lab-08-part2

November 11, 2025

[1]:

```python
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras import  layers, models
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
```

```
2025-11-11 22:41:26.790856: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to
STDERR
E0000 00:00:1762882886.865438   31808 cuda_dnn.cc:8579] Unable to register cuDNN
factory: Attempting to register factory for plugin cuDNN when one has already
been registered
E0000 00:00:1762882886.890510   31808 cuda_blas.cc:1407] Unable to register
cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has
already been registered
W0000 00:00:1762882887.023986   31808 computation_placer.cc:177] computation
placer already registered. Please check linkage and avoid linking the same
target more than once.
W0000 00:00:1762882887.024009   31808 computation_placer.cc:177] computation
placer already registered. Please check linkage and avoid linking the same
target more than once.
W0000 00:00:1762882887.024011   31808 computation_placer.cc:177] computation
placer already registered. Please check linkage and avoid linking the same
target more than once.
W0000 00:00:1762882887.024012   31808 computation_placer.cc:177] computation
placer already registered. Please check linkage and avoid linking the same
target more than once.
2025-11-11 22:41:27.042988: I tensorflow/core/platform/cpu_feature_guard.cc:210]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild
TensorFlow with the appropriate compiler flags.
```

[8]:

```python
data_dir = '/home/tazmeen/CV Lab/lab8/dataset'
```

[9]:

```
# class counts
import os
classes = sorted([d for d in os.listdir(data_dir) if os.path.isdir(os.path.
 ↪join(data_dir, d))])
counts = {c: sum(1 for f in os.listdir(os.path.join(data_dir, c)) if f.lower().
 ↪endswith(('.jpg','.jpeg','.png','.bmp','.gif'))) for c in classes}
print('Number of classes:', len(classes))
for c in classes:
    print(f'{c}: {counts[c]} images')
```

```
Number of classes: 5
daisy: 501 images
dandelion: 646 images
roses: 497 images
sunflowers: 495 images
tulips: 607 images
```

[10]:

```
IMG_SIZE = (224, 224)
BATCH_SIZE = 32
SEED = 42

# 70%
train_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.30,
    subset="training",
    seed=SEED,
    image_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
)

# 30%
valtest_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.30,
    subset="validation",
    seed=SEED,
    image_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
)

# Split 30% into 20% val and 10% test
valtest_batches = tf.data.experimental.cardinality(valtest_ds).numpy()
test_batches = valtest_batches // 3
test_ds = valtest_ds.take(test_batches)
val_ds  = valtest_ds.skip(test_batches)
val_ds   = valtest_ds.skip(test_batches)

data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.2),
    layers.RandomZoom(0.2),
])

rescale = tf.keras.layers.Rescaling(1./255)

# Apply augmentation to the training set ONLY
train_ds = train_ds.map(lambda x, y: (data_augmentation(x, training=True), y),␣
 ↪num_parallel_calls=tf.data.AUTOTUNE)
train_ds = train_ds.map(lambda x, y: (rescale(x), y), num_parallel_calls=tf.
 ↪data.AUTOTUNE)
val_ds   = val_ds.map(lambda x, y: (rescale(x), y), num_parallel_calls=tf.data.
 ↪AUTOTUNE)
test_ds  = test_ds.map(lambda x, y: (rescale(x), y), num_parallel_calls=tf.data.
 ↪AUTOTUNE)

print(f'Training batches: {tf.data.experimental.cardinality(train_ds).numpy()}')
print(f'Validation batches: {tf.data.experimental.cardinality(val_ds).numpy()}')
print(f'Testing batches: {tf.data.experimental.cardinality(test_ds).numpy()}')
```

```
Found 2746 files belonging to 5 classes.
Using 1923 files for training.
```

```
I0000 00:00:1762883214.184451   31808 gpu_device.cc:2019] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 3437 MB memory:  -> device: 0,
name: NVIDIA GeForce GTX 1050 Ti with Max-Q Design, pci bus id: 0000:01:00.0,
compute capability: 6.1
```

```
Found 2746 files belonging to 5 classes.
Using 823 files for validation.
Using 823 files for validation.
Training batches: 61
Validation batches: 18
Testing batches: 8
```

[11]:

```
num_classes = len(classes)

model = models.Sequential([

    layers.Input(shape=(IMG_SIZE[0], IMG_SIZE[1], 3)),


    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),


    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),


    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])

model.summary()
```

```
Model: "sequential_1"


 Layer (type)                      Output Shape                 Param #

 conv2d (Conv2D)                   (None, 222, 222, 32)             896

 max_pooling2d (MaxPooling2D)      (None, 111, 111, 32)               0

 conv2d_1 (Conv2D)                 (None, 109, 109, 64)          18,496

 max_pooling2d_1 (MaxPooling2D)    (None, 54, 54, 64)                 0

 conv2d_2 (Conv2D)                 (None, 52, 52, 128)           73,856

 max_pooling2d_2 (MaxPooling2D)    (None, 26, 26, 128)                0

 flatten (Flatten)                 (None, 86528)                      0

 dense (Dense)                     (None, 128)               11,075,712

 dropout (Dropout)                 (None, 128)                        0

 dense_1 (Dense)                   (None, 5)                        645


 Total params: 11,169,605 (42.61 MB)
 Trainable params: 11,169,605 (42.61 MB)
```

```
 Non-trainable params: 0 (0.00 B)
```

[12]:

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

[13]:

```
EPOCHS = 30

history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=EPOCHS
)
```

```
Epoch 1/30
```

```
WARNING: All log messages before absl::InitializeLog() is called are written to
STDERR
I0000 00:00:1762883241.459310   34193 service.cc:152] XLA service 0x745e18006090
initialized for platform CUDA (this does not guarantee that XLA will be used).
Devices:
I0000 00:00:1762883241.459353   34193 service.cc:160]   StreamExecutor device
(0): NVIDIA GeForce GTX 1050 Ti with Max-Q Design, Compute Capability 6.1
2025-11-11 22:47:21.669986: I
tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:269] disabling MLIR
crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
I0000 00:00:1762883242.607705   34193 cuda_dnn.cc:529] Loaded cuDNN version
91002
2025-11-11 22:47:24.468431: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.9 = (f32[32,32,222,222]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,3,224,224]{3,2,1,0} %bitcast.4699, f32[32,3,3,3]{3,2,1,0}
%bitcast.4706, f32[32]{0} %bitcast.5275), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:24.653720: I
```

```
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.10 = (f32[32,64,109,109]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,32,111,111]{3,2,1,0} %bitcast.5343, f32[64,32,3,3]{3,2,1,0}
%bitcast.4726, f32[64]{0} %bitcast.5403), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1_2/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:25.561731: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.11 = (f32[32,128,52,52]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,64,54,54]{3,2,1,0} %bitcast.5467, f32[128,64,3,3]{3,2,1,0}
%bitcast.4745, f32[128]{0} %bitcast.5527), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_2_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:30.644484: E
external/local_xla/xla/service/slow_operation_alarm.cc:73] Trying algorithm
eng0{} for conv %cudnn-conv-bw-filter.4 = (f32[64,32,3,3]{3,2,1,0}, u8[0]{0})
custom-call(f32[32,32,111,111]{3,2,1,0} %bitcast.5343,
f32[32,64,109,109]{3,2,1,0} %bitcast.5407), window={size=3x3},
dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBackwardFilter",
metadata={op_type="Conv2DBackpropFilter" op_name="gradient_tape/sequential_1_1/c
onv2d_1_2/convolution/Conv2DBackpropFilter"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false} is taking a while…
2025-11-11 22:47:30.750655: E
external/local_xla/xla/service/slow_operation_alarm.cc:140] The operation took
1.106208474s
Trying algorithm eng0{} for conv %cudnn-conv-bw-filter.4 =
(f32[64,32,3,3]{3,2,1,0}, u8[0]{0}) custom-call(f32[32,32,111,111]{3,2,1,0}
%bitcast.5343, f32[32,64,109,109]{3,2,1,0} %bitcast.5407), window={size=3x3},
dim_labels=bf01_oi01->bf01, custom_call_target="__cudnn$convBackwardFilter",
metadata={op_type="Conv2DBackpropFilter" op_name="gradient_tape/sequential_1_1/c
```

```
onv2d_1_2/convolution/Conv2DBackpropFilter"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false} is taking a while…
```

```
 1/61            15:57 16s/step - accuracy:
0.2500 - loss: 1.5975
```

```
I0000 00:00:1762883254.140725   34193 device_compiler.h:188] Compiled cluster
using XLA!  This line is logged at most once for the lifetime of the process.
```

```
60/61            0s 127ms/step -
accuracy: 0.2673 - loss: 1.9011
```

```
2025-11-11 22:47:42.478403: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.9 = (f32[3,32,222,222]{3,2,1,0}, u8[0]{0}) custom-
call(f32[3,3,224,224]{3,2,1,0} %bitcast.4699, f32[32,3,3,3]{3,2,1,0}
%bitcast.4706, f32[32]{0} %bitcast.5275), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:42.626758: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.10 = (f32[3,64,109,109]{3,2,1,0}, u8[0]{0}) custom-
call(f32[3,32,111,111]{3,2,1,0} %bitcast.5341, f32[64,32,3,3]{3,2,1,0}
%bitcast.4726, f32[64]{0} %bitcast.5401), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1_2/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:42.651746: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.11 = (f32[3,128,52,52]{3,2,1,0}, u8[0]{0}) custom-
call(f32[3,64,54,54]{3,2,1,0} %bitcast.5463, f32[128,64,3,3]{3,2,1,0}
%bitcast.4745, f32[128]{0} %bitcast.5523), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_2_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kNone","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
```

```
61/61            0s 191ms/step -
accuracy: 0.2682 - loss: 1.8960
```

```
2025-11-11 22:47:47.759564: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.9 = (f32[32,32,222,222]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,3,224,224]{3,2,1,0} %bitcast.526, f32[32,3,3,3]{3,2,1,0}
%bitcast.533, f32[32]{0} %bitcast.535), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:47.936507: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.10 = (f32[32,64,109,109]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,32,111,111]{3,2,1,0} %bitcast.542, f32[64,32,3,3]{3,2,1,0}
%bitcast.549, f32[64]{0} %bitcast.551), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1_2/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:48.816214: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.11 = (f32[32,128,52,52]{3,2,1,0}, u8[0]{0}) custom-
call(f32[32,64,54,54]{3,2,1,0} %bitcast.557, f32[128,64,3,3]{3,2,1,0}
%bitcast.564, f32[128]{0} %bitcast.566), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_2_1/convolution"
```

```
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:51.278447: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.9 = (f32[23,32,222,222]{3,2,1,0}, u8[0]{0}) custom-
call(f32[23,3,224,224]{3,2,1,0} %bitcast.526, f32[32,3,3,3]{3,2,1,0}
%bitcast.533, f32[32]{0} %bitcast.535), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:51.392507: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.10 = (f32[23,64,109,109]{3,2,1,0}, u8[0]{0}) custom-
call(f32[23,32,111,111]{3,2,1,0} %bitcast.542, f32[64,32,3,3]{3,2,1,0}
%bitcast.549, f32[64]{0} %bitcast.551), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_1_2/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
2025-11-11 22:47:51.990665: I
external/local_xla/xla/service/gpu/autotuning/conv_algorithm_picker.cc:549]
Omitted potentially buggy algorithm eng14{} for conv %cudnn-conv-bias-
activation.11 = (f32[23,128,52,52]{3,2,1,0}, u8[0]{0}) custom-
call(f32[23,64,54,54]{3,2,1,0} %bitcast.557, f32[128,64,3,3]{3,2,1,0}
%bitcast.564, f32[128]{0} %bitcast.566), window={size=3x3},
dim_labels=bf01_oi01->bf01,
custom_call_target="__cudnn$convBiasActivationForward",
metadata={op_type="Conv2D" op_name="sequential_1_1/conv2d_2_1/convolution"
source_file="/home/tazmeen/anaconda3/lib/python3.12/site-
packages/tensorflow/python/framework/ops.py" source_line=1200}, backend_config={
"operation_queue_id":"0","wait_on_operation_queues":[],"cudnn_conv_backend_confi
g":{"conv_result_scale":1,"activation_mode":"kRelu","side_input_scale":0,"leakyr
elu_alpha":0},"force_earliest_schedule":false}
```

```
61/61            35s 314ms/step -
accuracy: 0.2690 - loss: 1.8911 - val_accuracy: 0.4303 - val_loss: 1.3084
Epoch 2/30
Epoch 2/30
61/61            10s 148ms/step -
accuracy: 0.4512 - loss: 1.2937 - val_accuracy: 0.5644 - val_loss: 1.0238
Epoch 3/30
61/61            10s 142ms/step -
accuracy: 0.5603 - loss: 1.1196 - val_accuracy: 0.5220 - val_loss: 1.0826
Epoch 4/30
61/61            10s 141ms/step -
accuracy: 0.5791 - loss: 1.0602 - val_accuracy: 0.6173 - val_loss: 0.9481
Epoch 5/30
61/61            10s 145ms/step -
accuracy: 0.6025 - loss: 1.0207 - val_accuracy: 0.6384 - val_loss: 0.8854
Epoch 6/30
61/61            10s 154ms/step -
accuracy: 0.6000 - loss: 1.0192 - val_accuracy: 0.5256 - val_loss: 1.1137
Epoch 7/30
61/61            9s 134ms/step -
accuracy: 0.6203 - loss: 0.9380 - val_accuracy: 0.6226 - val_loss: 0.9474
Epoch 8/30
61/61            9s 136ms/step -
accuracy: 0.6212 - loss: 0.9432 - val_accuracy: 0.6155 - val_loss: 0.8725
Epoch 9/30
61/61            9s 138ms/step -
accuracy: 0.6585 - loss: 0.8757 - val_accuracy: 0.6402 - val_loss: 0.8595
Epoch 10/30
61/61            10s 141ms/step -
accuracy: 0.6622 - loss: 0.8529 - val_accuracy: 0.6526 - val_loss: 0.8192
Epoch 11/30
61/61            10s 150ms/step -
accuracy: 0.6896 - loss: 0.8291 - val_accuracy: 0.6384 - val_loss: 0.8219
Epoch 12/30
61/61            9s 137ms/step -
accuracy: 0.6951 - loss: 0.7843 - val_accuracy: 0.6772 - val_loss: 0.7539
Epoch 13/30
61/61            10s 141ms/step -
accuracy: 0.6614 - loss: 0.8345 - val_accuracy: 0.5979 - val_loss: 0.9361
Epoch 14/30
61/61            10s 146ms/step -
accuracy: 0.7117 - loss: 0.7631 - val_accuracy: 0.6861 - val_loss: 0.7628
Epoch 15/30
61/61            11s 162ms/step -
accuracy: 0.7130 - loss: 0.7529 - val_accuracy: 0.6667 - val_loss: 0.7851
Epoch 16/30
61/61            11s 167ms/step -
accuracy: 0.6970 - loss: 0.7792 - val_accuracy: 0.6755 - val_loss: 0.7990
```

```
Epoch 17/30
61/61             11s 173ms/step -
accuracy: 0.7073 - loss: 0.7204 - val_accuracy: 0.6684 - val_loss: 0.8137
Epoch 18/30
61/61             11s 170ms/step -
accuracy: 0.6972 - loss: 0.7572 - val_accuracy: 0.7037 - val_loss: 0.7402
Epoch 19/30
61/61             11s 165ms/step -
accuracy: 0.7282 - loss: 0.6902 - val_accuracy: 0.6984 - val_loss: 0.7273
Epoch 20/30
61/61             11s 167ms/step -
accuracy: 0.7249 - loss: 0.7031 - val_accuracy: 0.6896 - val_loss: 0.7740
Epoch 21/30
61/61             11s 161ms/step -
accuracy: 0.7213 - loss: 0.7140 - val_accuracy: 0.7160 - val_loss: 0.7785
Epoch 22/30
61/61             11s 161ms/step -
accuracy: 0.7412 - loss: 0.6943 - val_accuracy: 0.6878 - val_loss: 0.7813
Epoch 23/30
61/61             11s 159ms/step -
accuracy: 0.7315 - loss: 0.6655 - val_accuracy: 0.6772 - val_loss: 0.7817
Epoch 24/30
61/61             11s 164ms/step -
accuracy: 0.7363 - loss: 0.6778 - val_accuracy: 0.7019 - val_loss: 0.7890
Epoch 25/30
61/61             11s 170ms/step -
accuracy: 0.7136 - loss: 0.7461 - val_accuracy: 0.7249 - val_loss: 0.8068
Epoch 26/30
61/61             11s 169ms/step -
accuracy: 0.7460 - loss: 0.6698 - val_accuracy: 0.7425 - val_loss: 0.6943
Epoch 27/30
61/61             11s 173ms/step -
accuracy: 0.7401 - loss: 0.6584 - val_accuracy: 0.7231 - val_loss: 0.7005
Epoch 28/30
61/61             11s 164ms/step -
accuracy: 0.7439 - loss: 0.6664 - val_accuracy: 0.6931 - val_loss: 0.8007
Epoch 29/30
61/61             11s 156ms/step -
accuracy: 0.7366 - loss: 0.6606 - val_accuracy: 0.7178 - val_loss: 0.7118
Epoch 30/30
61/61             11s 159ms/step -
accuracy: 0.7591 - loss: 0.5902 - val_accuracy: 0.6931 - val_loss: 0.7135
```

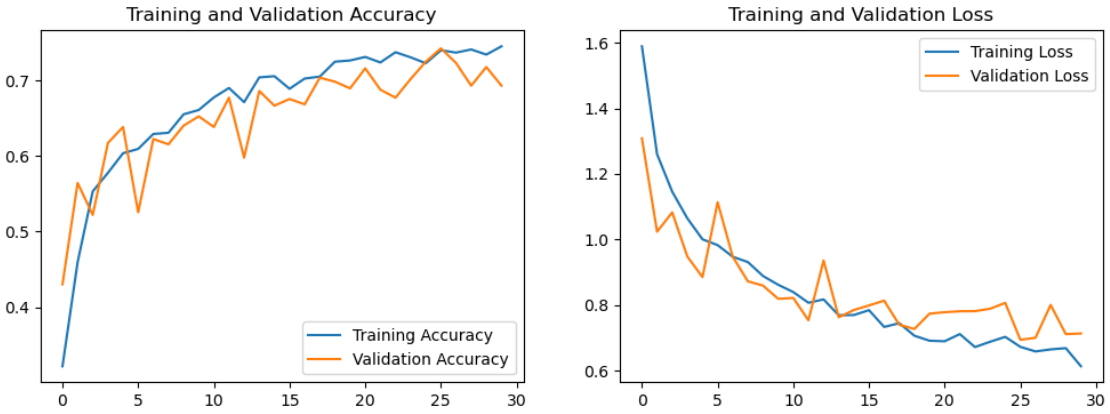
[14]:

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
```

```
epochs_range = range(EPOCHS)

plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



[15]:

```
test_loss, test_acc = model.evaluate(test_ds)
print(f"VGG16 Test Accuracy: {test_acc*100:.2f}%")
```

```
8/8             1s 51ms/step -
accuracy: 0.6810 - loss: 0.8349
VGG16 Test Accuracy: 71.09%
```

[ ]: