```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.python.keras.layers import Dense, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

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

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

```
!unzip /content/drive/MyDrive/foods_final1.zip
```

```
  inflating: train/popcorn_peanuts_seeds_related_snacks/9819.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/982.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9822.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9823.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9826.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9829.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/983.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9830.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9833.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9839.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9843.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9845.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/985.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9852.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9855.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/986.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/987.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9871.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9872.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9874.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9875.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9877.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9878.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9880.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9882.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9884.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/989.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9890.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9891.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9892.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9893.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9897.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9899.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/990.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9901.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9904.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9905.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9907.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9908.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/991.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9910.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9916.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/992.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/993.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/995.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9958.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/996.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/997.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9970.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9971.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9973.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9978.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/998.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9980.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9983.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9984.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/999.jpg
  inflating: train/popcorn_peanuts_seeds_related_snacks/9999.jpg
```

```
batch_size = 64
img_height = 180
img_width = 180
```

```
train_ds = tf.keras.utils.image_dataset_from_directory("train",
                                                       label_mode = 'categorical',
                                                       validation_split=0.2,
                                                       subset="training",
                                                       seed=37,
                                                       image_size=(img_height, img_width),
                                                       batch_size=batch_size,  )
```

```
Found 31751 files belonging to 6 classes.
Using 25401 files for training.
```

```
val_ds = tf.keras.utils.image_dataset_from_directory("train",
                                                       label_mode = 'categorical',
                                                       validation_split=0.2,
                                                       subset="validation",
                                                       seed=37,
                                                       image_size=(img_height, img_width),
                                                       batch_size=batch_size)
```

```
Found 31751 files belonging to 6 classes.
Using 6350 files for validation.
```

```
test_ds = tf.keras.utils.image_dataset_from_directory("test", labels = None,image_size=(img_height, img_width),shuffle = FALSE)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-36-445a4003a111> in <cell line: 1>()
----> 1 test_ds = tf.keras.utils.image_dataset_from_directory("test", labels = None,image_size=
(img_height, img_width),shuffle = FALSE)

NameError: name 'FALSE' is not defined
```

SEARCH STACK OVERFLOW

```
class_names = train_ds.class_names
print(class_names)
```

```
['cakes_cupcakes_snack_cakes', 'candy', 'chips_pretzels_snacks', 'chocolate', 'cookies_biscuits', 'popcorn_peanuts_seeds_related_sn
```

```
resnet_model = Sequential()

pretrained_model= tf.keras.applications.ResNet50(include_top=False,
                   input_shape=(180,180,3),
                   pooling='avg',classes=6,
                   weights='imagenet')
for layer in pretrained_model.layers:
        layer.trainable=False

resnet_model.add(pretrained_model)
resnet_model.add(Flatten())
resnet_model.add(Dense(512, activation='relu'))
resnet_model.add(Dense(6, activation='softmax'))
```

```
resnet_model.summary()
```

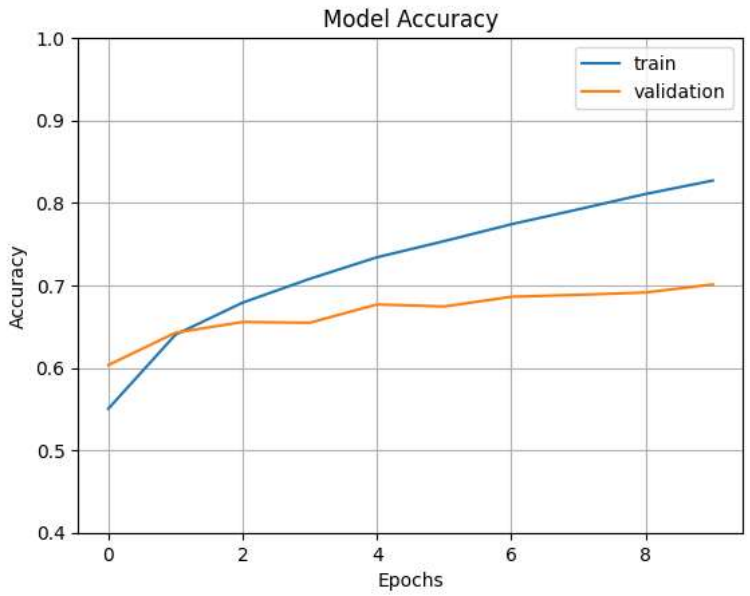
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 module_wrapper_3 (ModuleWra  (None, 2048)             0
 pper)

 module_wrapper_4 (ModuleWra  (None, 512)              1049088
 pper)

 module_wrapper_5 (ModuleWra  (None, 6)                3078
 pper)

=================================================================
Total params: 24,639,878
Trainable params: 1,052,166
Non-trainable params: 23,587,712
_________________________________________________________________
```

```
resnet_model.compile(optimizer=Adam(0.0001),loss='categorical_crossentropy',metrics=['accuracy'])
```

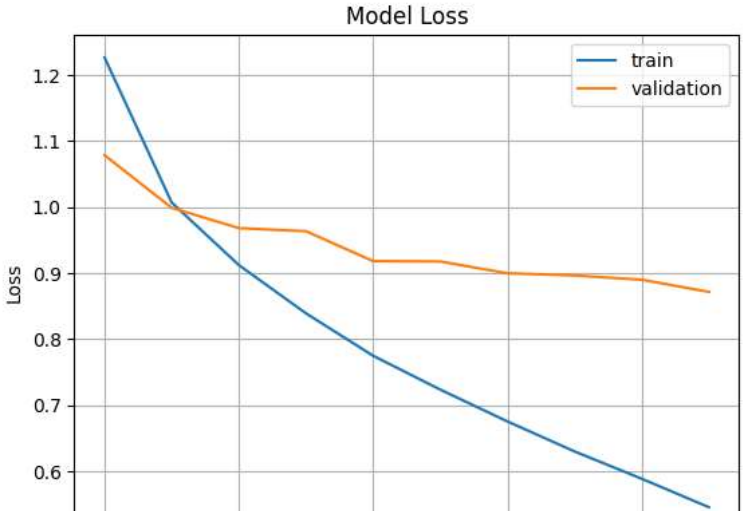
```
epochs=10
history = resnet_model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs
)
```

```
Epoch 1/10
397/397 [==============================] - 77s 184ms/step - loss: 1.2264 - accuracy: 0.5504 - val_loss: 1.0785 - val_accuracy: 0.60
Epoch 2/10
397/397 [==============================] - 72s 179ms/step - loss: 1.0067 - accuracy: 0.6401 - val_loss: 0.9986 - val_accuracy: 0.64
Epoch 3/10
397/397 [==============================] - 78s 195ms/step - loss: 0.9119 - accuracy: 0.6790 - val_loss: 0.9679 - val_accuracy: 0.65
Epoch 4/10
397/397 [==============================] - 71s 179ms/step - loss: 0.8388 - accuracy: 0.7079 - val_loss: 0.9632 - val_accuracy: 0.65
Epoch 5/10
397/397 [==============================] - 72s 180ms/step - loss: 0.7745 - accuracy: 0.7340 - val_loss: 0.9179 - val_accuracy: 0.67
Epoch 6/10
397/397 [==============================] - 71s 178ms/step - loss: 0.7232 - accuracy: 0.7536 - val_loss: 0.9173 - val_accuracy: 0.67
Epoch 7/10
397/397 [==============================] - 78s 195ms/step - loss: 0.6750 - accuracy: 0.7741 - val_loss: 0.8995 - val_accuracy: 0.68
Epoch 8/10
397/397 [==============================] - 71s 179ms/step - loss: 0.6295 - accuracy: 0.7922 - val_loss: 0.8962 - val_accuracy: 0.68
Epoch 9/10
397/397 [==============================] - 71s 179ms/step - loss: 0.5882 - accuracy: 0.8108 - val_loss: 0.8897 - val_accuracy: 0.69
Epoch 10/10
397/397 [==============================] - 71s 179ms/step - loss: 0.5451 - accuracy: 0.8271 - val_loss: 0.8713 - val_accuracy: 0.70
```

```
fig1 = plt.gcf()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.axis(ymin=0.4,ymax=1)
plt.grid()
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'validation'])
plt.show()
```




```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.grid()
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'validation'])
plt.show()
```




```
pred_gen_test = resnet_model.predict(test_ds)
```

```
111/111 [==============================] - 9s 72ms/step
```

```
pred_val = resnet_model.evaluate(val_ds)
```

```
100/100 [==============================] - 15s 145ms/step - loss: 0.8713 - accuracy: 0.7011
```

```
output_classes_test = [class_names[np.argmax(pred_gen_test[i])] for i in range(3525)]
```

```
idx_test = [test_ds.file_paths[i].split("/")[1].split(".")[0] for i in range(3525)]
```

```
import pandas as pd

data = {'idx': idx_test, 'pred_cat': output_classes_test}
df = pd.DataFrame(data)

csv_file_path = 'drive/MyDrive/model_03.csv'

df.to_csv(csv_file_path, index=False)
```

✓ 0s completed at 5:28 PM