            0         ['batch_normalization_6[0][0]']
 MaxPooling1D)

 global_max_pooling1d_7 (Global  (None, 16)            0         ['batch_normalization_7[0][0]']
 MaxPooling1D)

 concatenate_3 (Concatenate)    (None, 32)             0         ['global_max_pooling1d_6[0][0]',
                                                                  'global_max_pooling1d_7[0][0]']

 dense_6 (Dense)             (None, 20)                660       ['concatenate_3[0][0]']

 dense_7 (Dense)             (None, 39)                819       ['dense_6[0][0]']

==================================================================================================
Total params: 8,711
Trainable params: 8,647
Non-trainable params: 64
__________________________________________________________________________________________________
```

```
plot_model(model)
```




```
early = EarlyStopping(patience=4)
reduce_lr = ReduceLROnPlateau(factor=0.1,patience=1)
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
history = model.fit(x=train,epochs=20,callbacks=[early,reduce_lr])
```

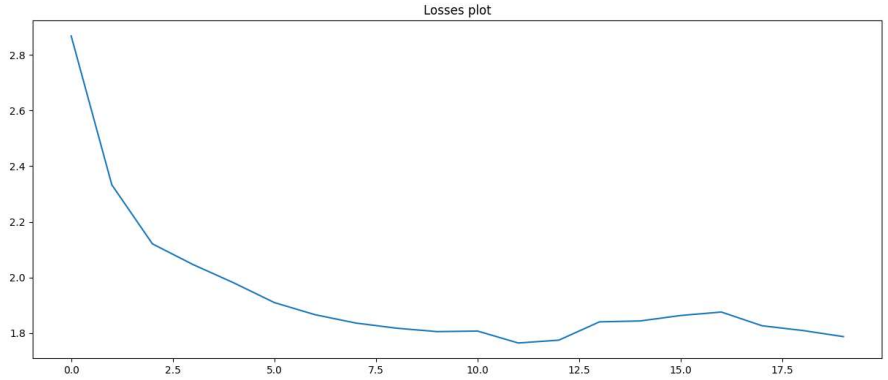
```
Epoch 1/20
1733/1733 [==============================] - ETA: 0s - loss: 2.8688 - accuracy: 0.2436WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 257s 144ms/step - loss: 2.8688 - accuracy: 0.2436 - lr: 0.0010
Epoch 2/20
1733/1733 [==============================] - ETA: 0s - loss: 2.3331 - accuracy: 0.3530WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 249s 144ms/step - loss: 2.3331 - accuracy: 0.3530 - lr: 0.0010
Epoch 3/20
1733/1733 [==============================] - ETA: 0s - loss: 2.1206 - accuracy: 0.4045WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 244s 141ms/step - loss: 2.1206 - accuracy: 0.4045 - lr: 0.0010
Epoch 4/20
1733/1733 [==============================] - ETA: 0s - loss: 2.0461 - accuracy: 0.4194WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 244s 141ms/step - loss: 2.0461 - accuracy: 0.4194 - lr: 0.0010
Epoch 5/20
1733/1733 [==============================] - ETA: 0s - loss: 1.9804 - accuracy: 0.4333WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 261s 151ms/step - loss: 1.9804 - accuracy: 0.4333 - lr: 0.0010
```

```
Epoch 6/20
1733/1733 [==============================] - ETA: 0s - loss: 1.9093 - accuracy: 0.4520WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 245s 142ms/step - loss: 1.9093 - accuracy: 0.4520 - lr: 0.0010
Epoch 7/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8658 - accuracy: 0.4625WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 244s 141ms/step - loss: 1.8658 - accuracy: 0.4625 - lr: 0.0010
Epoch 8/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8358 - accuracy: 0.4691WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 250s 144ms/step - loss: 1.8358 - accuracy: 0.4691 - lr: 0.0010
Epoch 9/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8173 - accuracy: 0.4764WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 250s 144ms/step - loss: 1.8173 - accuracy: 0.4764 - lr: 0.0010
Epoch 10/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8049 - accuracy: 0.4797WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 246s 142ms/step - loss: 1.8049 - accuracy: 0.4797 - lr: 0.0010
Epoch 11/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8069 - accuracy: 0.4799WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 244s 141ms/step - loss: 1.8069 - accuracy: 0.4799 - lr: 0.0010
Epoch 12/20
1733/1733 [==============================] - ETA: 0s - loss: 1.7640 - accuracy: 0.4924WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 249s 144ms/step - loss: 1.7640 - accuracy: 0.4924 - lr: 0.0010
Epoch 13/20
1733/1733 [==============================] - ETA: 0s - loss: 1.7742 - accuracy: 0.4892WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
1733/1733 [==============================] - 244s 141ms/step - loss: 1.7742 - accuracy: 0.4892 - lr: 0.0010
Epoch 14/20
1733/1733 [==============================] - ETA: 0s - loss: 1.8401 - accuracy: 0.4725WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics a
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy,lr
```

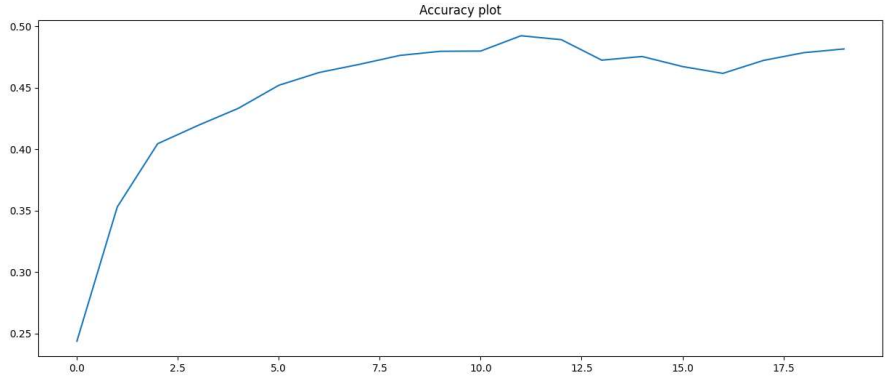
```
model.save('GRUModel.h5')
```

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(15,6))
plt.plot(history.history['loss'])
_=plt.title("Losses plot")
```




```
plt.figure(figsize=(15,6))
plt.plot(history.history['accuracy'])
_=plt.title("Accuracy plot")
```



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.